Keypoint-less Camera Calibration for Sports Field Registration in Soccer

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

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Our proposed framework for 3D sports field registration: (1) segment localization performs instance segmentation and selects appropriate points from a known 3D object, and (2) our main contribution \textit{TVCalib}, which predicts camera parameters \(\phi\) by iteratively minimizing the segment reprojection loss.}
\end{figure}

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. The calibration module does not require any training data and 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 various computer vision applications such as tracking systems, autonomous driving, robotics, augmented reality, and many more. 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. In the domain of sports, broadcast videos are a widely available data source. The ability to calibrate from a single, moving camera with unknown and changing camera parameters enables a diverse range of augmented reality [15] and sports analytics applications [12, 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 (2D sports field registration) instead of calibration since the estimation of camera parameters enables more 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-
standard 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 remainder of the paper is organized as follows. Sec. 2 provides an overview on 2D sports field registration and its relation to the calibration task. In Sec. 3, we describe the proposed \(TVCalib\) in detail. The experimental results, including 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 the majority of approaches for sports field registration is that they predict homography matrices in some way 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 \(H_{\text{init}}\), which is subsequently refined by the majority of approaches and considered as the relative (non-)affine image transformation \(H_{\text{rel}}\) between the segmented input image and the predicted or retrieved image, finally resulting in \(H = H_{\text{init}}H_{\text{rel}} \in \mathbb{R}^{3 \times 3}\).

We now describe 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 \([13, 19]\) 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 segment the sports field into four different areas \([30]\), or to 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 \([7]\) but on all individual field segments.

**Initial Estimation:** A grid of uniformly sampled and predicted points \([8, 27]\) or predicted keypoints \([11, 13]\) is the input for DLT (+RANSAC) \([18]\) to get usually a rough initial homography estimation, which needs to be refined \([27]\). Some methods use the segmented \([23]\) or raw \([32]\) image to directly predict the homography or to regress four points. Still, such approaches require many 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. \([30]\) retrieve the best candidate from 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 algorithms like the Lucas-Kanade algorithm \([3]\), also in combination with spatial pyramids \([16]\) with the assumption that the image transformation is small. For this reason, the Spatial Transformer Network (STN) was introduced in sports field registration which handles large non-affine transformations \([21]\), e.g., 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 refinement is performed on the homography. A database of synthetic templates [6, 30] allows for direct access to the camera pose as a projectional geometry is used to create template images, however, the smaller the database, the larger the reprojection error is without a refinement step. Despite focus on homographies, it allows us to access individual camera parameters, at least with homography decomposition [11, 18] (see Appx. B). Citraro et al. [11] decompose the initial estimated homography matrix to achieve temporal consistency and also apply a PoseNet [24] as a baseline to regress translation and quaternion vectors directly, but with inferior performance.

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 at image space. Finally, we introduce its direct application, the 3D sports field registration (Sec. 3.3) and segment localization (Sec. 3.4).

3.1. Calibration Object & Camera Model

Given a known 3D (calibration) object 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 \( \phi \) and potential lens distortion coefficients \( \psi \) that minimize its reprojection error.

Modeling the Calibration Object: Line segments are defined in the parametric form \( s_{\text{line}} = \{X_0 + \lambda X_1 | \lambda \in [0, 1]\} \) and point cloud segments as \( s_{\text{pc}} = \{X_j | j = 1, \ldots, s_{\text{pc}}\} \). Without loss of generality, we can define a labeled point segment as \( s_{\text{point}} = X = \mathbb{R}^3 \), 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 \( C: \mathbb{S} = \bigcup_{C \in \{\text{point, line, pc}\}} \{s^{(1)}_C, s^{(2)}_C, \ldots\} \). (Note: the exact number of segments per category might vary depending on the specific object model used.)

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 \( P = K[R[I - t]] \in \mathbb{R}^{3 \times 4} \) parameterized with the intrinsics \( K \in \mathbb{R}^{3 \times 3} \), which define the transformation from camera coordinates to image coordinates, and extrinsics \( [R \in \mathbb{R}^{3 \times 3}, t \in \mathbb{R}^3] \), 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, the only unknown variable in \( K \), we predict the Field of View (FoV) and transform the image space to Normalized Device Coordinates (NDC) for numerical stability (Appx. A.1). Following Euler’s angles convention, the rotation matrix \( R = R_z(\text{roll})R_x(\text{tilt})R_z(\text{pan}) \) 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 \( \phi = (\text{FoV}, t, \text{pan}, \text{tilt}, \text{roll}) \), and assume that \( \pi_0 : X \mapsto x \) 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 \( P' = K[R[I - t]] = H \in \mathbb{R}^{3 \times 3} \) is the respective homography matrix only able to map all points lying on one plane. Appx. B describes how to approximate \( \phi \) given a predicted \( H \) only.

Lens Distortion: As we do not want to restrict to a specific lens distortion model \( \psi \) (e.g., Brown [4]), we define \( \text{distort}_\psi(x) \) that distorts a point \( x \) at image level 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 \( \psi \) and \( \phi \).

