End-to-End Camera Calibration for Broadcast Videos

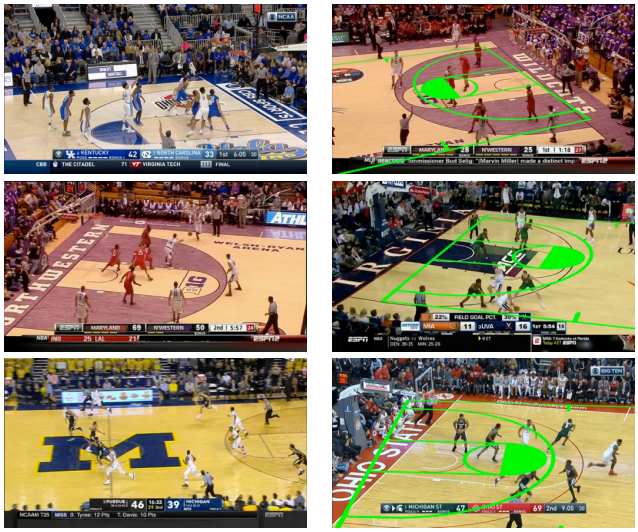
Long Sha Jennifer Hobbs Panna Felsen Xinyu Wei Patrick Lucey Sujoy Ganguly
Stats Perform
{long.sha,jennifer.hobbs,panna.felsen,xinyu.wei,patrick.lucey,sujoy.ganguly}@statsperform.com

Abstract

The increasing number of vision-based tracking systems deployed in production has necessitated fast, robust camera calibration. In the domain of sport, the majority of current work focuses on sports where lines and intersections are easy to extract, and appearance is relatively consistent across venues. However, for more challenging sports like basketball, those techniques are not sufficient. In this paper, we propose an end-to-end approach for single moving camera calibration across challenging scenarios in sports. Our method contains three key modules: 1) area-based court segmentation, 2) camera pose estimation with embedded templates, 3) homography prediction via a spatial transform network (STN). All three modules are connected, enabling end-to-end training. We evaluate our method on a new college basketball dataset and demonstrate the state of the art performance in variable and dynamic environments. We also validate our method on the World Cup 2014 dataset to show its competitive performance against the state-of-the-art methods. Lastly, we show that our method is two orders of magnitude faster than the previous state of the art on both datasets.

1. Introduction

Camera calibration is a fundamental task for computer vision applications such as tracking systems, SLAM, and augmented reality (AR). Recently, many professional sports leagues have deployed some version of a vision-based tracking system [26, 25]. Additionally, AR applications (e.g., Virtual 3 in NBA [3], First Down Line in NFL [15]) used during video broadcasts to enhance audience's engagement have become commonplace. All of these applications require high-quality camera calibration systems. Presently, most of these applications rely on multiple pre-calibrated fixed cameras or the real-time feed of pan-tilt-zoom (PTZ) parameters directly from the camera. However, as the most widely available data source in the sports domain is broadcast videos, the ability to calibrate from a single, moving camera with unknown and changing camera parameters would greatly expand the reach of player tracking data and fan-engagement solutions. Calibration of a single moving camera remains a challenging task as the approach must be accurate, fast, and generalizable to a variety of views and appearances. Our solution enables us to determine the camera homography of a single moving camera with only the frame and the sport.



(a) (b)

Figure 1. (a) Some of the critical challenges of camera calibration in highly dynamic environments, like basketball, are the different appearances from video to video, heavy occlusion on registration features caused by players, and motion blur due to fast camera movements (enlarge to see the blur). (b) In systems with moving cameras, small camera movements generate large transformations. Here the green lines are the projection caused by moving the camera a small amount. The changes in pan, tilt, and focal length are less than 3°, 3°, and 300 pixels, respectively. The change in camera location is less than 3 feet in each axis.

Current approaches mainly follow a framework based on field registration, template matching (i.e., camera pose initialization), and homography refinement. Although existing approaches based on this framework [5, 24, 6] have proven effective for specific sports, there are still some limitations that prevent them from being applied to more challenging scenarios. First, most of these approaches [5, 24, 6, 8, 9, 4] focus on sports where semantic information (i.e., key court

markings) is easy to extract, the field appearance is consistent across stadiums (i.e., green grass and white lines), and motion of the camera is relatively slow and smooth. These assumptions do not hold in more dynamic sports like basketball (Figure 1), where players occlude field markings, the field appearance varies wildly from venue to venue, and the camera moves quickly.

Furthermore, most existing works consist of separately trained or tuned modules. As a result, they cannot achieve the global optimal for such an optimization task. This issue further limits the performance of those methods in more challenging scenarios as error propagates through the system, module to module.

In this paper, we address these issues with a brand new end-to-end neural network (Figure 2). Our method follows a similar framework (semantic segmentation, camera pose initialization, homography refinement), but extends the approach to handle more challenging scenarios involving motion blur, occlusion, and large transformations. Our contributions are:

1. A method to use area-based semantics rather than lines for camera calibration, which is more robust for dynamic environments and those with highly variable appearance features.
2. Incorporation of a spatial transform network [16] for large transform learning, which reduces the number of required templates.
3. An end-to-end architecture for camera calibration, which allows us to train everything jointly and inference homography much more efficiently.
4. A well-curated basketball dataset that allows the community to study the calibration problem in a more challenging environment.

The structure of the paper is as follows. In section 2, we discuss the related work in the area, followed by section 3, where we detail our method. In section 4, we describe two experiments on both soccer and basketball datasets. Finally, in section 5, we conclude and discuss future directions.

