

Sports Camera Calibration via Synthetic Data

Jianhui Chen and James J. Little

Abstract— Calibrating sports cameras is important for autonomous broadcasting and sports analysis. Here we propose a highly automatic method for calibrating sports cameras from a single image using synthetic data. First, we develop a novel camera pose engine. The camera pose engine has only three significant free parameters so that it can effectively generate a lot of camera poses and corresponding edge (*i.e.*, field marking) images. Then, we learn compact deep features via a siamese network from paired edge image and camera pose and build a feature-pose database. After that, we use a novel two-GAN (generative adversarial network) model to detect field markings in real images. Finally, we query an initial camera pose from the feature-pose database and refine camera poses using truncated distance images. We evaluate our method on both synthetic and real data. Our method not only demonstrates the robustness on the synthetic data but also achieves the state-of-the-art accuracy on a standard soccer dataset and very high performance on a volleyball dataset.

I. INTRODUCTION

Camera calibration is a fundamental task for robotic applications such as visual SLAM and relocalization. It also provides training examples for data-driven applications such as autonomous broadcasting systems which replace human operators with robotic cameras to capture sports events. For example, companies like ESPN [1] train intelligent broadcast systems using accurate camera poses. Our task is to estimate the camera poses from a single color image in sports such as soccer games.

Many methods were developed to align images with templates (*e.g.* soccer pitch). The current trend is using highly/fully automatic methods to deal with large-scale data. For example, Wen *et al.* [2] first reconstructed a panoramic court image from a basketball video. Then, they warped the panoramic court to the court template. However, their method requires multiple images from well-textured stadiums (*e.g.* for basketball games).

Our work is closely related to [3] and [4] but we approach the problem from a very different perspective. The deep structured model (DSM) [3] uses semantic information such as grassland and field markings to relocalize the soccer field. On the other hand, Sharma *et al.* [4] uses a synthetic dictionary to query the camera pose. Both of them use the standard projection (2D to 2D homography or a standard pinhole camera model) in their methods. None of them approaches the problem by analyzing the specific form of sports camera poses. We will show that with a better decomposition of sports camera poses, we can obtain more accurate calibration results.

Jianhui Chen and James J. Little are with the Department of Computer Science, University of British Columbia, Vancouver, Canada. {jhchen14,little}@cs.ubc.ca

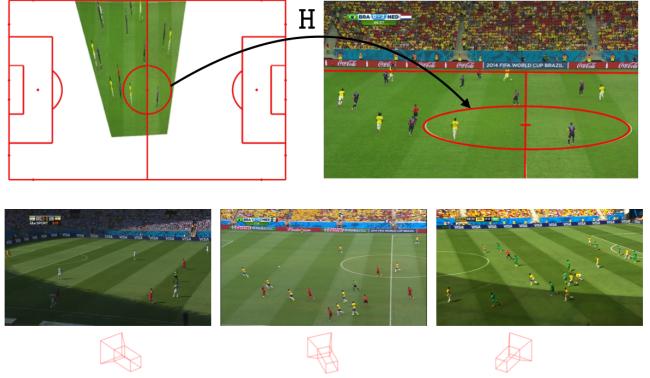


Fig. 1: Sports camera calibration and camera location prior. The first row shows our task of calibrating a sports camera from a single image. The second row shows our observation that these cameras have a strong prior of the location (roughly shown by red cameras). Our work develops a novel camera calibration method by using the location prior.

Our observation of sports images inspires our method. Figure 1 (second row) shows image examples from the WorldCup dataset [3]. Our first observation is that field markings (*e.g.* lines and circles) must be used in the calibration as they are the visible evidence of camera poses. The second observation is that the camera poses have some “default” settings. For example, the cameras are roughly located near and above the middle line. The cameras have a large range of pan angles (from left to right) and a small range of tilt angle (from top to down). By watching lots of broadcasting videos, we found these “default” settings hold for many sports videos. We will show how to use these “default” settings as a prior to simplify the camera model.

Our method extends the two-point method [5] by relaxing the constraint of the camera location. The two-point method assumes the camera location is *exactly* known in training/testing. However, this assumption generally does not hold when cameras change positions in different games. In this work, we only assume the camera location is *roughly* known. As a result, our method can be applied to more situations than the two-point method. For instance, the images in training and testing can be from different stadiums.

