TVCalib: Camera Calibration for Sports Field Registration in Soccer

https://mm4spa.github.io/tvcalib

Jonas Theiner[1] Ralph Ewerth[1,2]
[1] L3S Research Center, Leibniz University Hannover, Hannover, Germany
[2] TIB – Leibniz Information Centre for Science and Technology, Hannover, Germany
theiner@l3s.de ralph.ewerth@tib.eu

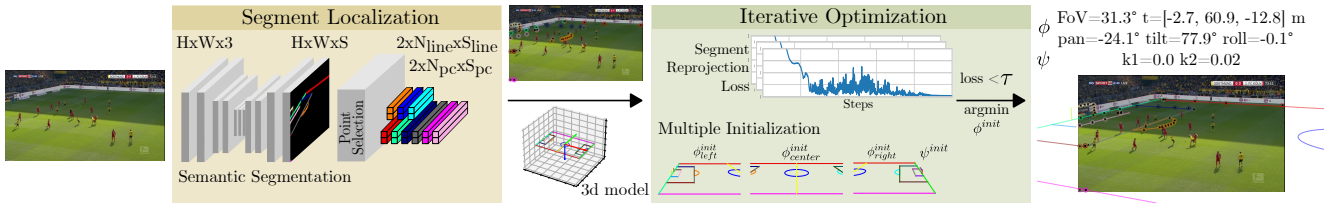

Figure 1: Our proposed framework for 3D sports field registration: (1) *segment localization* performs instance segmentation and selects appropriate points with respective label from a known calibration object (3D model), and (2) our main contribution, the calibration module, which predicts camera parameters ϕ by iteratively minimizing the *segment reprojection loss*.

Abstract

Sports field registration in broadcast videos is typically interpreted as the task of homography estimation, which provides a mapping between a planar field and the corresponding visible area of the image. In contrast to previous approaches, we consider the task as a camera calibration problem. First, we introduce a differentiable objective function that is able to learn the camera pose and focal length from segment correspondences (e.g., lines, point clouds), based on pixel-level annotations for segments of a known calibration object. The calibration module iteratively minimizes the segment reprojection error induced by the estimated camera parameters. Second, we propose a novel approach for 3D sports field registration from broadcast soccer images. Compared to the typical solution, which subsequently refines an initial estimation, our solution does it in one step. The proposed method is evaluated for sports field registration on two datasets and achieves superior results compared to two state-of-the-art approaches.

1. Introduction

Camera calibration is fundamental for numerous computer vision applications such as tracking, autonomous driving, robotics, augmented reality, etc. Existing literature has extensively studied this problem for fully calibrated, partially calibrated, and uncalibrated cameras in various settings [22], for different types of data (e.g. monocular images, image sequences, RGB-D images, etc.), and related tasks like 3D reconstruction. Broadcast videos of sports events are a widely available data source. The ability to calibrate from a single, moving camera with unknown and changing camera parameters enables various augmented reality [15] and sports analytics applications [13, 29].

The sports field serves as a calibration object (known dimensions according to the game rules). However, the non-visibility of appropriate keypoints in broadcast soccer videos [11] and the unknown focal length prevent a sufficiently accurate direct computation of a homography or intrinsics and extrinsics from 2D-3D (keypoint) correspondences [2, 18, 37, 38]. It has been shown that line [20, 27], area [6, 27, 30], point features with additional information [8, 11, 27] are more suitable for accurate sports field registration. Previous approaches [6, 8, 23, 27, 30, 32] treat the task as homography estimation instead of calibration despite the estimation of camera parameters enables further applications (e.g., virtual stadiums, automatic camera control, or offside detection). To date, homography-based approaches may provide camera parameters for a first coarse initial estimation, but the more accurate results are usually based on homography refinements.

In this paper, we suggest to consider sports field registration as a calibration task and estimate individual camera parameters (position, rotation, and focal length) of the stan-

dard pinhole camera model (and potential radial lens distortion coefficients) from an image without relying on keypoint correspondences between the image and 3D scene. Contrary to the dominant direction of first estimating an initial result and then refining it, our method does both in one step without relying on training data for the calibration part. Further, we use a dense representation of the visible field, i.e., directly leverage a small fraction of labeled pixel representing field segments instead of a (deep) image representation for both initial estimation [6, 30] or refinement [6, 8, 11, 23, 27, 30, 32].

We propose (1) a generic differentiable objective function that exploits the underlying primitives of a 3D object and measures its reprojection error. We additionally suggest (2) a novel framework for 3D sports field registration (*TVCalib*) from TV broadcast frames (Fig. 1), including semantic segmentation, point selection, the calibration module, and result verification, where the calibration module iteratively minimizes the *segment reprojection loss*. The effectiveness of our method is evaluated on two real-world soccer broadcast datasets (*SoccerNet-Calibration* [10] and *World Cup 2014* (WC14) [20]), and we compare to state of the art in 2D sports field registration.

The rest of the paper is organized as follows. Sec. 2 provides an overview on 2D sports field registration and the related calibration task. In Sec. 3, we describe the proposed *TVCalib* in detail. Experimental results and a comparison with the state of the art are reported in Sec. 4, while Sec. 5 concludes the paper and outlines areas of future work.

2. Related Work on Sports Field Registration

Common to most approaches for sports field registration is that they predict homography matrices from main broadcast videos in team sports while the focus is on soccer. Early approaches rely on local feature matching in combination with Direct Linear Transform (DLT) for homography estimation [5, 16, 17, 28], and both line and ellipse features are already used (e.g., [17, 20, 27, 30]). More recent approaches rely on learning a representation of the visible sports field by performing different variants of semantic segmentation. Approaches directly predict or regress an initial homography matrix [8, 23, 27, 32] or search for the best matching homography in a reference database [6, 30, 31, 36] containing synthetic images with known homography matrices or camera parameters. This estimation is called initial estimation $\hat{\boldsymbol{H}}_{init}$ which is subsequently refined by the majority of approaches and considered as the relative (non-)affine image transformation $\hat{\boldsymbol{H}}_{rel}$ between the segmented input image and the predicted or retrieved image, finally resulting in $\hat{\boldsymbol{H}} = \hat{\boldsymbol{H}}_{init}\hat{\boldsymbol{H}}_{rel} \in \mathbb{R}^{3\times3}$.

Next, we review existing approaches regarding segmentation, initial estimation, refinement, and finally discuss how to access camera parameters.