2. Related Work

Field Registration Field registration is a critical component of camera calibration in sport. It enables the generation of reliable real-world or "top-down" tracking data, which is widely used in sports analytics [23, 30, 13]. Mathematically, the task is to find the homography that can map the 2D field from the observed camera perspective to a known overhead perspective. Many classical methods exist to find correspondence between points or line segments [20, 8, 9, 4, 10]. Others [21, 9, 8, 4, 10] followed a frame-to-frame scheme where they calibrated each sequence using the initial homography and frame to frame matching. These approaches often required human intervention and venue specific priors.

Methods based on court segmentation fully-automated this process by applying court segmentation on a synthesized panoramic image [28, 29]. Based on the work in [21], Hess *et al*. [12] eliminated the need for manual initialization by pre-defining a venue-specific overhead (i.e., top-down) field model so that every frame could be matched to the field model directly. Recently [22], convolutional neural networks (CNN) have been introduced for better semantic extraction. Homayounfar *et al*. [14] extended the works of [10, 11] with CNN-based semantic detection to more precisely estimate vanishing points of a field in the image plane. Chen *et al*. [5] extended this work by using two generative adversarial networks (GANs) to extract the edges on a field, producing better reference image matching.

Pan-Tilt-Zoom Formulation Recently an increasing number of approaches exploit a pan-tilt-zoom (PTZ) camera configuration to constrain the registration problem in broadcast video. Chen *et al*. [6] leveraged the methods of [27] and [18] to estimate the base parameters and focal length for the camera. Sharma *et al*. [24] and Chen *et al*. [5] estimated the ranges of pan, tilt, and zoom from training data and uniformly sampled a large number (100k) of potential camera poses. Using an overhead field model and the sampled camera poses, they generate semantic images from a camera perspective to act as templates. By constructing a dictionary of templates, they reduced the problem of field registration to the nearest neighbor search task, followed by fine-tuning and refinement.

Homography Refinement Previous approaches use methods like the Lucas-Kanade algorithm [1] or Inexact Augmented Lagrangian Method (IALM) to refine the homography after its initialization from a matched template or reference frame. A fundamental assumption of these approaches is that the transformation is small and local. To satisfy these assumptions, Sharma *et al*. [24] and Chen *et al*. [5] used 100k templates, ensuring the transform between the input image and the matched template was very small, whereas Carr *et al*. [4] and Ghanem *et al*. [8] formulated this problem as a non-linear optimization task, optimizing the homography through the loss of image warping.

Jaderberg *et al*. [16] proposed a spatial transformer network (STN) that learned the affine transform for handwritten digit patches and improved recognition accuracy. Bhagavatula *et al*. [2] and Lin *et al*. [19] also showed that the STN could handle some perspective transforms on faces and rigid objects. We incorporate these methods to address the need to handle large perspective transforms during refinement, and create a fully neural network solution.

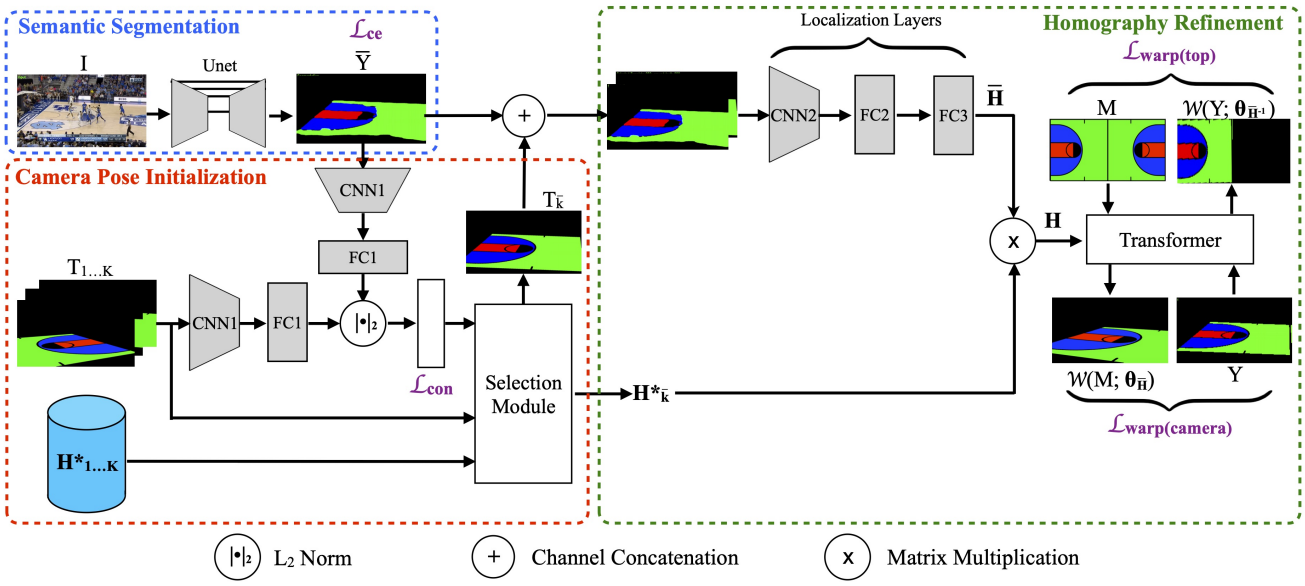


Figure 2. Given a single input image, we calculate the homography $\mathbf{H}$. First, we find the court features Y using semantic segmentation (blue box). Then we use the semantic map to select an appropriate template $T_{\bar{k}}$ from a set of templates. We concatenate $T_{\bar{k}}$ and $\bar{Y}$ and predict the relative homography $\bar{\mathbf{H}}$ between the template and the semantic map. Next $\mathbf{H} = \bar{\mathbf{H}}\mathbf{H}^*_{\bar{k}}$ is used to generate the real-world to image and the image to real-world warping. The gray blocks are neural network layers, and the blocks with the same name share the same weights. This network architecture has four distinct loss functions $\mathcal{L}_{\text{ce}}$ (Eq. 3), which comes from the semantic segmentation module, $\mathcal{L}_{\text{con}}$ (Eq. 6), which comes from the camera pose initialization module, and two warping losses $\mathcal{L}_{\text{warp(camera)}}$ and $\mathcal{L}_{\text{warp(top)}}$ which come from the homography refinement module (Eq. 8). Since all the losses are fully differentiable with respect to the network parameters, this network can be trained end-to-end.