Figure 2 shows the pipeline of our method. We first build a feature-pose database in training. In the database, the camera poses are generated by a novel camera pose engine. The features are learned from a siamese network. In testing, our method detects field markings from the input image. Then, an initial camera pose is retrieved from the database and is refined using distance images. In summary, our paper has

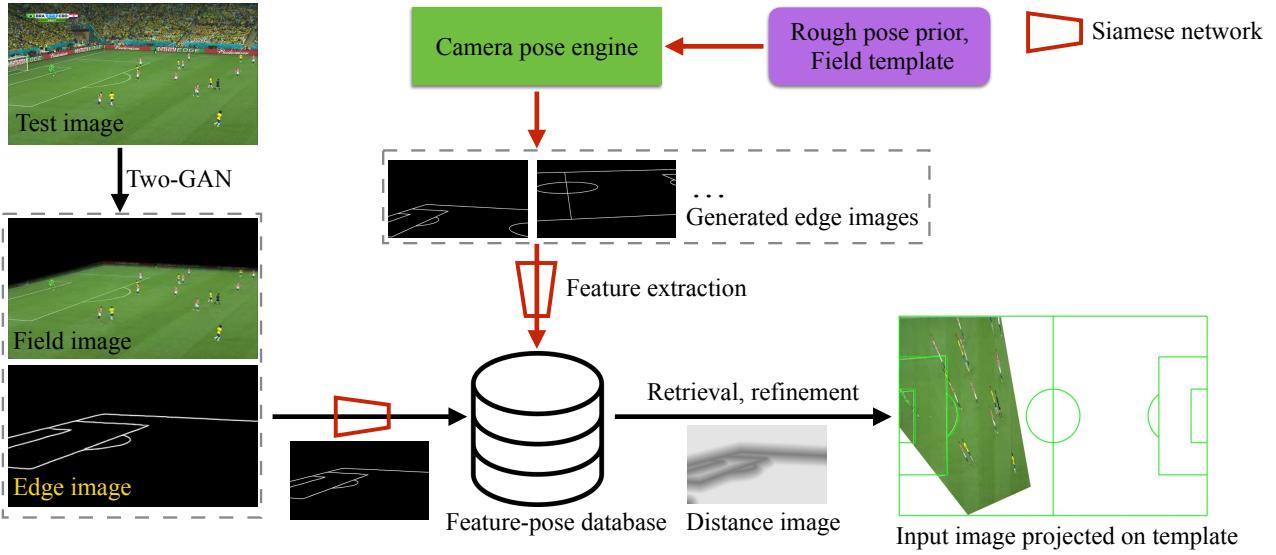


Fig. 2: Sports camera calibration pipeline. In training (red arrows), we use a novel camera pose engine to synthesize a set of edge images which are embedded into a low dimensional feature space. In this way, we obtain a feature-pose database. In testing (black arrows), we detect field markings using a two-GAN model from a testing image. The camera pose is quickly retrieved from the database and is refined using the distance image. Best viewed in color.

three contributions:

- We propose a novel sports camera pose engine that only has three significant free parameters.
- We also propose an effective feature extraction method for edge images and an end-to-end two-GAN model to detect field markings.
- We demonstrate the state-of-the-art performance on a standard soccer dataset and very high performance on a volleyball dataset.

In the following, we start with a discussion of related work and then describe our method. Finally, we demonstrate our method on both synthetic and real datasets.

II. RELATED WORK

Camera Relocalization and Sports Camera Calibration: Camera relocalization has been widely studied in the context of global localization for robots using edge images [6], random forests [7], [8] and deep networks [9], [10], [11], [12].

In sports camera calibration, researchers assume the playing surface is flat so that camera calibration is equivalent to estimating the homography from the ground to the image. Previous work [13], [14], [15] first manually annotates several reference images. Then, they calibrate other images by finding correspondences from reference images. Fully-automatic methods are emerging because they require no or fewer user interactions [2], [16]. For example, Homayounfar *et al.* [3] formulate the problem as a branch and bound inference in a Markov random field where an energy function is defined in terms of semantic cues (*e.g.* field surface, lines and circles).

Recently, Sharma *et al.* [4] formulate the problem as a nearest neighbour search problem over a synthetic dictionary

of edge images. The method first warps the training image to the field template in which the training image becomes a convex quadrilateral. Then, it simulates pan, tilt and zoom by manipulating the quadrilateral. This approach also extensively studies different representations for edge images and finally chooses the histogram of gradients (HOG) features. Our method is significantly different from this method in two points. First, we propose a novel camera pose engine to generate edge images. The camera pose engine is physically interpretable and only requires very rough information of camera locations. Second, we learn features of edge images using a siamese network that provides much more compact features than HOG.