3.2. Segment Reprojection Loss

Perspective-n-Point (PnP) refers to the problem of estimating the camera pose (extrinsics) from a calibrated camera \( K \) given \( n \) 2D-3D point correspondences. Geometric solvers for PnP or PnP(f), that also estimate the focal length, approximate the projection matrix \( P \) through the geometric or algebraic reprojection error for \( \text{argmin}_{P} d(x, \pi_{P}(X)) \) where \( d(x, \hat{x}) \) is the Euclidean distance between two pixels. However, accurate correspondences are assumed to be known and the focal length in \( K \) 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 \( \phi \) (and potential
lens distortion coefficients $\psi$) by minimizing the Euclidean distance between all reprojected segments and respective annotated (or predicted) pixels (see Sec. 3.4 for segment localization). Our segment reprojection 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 $x^{(c)} \in \mathbb{R}^{T \times 2}$ for each (visible) segment label $c \in \mathbb{S}$ at (undistorted) image space. For a respective line segment $s^{(c)}_{\text{line}}$, the perpendicular distance to its respective re-projected line $\hat{s}_{\text{line}} = \{\pi_\phi(X_0^{(c)}) + \lambda \pi_\phi(X_1^{(c)}) | \lambda \in \mathbb{R}\}$ can be computed for each $p \in x^{(c)}$:

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

(1)

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

$$\mathcal{L} := \arg\min_{\phi} \pi(\psi) \frac{1}{|\mathcal{S}|} \sum_{c \in \mathcal{S}} d_{\text{mean}}(\text{undistort}_\phi(x^{(c)}), \pi_\phi(s^{(c)}))$$

(2)

Please note that $\pi$ 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 $\leftrightarrow$ point, point $\leftrightarrow$ line, or point $\leftrightarrow$ 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 $C$ (e.g., lines) is $x_c \in \mathbb{R}^{T \times S_C \times N_c \times 2}$, where $N_C$ represents the number of selected pixels ($N_{\text{keypoint}} = 1$), $S_C$ 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 $m_c \in \{0, 1\}^{T \times S_{pc} \times N_{pc} \times 3}$. To reproject the 3D object, all points are projected from the following input dimension per segment type $X_{\text{line}} \in \mathbb{R}^{T \times S_{pc} \times 2 \times 3}$, $X_{pc} \in \mathbb{R}^{T \times S_{pc} \times N_{pc} \times 3}$, and $X_{\text{keypoint}} \in \mathbb{R}^{T \times S_{pc} \times 1 \times 3}$ 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 $m_c$, 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), rather aim to predict all unknown variables $\phi = \{\text{FoV}, \text{pan, tilt, roll}, \theta\}$ 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 $U(a, b)$, we transform to a normal distribution $N(\mu, \sigma)$, so that $\sigma$ 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 according to Eq. (2) (argmin).

**Self-Verification:** Self-verification aims to identify all images in which the model is unable to calibrate or estimate the homography. While Nie et al. [27] use the mean point reprojection error and Citaro et al. [11] verify geometrical constraints, 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 rate 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 TVCalib as well as manual annotations. Please recall that the expected dimension for each segment category $C$ is $X_C \in \mathbb{R}^{T \times S_C \times N_C \times 2}$. Ideal lines are sufficiently represented by two points, however, we have noticed more stable gradients if more than two points are selected. Furthermore, we want to allow potential lens distortion correction based on the extracted points which may show a curved polyline. For sports field registration in soccer, we set $|N_{line}| = |N_{pc}| = 4$. We use the common DeepLabV3 ResNet [7] architecture 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]. Additional points for line segments are selected randomly, similar to circle segments.

4. Experiments

The experimental setup including all baselines, metrics, datasets, and hyperparameters of the TVCalib is introduced in Sec. 4.1. We conduct ablative studies for the proposed self-verification, multiple camera initialization, lens distortion, and segment localization. The individual results per dataset, including ablation studies and comparison to state of the art are reported and discussed in Sec. 4.2 and Sec. 4.3, while limitations of the proposed approach for 3D sports field registration are presented 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 address the homography estimation [8, 27, 30, 32]. A reasonable approach is therefore the homography decomposition (see Appx. B for details) denoted as HDecomp, which allows the approximation of the camera pose and focal length. Baseline: 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 ($\tilde{H}_{line}(DLT)$) [26].

Comparison to State of the Art: As the majority of approaches estimate homography matrices, it is reasonable to apply decomposition on both (1) predicted matrices ($\tilde{H}$) or (2) already manually annotated or ground-truth matrices ($\tilde{H}$). More concrete, we have reimplemented the approach from Chen and Little [6] as they rely on synthetic data for homography estimation. To foster generalization properties on other datasets, we additionally test different variants for camera parameter distributions during training [34]. As the second approach, we apply the official implementation from Jiang et al. [23] for homography estimation denoted as $H$ (25]).

Despite the following variant does not represent the calibration task, the homography matrices map all segments lying on a plane (see Sec. 3.1). Hence, we are able to effectively compare with 2D sports field registration and especially neglect errors induced by HDecomp.

4.1.2 Datasets

SN-Calib Dataset: SoccerNetV3-Calibration [10] (SN-Calib) consists of 20,028 images taken from the SoccerNet [14] videos (500 matches). In contrast to other (private) datasets [27, 30, 34] or WC14 [20], which cover the main broadcast camera (here denoted as center), the SN-Calib dataset covers more camera locations placed in the stadium. 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 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. [10] provide annotation for all segments of the soccer field, i.e., lines, circle segments, and goal posts, and each visible segment has at least two annotated positions optimally representing the segment (i.e., corner and border points).

WC14 Dataset: The WC14 dataset [20] is the traditional benchmark dataset for sports field registration in soccer, containing 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, that are considered as ground truth ($\tilde{H}$).