Semantic Segmentation: Some approaches use handcrafted methods to detect lines, edges, ellipses, vanishing points (lines) or to perform area segmentation (see [12, 19] for an overview). Convolutional Neural Networks with increased receptive field (e.g., via dilated convolutions [7] or non-local blocks [35]) are used perform various types of image segmentation tasks, e.g., keypoint prediction, line segmentation, or area masking. Chen and Little [6] first remove the background and then predict a binary mask representing all field markings. Homayounfar et al. [20] predict points from specific line and circle segments. Other approaches segment the sports field into four different areas [30], or detect appropriate field keypoints and player positions [11]. Nie et al. [27] aim to learn a strong field representation by jointly predicting uniformly sampled grid points, line features, and area features. Inspired by predicting a dense grid of points [27], Chu et al. [8] formulate the task as an instance segmentation problem. We also apply instance segmentation [8] but on all individual field segments.

Initial Estimation: A grid of uniformly sampled and predicted points [8, 27] or predicted keypoints [11, 12] is the input for DLT (and variants) [18] to get usually a rough initial homography estimation. Segmented [23] or raw [32] images are used to directly predict the homography or to regress four points. Still, such approaches require annotated homography matrices for training [27]. Sharma et al. [31] develop a large synthetic dataset of camera poses, whereby Chen and Little [6] train a Siamese network to learn a representation of the respective segmentation mask and retrieve the nearest neighbor given an input mask. Sha et al. [30] use a much smaller database and consequently leave the refinement module to perform large non-affine transformations to the semantic input image.

Homography Refinement: Homography refinement is a crucial step in order to obtain a more accurate estimate, if necessary [8]. Previous approaches [6, 36] use the Lucas-Kanade algorithm [3], also in combination with spatial pyramids [16] with the assumption that the image transformation is small. To handle large non-affine transformations, the Spatial Transformer Network (STN) [21] was introduced in sports field registration. Refinement is performed during one feed-forward step [30] or by iteratively minimizing the difference between the input image and the initial estimation [23, 27].

Accessing Individual Camera Parameters: Carr et al. [5] leverage a gradient-based image alignment algorithm to estimate camera and lens distortion parameters, but the refinement is performed on the homography. A database of synthetic templates [6, 30] allows for direct access to the

camera pose as projective geometry is used to create template images. However, the smaller the database, the larger the reprojection error is without a refinement step. Despite the focus on homographies, it allows us to access individual camera parameters, at least with homography decomposition [11, 18]. Citraro et al. [11] decompose the initial estimated homography matrix to achieve temporal consistency and also apply a *PoseNet* [24] to regress translation and quaternion vectors.

3. *TVCalib*: Keypoint-less Calibration

After modeling the calibration object and camera model (Sec. 3.1), we propose the differentiable objective function (Sec. 3.2) that aims to approximate individual camera parameters given segment correspondences by iteratively minimizing the *segment reprojection loss* in 2D image space. Finally, we introduce its direct application, the 3D sports field registration (Sec. 3.3) and required segment localization (Sec. 3.4). The main workflow is summarized in Fig. 1.

3.1. Calibration Object & Camera Model

Given a calibration object (with known dimensions) that can be divided into individual labeled sub-objects of fundamental primitives (in this paper called *segments*) like *points*, *lines*, or *point clouds*, the aim is to predict the underlying camera parameters ϕ and potential lens distortion coefficients ψ that minimize its reprojection error.

Modeling the Calibration Object: Line segments are defined in the parametric form $s_{line} = \{X_0 + \lambda X_1 | \lambda \in [0, 1]\}$ and point cloud segments as $s_{pc} = \{X_j \in \mathbb{R}^3 | j = 1, \ldots, |s_{pc}|\}$. Without loss of generality, we define a labeled point segment as $s_{point} = X \in \mathbb{R}^3$, resulting in the traditional Perspective-n-Point (PnP) formulation where 2D-3D point correspondences are given. Finally, the calibration object is the composition of all individual segments per segment category $\mathcal{C}$: $\mathbb{S} = \bigcup_{\mathcal{C} \in \{\text{point, line, pc}\}} \{s_{\mathcal{C}}^{(1)}, s_{\mathcal{C}}^{(2)}, \ldots\}$

Modeling the Soccer Field: A soccer field is composed of lines and circle segments (modeled as point clouds), representing all field markings, goal posts, and crossbars. Please note that keypoint correspondences are not directly used in our approach, since all potential visible keypoints are part of line segments. Nevertheless, we do not intend to exclude the possible explicit use of them here beforehand. We follow the segment definitions of Cioppa et al. [10], but modify the *central circle* and split it into two parts from a heuristic in a post-processing step after semantic segmentation to induce context information. In case of a vertically oriented *middle line*, all points of the *central circle* that lie on the left are assigned to a sub-segment *left*, otherwise they are assigned to the sub-segment *right*.

Modeling the Pinhole Camera: We use the common pinhole camera model $\boldsymbol{P} = \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I}| - \boldsymbol{t}] \in \mathbb{R}^{3\times4}$ parameterized with the intrinsics $\boldsymbol{K} \in \mathbb{R}^{3\times3}$, which define the transformation from camera coordinates to image coordinates, and extrinsics $\left[\boldsymbol{R} \in \mathbb{R}^{3\times3}, \boldsymbol{t} \in \mathbb{R}^3\right]$, defining the camera pose transformation from the scene coordinates to the camera coordinates. We assume square pixels, zero skew and set the principal point to the center of the image. Instead of predicting the focal length directly, i.e., the only unknown variable in $\boldsymbol{K}$, we predict the Field of View (*FoV*) and transform the image coordinates to Normalized Device Coordinates (*NDC*) for numerical stability (Appx. A.1). Following Euler's angles convention, the rotation matrix $\boldsymbol{R} = \boldsymbol{R}_z(roll)\boldsymbol{R}_x(tilt)\boldsymbol{R}_z(pan)$ is the composition of individual rotation matrices, encoding the *pan*, *tilt*, and *roll* angles (in radians) of the camera base according to a defined reference axis system. Intrinsics and extrinsics are thus only parameterized by $\phi = (\textit{FoV}, \boldsymbol{t}, pan, tilt, roll)$, and assume that $\pi_\phi : X \mapsto x$ projects any scene coordinate $X \in \mathbb{R}^3$ to its respective image point $x \in \mathbb{R}^2$.