Edge Image Detection and Representation: An edge image is the projection of a template (field markings) in an image. Previous work developed color-based kernel [17], line and ellipse detection [18] to distinguishes field-marking pixels from other pixels. Recently, Sharma *et al.* [4] uses a conditional generative adversarial network (CGAN) [19] to directly generate edge images from an RGB image.

Edge images are represented by features that are much more efficient than raw edge images in search [20], [3]. For example, Sharma *et al.* [4] used Chamfer transformation and HOG to represent edge images for soccer games. In an early version of [4], deep features have been reported with good results on synthetic experiments but poor results on real data experiments.

Metric learning was used to learn deep features from paired/triplet images [21], [22]. For example, Wohlhart and Lepetit [23] learns deep features from a triplet network to estimate 3D poses for small indoor objects. Doumanoglou *et al.* [24] analyze the influence of different objectives such as regression loss of poses and contrastive loss of paired images. To the best of our knowledge, few previous methods

focus on edge images which are texture-less and much more challenging than well-textured images.

III. METHOD

Our method has a camera pose engine, a deep feature extractor, a two-GAN model for field marking detection and a camera pose refinement process. Limited by pages, intermediate results from each step are provided in the supplementary video.

A. Camera Pose Engine

We use the pinhole camera model to describe the projective aspects of a camera

$$P = KR[I] - C, \quad (1)$$

where K is the intrinsic matrix, R is a rotation matrix from world to camera coordinates, I is an identity matrix and C is the camera's center of projection in the world coordinate. To simplify the problem, we assume square pixels, a principal point at the image center and no lens distortion. We found these assumptions hold well for our problem. As a result, the focal length f is the only unknown variable in the intrinsic matrix.

Most sports cameras are pan-tilt-zoom (PTZ) cameras [20], [5]. As a result, we decompose the rotation matrix R in (1):

$$P = \underbrace{KQ_\phi Q_\theta S}_{\text{PTZ}} \underbrace{[I] - C}_{\text{prior}}. \quad (2)$$

The combination of $Q_\phi Q_\theta S$ describes rotations from world to camera coordinates. First, it rotates the camera to the PTZ camera base (*e.g.* a big tripod) by S . Then the camera pans by Q_θ and tilts by Q_ϕ . P can be separated into two parts. The right part $S[I] - C$ is time invariant because camera bases are generally fixed to capture stable videos in sports.

To reduce the number of free parameters, we further decompose the base rotation S to

$$S = S_{\theta'} S_{\rho} S_{\phi'}, \quad (3)$$

where ϕ' , ρ , and θ' are tilt, roll and pan angles of the camera base. We simplify the formation by eliminating $S_{\theta'}$ since $Q_\theta S_{\theta'}$ can be represented by Q_θ with $\theta = \theta + \theta'$. Thus, the camera model becomes:

$$P = \underbrace{KQ_\phi Q_\theta S_\rho}_{\text{PTZ}} \underbrace{S_{\phi'}[I] - C}_{\text{prior}}. \quad (4)$$

Without loss of generality, we set the world and camera coordinates as in Figure 3. The world origin is at the left bottom of the field template. When pan and tilt are zeros, the camera looks along the Y-axis in the world coordinate.

At first glance, (4) seems to have many free parameters (3 for location, 4 for rotation and 1 for focal length). However, the number of significant free parameters is only three. In (4), we set $\phi' = -90^\circ$ with which cameras are set up to be “level” because the effect of $S_{\phi'}$ will be canceled by Q_ϕ . ρ' varies in a small range ($\pm 0.1^\circ$) because the camera base prevents the camera from rotating about its direction-of-view.

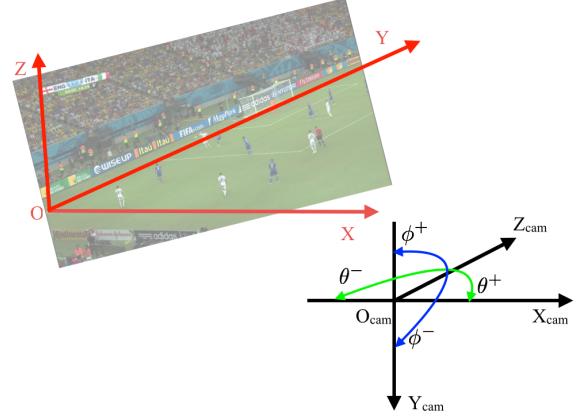


Fig. 3: World (red) and camera (black) coordinates used in (4). The origin of the world coordinate is at the left bottom of the soccer template. The Z-axis in the camera coordinate is aligned with the Y-axis in the world coordinate. Best viewed in color.