We have annotated all images from the test split similar as data for homography estimation.
in SN-Calib [10], especially to evaluate the quality of the provided homography matrices.

4.1.3 Metrics

The quality of estimated camera parameters or homography can be evaluated both (1) at image level by measuring an image reprojection error, and (2) in world space by measuring a projection error.

**Accuracy@threshold** [26]: The evaluation is based on the reprojection error which is defined as the Euclidean distance between one annotated ground-truth (GT) position and the segment to which the point belongs. As this metric does not account well for false positives (FP) and false negatives (FN) (missing lines), a more suitable metric is proposed [26]. Segments are projected to the image from dense sampled points resulting in a polyline. A predicted element is a true positive (TP), if all the Euclidean distances between its GT points and the predicted polyline are below a certain threshold. A FP contains elements that were detected with a segment label that do not belong to the GT segments, and elements with valid segments which are distant from at least t pixels from one of the GT points associated to the element. Segments that are only present in the ground truth are counted as FN. There are no true negatives. The accuracy for a threshold of \( t \in \{5, 10, 20\} \) pixel is given by:

\[
AC@t = TP / (TP + FN + FP)
\]

**Completeness (CR):** We also measure the completeness rate as the number of camera parameters provided divided by the number of reserved images for evaluation. **Compound Score (CS):** To summarize the above four scores, they are weighted as follows [26]:

\[
CS = (1 - e^{-4CR}) \left( \sum_{t \in [5, 10, 20], w \in [0, 0.5, 0.35, 0.15]} wAC@t \right)
\]

**Intersection over Union (IoU) [20]:** The accuracy for homography estimation for sports fields is traditionally evaluated on the IoU\(_{part}\) and IoU\(_{whole}\) metric. They measure 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.

4.1.4 Hyperparameters

To optimize the camera parameters \( \phi \), we use AdamW [25] with a learning rate of 0.05 and weight decay of 0.01, optimize for 2000 steps using the one-cycle learning rate scheduling [33] with \( pcls\text{start} = 0.5 \). These parameters were found on the SN-Calib-valid split by a visual exploration of qualitative examples. 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 on all datasets. The

![Figure 2: Aggregated results on SN-Calib-test for the calibration task: Different variants of TVCalib are evaluated for several self-verification thresholds \( \tau \) and compared to the best performing model from Chen and Little [6] (\( U_{FoV} + U_{xyz} \)).](image)

**SN-Calib** dataset is initialized from three camera type distributions, where we only change the horizontal camera position, to cover the main camera center, left, and right. For **segment localization**, the training data are derived from the provided point annotations of the SN-Calib dataset. For training details we refer to Appx. C.

4.2. Results on SN-Calib

The results for the ablation studies are mainly summarized for the full test set in Fig. 2. Performance comparison for the main camera center and comparison to state of the art is provided in Table 2.

**Impact of Segment Localization:** Since we want to find an upper limit for the performance of our method, we use the provided annotations and compare with the predicted segments from our segment localization model (GT vs. Pred). This yields very strong results as errors from the segment localization are ignored. As shown in the randomly selected examples (Fig. 3) visual similar results are achieved.

**Self-verification:** The more the parameter \( \tau \) is restricted, the less the completeness rate decreases, with increasing accuracy that at some point saturates (Fig. 2). A concrete choice of \( \tau \) is roughly valid for all splits (Fig. 4). For the rest of the paper, we set \( \tau = 0.017 \) globally based on the maximum CS on SN-Calib-valid (argmin and predicted segment localization), but please note that the optimal value can be chosen for each dataset.

**Multiple Initialization:** As our solution aims to optimize the camera parameters for multiple camera locations (center, left, right), (a) the question arises whether one initialization (center) is sufficient or multiple initialization (one per camera location) are preferred. And (b), 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 is the optimization from multiple initializations and take the best result (argmin). As shown in Fig. 2, initializing from three camera positions (argmin and stacked) is noticeably better than only using one initialization (center) and selecting the best result (argmin) is slightly better than
Table 2: Results on SN-Calib-test (center) only evaluating where the main camera center is shown (1454 images): When evaluating the homography estimation, all segments not lying one the plane are ignored.