Relation to the Homography Matrix: If $X_z = 0.0$ then $\boldsymbol{P}^{3\times[1,2,4]} = \boldsymbol{K}\,\boldsymbol{R}^{3\times[1,2]}[\boldsymbol{I}| - \boldsymbol{t}] = \boldsymbol{H} \in \mathbb{R}^{3\times3}$ is the respective homography matrix only able to map all points lying on one plane. Appx. B describes how to approximate ϕ given a predicted $\hat{\boldsymbol{H}}$ only.

Lens Distortion: As we do not want to restrict to a specific lens distortion model ψ (e.g., Brown [4]), we define $\texttt{distort}_\psi(x)$ that distorts a pixel x and `undistort` for its inverse function. In case lens distortion coefficients are not known *a priori*, we assume that `undistort` is differentiable which enables the possibility to jointly optimize ψ and ϕ.

3.2. Segment Reprojection Loss

Perspective-n-Point (PnP) refers to the problem of estimating the camera pose (extrinsics) from a calibrated camera $\boldsymbol{K}$ given n 2D-3D point correspondences. Geometric solvers for PnP or PnP(f), that also estimate the focal length, approximate the projection matrix $\boldsymbol{P}$ through the geometric or algebraic reprojection error for $argmin_{\boldsymbol{P}}\, d(\boldsymbol{x}, \pi_{\boldsymbol{P}}(\mathbf{X}))$ where $d(x, \hat{x})$ is the Euclidean distance between two pixels. However, accurate correspondences are assumed to be known, the focal length in $\boldsymbol{K}$ needs to be estimated, and there are some further requirements (e.g., minimum number of points, number of points that are allowed to be on one plane, etc.) need to be considered [18].

Instead, we aim to learn the underlying camera parameters ϕ (and potential lens distortion coefficients ψ) by minimizing the Euclidean distance between all reprojected segments and respective annotated (or predicted) pixels (see Sec. 3.4 for segment localization). Our *segment reprojec-*

tion loss is based on the Euclidean distance between annotated pixels with respective segment label and reprojected segments of the calibration object.

Let us consider a sample-dependent number of pixel annotations $\boldsymbol{x}^{(c)} \in \mathbb{R}^{?\times 2}$ for each (visible) segment label $c \in \mathbb{S}$. For a respective line segment $s^{(c)}_{line}$, the perpendicular distance to its respective reprojected line $\hat{s}^{(c)}_{line} = \{\pi_\phi(X^{(c)}_0) + \lambda\pi_\phi(X^{(c)}_1) | \lambda \in \mathbb{R}\}$ can be computed for each $p \in \boldsymbol{x}^{(c)}$:

$$d(p, \hat{s}_{line}) = \frac{|det((\pi_\phi(X_1) - \pi_\phi(X_0)); (\pi_\phi(X_0) - p))|}{|\pi_\phi(X_1) - \pi_\phi(X_0)|} \tag{1}$$

and hence describes the point-line distance. The distance between a pixel $p^c \in \mathbb{R}^2$ and its corresponding reprojected point cloud $\hat{s}^c_{pc} = \{\pi_\phi(X_j) | j = 1, \ldots, |s^c_{pc}|\}$ is the minimum Euclidean distance for each $p \in \boldsymbol{x}^{(c)}$. The `mean` distance over all annotated points $\boldsymbol{x}$ is taken to aggregate one segment c. Finally, the *segment reprojection loss function* needs to be minimized where each segment contributes equally:

$$\mathcal{L} := \underset{\phi,\,(\psi)}{argmin} \quad \frac{1}{|\mathbb{S}|} \sum_{c \in \mathbb{S}} d_{\texttt{mean}}(\texttt{undistort}_\psi(\boldsymbol{x}^{(c)}), \pi_\phi(s^{(c)})) \tag{2}$$

Please note that π in Eq. (2) represents the reprojection of an arbitrary segment $\hat{s} = \pi_\phi(s)$ to the image to simplify the notation. Depending on the segment type, point↔point, point↔line, or point↔point-cloud distances are computed. Without lens distortion correction, `undistort` can be considered as identity function.

Implementation details: All computations (image projection and distance calculation) can be performed on tensor operations, which allows for more efficient computation and parallelization. The input dimension of annotated or predicted pixels for each segment category $\mathcal{C}$ (e.g., lines) is $\hat{\mathbf{x}}_\mathcal{C} \in \mathbb{R}^{T\times S_\mathcal{C}\times N_\mathcal{C}\times 2}$, where $N_\mathcal{C}$ represents the number of selected pixels ($N_{\text{keypoint}} = 1$), $S_\mathcal{C}$ is the number of segments for the specific segment category, and T is an optional batch or temporal dimension. However, we need to pad the input if the number of provided pixels per segment differ, and remember its binary *padding mask* $\mathbf{m}_\mathcal{C} \in \{0,1\}^{T\times S_\mathcal{C}\times N_\mathcal{C}}$. To reproject the 3D object, all points are projected from the following input dimension per segment type $\mathbf{X}_{\text{line}} \in \mathbb{R}^{T\times S_{\text{line}}\times 2\times 3}$, $\mathbf{X}_{\text{pc}} \in \mathbb{R}^{T\times S_{\text{pc}}\times N^*_{pc}\times 3}$, and $\mathbf{X}_{\text{keypoint}} \in \mathbb{R}^{T\times S_{\text{keypoint}}\times 1\times 3}$ where N^*_{pc} is the number of sampled 3D points for each point cloud. After distance calculation for each segment type, the distance of padded input pixels are set to zero according to the *padding mask* of each segment category $\mathbf{m}_\mathcal{C}$, implying that the distance of non-visible segments is also set to zero. Aggregating the S and N dimension via `sum` and dividing by the number of actually provided pixels of the input is equivalent to Eq. (2), where each segment contributes equally.

3.3. Gradient-based Iterative Optimization

Given human annotations or a model (Sec. 3.4) that predicts pixel positions with corresponding segment label, one way is to directly optimize the proposed objective function (Eq. (2)) via gradient descent.

Initialization: We do not further encode the camera parameters nor modify the modeled pinhole camera (Sec. 3.1), but rather aim to predict all unknown variables $\phi = \{FoV, pan, tilt, roll, \boldsymbol{t}\}$ in a direct manner. However, it is beneficial to initialize an optimizer with an appropriate set of parameters. We introduce some prior information restricting possible camera ranges. Raw camera parameters are standardized to a zero mean and provided standard deviation. For uniformly distributed camera ranges $\mathcal{U}(a, b)$, we transform to a normal distribution $\mathcal{N}(\mu, \sigma)$, so that σ covers the 95% confidence interval, given $\mu = a + (b - a)/2$ and finally initialize with zeros. Roughly speaking, this initialization corresponds to the mean image, e.g., a central view of the calibration object.