For sports fields, C is further constrained in practice, for example, most cameras are above and along the center line for soccer games. As a result, the significant free parameters are f, ϕ, θ whose ranges are known (*e.g.* from training data). We generate lots of camera poses and paired edge images by uniformly sampling these free parameters.

B. Deep Edge Feature Extraction

Given edge images and their poses, we learn compact features via a siamese network [21]. The input of the network is a pair of edge images. The label is similar or dissimilar. A pair of edge images is set as similar if their pan, tilt and focal length differences are within pre-defined thresholds, and vice versa. The siamese network has two branches each of which is a convolutional neural network $f_w(\cdot)$. We want to ensure an image x_1 is closer to another image x_2 when the camera poses are similar. The loss function is:

$$\begin{aligned} \mathcal{L}(w, x_1, x_2, y) &= y D_w(x_1, x_2) \\ &\quad + (1 - y) \max(0, m - D_w(x_1, x_2)), \end{aligned} \quad (5)$$

where $D_w(x_1, x_2) = \|f_w(x_1) - f_w(x_2)\|_2^2$, y is the similar/dissimilar label and m is a margin (1 in this work).

We have tried many network structures for the siamese network. We found max-pooling and batch normalization have slightly negative impacts on the performance, which agrees with [24]. The network consists of 5 stride-2 convolutions (kernel size 7, 5, 3, 3, 3) followed by a 6×10 convolution and a $L2$ normalization layer. The learned feature dimension is experimentally set as 16. The siamese network also extracts features from detected edge images in the testing.

C. Two-GAN Model for Field Marking Detection

We extend the work of [4] to detect field markings using a two-GAN model. We first use a segmentation GAN to segment the playing surface from the whole image. Then, we use a detection GAN to detect field markings from the playing surface. The motivation of using two GANs is to

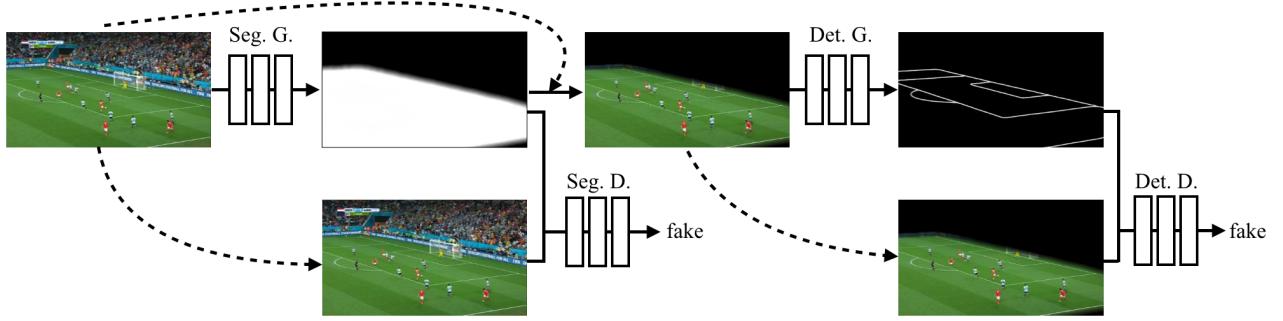


Fig. 4: The two-GAN model. We use a two chained conditional GAN to detect field markings. “Seg.” and “Det.” are short for segmentation and detection, respectively. “G.” and “D.” are short for generative network and discriminative network, respectively.

avoid the negative influence of background objects (*e.g.* white lines on commercial boards).

Figure 4 shows the structure of our two-GAN model in which each GAN is a conditional GAN [19]. The first GAN (segmentation GAN) segments foreground (*e.g.* grassland in soccer games) areas from the input RGB image and outputs a mask image. The second GAN (detection GAN) detects field markings from the foreground image. Each GAN has a discriminator network. The segmentation discriminator predicts whether the mask image is real or fake. The detection discriminator predicts whether the edge image is real or fake.