| Calibration | Homography | Seg | AC@5 [%] | AC@10 [%] | AC@20 [%] | CR |
|-------------|------------|-----|----------|-----------|-----------|----|
| **Evaluating the Calibration (φ)** | | | | | | |
| TVCalib | Pred | 52.4 | 76.3 | 89.0 | 100.0 | |
| TVCalib(τ) | Pred | 57.1 | 81.6 | 93.6 | 85.5 | |
| Jiang et al. [23] | [23] | 24.7 | 42.2 | 61.3 | 76.5 | |
| TVCalib | GT | 65.6 | 84.3 | 92.6 | 100.0 | |
| TVCalib(τ) | GT | 69.5 | 88.4 | 96.2 | 91.0 | |
| Chen and Little [6] | [6] (HDecomp) | 51.9 | 73.3 | 84.2 | 74.6 | |
| [6] (HDecomp + 2x cam. [34]) | GT | 52.5 | 75.5 | 87.0 | 81.8 | |
| [6] (HDecomp + U_{xy} [34]) | GT | 55.1 | 78.0 | 88.9 | 77.9 | |
| [6] (HDecomp + U_{xy} [34]) | GT | 53.7 | 77.5 | 88.4 | 80.3 | |
| **Evaluating the Homography (H)** | | | | | | |
| TVCalib | Pred | 49.5 | 72.7 | 87.6 | 100.0 | |
| TVCalib(τ) | Pred | 53.7 | 77.7 | 92.6 | 85.5 | |
| Jiang et al. [23] | [23] | 28.5 | 45.0 | 62.9 | 76.5 | |
| TVCalib | GT | 61.9 | 81.8 | 92.0 | 100.0 | |
| TVCalib(τ) | GT | 51.6 | 66.3 | 74.0 | 100.0 | |
| Chen and Little [6] | [6] (HDecomp) | 56.8 | 74.6 | 82.4 | 100.0 | |
| [6] (HDecomp + 2x cam. [34]) | GT | 56.8 | 72.7 | 80.1 | 100.0 | |
| [6] (HDecomp + U_{xy} [34]) | GT | 57.3 | 76.0 | 83.7 | 100.0 | |
| TVCalib(τ) | GT | 65.7 | 85.9 | 95.7 | 91.0 | |
| HDecomp | Chen and Little [6] | 58.2 | 76.4 | 85.0 | 74.6 | |
| [6] (HDecomp + 2x cam. [34]) | GT | 59.5 | 79.7 | 88.5 | 81.8 | |
| [6] (HDecomp + U_{xy} [34]) | GT | 62.3 | 81.7 | 89.6 | 77.9 | |
| [6] (HDecomp + U_{xy} [34]) | GT | 61.1 | 81.2 | 89.4 | 80.3 | |

Knowing the camera type before (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.

**Comparison to State of the Art:** We compare to Chen and Little [6] as the retrieval and refinement module solely rely on synthetic data and we have replaced the segmentation module (pix2pix models) with the ground-truth masks generated from the SN-Calib annotations. Further we apply the official implementation from Jiang et al. [23]. Other state-of-the-art approaches like [8, 27, 30] rely on annotated homography matrices which are not available for SN-Calib.

If the same ground-truth segmentation is used as input for TVCalib and [6] (and its variants [34]), TVCalib achieves superior results when evaluating both, the calibration task and the homography estimation.

The main reason for the performance drop for Jiang et al. [23] compared to our method and to the WC14 dataset is explained by the fact that training was only performed on WC14 with a small number of images and not as much diverse dataset (stadiums, lightning conditions, etc.).

4.3. Results on WC14

**Evaluating the Calibration Task:** This task represents the main task of estimating individual camera parameters φ where the reprojection error (AC@τ) induced by φ is evaluated. The results on the WC14-test are presented in Table 3.

TVCalib from GT Segments vs. HDecomp from Annotated Homography: The results show (first four rows) that the performance of the decomposition from annotated ho-
homographies $\tilde{H}$ is much lower compared to our method using ground-truth segment annotations as input.

**TVCali vs. HDecomp from Predicted Matrices:** Compared to [6, 23] our method achieves superior results. 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 matrix in the following and thus exclude the influence of HDecomp.

**Evaluating the Homography Estimation:** We can also evaluate the reprojection error without using the homography decomposition and hence use the raw homography matrix that only maps all planar points and segments like the goal posts are ignored.

**TVCali from GT Segments vs. Annotated Homography:** The reprojection error given the provided annotated homographies $\tilde{H}$ is comparable with our results at 5 and 10 pixels thresholds, indicating that the individual reprojected segments do not fit the actual segments ideally.

**TVCali vs. Predicted Matrices:** Our method achieves slightly better results than related work [6, 23], close to the corresponding results where the calibration task is evaluated. This shows that decomposition induces additional errors. Based on the per-segment accuracy, we found that in particular a larger reprojection error is frequently visible for the goal segments.

Please recall that for the evaluation in world space (via $IoU$), the comparison is performed given a predicted homography and a ground-truth homography. Since the reprojection error from the annotated homographies $\tilde{H}$ is comparable with our results, but not ideal, this indicates there is a bias for the $IoU$ (projection metric). This annotation bias is explained in Appx. E. However, TVCalib achieves very comparable results compared to state-of-the-art approaches without performing training or fine-tuning on this dataset.

### 4.4. Limitations

Despite strong results, even for a small fraction of given ground-truth segment annotations (e.g., Table 3 TVCalib($\tau$) and Fig. 4), some samples are rejected. This is mainly caused by local minima due to the nature of a gradient-based iterative optimization [1]. Related to the camera initialization, we have not investigated other cameras, obviously resulting in a lower total completeness rate on SN-Calib. TVCalib relies on an accurate segment localization, but no regularization term is included that allows outliers. We have already shown the upper limit of the performance for TVCalib when using ground-truth annotations, indicating that the segment localization needs improvement. Jointly learning lens distortion coefficients has not been deeply investigated.