Multiple Initialization: In case there is a large variance for some parameter, for instance, the camera location, it is reasonable to provide multiple sets of camera distributions. Suppose this information is *a priori*, for instance, the main broadcast camera. In that case, a user can select the correct set, or this information is known from shot boundary and shot type classification (later denoted as `stacked`). Otherwise, we propose to run the optimization with multiple candidates and the best result is taken automatically by selecting the one with minimum loss (`argmin`) according to Eq. (2).

Self-Verification: Self-verification aims to identify all images in which the model is unable to calibrate or estimate the homography. While other approaches use the mean point reprojection error (e.g., [27]) or verify geometrical constraints [11], we can directly reject all samples whose loss (Sec. 3.2) is below a threshold $\tau \in \mathbb{R}^+$. This user-defined threshold controls the trade-off between accuracy and completeness ratio and can be found empirically, e.g., by taking the best global result on a target metric for a dataset. This procedure might be necessary for invalid input images, e.g., out of camera distribution, erroneous semantic segmentation, or internal errors during optimization such as local minima.

3.4. Segment Localization & Point Selection

The output of any model for the segment localization which provides pixel annotations for each visible segment

given a raw input image can serve as input for the calibration module as well as manual annotations. We use the *DeepLabV3 ResNet* [7] (Residual Networks) to perform instance segmentation for each visible line or circle segment and do not directly predict appropriate pixels per segment. Pixel selection is then a post-processing step, aiming to select, for instance, at least two points for a line segment with maximum distance, best representing a line where we follow a non-differentiable implementation [26]. Ideal lines are sufficiently represented by two points, however, we have noticed more stable gradients if more than two points are selected. Further, we want to allow potential for lens distortion correction based on the extracted points which may show a curved polyline.

4. Experiments

The experimental setup including the baselines, metrics, datasets, and hyperparameters is introduced in Sec. 4.1. The results and comparisons to the state of the art are presented in Sec. 4.2. We conduct ablation studies for the proposed (1) segment localization, (2) self-verification, (3) multiple camera initialization, and (4) lens distortion (Sec. 4.3), while limitations are discussed in Sec. 4.4.

4.1. Experimental Setup

4.1.1 Baselines & State of the Art

Team sports such as soccer are played on an approximately planar field, hence many approaches assume a 2D area and use homography estimation [8, 27, 30, 32] to map all segments lying on this plane. To additionally estimate the camera pose and focal length, a reasonable approach is therefore the homography decomposition (see Appx. B for details) denoted as HDecomp.

Since in TV broadcasts of games like soccer or basketball, individual field segments are primarily visible, rather than keypoints, a suitable baseline is homography estimation via DLT from line segments [26]. Further, we compare to Chen and Little [6] for homography estimation. As their retrieval and refinement module solely relies on synthetic data, we can test different variants for camera parameter distributions during training [34]. For a fair comparison, we neglect the impact of the original segment localization by using ground-truth masks generated from the SN-Calib annotations or use the predicted masks from our segmentation model. As a second approach, we apply the official implementation from Jiang et al. [23]. Jiang et al. [23] and other recent approaches [8, 27, 30] rely on annotated homography matrices for training.

Table 1: Dataset comparison regarding camera type distribution, number of images, and resolution. The values labeled with * are approximated from 100 images since our calibration module does not require training data.

Dataset	Split	Images	Reso.	Camera Type Distr. [%] Center	Left	Right	Other
SN-Calib	train	14513	540p	*48.0	*14.0	*15.0	*23.0
	valid	2796	540p	52.7	10.2	9.5	27.7
	test	2719	540p	53.5	8.5	9.5	28.5
WC14	train/valid	209	720p	100.	0.0	0.0	0.0
	test	186	720p	100.	0.0	0.0	0.0

4.1.2 Datasets

SN-Calib dataset: The SoccerNetV3-Calibration dataset [10] consists of 20 028 images taken from the SoccerNet [14] videos (500 matches) and covers more camera locations in addition to the main broadcast camera. An example setting may consist of two cameras that are placed also on the same tribune as the central broadcast camera, but are closer located to the side lines (main camera left and right). In addition, there are other cameras, e.g., *behind the goal* and *inside the goal*, or *above the field* (spider cam). We have manually annotated these camera locations used in this paper to get an overview. Table 1 summarizes the camera type distribution and number of images per split (train, validation, test) without stadium overlap. Cioppa et al. provide annotation for all segments of the soccer field [10], i.e., lines, circle segments, and goal posts. Each visible segment has at least two annotated positions optimally representing the segment (i.e., corner and border points) in form of a polyline.

WC14 dataset: The WC14 dataset [20] is the traditional benchmark for sports field registration in soccer and contains images from broadcast TV videos (only central main camera without large zoom) from the FIFA World Cup 2014 and the corresponding manually annotated homography matrices. We have additionally annotated the segments in the test split according to the guidelines in SN-Calib [10].

4.1.3 Metrics

The quality of estimated camera parameters or homography matrices can be evaluated both at 2D image space by measuring a *reprojection error*, and in world space by measuring a *projection error*.

Accuracy@threshold [26]: The evaluation is based on the distance of the reprojection of each soccer field segment and the corresponding annotated polyline. Segments are reprojected from the predicted camera parameters ϕ (and ψ) to the image from dense sampled points of the 3D model resulting in one polyline for each segment. A polyline

corresponding to a soccer field segment s is detected as a true positive (TP), if the Euclidean distance between **every** point of the annotated polyline of segment $\tilde{s}$ and the reprojected polyline $\pi_\phi(s)$ is less than t pixels: $\forall p \in \tilde{s} : d(p, \pi_\phi(s)) < t$. If the distance of one annotated point to its corresponding projected polyline is greater than t pixels, this segment is counted as a false positive (FP), along with the projected polyline that does not appear in the annotations. Segments that are only present in the annotations are counted as false negatives (FN). Finally, the accuracy for a threshold of $t \in \{5, 10, 20\}$ pixels is given by: $AC@t = TP/(TP+FN+FP)$. If the camera calibration or the homography estimation may fail for some images, the **Completeness Ratio** (CR) measures the number of provided parameters divided by the number of images of the dataset. **Compound Score** (CS): To summarize the above four scores, they are weighted as follows [26]:

$$CS := (1 - e^{-4CR}) \quad (\sum_{t \in [5,10,20], w \in [0.5,0.35,0.15]} wAC@t) \tag{3}$$

Intersection over Union (IoU) [20]: The accuracy for homography estimation for sports fields is traditionally evaluated on the IoU_{part} and IoU_{whole} metrics that measure the projection error. They calculate the binary IoU of the projected templates from predicted homography and a ground-truth homography in world (top view / bird view) space for the visible area (*part*) and the full (*whole*) area of the sports field, respectively. Due to the absence of ground-truth information like camera parameters, the evaluation can only be performed given *annotated* homography matrices [20] that are obtained from the visible sports field in the image (e.g., via DLT). Hence, projection correctness can be guaranteed only for the visible area and we prefer the usage of IoU_{part} similar to Nie et al. [27].