The two-GAN model is trained from scratch. The two GANs are first trained independently and then are merged via joint training. In training, we found the segment boundary (grassland vs. non-grassland) has considerable influence on the detection. When a binary segmentation boundary is used, the detection GAN tends to memorize the boundary and that will cause artifacts in testing. We solve this problem by using a soft (alpha-blending) boundary. The hard foreground is 1 and the hard background is 0, with linearly interpolated values between them. We randomly set the width of the alpha-blending band in the range of [30, 50] pixels to prevent the detection GAN from memorizing the bandwidth.

D. Camera Pose Optimization

Given a detected edge image, we first extract its feature using the siamese network from Section III-B. Then we retrieve the initial camera pose from the feature-pose database. After that, we optimize the camera pose by applying the Lucas-Kanade algorithm [25] on two distance images that are from the testing edge image and the retrieved edge image.

We use a truncated distance (*i.e.*, L2 norm) images [26] to overcome the narrow range of gradients in edge images [27], [28]. With the distance image, we use the Lucas-Kanade algorithm [25] to estimate the homography matrix from the nearest neighbor image to the testing image and use the chain rule to get the refined camera pose.

IV. EVALUATION AND EXPERIMENTS

World Cup dataset This dataset was collected by Homayounfar *et al.* [3] from World Cup 2014. The dataset has 10 games of 209 images for training and 186 images from 10

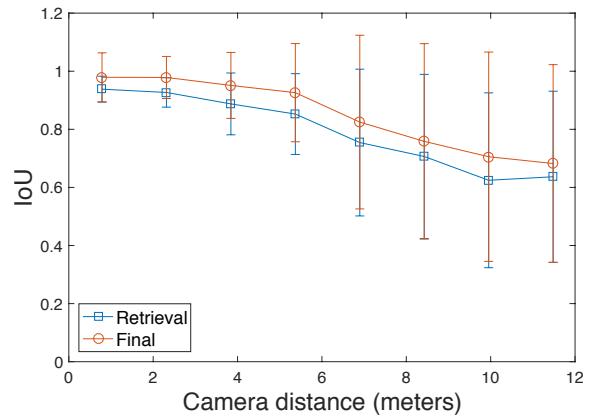


Fig. 5: Calibration accuracy (IoU_{part}) as a function of quantized camera distance. The blue line shows the retrieval results. The red line shows the refined results. Error bars show standard deviation. Best viewed in color.

other games for testing. The images consist of different views of the field with different grass textures, lighting patterns and heavy shadows. We pre-process the training set to obtain following camera configurations: camera location distribution $\mathcal{N}(\mu, \sigma^2)$ ($\mu \approx [52, -45, 17]^T$ and $\sigma \approx \pm[2, 9, 3]^T$ meters); pan, tilt and focal length ranges ($[-35^\circ, 35^\circ]$, $[-15^\circ, -5^\circ]$ and [1000, 6000] pixels, respectively). We found these camera parameters are typical settings for many cameras in soccer games.

Error metric We use the intersection over union (IoU) score to measure the calibration accuracy of different methods. The IoU is calculated by warping the projected model to the top view by the ground truth camera and the estimated camera. Homayounfar *et al.* [3] measures the IoU on the whole area of the model, while Sharma *et al.* [4] measures the IoU only on the area that is visible in the image. We denote these two metrics by $\text{IoU}_{\text{whole}}$ and IoU_{part} , respectively. For a fair comparison, we report both metrics for our method.

A. Synthetic Data Experiments

We conducted experiments on synthetic data to test the robustness of our method. We fix the training camera location

and we vary the displacement between the testing camera and the training camera. This experiment measures the robustness of our method to the displacement of the camera.

First, we generate a training set in which the cameras are located at the same mean location μ as the World Cup dataset. Then, we generate a testing dataset whose camera locations are $\mu + \mathbf{d}$ in which \mathbf{d} is a random displacement. We randomly generate other parameters (*e.g.* pan, tilt and focal length ranges) as those on the World Cup dataset. For example, the focal length is uniformly sampled from the range of [1000, 6000]. The training dataset size is 10,000 and the testing dataset size 1,000. The image resolution is 1280×720 . Then, we estimate the camera poses of the testing set using our method. We report the calibration accuracy as a function of camera distances (L2 norm of camera displacement \mathbf{d}).