### Table 3: Results on WC14-test

| Calibration | Homography Set | AC@5 [%] | IoU($\tilde{H}$) mean med. |
|-------------|----------------|----------|-----------------------------|
| TVCalib     | GT             | 64.1     | 90.5 | 93.0 | 95.0 | 97.0 |
| HDecomp     | TVCalib_GT     | 60.4     | 92.6 | 94.8 | 96.8 | 98.8 |
| TVCalib($\tau$) | GT         | 64.7     | 90.5 | 93.4 | 95.4 | 97.4 |
| HDecomp     | $\tilde{H}$ (DLT) GT | 52.3     | 78.9 | 91.5 | 92.3 | 93.0 |
| TVCalib($\tau$) | Pred | 35.7     | 69.1 | 97.9 | 99.5 | 100.0 |
| HDecomp($\tau$) (1k) | [23] [23] | 32.4     | 58.5 | 75.3 | 99.0 | 100.0 |
| TVCalib($\tau$) | Pred | 37.1     | 70.8 | 98.9 | 99.9 | 100.0 |
| HDecomp($\tau$) (DLT) Pred | [6] [23] | 52.7     | 67.3 | 83.1 | 83.9 | 84.9 |
| $\tilde{H}$ (DLT) Pred | | 26.2     | 33.0 | 40.4 | 73.7 | 74.7 |

### 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 sports field registration, our method achieves 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.

A. Camera Model

A.1. Field of View at NDC and Raster

For numerical stability the input image or pixel are normalized w.r.t NDC coordinates (image dimensions from $[-1, 1]$) and the Field of View (FoV) in radians is predicted instead of the focal length resulting in $f_x^{NDC} = \frac{1}{\tan(0.5 \times \text{FoV})}$ and $f_y^{NDC} = a \times \frac{0.5}{\tan(0.5 \times \text{FoV})}$, where $a$ is the original image aspect ratio. To access the true focal length, we know square pixel and thus, use $f_x^I = f_y^I = w \times \frac{0.5}{\tan(0.5 \times \text{FoV})}$ where $w$ is the original image width in pixel.

A.2. Camera Distribution

The following camera parameter distribution cover a variety of stadiums over the world for the main tribune and is coarser distribution as used in [6]: $\text{pan} \in \mathcal{U}(-45^\circ, 45^\circ)$, $\text{till} \in \mathcal{U}(45^\circ, 90^\circ)$, $\text{roll} \in \mathcal{U}(-10^\circ, 10^\circ)$, $\text{aov} \in \mathcal{U}(8.2^\circ, 90^\circ)$, $t_x \in \mathcal{U}(-40\text{~m}, -5\text{~m})$, $t_y \in \mathcal{U}(40\text{~m}, 110\text{~m})$, $t_z \in \mathcal{U}(-36 - 16.5\text{~m}, -36 + 16.5\text{~m})$ for main camera left and $t_y \in \mathcal{U}(36 - 16.5\text{~m}, 36 + 16.5\text{~m})$ for main camera right.

A.3. Note on the World Reference Axis System

Given a world reference coordinate system and the definition of the pinhole camera model ($KR[I - t]$) including the principal axis, the decomposition of $H$ in $R$ and $t$ and individual rotation angles must follow the concrete definition in order to derive expected values.

In case where the world axis system differ, other units (yards instead of meters) or world camera center is used as in [6, 23], the provided homography matrices can be aligned.

Alignment with Chen and Little [6] Because Chen and Little [6] place the coordinate system differently through the sports field, the output $H_{[6]}$ of the reproduced model (reimplemented using the official code snippets from the authors) is aligned to the SN-Calib axis system according to

$$
\hat{H} = R(TH_{[6]}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \left( \begin{bmatrix} 1 & 0 & -105/2 \\ 0 & 1 & -68/2 \\ 0 & 0 & 1 \end{bmatrix} H_{\text{WC14}} \right)
$$

where first the coordinate center is moved to the middle of the sports field and only the direction of the $y$-axis is swapped.

Alignment with original WC14 [20] homography matrices The provided homography matrices from the WC14 dataset ($\hat{H}$) are aligned to the SoccerNet coordinate system as follows: The scene coordinate center needs to be moved to the center of the sports field and the dimensions need to be scaled from yards to meters ($y2m \approx 0.9144$):

$$
\hat{H} = S(T\hat{H}_{\text{WC14}}) = \begin{bmatrix} y2m & 0 & 0 \\ 0 & y2m & 0 \\ 0 & 0 & 1 \end{bmatrix} \left( \begin{bmatrix} 1 & 0 & -115/2 \\ 0 & 1 & -74/2 \\ 0 & 0 & 1 \end{bmatrix} H_{\text{WC14}} \right)
$$

Alignment with Jiang et al. [23] As Jiang et al. [23] use other dimensions for the image and sports field template ($(-0.5, 0.5)$) and origin, the output $H_{[23]}$ of their officially provided model is first aligned to WC14 [20] ($H_{[23]}^*$) which is subsequently aligned to SoccerNet by scaling (1) the image to the original resolution (W, D), (2) scaling the template image to the used 720p resolution:

$$
\hat{H} = \begin{bmatrix} W & 0 & W/2 \\ 0 & H & 1 \end{bmatrix} H_{[23]}^* = \begin{bmatrix} 1280 & 0 & 640 \\ 0 & 720 & 360 \end{bmatrix}^{-1}
$$

B. Homography Decomposition: From Homography to Camera Parameters

This section describes how to extract the camera position $t$, orientation (pan, tilt, roll) and focal length from a plane homography according to the pinhole camera model as described in Sec. 3.1 assuming square pixel, zero skew, and a centered principal point. In general, given a calibration matrix $K$ and a homography matrix $H$ that describes the mapping between two planes (e.g., derived from point correspondences), rotation matrix $R$ and translation vector $t$, can be derived [18]. The procedure described below is in general in line with [11, 26] and we mainly follow the implementation from [26] with minor adjustments.