4.1.4 Hyperparameters

Optimization: We use AdamW [25] with a learning rate of 0.05 and weight decay of 0.01 to optimize the camera parameters ϕ for 2000 steps using the one-cycle learning rate scheduling [33] with $pct_{start} = 0.5$. These parameters were found on the SN-Calib-valid split through a visual exploration of qualitative examples. **Calibration Object & Camera Parameter Distribution:** Furthermore, we set the number of sampled points for each point cloud to $N^*_{pc} = 128$ ($0.45\,m$ point density for the central circle). We use a very coarse camera distribution (see Appx. A.2) of the main camera center and apply it to all datasets. **Segment Localization:** The training data are derived from the provided annotations of the SN-Calib-train dataset. For training details we refer to Appx. C. Please recall that the expected dimension for each segment category $\mathcal{C}$ is $\hat{\mathbf{x}}_\mathcal{C} \in \mathbb{R}^{T \times S_\mathcal{C} \times N_\mathcal{C} \times 2}$. We set $|N_{line}| = 4$ and $|N_{pc}| = 8$ following initial considerations (Sec. 3.4) which is in general in line with the number of annotated points per segment in SN-Calib.

Self-Verification: We set the parameter $\tau = 0.019$ (Sec. 3.3) globally for all experiments based on the maximum *CS* on SN-Calib-valid-center using the predicted segment localization (from $\tau \in [0.013; 0.025]$ with a step size of 10^{-3}; see Fig. 2 for visual verification).

4.2. Results & Comparison to State of the Art

Previous approaches focus on the (1) main camera center and (2) homography estimation. Hence, we (1) compare on the subset of SN-Calib-test and (2) measure both the camera calibration performance induced by the predicted camera parameters and the homography estimation.

Reprojection Error for Camera Calibration: This task represents the main task of estimating individual camera parameters ϕ where the reprojection error (*AC@t*) induced by ϕ is evaluated. The results on the test splits on SN-Calib-center and WC14-test are presented in Table 2 (top) and Table 3, respectively.

Pred vs. Ground Truth (GT) Segmentation: If the same ground-truth segmentation is used as input, our method outperforms the best variant from Chen and Little [6] ($\mathcal{U}_{FoV}$+$\mathcal{U}_{xyz}$ [34]) and the baseline on both datasets.

Self-Verification: The homography decomposition also contains a kind of self-verification resulting in a higher reprojection accuracy (AC@t) but lower completeness ratio (CR), as shown in Tables 2 and 3. Hence, we can compare these approaches with our results after self-verification of *TVCalib*. Superior results are achieved for all variants of segmentation and on both datasets.

Reprojection Error for Homography Estimation: To investigate whether the quality of the homography estimation or the decomposition are the reason for the results, we examine the plain performance of the homography estimation and thus exclude the impact of the homography decomposition. We measure the same metrics, but only map all segments lying on one plane, i.e., ignore goal posts and crossbars. The results for the estimated as well as the ground-truth homography matrices are presented in Table 2 (bottom) and Table 4.

Influence of the Homography Decomposition: Compared to the reprojection error for the calibration task, noticeably better results are achieved indicating that the decomposition introduces additional errors. Based on the per-segment accuracy, we found that in particular a larger projection error is frequently visible for goal segments since the height information is missing but not the only reason for higher errors (e.g., DLT Lines with and without HDecomp).

Pred vs. GT Segmentation: Similar to the evaluation of camera calibration performance, superior results are

Table 2: Results on SN-Calib-test-center only evaluating where the main camera center is shown (1454 images): When evaluating the homography, all segments not lying one the plane (goal posts and crossbars) are ignored.

Calibration	Seg.	AC@ [%] 5	10	20	CR	CS
Evaluating the Camera Calibration ($\hat{\phi}$)						
TVCalib(τ)	GT	68.7	88.0	96.1	92.8	76.9
TVCalib	GT	65.3	84.2	92.6	100.0	75.5
HDecomp + [6] ($\mathcal{U}_{FoV}$+$\mathcal{U}_{xyz}$)	GT	53.7	77.5	88.4	80.3	65.1
HDecomp + DLT Lines	GT	48.1	68.5	84.6	79.8	60.2
TVCalib(τ)	Pred	57.6	81.7	93.2	93.7	72.6
TVCalib	Pred	54.8	78.5	90.4	100.0	71.4
HDecomp + DLT Lines	Pred	40.6	63.2	80.4	79.6	55.9
HDecomp + [6] ($\mathcal{U}_{FoV}$+$\mathcal{U}_{xyz}$)	Pred	34.4	64.6	81.3	66.6	52.0
Evaluating the Homography Estimation $\hat{H}$						
TVCalib(τ)	GT	65.0	85.4	95.6	92.8	75.5
TVCalib	GT	61.7	81.6	92.0	100.0	73.9
[6] ($\mathcal{U}_{FoV}$+$\mathcal{U}_{xyz}$)	GT	57.3	76.0	83.7	100.0	68.0
HDecomp + [6] ($\mathcal{U}_{FoV}$+$\mathcal{U}_{xyz}$)	GT	61.1	81.2	89.4	80.3	67.5
DLT Lines	GT	54.7	69.9	81.6	97.6	64.4
HDecomp + DLT Lines	GT	56.5	74.3	86.3	79.8	63.6
TVCalib(τ)	Pred	54.6	78.3	92.4	93.7	70.8
TVCalib	Pred	51.9	75.2	89.4	100.0	69.5
DLT Lines	Pred	46.9	66.5	79.3	97.9	61.3
HDecomp + DLT Lines	Pred	46.5	68.5	83.0	79.6	59.2
[6] ($\mathcal{U}_{FoV}$+$\mathcal{U}_{xyz}$)	Pred	32.9	59.0	72.5	100.0	54.6
HDecomp + [6] ($\mathcal{U}_{FoV}$+$\mathcal{U}_{xyz}$)	Pred	40.1	68.3	82.3	66.6	54.0

achieved on both datasets with a noticeable drop when using the segment localization model instead of ground truth. The evaluation on the WC14 dataset (Table 4) yields better results when using segment localization from the individual approaches [6, 23] trained on this dataset, but still the *TVCalib* approach outperforms these variants.