Figure 5 shows the results. In the figure, the horizontal axis is the quantized camera distance and the vertical axis is the calibration accuracy (mean and standard deviation of IoU_{part}). The figure shows that our method achieves high accuracy ($\text{IoU}_{\text{part}} \geq 92\%$) when the camera displacement is within 5 meters. We believe this is useful in practice because the camera location generally changes its location in a small range for different games. When the camera location is far from the ground truth camera (*e.g.* about 10 meters), the calibration accuracy gradually drops to about 70%.

B. Real Data Experiments

We evaluate our method on the World Cup dataset and compare with state-of-the-art methods [3] and [4].

a) Training and testing process: In training, we sample 100,000 camera poses using our camera pose engine. The camera centers are sampled from the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. The pan, tilt and focal length values are sampled from the uniform distributions of $[-35^\circ, 35^\circ]$, $[-15^\circ, -5^\circ]$ and $[1000, 6000]$ pixels, respectively. The tilt of camera base ϕ' is fixed (-90°) and the roll angle is a random value from $[-0.1^\circ, 0.1^\circ]$.

We generate edges images (1280×720 resolution) for sampled camera poses. Then, the edge images are resized to 320×180 and are used to train a siamese network. The output of the siamese network is a 16-dimension deep feature. The deep features and the camera poses are stored in a feature-pose database.

In testing, we first detect field markings using the two-GAN model on the resized images (256×256). Then, we query an initial camera pose from the feature-pose database using the deep feature of the detected edge image. After that, we refine the camera pose by the distance image and the LK algorithm.

b) Main results: Table I shows the main result on the World Cup dataset. We compare our method with previous methods on mean and median IoU scores. Our method is significantly more accurate (89.4 vs. 83) than the DSM method [3] in mean $\text{IoU}_{\text{whole}}$. Our method is also more accurate (94.5 vs. 91.4 in mean IoU_{part}) than [4]. Comparing with [4], our

Method	$\text{IoU}_{\text{whole}}$		IoU_{part}	
	Mean	Median	Mean	Median
DSM [3]	83	—	—	—
Dict. + HOG [4]	—	—	91.4	92.7
Ours	89.4	93.8	94.5	96.1

TABLE I: Comparison on the World Cup dataset [3]. The results of DSM and Dict. + HOG are taken from their respective papers. The best performance is highlighted by bold.

Method	Mean IoU_{part} (%)	Δ
Ours	94.5	—
Ours w/o segmentation	90.2	4.3
Ours w/o LK warp	91.5	3.0
Ours w/o segmentation and LK warp	88.9	5.6
Ours replace deep feature with HOG	94.5	0.0

TABLE II: Components analysis on the World Cup dataset. The last column shows the improvements.

method also has a much simple and interpretable camera pose engine.

Table II shows the component analysis of our method. It shows that both the segmentation GAN and the LK warp contribute to the final result. On this dataset, it is interesting to see that the HOG feature is as good as our method in accuracy but is much less compact (1,860 vs. 16). It shows that the HOG feature is a strong baseline on this dataset.

c) Qualitative results: Figure 6 shows the qualitative results of our method on the soccer dataset. Our method achieves very high accuracy when the field-of-view is wide (see the first and the third column) or when more field markings are visible (see the second column). When the field markings are not evenly distributed in the image, our method has a lower accuracy (*e.g.* the fifth column in which the bottom touch lines are not visible in the image). We also found that the heavy shadow is the main reason for incorrect field marking detection. However, our method can estimate camera poses in these difficult situations (see the third column). More results are in the supplementary video.

Volleyball dataset results: This work focuses on soccer games. However, we also tested our method on a volleyball dataset. The volleyball dataset is collected from the volleyball action recognition dataset of Ibrahim *et al.* [29]. The dataset has 47 games and each game has 10 images. We randomly separated these games into two sets: 24 games for training and 23 games for testing. This data is less challenging than the soccer dataset in terms of lighting conditions. However, it is more challenging in terms of the number of visible field markings. We set the parameters of our method similar to those on the soccer dataset except the camera location, pan, tilt and focal length ranges. Our method achieves very high performance with mean and median IoU_{part} of 97.6% and 98.8%, respectively. Figure 7 shows qualitative results on the volleyball dataset.

d) Implementation details: Our approach (except the two-GAN model) is implemented in Matlab on an Intel 3.0 GHz CPU, 16GB memory Mac system. In the current implementation, the speed is not optimized. In testing, the