3. Method

The goal of our method is to find a homography $\mathbf{H}$ that can register the target ground-plane surface of any frame I from a broadcast video with a top view field model M. The standard objective function for computing homography with point correspondences is

$$\mathbf{H} = \arg\min_{\mathbf{H}} \frac{1}{|\mathcal{X}|} \sum_{(\mathbf{x}'_i, \mathbf{x}_i) \in \mathcal{X}} |\mathbf{H}\mathbf{x}'_i - \mathbf{x}_i|_2, \tag{1}$$

where $\mathbf{x}_i$ is the (x, y) location of pixel i in the (broadcast) image I and $\mathbf{x}'_i$ is the corresponding pixel location on the model "image" M. $\mathcal{X}$ is a set of point correspondences between the two images I and M.

Our method leverages three major techniques; semantic segmentation, camera pose initialization and homography refinement. Because each task can be accomplished with neural networks, all three can be integrated into a single network architecture (Figure 2) and trained end-to-end.

3.1. Semantic Segmentation

Semantic segmentation is usually used to extract key features and remove irrelevant information from the image, providing a venue agnostic appearance $\bar{Y}$ that can be used to determine the point correspondences. Thus the objective function (Equation 1) can be rewritten as

$$\theta_{\mathbf{H}} = \arg\min_{\theta_{\mathbf{H}}} \mathrm{L}(\bar{Y}, \mathcal{W}(M; \theta_{\mathbf{H}})), \tag{2}$$

where $\theta_{\mathbf{H}}$ is a vector of the 8 homography parameters, $\mathcal{W}(;\theta)$ is the warping function with transform parameter θ, and $\mathrm{L}()$ is any loss function that measures the difference between two images, in this case the predicted semantic map $\bar{Y}$ and the warped overhead model M.

We conduct area-based segmentation on the field to address the challenges shown in Figure 1. The field is divided into four regions, making the overhead field model M a 4-channel image as seen in Figure 3; the goal of this module is to classify each pixel in I into one of the four classes. To generate the area-based semantic labels of each image, we warp the overhead model with the associated ground truth homography, thus providing ground truth semantic labels for training.

For the segmentation task we use a Unet [22] style auto-encoder (see detailed architecture in the Appendix Section A) which takes an image I and outputs a semantic map $\bar{Y}$ as needed by the final objective function (Equation 2). To train the Unet, we use the cross-entropy loss

$$\mathcal{L}_{\text{ce}} = -\frac{1}{|\bar{Y}||C|} \sum_{\bar{y}^c_i \in \bar{Y}} \sum_{c \in C} y^c_i \log(\bar{y}^c_i), \tag{3}$$

where C is the set of classes, and y^c_i is the ground truth

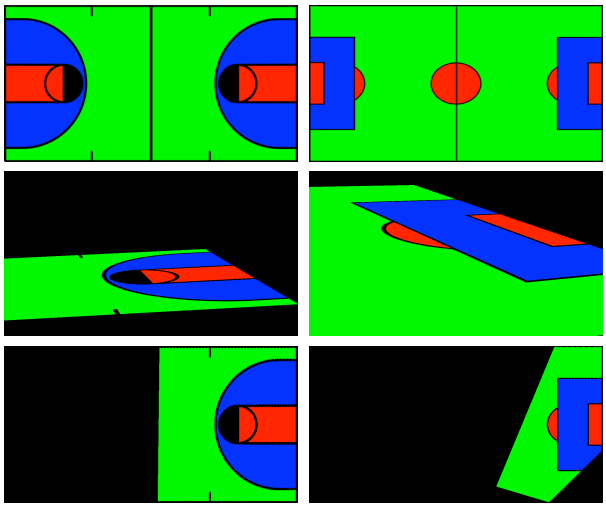


Figure 3. The top row shows our top-down view field model for basketball (left) and soccer (right). The middle row shows our semantic labels for one image by using the field models. These images are generated by warping the field model $\mathcal{M}$ using the ground truth homography. These images are then used to train the semantic segmentation module. The bottom row shows the polygonal area of the middle row from top-down perspective, showing the fraction of the field model in the camera view. The top-down views of basketball and soccer are resized here to the same dimensions for display purpose only.

label and $\bar{y}_i^c$ is the likelihood of pixel i belonging to class c.

3.2. Camera Pose Initialization

Since we assume a PTZ camera we can generate a camera pose dictionary (i.e. set of templates) based on the range of possible pan, tilt and focal length parameters. We use a siamese network to determine the best template for each input semantic image.

3.2.1 Camera Pose Dictionary Generation

For a PTZ camera, the projective matrix $\mathbf{P}$ can be expressed as

$$\mathbf{P} = \mathbf{KR}[\mathbf{I}|-\mathbf{C}] = \mathbf{KQS}[\mathbf{I}|-\mathbf{C}], \tag{4}$$

where $\mathbf{Q}$ and $\mathbf{S}$ are decomposed from rotation matrix $\mathbf{R}$, $\mathbf{K}$ are the intrinsic parameters of the camera, $\mathbf{I}$ is the $3X3$ identity matrix and $\mathbf{C}$ is the camera translation. The matrix $\mathbf{S}$ describes the rotation from the world coordinate to the PTZ camera base, and $\mathbf{Q}$ represents the camera rotation due to pan and tilt. In our case, we define $\mathbf{S}$ to rotate around world x-axis by -90° so that the camera looks along the y-axis in the world plane; this means the camera is level and its projection is parallel to the ground.