Projection Error (IoU): *TVCalib* achieves very similar results compared to the reproduced approaches and other state-of-the-art approaches without performing training or fine-tuning on this dataset. The reprojection error measured via AC@t from the annotated homographies $\boldsymbol{H}$ is comparable with our results (Table 4), but not ideal, demonstrating bias on the IoU metrics since $\boldsymbol{H}$ is used to evaluate the projection error.

4.3. Ablation Studies

Impact of Segment Localization (Pred vs. GT): Because we want to find the upper limit for the performance of our method, we use the provided annotations and compare with the predicted segments from our segment localization model. The lower performance (Tables 2 and 3) when using the predicted segments shows that the *segment localization* module (Sec. 3.4) needs improvement despite the visually similar results for the majority of images (Fig. 4).

Table 3: Evaluating the reprojection error induced by the camera parameters (ϕ) on WC14-test dataset (186 images).

Calibration	Seg.	AC@ [%] 5	10	20	CR	CS
TVCalib	GT	64.4	86.7	96.0	100.0	86.4
HDecomp + [6]	GT	52.8	78.8	91.3	90.9	79.0
HDecomp + $\boldsymbol{H}$ [20]	✗	48.1	78.9	91.5	90.9	78.4
HDecomp + DLT Lines	GT	32.0	54.0	73.1	73.7	57.1
TVCalib	Pred	39.9	71.9	90.5	100.0	75.0
HDecomp + [6] ζ=1k)	[6]	29.0	59.8	79.0	100.0	63.6
HDecomp + [23] (ζ=1k)	[23]	32.4	58.5	75.3	99.5	61.8
TVCalib(τ)	Pred	41.3	73.6	91.4	95.7	76.0
HDecomp + [6]	[6]	32.7	67.3	87.3	81.7	69.4
HDecomp + [23]	[23]	36.9	66.4	83.9	84.9	68.4
HDecomp + [6]	Pred	28.1	60.6	80.8	78.5	63.0
HDecomp + DLT Lines	Pred	26.9	53.3	72.7	74.2	56.0

Table 4: Evaluating the homography estimation on WC14-test: IoU_{part} compares the projection error (top view) using annotated homography matrices ($\boldsymbol{H}$ [20]). Grayed out: Results taken from the respective paper.

Approach	Seg.	AC@ [%] 5	10	20	CR	CS	IoU_{part} mean	med.
TVCalib(τ)	GT	62.7	84.9	95.5	100.0	85.3		
HDecomp + [6]	GT	56.1	80.6	91.1	90.9	80.0		
HDecomp + $\boldsymbol{H}$ [20]	✗	50.6	79.4	91.1	90.9	78.8		
HDecomp + DLT Lines	GT	35.8	57.6	74.2	73.7	59.4		
TVCalib	GT	62.7	84.9	95.5	100.0	85.3	96.1	97.1
$\boldsymbol{H}$ [20]	✗	54.1	82.9	92.4	100.0	81.8	100.	100.
Chen and Little [6]	GT	61.2	82.5	90.6	100.0	81.8	95.2	97.3
DLT Lines	GT	39.2	57.4	72.1	89.8	60.3	82.6	96.5
TVCalib	Pred	38.8	69.1	89.4	100.0	73.3	95.3	96.6
Chen and Little [6]	[6]	35.8	66.3	84.4	100.0	69.5	94.6	96.3
Jiang et al. [23]	[23]	36.9	62.9	81.5	100.0	67.1	95.2	97.1
Chen and Little [6]	Pred	28.8	58.0	77.3	100.0	62.1	91.7	94.9
DLT Lines	Pred	31.4	55.9	71.9	87.6	58.4	83.7	95.4
Cioppa et al. [9]	[9]	✗	✗	✗	100.	✗	88.5	92.3
Sha et al. [30]	[30]	✗	✗	✗	100.	✗	93.2	96.1
Chu et al. [8]	[8]	✗	✗	✗	100.	✗	96.0	97.0
Shi et al. [32]	[32]	✗	✗	✗	100.	✗	96.6	97.8

Choice of the Self-Verification Parameter: Please recall that τ is a user-defined threshold able to reject images based on the *reprojection loss*. For simplicity, we have set this value once globally based on the maximum *CS* on SN-Calib-valid-center (predicted segment localization), but the optimal value can be chosen for each dataset and configuration individually or specified manually. This value is roughly valid across multiple datasets, camera distributions, and splits (see Fig. 2). The projection performance is shown in Fig. 3 for multiple configurations of *TVCalib* by varying this parameter. In general, the more τ is restricted, the less the completeness ratio decreases, with increasing accuracy that at some point saturates.

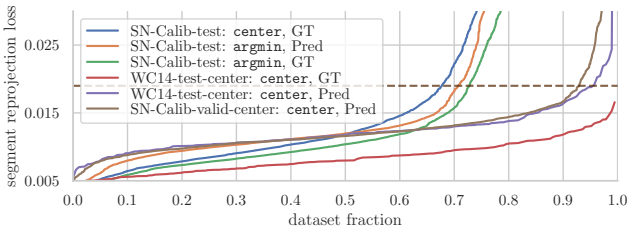

Figure 2: *Segment reprojection loss* per sample for several dataset splits and configurations.

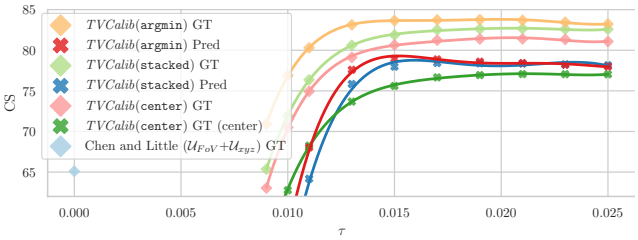

Figure 3: Aggregated results on SN-Calib-test (all) for the calibration task: Different variants of *TVCalib* are shown for several self-verification thresholds τ.