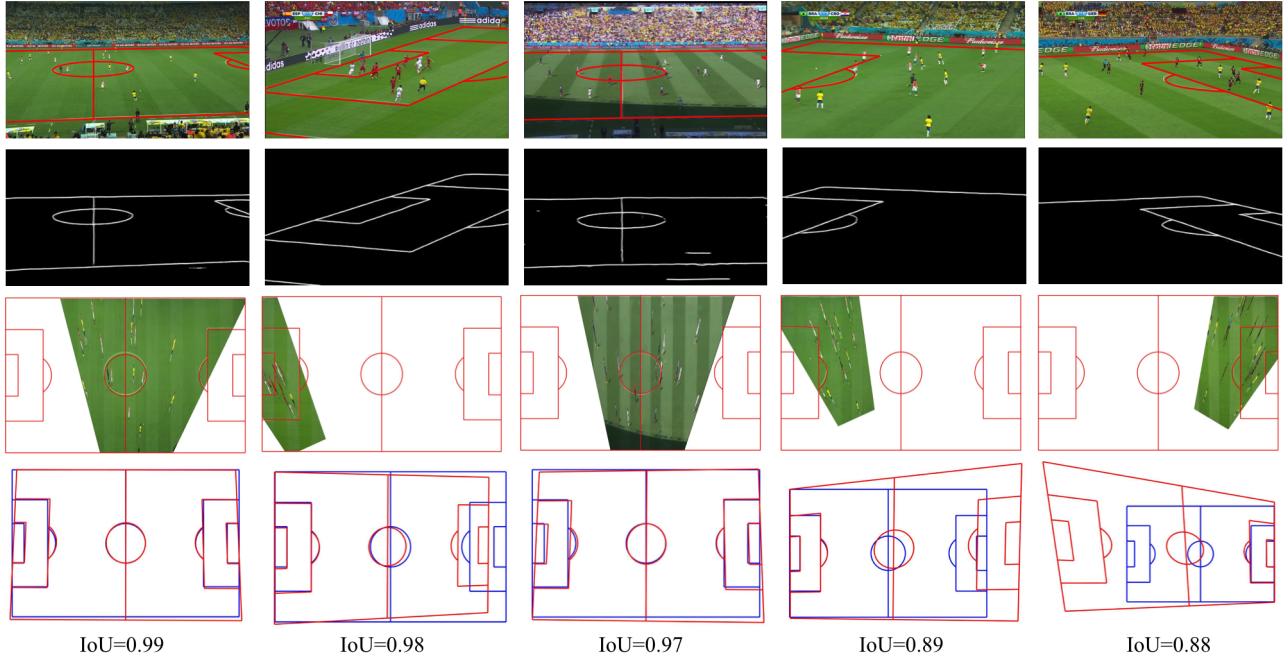


Fig. 6: Qualitative results. First row: field template overlaid on the image using the estimated camera poses. Second row: detected field markings. Third row: warped image on the template. Fourth row: IoU_{part} .



Fig. 7: Qualitative results on the volleyball dataset. The green lines are overlaid by estimated camera poses.

running time is about 0.5 seconds/frame. We will make the implementation publicly available.

The two-GAN model implementation is based on the pix2pix network [19] (PyTorch version). In the pix2pix network, both generator and discriminator use modules of the form convolution-BatchNorm-Relu. The generator uses skip connections and follows the shape of a U-Net [30]. In training, we set the weight of L1 loss $\lambda = 100$. We alternate between one gradient descent step on D , then one step on G . The losses of the segmentation GAN and the detection GAN have equal weights. We set the batch size to 1 because of limited GPU memory (12GB). The number of epochs is 200 and the learning rate linearly decays from 0.0002. We augment the training set by first resizing the image to 300×300 then randomly cropping it to 256×256 .

The final result is not sensitive to the parameters of our method. For example, we found 30-50 pixels work well for the distance threshold in the truncated distance image. For the siamese network, the input dimension is 320×180 and output dimension is 16. For the similar camera poses, the thresholds of pan, tilt and focal length differences are 1° , 0.5° and 30 pixels, respectively.

e) Discussions: Although our method achieves the state-of-the-art accuracy on a standard soccer dataset, it has several limitations at the current stage. For example, our method still requires human annotations to train the field marking detection network. One possible improvement is to train the field marking detection network using purely synthetic data, for example, from realistic rendering engines [31]. Moreover, the accuracy of our method relies on the field marking detection result. It fails when there is heavy rain/snow in the playing ground.