For each image, we assume a center principle point, square pixels, and no lens distortion. Camera rolling is ignored in our work since the rolling angle is observed to be very small (less than 1°), leaving 6 parameters in total: the focal length, 3D camera location, pan and tilt angles.

Algorithm 1 GMM-based clustering algorithm

Pre-define covariance Σ
for $K = [100, 110, 120, ..., N]$ **do**
  Initialize μ_k for K GMM components
  while μ not converge **do**
    Compute $\gamma_k(\lambda_n) = \frac{\pi_k \mathcal{N}(\lambda_n;\mu_k,\Sigma)}{\sum_j \pi_k \mathcal{N}(\lambda_n;\mu_k,\Sigma)}$
    Update $\mu_k = \frac{\sum_n \gamma_k(\lambda_n)\lambda_n}{\sum_n \gamma_k(\lambda_n)}$
  end while
  if $\frac{1}{N}\sum_n \max_k \frac{\mathcal{N}(\lambda_n;\mu_k,\Sigma)}{\mathcal{N}(\mu_k;\mu_k,\Sigma)} > threshold$ **then**
    break
  end if
end for
Return GMM

We use Zhang's method [31] and the ground truth homography to initialize the intrinsic camera matrix $\mathbf{K}$, camera location $\mathbf{C}$, and rotation matrix $\mathbf{R}$. With this initialization, we use the Levenberg–Marquardt algorithm [17] to find the optimal focal length, 3D camera location, and rotation angles. Once $\mathbf{K}$, $\mathbf{C}$, $\mathbf{R}$ and $\mathbf{S}$ are determined, $\mathbf{Q}$ is computed. The Rodrigues formula [7] is applied to $\mathbf{Q}$ to compute the pan and tilt angles. Thus, the 6-dimensional camera configuration (pan, tilt, zoom, and 3D camera location) λ is determined. Although the estimation of the camera parameters from a single image is not very precise, it is sufficient for camera pose dictionary generation.

After the camera configuration λ is estimated for each training image, we generate a dictionary of possible camera poses Λ in one of two ways. The first method entails uniform sampling from the range of possible camera poses. We determine the ranges of pan, tilt, focal length, and camera location from training data and uniformly sample the poses from a 6-dimensional grid. The advantage of this method is that it covers all camera poses even if the training set is small. Additionally, using a small grid simplifies the homography refinement since the maximum scale of the transformation required is on the scale of the grid size. However, this also creates many templates that are not realistic.

Alternatively, when the training set has sufficient diversity, Λ can be learned directly from training data via clustering. We chose to treat Λ as a multi-variant normal distribution and apply a Gaussian mixture model (GMM) to build our camera pose set. We fix the mixing weights π as equal for each component and fix the covariance matrix Σ for each distribution. Here the characteristic scale of Σ sets the scale of the transformations that handled by the homography refinement module. In contrast with traditional GMM's, instead of setting the number of components K, the GMM learning algorithm finds the number of components K and the mean μ_k of each distribution given the mixing weights π and covariance matrix Σ. The identical Σ and π for each component ensures the GMM components are sampled uniformly from the manifold of the training data.

Algorithm 1 shows the GMM clustering procedure. Because we fix Σ, we only update μ during the maximization step (M-step). We gradually increase K until the stopping criteria are satisfied. The stopping criteria (line 8) aims to generate enough components so that every training example is close to the mean of one component in the mixture. The camera pose dictionary Λ is formed utilizing all components $[\mu_1, ..., \mu_K]$.

Given the dictionary of camera poses Λ, the homography for each pose can be computed and used to warp the overhead field model M. Therefore, a set of image templates $\mathcal{T} = [T_1, ..., T_K]$ and their corresponding homography matrices $\mathcal{H}^* = [\mathbf{H}_1^*, ..., \mathbf{H}_K^*]$ are determined and used in the camera pose initialization module.

3.2.2 Camera Pose Search

Given the semantic segmentation image $\bar{Y}$ and a set of template images $\mathcal{T}$, a siamese network is used to compute the distance between each input and template pair $(\bar{Y}, T_k)$. The target/label for each pair is *similar* or *dissimilar*. For the grid sampled camera pose dictionary, a template T_k is similar to the image if its pose parameters are the nearest neighbor in the grid. For the GMM-based camera pose dictionary, a template T_k is labeled as similar to an image if the corresponding distribution of the template $\mathcal{N}(; \mu_k, \Sigma)$ gives the highest likelihood to the pose parameters λ of the input image. This procedure generates a template similarity label for every image in the training set.

The red box in Figure 2 shows the steps of the camera pose search process. Once the input semantic image $\bar{Y}$ and the template images $\mathcal{T}$ are encoded (after FC1), the latent representations are used to compute the L2 distance between the input image and each template. A selection module finds the target camera pose index $\bar{k}$ and retrieves its template image $T_{\bar{k}}$ and homography $\mathbf{H}_{\bar{k}}^*$ as output, according to

$$\bar{k} = \arg\min_k |f(\bar{Y}) - f(T_k)|_2, \tag{5}$$

where $f()$ is the encoding function of siamese network.