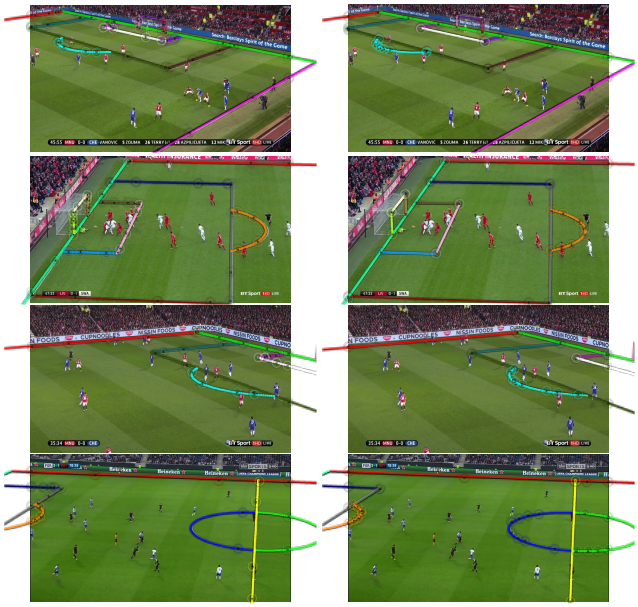

Figure 4: Random samples for *TVCalib* (`argmin`) on SN-Calib-test using predicted (left) and GT (right) segments.

Multiple Initialization: As our solution aims to optimize the camera parameters for multiple camera locations (`center`, `left`, `right`), (1) the question arises whether one initialization (`center`) is sufficient or multiple initialization (one per camera location) are preferred, and (2), if the camera position is known *a priori*, one variant is to use only the respective initialization and for this experiment to stack the results (`stacked`). The other variant utilizes the optimization from multiple initializations and takes the best result (`argmin`). As shown in Fig. 3, initializing from three camera positions (`argmin` and `stacked`) is noticeably better than using only one initialization (`center`), and selecting the best result (`argmin`) is slightly better than knowing the camera type in advance (`stacked`). Due to the iterative optimization process, the ability to start from several locations enables the chance to find better minima.

Lens Distortion: The results when camera and radial lens distortion parameters were learned jointly are presented and discussed in Appx. D. In summary, results can be improved at *AC@5* for samples where radial lens distortion is visible.

4.4. Limitations

Despite strong results, for a small fraction of given ground-truth segment annotations, some samples are rejected. This is mainly caused by local minima due to the nature of gradient-based iterative optimization [1]. Related to the camera initialization, we have not investigated any cameras other than those on the main tribune. The *TVCalib* approach relies on an accurate segment localization, but no regularization term is included that allows for outliers. Finally, jointly learning lens distortion coefficients has not been deeply investigated.

5. Conclusions

We have presented an effective solution to learn individual camera parameters from a calibration object that is modeled by point, line, and point cloud segments. Furthermore, we have successfully demonstrated its direct application to 3D sports field registration in soccer broadcast videos. In the target task of 3D as well as for 2D sports field registration, our method has achieved superior results compared to two state-of-the-art approaches [6, 23] for 2D sports field registration in terms of the image reprojection error.

Future work could investigate the integration of temporal consistency and associated speedup, the application to other sports, and finally the incorporation into a deep neural network to estimate the camera parameters in one feed-forward step or full end-to-end learning.

Acknowledgement

Thanks to Wolfgang Gritz and Eric Müller-Budack for reviewing this paper, Jim Rhotert for the segmentation module and Markos Stamatakis for enriching the WC14 dataset. This project has received funding from the German Federal Ministry of Education and Research (BMBF – Bundesministerium für Bildung und Forschung) under 01IS20021B.

References

[1] Raul Acuna and Volker Willert. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint*, abs:1803.03025, 2018. URL http://arxiv.org/abs/1803.03025.

[2] Nabih M Alem, John W Melvin, and Garry L Holstein. Biomechanics applications of direct linear transformation in close-range photogrammetry, 1978. URL https://doi.org/10.1016/b978-0-08-022678-1.50056-4.

[3] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision, IJCV*, 56(3):221–255, 2004.

[4] Duane C Brown. Decentering distortion of lenses. *Photogrammetric Engineering and Remote Sensing*, 1966.

[5] Peter Carr, Yaser Sheikh, and Iain A. Matthews. Point-less calibration: Camera parameters from gradient-based alignment to edge images. In *Workshop on Applications of Computer Vision, WACV*, pages 377–384. IEEE Computer Society, 2012. URL https://doi.org/10.1109/WACV.2012.6163012.

[6] Jianhui Chen and James J Little. Sports camera calibration via synthetic data. In *Conference on Computer Vision and Pattern Recognition Workshops, CVPRW*. CVF/IEEE, 2019.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. URL http://arxiv.org/abs/1706.05587.

[8] Yen-Jui Chu, Jheng-Wei Su, Kai-Wen Hsiao, Chi-Yu Lien, Shu-Ho Fan, Min-Chun Hu, Ruen-Rone Lee, Chih-Yuan Yao, and Hung-Kuo Chu. Sports field registration via keypoints-aware label condition. In *Conference on Computer Vision and Pattern Recognition Workshops, CVPRW*, pages 3523–3530. IEEE/CVF, 2022. URL https://doi.org/10.1109/CVPRW56347.2022.00396.

[9] Anthony Cioppa, Adrien Deliege, Floriane Magera, Silvio Giancola, Olivier Barnich, Bernard Ghanem, and Marc Van Droogenbroeck. Camera calibration and player localization in soccernet-v2 and investigation of their representations for action spotting. In *Conference on Computer Vision and Pattern Recognition Workshops, CVPRW*, pages 4537–4546. CVF/IEEE, 2021. URL https://doi.org/10.1109/CVPRW53098.2021.00511.

[10] Anthony Cioppa, Adrien Deliège, Silvio Giancola, Bernard Ghanem, and Marc Van Droogenbroeck. Scaling up soccernet with multi-view spatial localization and re-identification. *Scientific Data*, 9(1):1–9, 2022.

[11] Leonardo Citraro, Pablo Márquez-Neila, Stefano Savare, Vivek Jayaram, Charles Dubout, Félix Renaut, Andres Hasfura, Horesh Ben Shitrit, and Pascal Fua. Real-time camera pose estimation for sports fields. *Machine Vision and Applications*, 31(3):16, 2020. URL https://doi.org/10.1007/s00138-020-01064-7.