(1) Computing the Focal Length: As $H$ already describes the relation between two planes, and the focal length is the only unknown parameter in $K$, the first step is to approximate the focal length (see Algorithm 8.2 in Hartley and Zisserman [18]) given constraints from the homography matrix and our assumptions on $K$.

(2) Computing the Rotation Matrix: Leveraging the relation between the approximated calibration matrix and provided homography, orientation (first rotation matrix $R$) and translation $t$ (camera position) are then approximated as we know that $H X_\text{vec} = 0 P_{\times [1, 2, 4]} = K R_{\times [1, 2]} [I - t]$.

Since $K$ is already given, $K^{-1} H$ yields individual column vectors $[r_1', r_2', -t']$ encoding rotation and translation. After normalizing $r_1', r_2'$ to unit length, the third column $r_3'$ of the rotation matrix $R' = [r_1, r_2, r_3]$ can be approximated from $r_1 \times r_2$, since we expect orthogonality for $R$ (constructed from per axis rotations, i.e., $R_z(\text{roll})R_x(\text{till})R_y(\text{pan})$). Singular value decomposition is applied $USV^T = R'$ and since one property is that $U, V$ are real orthogonal matrices, the estimated rotation matrix is $R = UV^T$. 


(3) Computing the Camera Position: The translation vector is finally derived from \( t = -R^T (t' + \sqrt{r_1^2 \times r_2^2}) \).
(4) Refining \( R \) and \( t \): Once \( K \), \( R \), and \( t \) are roughly approximated, the camera pose can be refined given reprojected keypoints from \( H \) (2D-3D point correspondences) via non-linear least-squares minimization (Levenberg-Marquardt refinement, see \texttt{cv2.solvePnPRefineLM()}). To provide only a reasonable set of keypoint correspondences, a keypoint is only considered if a point of the homography is visible in the image with a tolerance of \( 0.1 \times \) image width and height, respectively. The tolerance is motivated by a simple example and is not part of Magera et al. [26]: Assume the keypoint in the middle of the central circle which is close outside the visible image. It is a valuable information despite it is not visible. As the Levenberg-Maquardt algorithm is not able to handle large redefinitions, a point is not considered if its reprojection error between initial estimation and homography is larger than \( \zeta = 100 \) pixels [26]. The impact of \( \zeta \) is shown in Table 4.

In case the number of point correspondences is smaller than three, the refinement algorithm cannot be performed. We reject the entire sample and do not return the initial estimation as the difference between the decomposition and the original estimated homography is too large.

(5) Accessing Individual Rotation Angles: As \( R \) is composed of individual per axis rotations representing pan, tilt, and roll of the camera of known order (i.e., a known scene coordinate system) and given principal axis, individual rotation angles can be extracted by solving \( R = R_z(\text{roll})R_y(\text{tilt})R_x(\text{pan}) \) for pan, tilt, and roll angles. However, as there are two solutions, we exploit world knowledge and take the solution where the roll parameter is minimal [26].

C. Segment Localization

We use the common DeepLabv3 ResNet-101 [7] architecture to perform instance segmentation on all sports field segments. To train this model, we use the training data from the SN-Calib train split and validate on the respective validation split while keeping the model with the lowest loss on validation. During training, images are resized to a height of 256 pixels. Basically following the vanilla training script and suggested parameters, we train for max. 30 epochs using a batch size of 8, SGD (momentum: 0.9, weight decay: \( 1^{-3} \)), learning rate of 0.01, initialized with ImageNet1k weights, and auxiliary loss.

D. Radial Lens Distortion Correction

As only radial lens distortion seems to be present for some samples, optimization of radial lens distortion coefficients \( \psi = \{k_1, k_2\} \) may also be performed where we follow kornia’s implementation of lens distortion models.

To first focus on learning the camera parameters \( \phi \), lens distortion coefficients \( \psi \) are optimized with its own optimizer (also AdamW but with a learning rate of \( 1e^{-3} \)) and one-cycle learning rate scheduling (\( pct_{start} = 0.33 \)). However, we have observed this process works for many samples where radial lens distortion is present (Table 5, Fig. 5 A and B) with significantly better results on WC14, but noticed an issue on the SN-Calib dataset and specific samples (e.g., Fig. 5 C and D, mainly images with a low FoV). WC14 is not affected as usually a larger FoV is shown. Selected points are transformed via undistort and in some cases the points are distorted too much (towards the principal point) or the FoV explodes, resulting in local minima.

Transforming the reprojected points (distort) instead of the selected points is reasonable (i.e., \( d(ax^i, \text{distort}_{\phi, \psi}(s^i)) \)), but the distance calculation for ideal lines (on undistorted image space) is very effective and needs to be adjusted otherwise. We continue to investigate this issue to find a practical solution. Results indicate that the performance can be increased for samples where radial lens distortion is present.