As is standard practice, contrastive loss is used to train the siamese network:

$$\begin{aligned}\mathcal{L}_{con} = a|f(\bar{Y}) - f(T_k)|_2^2 \\ + (1-a)\max(0, m - |f(\bar{Y}) - f(T_k)|_2^2),\end{aligned} \tag{6}$$

where a is the binary similarity label for the image pair $(\bar{Y}, T_k)$ and m is the margin for contrastive loss.

3.3. Homography Refinement

After determining the target template and camera pose, the last step is to refine the homography by finding the relative transform between the selected template and the input image. We introduce the spatial transformer network (STN) for this task so that we can handle large non-affine transformation and use a smaller camera pose dictionary.

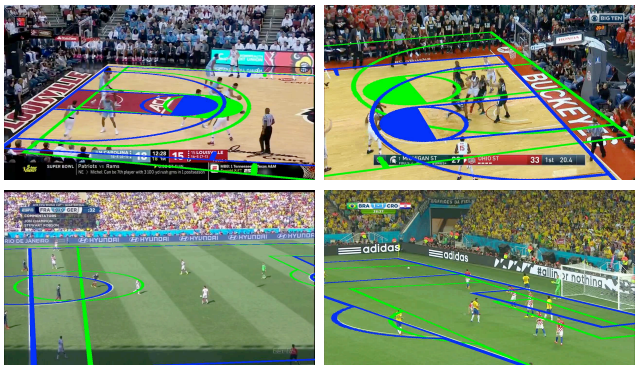


Figure 4. Here, the green field lines are the projection with the initialized camera pose parameters (before refinement), and the blue field lines come after refinement of the homography. We can see that the STN can handle relatively large transforms, which enables the network to use far fewer templates for camera pose initialization.

The green box in Figure 2 shows the process of homography refinement. To compute the relative transform between the input semantic image $\bar{Y}$ and the selected template image $T_{\bar{k}}$, we stack them into an 8-channel image, forming the input to the localization layers of an STN. The output of the localization layers is the 8 parameters of the relative homography $\bar{\mathbf{H}}$ that maps the semantic image $\bar{Y}$ to the template T_k.

Importantly, we initialize the last of the localization layers (FC3 in Figure 2) such that all elements in the kernel are zero, and the bias is to the first 8 values of a flattened identity matrix. Therefore, at the start of training, the input image is assumed to be identical to the template, providing a good initialization for the STN optimization. Therefore, the final homography is $\mathbf{H} = \mathbf{H}_{\bar{k}}^* \bar{\mathbf{H}}$.

Once $\mathbf{H}$ is computed, the transformer can warp the overhead model M to the camera perspective or vice versa, which allows us to compute the loss function in Equation 2. We use the Dice coefficient loss:

$$Dice(U, V) = \frac{1}{|C|} \sum_{c \in C} \frac{2||U^c \circ V^c||}{||U^c|| + ||V^c||}, \tag{7}$$

where U, V are semantic images, C is the number of channels, $\circ$ is the element-wise multiplication, and $||\cdot||$ is the sum of pixel intensity in an image. Here, the intensity in each channel is the likelihood that the pixel belongs to a channel c. One of the major advantages of using area-based segmentation, as opposed to line-based, is that it is robust to occlusions and makes better use of the network capacity since a larger fraction of image pixels belong to a meaningful class.

However, a limitation of IoU-based loss is that as the fraction of the field in view decreases, the IoU loss becomes sensitive to segmentation errors. For example, if the field

only occupied a tiny proportion of the image, a small transform could reduce the IoU dramatically. Figure 3 shows two examples of the occupancy fraction in basketball and soccer from two perspectives. Soccer has a higher occupancy fraction from the camera perspective, while top view has a lower occupancy fraction, and basketball is the opposite. Therefore, we use the Dice loss on the warped field in both perspectives; the high occupancy perspective can achieve coarse registration quickly while the low occupancy perspective can provide strong constraints on fine-tuning. Thus, we define the loss function in Equation 2 as,

$$\begin{aligned}\mathcal{L}_{warp} = \delta Dice(Y, \mathcal{W}(M, \theta_{\mathbf{H}})) \\ + (1-\delta) Dice(M', \mathcal{W}(Y, \theta_{\mathbf{H}^{-1}})),\end{aligned} \tag{8}$$

where Y is the ground truth semantic image and M' is masked overhead field model so that loss is only computed for the area shown in the image. Losses from the two perspectives are weighted by δ, where the weight for the lower occupancy fraction perspective is always higher.

Figure 4 shows some example homography refinement results. The green field lines are projected with the initial camera poses from the selected templates, while the blue projections use the refined homographies. Those results showcase the ability of STN to learn relatively large transformations, which allows us to use a much smaller camera pose set in our method.

3.4. Learning

Since each module uses the output of other modules as input, the three modules can be connected into a single neural network, as shown in Figure 2. The total loss of the network becomes

$$\mathcal{L} = \alpha \mathcal{L}_{ce} + \beta \mathcal{L}_{con} + (1-\alpha-\beta)\mathcal{L}_{warp}, \tag{9}$$

where $\alpha, \beta \in [0, 1)$.

We turn on the training of the entire network incrementally, module-by-module, so the siamese network and STN can start training with reasonable inputs. Training starts with a 20-epoch warm-up for the Unet. Then the siamese network training is turned on with $\alpha = 0.1$ and $\beta = 0.9$. After another 10 epochs, the STN is turned on with $\alpha = 0.05$ and $\beta = 0.05$. The full network continues joint training until convergence.