V. CONCLUSION

In this work, we proposed an automatic sports camera calibration method by developing a novel camera pose engine. The method is highly automatic and requires very few human interactions. It has achieved the best performance on a standard soccer dataset and very high performance on a volleyball dataset. For future work, we would like to synthesize data from realistic camera pose engines (*e.g.* from sports video games). We also would like to extend our method to other sports such as basketball, ice hockey and US football.

REFERENCES

- [1] J. Chen and J. J. Little, "Where should cameras look at soccer games: Improving smoothness using the overlapped hidden Markov model," *Computer Vision and Image Understanding (CVIU)*, vol. 159, pp. 59–73, 2017.
- [2] P.-C. Wen, W.-C. Cheng, Y.-S. Wang, H.-K. Chu, N. C. Tang, and H.-Y. M. Liao, "Court reconstruction for camera calibration in broadcast basketball videos," *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 22, no. 5, pp. 1517–1526, 2016.
- [3] N. Homayounfar, S. Fidler, and R. Urtasun, "Sports field localization via deep structured models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [4] R. A. Sharma, B. Bhat, V. Gandhi, and C. Jawahar, "Automated top view registration of broadcast football videos," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.
- [5] J. Chen, F. Zhu, and J. J. Little, "A two-point method for PTZ camera calibration in sports," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.
- [6] K. Qiu, T. Liu, and S. Shen, "Model-based global localization for aerial robots using edge alignment," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1256–1263, 2017.
- [7] L. Meng, J. Chen, F. Tung, J. J. Little, J. Valentin, and C. W. de Silva, "Backtracking regression forests for accurate camera relocalization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [8] T. Cavallari, S. Golodetz, N. A. Lord, J. Valentin, L. Di Stefano, and P. H. S. Torr, "On-the-fly adaptation of regression forests for online camera relocalisation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [9] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [10] A. Rubio, M. Villamizar, L. Ferraz, A. Penate-Sanchez, A. Ramisa, E. Simo-Serra, A. Sanfeliu, and F. Moreno-Noguer, "Efficient monocular pose estimation for complex 3D models," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [11] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-DoF global localization in outdoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [12] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [13] J. Puwein, R. Ziegler, J. Vogel, and M. Pollefeys, "Robust multi-view camera calibration for wide-baseline camera networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2011.
- [14] 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.
- [15] B. Ghanem, T. Zhang, and N. Ahuja, "Robust video registration applied to field-sports video analysis," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.
- [16] R. Zeng, R. Lakemond, S. Denman, S. Sridharan, C. Fookes, and S. Morgan, "Calibrating cameras in poor-conditioned pitch-based sports games," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
- [17] J.-B. Hayet, J. Piater, and J. Verly, "Robust incremental rectification of sports video sequences," in *British Machine Vision Conference (BMVC)*, 2004.
- [18] V. Pătrăucean, P. Gurdjos, and R. G. Von Gioi, "A parameterless line segment and elliptical arc detector with enhanced ellipse fitting," in *European Conference on Computer Vision (ECCV)*, 2012.
- [19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [20] G. Thomas, "Real-time camera tracking using sports pitch markings," *Journal of Real-Time Image Processing*, vol. 2, no. 2-3, pp. 117–132, 2007.
- [21] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [22] A. Loquercio, M. Dymczyk, B. Zeisl, S. Lynen, I. Gilitschenski, and R. Siegwart, "Efficient descriptor learning for large scale localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [23] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3D pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [24] A. Doumanoglou, V. Balntas, R. Kouskouridas, and T.-K. Kim, "Siamese regression networks with efficient mid-level feature extraction for 3D object pose estimation," *arXiv preprint arXiv:1607.02257*, 2016.
- [25] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *International Journal of Computer Vision (IJCV)*, vol. 56, no. 3, pp. 221–255, 2004.
- [26] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of Computing*, vol. 8, no. 19, pp. 415–428, 2012.
- [27] P. Carr, Y. Sheikh, and I. Matthews, "Point-less calibration: Camera parameters from gradient-based alignment to edge images," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2012.
- [28] K. Rematas, I. Kemelmacher-Shlizerman, B. Curless, and S. Seitz, "Soccer on your tabletop," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [29] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference On Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2015.
- [31] A. Shafaei, J. J. Little, and M. Schmidt, "Play and learn: Using video games to train computer vision models," in *British Machine Vision Conference (BMVC)*, 2016.