Table 4: Results on WC14-test varying the \( \zeta \) parameter which is responsible for appropriate keypoint selection.

| Calibration | Homography | \( \zeta \) [px] | AC@ [%] | 5 | 10 | 20 | CR |
|-------------|------------|----------------|--------|----|----|----|----|
| HDecomp \( \hat{H} \) | 100 | 32.7 | 67.3 | 87.3 | 81.7 | | | |
| | 300 | 31.6 | 64.9 | 85.4 | 90.3 | | | |
| | 1000 | 29.0 | 59.8 | 79.0 | 100.0 | | | |
| HDecomp \( \hat{H}(\text{Chen and Little [6]} \) | 100 | 32.7 | 67.3 | 87.3 | 81.7 | | | |
| | 300 | 31.6 | 64.9 | 85.4 | 90.3 | | | |
| | 1000 | 29.0 | 59.8 | 79.0 | 100.0 | | | |

Table 5: Ablation study for radial lens distortion correction (LD)

| Dataset | Calibration | Segm. LD | AC@ [%] | 5 | 10 | 20 | CR |
|---------|-------------|----------|--------|----|----|----|----|
| WC14-test | TVCalib | GT | 64.0 | 86.6 | 95.2 | 100.0 | |
| | TVCalib | GT | 69.0 | 87.7 | 95.7 | 100.0 | |
| | TVCalib(r) | GT | 64.2 | 87.0 | 95.5 | 98.9 | |
| | TVCalib(r) | GT | 69.3 | 88.0 | 95.9 | 98.9 | |
| SN-Calib-val (center) | TVCalib | Pred | 51.3 | 75.6 | 89.1 | 99.6 | |
| | TVCalib | Pred | 52.0 | 76.2 | 89.6 | 46.7 | |
| | TVCalib(r) | Pred | 55.3 | 79.7 | 92.2 | 86.8 | |
| | TVCalib(r) | Pred | 54.9 | 79.3 | 93.2 | 41.6 | |
| SN-Calib-val (center) | Vanilla | GT | 61.8 | 81.4 | 91.1 | 99.3 | |
| | Vanilla | GT | 63.5 | 82.4 | 92.2 | 80.9 | |
| | TVCalib(r) | GT | 66.5 | 86.5 | 95.7 | 89.1 | |
| | TVCalib(r) | GT | 66.7 | 85.7 | 94.8 | 74.3 | |
Figure 5: A, B: Samples where radial lens distortion is present (left: with correction, right: without correction for comparison); C, D: samples with lower FoV result in trivial local minima when jointly optimizing distortion coefficients and camera parameters.

E. Why do we not recommend the use of the IoU metrics on WC14 [20]?

The accuracy for homography estimation in sports fields is traditionally evaluated on the \( \text{IoU}_{\text{part}} \) and \( \text{IoU}_{\text{full}} \) metric [20] which measures the binary IoU of the projected templates from predicted homography and a ground-truth homography at world (top view/bird view) space for the visible area (part) and full (whole) area of the field. We highlight the major disadvantage of this metric and refer to [11] for further additional drawbacks. However, homography matrices are annotated at image space, but the IoU metric compares at the top-view, hence there is an induced annotation bias.

We claim that annotating homography matrices (e.g., at least by four points and then using DLT) is in some cases tricky and can not cover the entire visible area of the sports field leading to additional bias. To confirm this claim, we use the provided homography matrices from the WC14 dataset and evaluate its reprojection error (\( \text{AC}@t \)). The results are already available in the main paper (Table 3 evaluating the homography, \( \hat{H} \)) and confirm our claim.

References

[1] R. Acuna and V. Willert. Insights into the robustness of control point configurations for homography and planar pose estimation. arXiv preprint arXiv:1803.03025, 2018.
[2] N. M. Alem, J. W. Melvin, and G. L. Holstein. Biomechanics applications of direct linear transformation in close-range photogrammetry, 1978.
[3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision, IJCV, 56(3):221–255, 2004.
[4] D. C. Brown. Decentering distortion of lenses. Photogrammetric Engineering and Remote Sensing, 1966.
[5] P. Carr, Y. Sheikh, and I. A. Matthews. Point-less calibration: Camera parameters from gradient-based alignment to edge images. In Workshop on Applications of Computer Vision, WACV, 2012, pages 377–384. IEEE Computer Society, 2012. doi: 10.1109/WACV.2012.6163012.
[6] J. Chen and J. J. Little. Sports camera calibration via synthetic data. In Conference on Computer Vision and Pattern Recognition workshops, CVPRW. CVF/IEEE, 2019.
[7] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Re-thinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587, 2017.
[8] Y.-J. Chu, J.-W. Su, K.-W. Hsiao, C.-Y. Lien, S.-H. Fan, M.-C. Hu, R.-R. Lee, C.-Y. Yao, and H.-K. Chu. Sports field registration via keypoints-aware label condition. In Conference on Computer Vision and Pattern Recognition workshops, CVPRW, 2022, pages 3523–3530. IEEE/CF, 2022.
[9] A. Cioppa, A. Deliege, F. Magera, S. Giancola, O. Barnich, B. Ghanem, and M. 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, 2021, pages 4537–4546. CVF/IEEE, 2021.
[10] A. Cioppa, A. Deli`ege, S. Giancola, B. Ghanem, and M. Van Droogenbroeck. Scaling up soccernet with multi-view spatial localization and re-identification. Scientific Data, 9(1):1–9, 2022.
[11] L. Citraro, P. Márquez-Neila, S. Savare, V. Jayaram, C. Dubout, F. Renaut, A. Hasfura, H. B. Shitrit, and P. Fua. Real-time camera pose estimation for sports fields. Mach. Vis. Appl., 31(3):16, 2020. doi: 10.1007/s00138-020-01064-7.
[12] C. Cuevas, D. Quilón, and N. García. Techniques and applications for soccer video analysis: A survey. Multimedia Tools and Applications, 79(39):29685–29721, 2020.
[13] C. Cuevas, D. Quilón, and N. García. Automatic soccer field of play registration. 103:107278, 2020. doi: 10.1016/j.patcog.2020.107278.
[14] A. Deliege, A. Cioppa, S. Giancola, M. J. Seikavandi, J. V. Dueholm, K. Nasrollahi, B. Ghanem, T. B. Moeslund, and M. Van Droogenbroeck. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In Conference on Computer Vision and Pattern Recognition,
CVPR, 2021, pages 4508–4519. IEEE/CVF, 2021. 