4. Evaluation And Experiments

4.1. Dataset

College Basketball Dataset We create a dataset from 13 NCAA basketball games. We use 10 games for training and the remaining 3 for testing. Different games have different camera locations, and each game was played in a unique venue; this means the field appearance is very different from game to game. For each game, we selected 30-60 frames for annotations with a high camera pose diversity. Professional annotators clicked 4-6 point correspondences in each image to compute the ground truth homography. These annotations produced 526 images for training and 114 images for testing. We further enrich the training data by flipping the images horizontally, which gives us 1052 training examples in total.

World Cup 2014 Dataset A soccer dataset was collected by Homayounfar *et al.* [14] from 20 games of the World Cup 2014. Those games were held in 9 different stadiums during day and night, and the images consist of different perspectives and lighting conditions. There are 209 training images collected from 10 games and 186 testing images from the other 10 games.

4.2. Implementation

College Basketball Dataset Since the training set for the basketball dataset is large and diverse, we use the GMM-based method to generate camera pose templates from 1052 training images. The standard deviation for pan, tilt, focal length, and camera locations (x, y, z) are set to $5°$, $5°$, 1000 pixels, and 15 feet respectively. The non-diagonal elements are set to zero as we assume those camera configurations are independent of each other. The threshold for the stopping criteria was set to 0.6 and the clustering algorithm-generated 210 components.

For the warping loss, $\mathcal{L}_{warp}$ δ is set to 0.8 because the camera perspective has a lower field occupancy rate than the top view perspective.

World Cup Dataset Because the soccer field is much larger than basketball, a high grid resolution is used for template generations: we set the resolution of pan, tilt, and focal length to $5°$, $2.5°$, and 500 pixels. The camera location is fixed at (560, 1150, 186) yard relative to the top left corner of the field since camera locations are very similar among different games. The soccer dataset has an insufficient number of examples to use the GMM-based camera pose estimation. Therefore, we used a uniform sampling for this dataset with estimated pan, tilt, and focal length range ($[-35°, 35°]$, $[5°, 15°]$, $[1500, 4500]$ pixels respectively), which generates 450 templates for camera pose initialization.

It is worth noting that the sampling resolutions we selected for both soccer and basketball are **NOT** guaranteed to be optimal. Using different resolutions may lead to better or worse performance, but in this paper, we focus on demonstrating the outstanding performance of our method with a much smaller template set. Investigating the optimal size of template set is out of the scope of this work.

Due to the small number of training examples in the World Cup dataset, we use synthesized data to warm up the camera pose initialization module and the homography refinement module. Apart from the Unet, the rest of the net-

work uses the semantic image as input so that we can synthesize an arbitrary number of semantic images to pre-train those parts of the network. We generated 2000 semantic images by uniformly sampling the pan, tilt, and focal length parameters. For each synthesized image, their ground truth homography is known, and template assignment can be easily found by down-sampling the grid. Thus, the camera pose initialization module, and STN can be pre-trained individually. After these two modules are warmed up, we follow the training procedure in section 3.4 to train the network with real data. Because soccer's top view has a lower field occupancy rate, δ was set to 0.2 for training.

4.3. Quantitative evaluation

We use the intersection over union (IoU) score as our evaluation metric. We compute the IoU in the top view and compare the intersection between the ground truth and the estimated homography. Previous works [14, 24, 5] measured the IOU on either the entire field IoU_{entire} or on only the polygonal area that appeared in the image IoU_{part}. We report our results under both approaches for ease of comparison. Our approach is implemented with Tensorflow on an Ubuntu system with an Intel 3.6GHz CPU, 48GB memory, and an Nvidia Titan RTX GPU.

Table 1 compares our method to Chen *et al*. [5] on the basketball dataset. The method in [5] is implemented with their released code. To ensure a fair comparison of the various calibration methods, we use the same template set (210 camera poses) for [5]. Apart from the direct comparison between [5] and our method, we also created two impractical variants of [5]; first, we provide perfect line extraction, and second, we supply perfect templates. We designed these variants to show the impact of these factors on calibration performance. To provide perfect templates, we created one template per example in the dataset, so every image has a perfect match, and no homography refinement is required. By providing the perfect segmentation or templates, we can see the best theoretical result that can be achieved by Chen *et al*.'s method. We also compare the performance of joint and separate training of each module to evaluate the benefit of end-to-end training on our network.

The result in Table 1 shows our method is a significant improvement (5% - 15%) over Chen *et al*. [5], even when provided with ground truth line segmentation or perfect templates. When we provide ground truth line extraction to the baseline, the IoU_{part} improves because camera pose initialization becomes trivial, but IoU_{entire} remains low because refinement based on Lucas-Kanade cannot handle the large transformations required by a small template set. In contrast, an infinitely large template set (perfect templates) addresses the error in the refinement step, so IoU_{entire} increases substantially, although the line extraction approach still limits the performance. Those results verify that in order to perform camera calibration in challenging dynamic environments, the networks need better semantic segmentation and homography refinement methods. Our method has an approximately 2% improvement over even the impractical variants of the previous state of the art.

Table 1. Evaluation results on College Basketball dataset. For the baseline method, 'GT' means the ground truth line extraction is provided while 'Per' means perfect templates are given. For our method, results from training each module separately and end-to-end training are reported. Here our method, when trained end-to-end is significantly better than the previous state of the art (Chen *et al*. [5]), even when the previous state of the art is provided with the best possible set of templates (Chen *et al*. + Per).

Method	IoU_{entire}		IoU_{part}	
	Mean	Med.	Mean	Med.
Chen *et al*.	62.6	68.4	85.0	90.1
Chen *et al*. + GT	67.2	71.2	91.7	91.8
Chen *et al*. + Per	80.5	82.3	91.9	94.5
Ours (Modular)	81.1	81.7	92.6	93.8
Ours (End-to-End)	**83.2**	**84.6**	**94.2**	**95.4**

Table 2. Evaluation results on World Cup dataset [14]. Baseline methods are take from their papers. These results indicate that our method is significantly better than Homayounfar *et al*. [14] and Sharma *et al*. [24], but marginally worse than Chen*et al*. [5] due to insufficient training data.