[12] Carlos Cuevas, Daniel Quilon, and Narciso García. Automatic soccer field of play registration. *Pattern Recognition*, 103:107278, 2020. URL https://doi.org/10.1016/j.patcog.2020.107278.

[13] Carlos Cuevas, Daniel Quilón, and Narciso García. Techniques and applications for soccer video analysis: A survey. *Multimedia Tools and Applications*, 79(39):29685–29721, 2020. URL https://doi.org/10.1007/s11042-020-09409-0.

[14] Adrien Deliege, Anthony Cioppa, Silvio Giancola, Meisam J Seikavandi, Jacob V Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B Moeslund, and Marc Van Droogenbroeck. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *Conference on Computer Vision and Pattern Recognition Workshops, CVPRW*, pages 4508–4519. IEEE/CVF, 2021. URL https://doi.org/10.1109/CVPRW53098.2021.00508.

[15] Tiziana D'Orazio and Marco Leo. A review of vision-based systems for soccer video analysis. *Pattern recognition*, 43(8):2911–2926, 2010. URL https://doi.org/10.1016/j.patcog.2010.03.009.

[16] B Ghanem, T Zhang, and N Ahuja. Robust video registration applied to field-sports video analysis. *International conference on acoustics, speech, and signal processing, ICASSP*, 2012.

[17] Ankur Gupta, James J. Little, and Robert J. Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *Canadian Conference on Computer and Robot Vision, CRV*, pages 32–39. IEEE Computer Society, 2011. URL https://doi.org/10.1109/CRV.2011.12.

[18] Andrew Harltey and Andrew Zisserman. *Multiple view geometry in computer vision (2. ed.)*. Cambridge University Press, 2006. ISBN 978-0-521-54051-3.

[19] J-B Hayet, Justus H Piater, and Jacques G Verly. Fast 2d model-to-image registration using vanishing points for sports video analysis. In *International Conference on Image Processing, ICIP*. IEEE, 2005.

[20] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun. Sports field localization via deep structured models. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4012–4020. IEEE Computer Society, 2017. URL http://doi.ieeecomputersociety.org/10.1109/CVPR.2017.427.

[21] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *Advances in Neural Information Processing System, NIPS*, pages 2017–2025, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/33ceb07bf4eeb3da587e268d663aba1a-Abstract.html.

[22] Yoonwoo Jeong, Seokjun Ahn, Christopher Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Self-calibrating neural radiance fields. In *International Conference on Computer Vision, ICCV, 2021*, pages 5846–5854, 2021. URL https://doi.org/10.1109/ICCV48922.2021.00579.

[23] Wei Jiang, Juan Camilo Gamboa Higuera, Baptiste Angles, Weiwei Sun, Mehrsan Javan, and Kwang Moo Yi. Optimizing through learned errors for accurate sports field registration. In *Winter Conference on Applications of Computer Vision, WACV*, pages 201–210. IEEE, 2020. URL https://doi.org/10.1109/WACV45572.2020.9093581.

[24] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *International Conference on Computer Vision, ICCV*, pages 2938–2946. IEEE Computer Society, 2015. URL https://doi.org/10.1109/ICCV.2015.336.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[26] Floriane Magera, Anthony Cioppa, and Silvio Giancola. SoccerNet Pitch Element Localization and Camera Calibration Challenge. https://github.com/SoccerNet/sn-calibration, 2022. [Online; accessed 01-June-2022].

[27] Xiaohan Nie, Shixing Chen, and Raffay Hamid. A robust and efficient framework for sports-field registration. In *Winter Conference on Applications of Computer Vision, WACV*, pages 1935–1943. IEEE, 2021. URL https://doi.org/10.1109/WACV48630.2021.00198.

[28] Jens Puwein, Remo Ziegler, Julia Vogel, and Marc Pollefeys. Robust multi-view camera calibration for wide-baseline camera networks. In *Workshop on Applications of Computer Vision, WACV*, pages 321–328. IEEE, 2011. URL https://doi.org/10.1109/WACV.2011.5711521.

[29] 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 Human-Computer Interaction*, 25(2), 2018. URL https://doi.org/10.1145/3185596.

[30] Long Sha, Jennifer A. Hobbs, Panna Felsen, Xinyu Wei, Patrick Lucey, and Sujoy Ganguly. End-to-end camera calibration for broadcast videos. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 13624–13633. IEEE/CVF, 2020. URL https://doi.org/10.1109/CVPR42600.2020.01364.

[31] Rahul Anand Sharma, Bharath Bhat, Vineet Gandhi, and C. V. Jawahar. Automated top view registration of broadcast football videos. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV*, pages 305–313. IEEE Computer Society, 2018. URL https://doi.org/10.1109/WACV.2018.00040.

[32] Feng Shi, Paul Marchwica, Juan Camilo Gamboa Higuera, Mike Jamieson, Mehrsan Javan, and Parthipan Siva. Self-supervised shape alignment for sports field registration. In *Winter Conference on Applications of Computer Vision, WACV*, pages 3768–3777. IEEE, 2022. URL https://doi.org/10.1109/WACV51458.2022.00382.

[33] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, pages 369–386. SPIE, 2019.

[34] Jonas Theiner, Wolfgang Gritz, Eric Müller-Budack, Robert Rein, Daniel Memmert, and Ralph Ewerth. Extraction of positional player data from broadcast soccer videos. In *Winter Conference on Applications of Computer Vision, WACV*, pages 1463–1473. IEEE, 2022. URL https://doi.org/10.1109/WACV51458.2022.00153.

[35] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 7794–7803. Computer Vision Foundation / IEEE Computer Society, 2018. URL http://doi.org/10.1109/CVPR.2018.00813.

[36] Neng Zhang and Ebroul Izquierdo. A high accuracy camera calibration method for sport videos. In *International Conference on Visual Communications and Image Processing, VCIP*, pages 1–5. IEEE, 2021. URL https://doi.org/10.1109/VCIP53242.2021.9675379.

[37] Zhengyou Zhang. A flexible new technique for camera calibration. *Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. URL https://doi.org/10.1109/34.888718.

[38] Yinqiang Zheng, Shigeki Sugimoto, Imari Sato, and Masatoshi Okutomi. A general and simple method for camera pose and focal length determination. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pages 430–437. IEEE Computer Society, 2014. URL https://doi.org/10.1109/CVPR.2014.62.