[15] T. D’Orazio and M. Leo. A review of vision-based systems for soccer video analysis. Pattern recognition, 43(8):2911–2926, 2010. 

[16] B. Ghanem, T. Zhang, and N. Ahuja. Robust video registration applied to field-sports video analysis. ICASSP, 2, 3 2012. 

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

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

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

[20] N. Homayounfar, S. Fidler, and R. Urtasun. Sports field localization via deep structured models. In Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pages 4012–4020. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.427. 

[21] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. Advances in neural information processing systems, NIPS, 2015. 

[22] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park. Self-calibrating neural radiance fields. In International Conference on Computer Vision, ICCV, 2021, pages 5846–5854, 2021. 

[23] W. Jiang, J. C. G. Higuera, B. Angles, W. Sun, M. Javan, and K. M. Yi. Optimizing through learned errors for accurate sports field registration. In Winter Conference on Applications of Computer Vision, WACV, 2020, pages 201–210. IEEE, 2020. doi: 10.1109/WACV45572.2020.9093581. 

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

[25] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1710.01901, 2017. 

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

[27] X. Nie, S. Chen, and R. Hamid. A robust and efficient framework for sports-field registration. In Winter Conference on Applications of Computer Vision, WACV, 2021, pages 1935–1943. IEEE, 2021. doi: 10.1109/WACV48630.2021.00198. 

[28] J. Puwein, R. Ziegler, J. Vogel, and M. Pollefeys. Robust multi-view camera calibration for wide-baseline camera networks. In Workshop on Applications of Computer Vision, WACV, 2011, pages 321–328. IEEE, 2011. 

[29] L. Sha, P. Lucey, Y. Yue, X. Wei, J. Hobbs, C. Rohlf, and S. Sridharan. Interactive sports analytics. 25:1–32, 2018. 

[30] L. Sha, J. A. Hobbs, P. Felsen, X. Wei, P. Lucey, and S. Gan-guly. End-to-end camera calibration for broadcast videos. In Conference on Computer Vision and Pattern Recognition, CVPR, pages 13624–13633. IEEE/CVF, 2020. doi: 10.1109/CVPR42600.2020.01364. 

[31] R. A. Sharma, B. Bhat, V. Gandhi, and C. V. Jawahar. Automated top view registration of broadcast football videos. In 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018, pages 305–313. IEEE Computer Society, 2018. doi: 10.1109/WACV.2018.00040. 

[32] F. Shi, P. Marchwica, J. C. G. Higuera, M. Jamieson, M. Javan, and P. Siva. Self-supervised shape alignment for sports field registration. In Winter Conference on Applications of Computer Vision, WACV, 2022, pages 3768–3777. IEEE, 2022. doi: 10.1109/WACV51458.2022.00382. 

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

[34] J. Theiner, W. Gritz, E. Müller-Budack, R. Rein, D. Memmert, and R. Ewerth. Extraction of positional player data from broadcast soccer videos. In Winter Conference on Applications of Computer Vision, WACV, pages 823–833, 2022. 

[35] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pages 7794–7803, 2018. 

[36] N. Zhang and E. 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. 

[37] Z. Zhang. A flexible new technique for camera calibration. 22:1330–1334, 2000. doi: 10.1109/34.888718. 

[38] Y. Zheng, S. Sugimoto, I. Sato, and M. Okutomi. Interactive sports analytics. 25:1–32, 2018. doi: 10.1145/3185596. 

[39] L. Sha, J. A. Hobbs, P. Felsen, X. Wei, P. Lucey, and S. Gan-guly. End-to-end camera calibration for broadcast videos. In Conference on Computer Vision and Pattern Recognition, CVPR, pages 13624–13633. IEEE/CVF, 2020. doi: 10.1109/CVPR42600.2020.01364. 

[40] R. A. Sharma, B. Bhat, V. Gandhi, and C. V. Jawahar. Automated top view registration of broadcast football videos. In 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018, pages 305–313. IEEE Computer Society, 2018. doi: 10.1109/WACV.2018.00040. 

[41] F. Shi, P. Marchwica, J. C. G. Higuera, M. Jamieson, M. Javan, and P. Siva. Self-supervised shape alignment for sports field registration. In Winter Conference on Applications of Computer Vision, WACV, 2022, pages 3768–3777. IEEE, 2022. doi: 10.1109/WACV51458.2022.00382. 

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

[43] J. Theiner, W. Gritz, E. Müller-Budack, R. Rein, D. Memmert, and R. Ewerth. Extraction of positional player data from broadcast soccer videos. In Winter Conference on Applications of Computer Vision, WACV, pages 823–833, 2022. 

[44] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pages 7794–7803, 2018. 

[45] N. Zhang and E. 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. 

[46] Z. Zhang. A flexible new technique for camera calibration. 22:1330–1334, 2000. doi: 10.1109/34.888718. 

[47] Y. Zheng, S. Sugimoto, I. Sato, and M. Okutomi. Interactive sports analytics. 25:1–32, 2018. doi: 10.1145/3185596.