Method	IoU_{entire}		IoU_{part}	
	Mean	Med.	Mean	Med.
DSM [14]	83	-	-	-
Sharma *et al*. [24]	-	-	91.4	92.7
Chen *et al*. [5]	**89.4**	**93.8**	**94.5**	**96.1**
Ours	88.3	92.1	93.2	**96.1**

Table 3. Comparison of inference time and the size of effective search space between different methods

Method	Mean Time(s)	# of templates/iter
DSM [14]	0.44	3328
Sharma *et al*. [24]	-	100,000
Chen *et al*. [5]	0.5	100,000
Ours	**0.004**	**450**

Table 2 shows the results of our method on the World Cup dataset. We compare our method with end-to-end training against previous methods under both metrics using both mean and median. Our method performed significantly better than Homayounfar *et al*. [14] and Sharma *et al*. [24], but approximately equal to Chen *et al*. [5]. On this dataset, our method suffers the insufficient training data, particularly for the semantic segmentation.

In Table 3, we also report the average inference time per image and the size of the search space in different methods. The method of [14] requires a search of a $300^2 \times 600^2$ grid, although they use branch and bound techniques to reduce this search space substantially. Thus, we only compare to

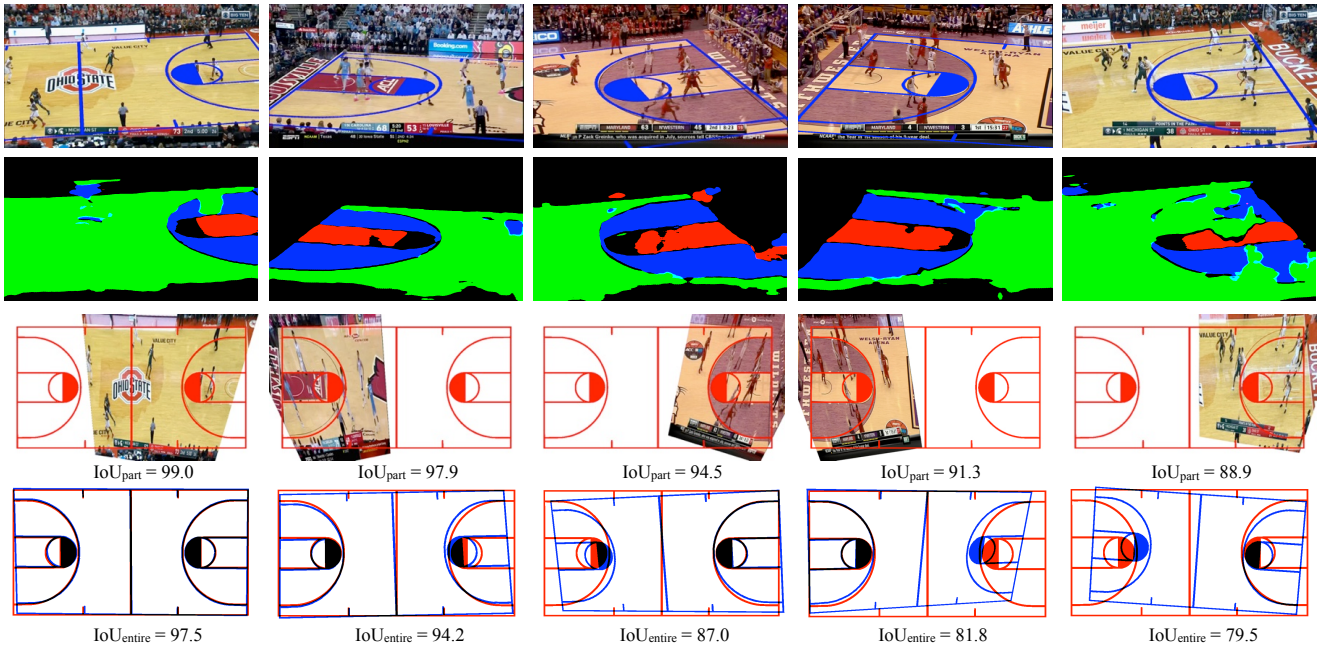


Figure 5. In the first row, we show the field projection (blue lines) generated by the predicted homography. The second row shows the semantic segmentation output. The third row shows the IoU_{part}, and the fourth row shows the IoU_{entire}, where red is the ground truth field, and the blue is the field warped by the predicted homography.

their effective search space, which is the average number of iterations required for homography estimation. Our speed is 2 orders of magnitude faster than [14] and [5] due to the end-to-end architecture and reduced search space, allowing our method to calibrate a moving camera live.

4.4. Qualitative Evaluation

Figure 5 shows some output examples from the basketball dataset; soccer figures can be found in Appendix Section B. Our method works quite well as long as the semantic segmentation is reasonable. For the rightmost example, calibration failed due to poor segmentation as a result of lighting variability. In soccer, the large shadows in the stadium similarly lead to poor segmentation and calibration results. More training data is required to enable the Unet to generalize to these extreme conditions. However, typically, the semantic segmentation module performs very well, leading to near-perfect calibration results. In fact, due to the end-to-end training of the network, we reduce the effect of small errors in semantic segmentation compared to the previous state of the art.

Thought the IoU_{part} is similar between soccer and basketball, the IoU_{entire} is quite different. The reason is twofold. Firstly, the top-down view of basketball is the higher occupancy perspective. Therefore, a small error in homography does not influence the IoU_{part} but can limit the performance in IoU_{entire}. Secondly, the basketball field has a larger width-to-height ratio than soccer, so a small error on one side can lead to a larger error on the out of view side.

5. Conclusion

In this work, we present an novel method for broadcast camera calibration in a dynamic environment which integrates semantic segmentation, camera pose initialization, and homography refinement into one neural network, which enables end-to-end training and inference. Furthermore, we use area-based rather than line-based semantics, which allows our method to handle noisy scenarios where there is significant occlusion of the court. We also used a spatial transformer network for the homography refinement task, allowing the refinement module to handle large transformations, thereby reducing the search space for camera pose initialization. The evaluation results show that our method outperforms the previous state-of-the-art in challenging scenarios like basketball and achieves competitive performance in relatively static environments like soccer.

One drawback of our method is that the selection step in the camera pose initialization module is not differentiable due to the $\arg\min$ operation. Thus, the back-propagation from the homography refinement module cannot flow into the camera pose initialization module. This limitation prevents us from using self-consistency between the warped image in the STN and the output from the Unet, which should be identical. Therefore, if the selection step was differentiable, we could train our network in a weakly supervised fashion. The weakly surprised training would also address the need for more substantial training datasets; we leave this as a challenge for our future work.

References

[1] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004. 2

[2] Chandrasekhar Bhagavatula, Chenchen Zhu, Khoa Luu, and Marios Savvides. Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3980–3989, 2017. 2

[3] Ben Cafardo. 'espn virtual 3' technology to debut on nba saturday primetime on abc, Jan 2016. 1

[4] Peter Carr, Yaser Sheikh, and Iain Matthews. Point-less calibration: Camera parameters from gradient-based alignment to edge images. In *2012 IEEE Workshop on the Applications of Computer Vision (WACV)*, pages 377–384. IEEE, 2012. 1, 2

[5] Jianhui Chen and James J Little. Sports camera calibration via synthetic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2, 7, 8

[6] Jianhui Chen, Fangrui Zhu, and James J Little. A two-point method for ptz camera calibration in sports. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 287–295. IEEE, 2018. 1, 2

[7] Olivier Faugeras and OLIVIER AUTOR FAUGERAS. *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993. 4

[8] Bernard Ghanem, Tianzhu Zhang, Narendra Ahuja, et al. Robust video registration applied to field-sports video analysis. 2012. 1, 2

[9] Ankur Gupta, James J Little, and Robert J Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *2011 Canadian Conference on Computer and Robot Vision*, pages 32–39. IEEE, 2011. 1, 2

[10] Jean-Bernard Hayet and Justus Piater. On-line rectification of sport sequences with moving cameras. In *Mexican International Conference on Artificial Intelligence*, pages 736–746. Springer, 2007. 2

[11] Jean-Bernard Hayet, Justus Piater, and Jacques Verly. Robust incremental rectification of sports video sequences. In *British Machine Vision Conference (BMVC'04)*, pages 687–696. Citeseer, 2004. 2

[12] Robin Hess and Alan Fern. Improved video registration using non-distinctive local image features. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[13] Jennifer Hobbs, Paul Power, Long Sha, and Patrick Lucey. Quantifying the value of transitions in soccer via spatiotemporal trajectory clustering. MIT Sloan Sports Analytics Conference, 2018. 2

[14] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. Sports field localization via deep structured models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5212–5220, 2017. 2, 6, 7, 8

[15] Stanley K Honey, Richard H Cavallaro, Jerry Neil Gepner, Edward Gerald Goren, and David Blyth Hill. Method and apparatus for adding a graphic indication of a first down to a live video of a football game, Oct. 31 2000. US Patent 6,141,060. 1

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2

[17] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 4

[18] Yunting Li, Jun Zhang, Wenwen Hu, and Jinwen Tian. Method for pan-tilt camera calibration using single control point. *JOSA A*, 32(1):156–163, 2015. 2

[19] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9455–9464, 2018. 2

[20] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2

[21] Kenji Okuma, James J Little, and David G Lowe. Automatic rectification of long image sequences. In *Asian Conference on Computer Vision*, volume 9, 2004. 2

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 3

[23] Long Sha, Patrick Lucey, Yisong Yue, Xinyu Wei, Jennifer Hobbs, Charlie Rohlf, and Sridha Sridharan. Interactive sports analytics: An intelligent interface for utilizing trajectories for interactive sports play retrieval and analytics. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 25(2):13, 2018. 2

[24] Rahul Anand Sharma, Bharath Bhat, Vineet Gandhi, and CV Jawahar. Automated top view registration of broadcast football videos. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 305–313. IEEE, 2018. 1, 2, 7

[25] SportsLogiq. https://www.sportlogiq.com. 1

[26] StatsPerform. https://www.stats.com/sportvu-football. 1

[27] Graham Thomas. Real-time camera tracking using sports pitch markings. *Journal of Real-Time Image Processing*, 2(2-3):117–132, 2007. 2

[28] Pei-Chih Wen, Wei-Chih Cheng, Yu-Shuen Wang, Hung-Kuo Chu, Nick C Tang, and Hong-Yuan Mark Liao. Court reconstruction for camera calibration in broadcast basketball videos. *IEEE transactions on visualization and computer graphics*, 22(5):1517–1526, 2015. 2

[29] Rui Zeng, Ruan Lakemond, Simon Denman, Sridha Sridharan, Clinton Fookes, and Stuart Morgan. Calibrating cameras in poor-conditioned pitch-based sports games. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1902–1906. IEEE, 2018. 2

[30] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using pro-

grammatic weak supervision. *The International Conference on Learning Representations (ICLR)*, 2018. 2

[31] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000. 4