Efficient One-Shot Sports Field Image Registration with Arbitrary Keypoint Segmentation

Nicolas Jacquelin, Romain Vuillemot, Stefan Duffner

EFFICIENT ONE-SHOT SPORTS FIELD IMAGE REGISTRATION WITH ARBITRARY KEYPOINT SEGMENTATION

Nicolas Jacquelin, Romain Vuillemot, Stefan Duffner

Centrale Lyon, INSA Lyon, CNRS, UCBL
LIRIS - UMR5205, F-69081 Ecully, France
{nicolas.jacquelin, romain.vuillemot, stefan.duffner}@liris.cnrs.fr

ABSTRACT

Automatic sports field registration aims at projecting a given image taken with unknown camera parameters to a known 3D coordinate system in order to obtain higher-level information like the position and speed of players. Existing methods generally detect specific visual landmarks on the field and then use an iterative refinement to get closer to the desired calibration. They are usually only compared in terms of precision on a standard benchmark without considering other metrics. However, execution speed is also important, mainly in the context of live broadcast TV and sports analysis. This work introduces a new automatic field registration method achieving excellent performance on the WorldCup Soccer benchmark, while neither depending on specific visible landmarks nor any refinement, resulting in a very high execution speed one-shot model. Finally, to complement the usual Soccer benchmark, we introduce a new Swimming Pool registration benchmark which is more challenging for the task at hand. Code and dataset available at `https://github.com/njacquelin/sports_field_registration`.

***Index Terms*—** registration, real-time, sports, dataset

1. INTRODUCTION

Field registration designates the common method to align the visible field in a frame to an absolute field template. It can convert the position of players in an image into their position in the field, inferring their speed and acceleration. As sports fields are planar, this is a linear projection called homography. To compute the homography matrix, one can map points from the original image to positions on the template. This gives a first projection, that requires refinement to fit more precisely the image to the template. Automatic methods [1, 2, 3, 4, 5] tend to decompose the task into a similar two-stage process: first getting an initial projection, then several refinement steps to get more precise results. This second stage takes much longer, 96% of the total processing time according to [4].

Our work introduces an automatic field registration method which does not need this costly refinement step. It learns to segment the input image into a map that highlights a specific (grid-like) pattern corresponding to points on the 3D field plane (see Fig. 2). Our approach can be applied to any type of 2D sports field with TV streams or side stadium view. While maintaining excellent precision on the WordCup Soccer benchmark [6], it achieves an inference speed of around 50 FPS on rather modest hardware (see Figure 2). This is important as it is critical to calibrate a field in real time, e.g. 1) for live-TV visualisation tools, or 2) for athletes to get a quick feedback on their performance during training, and 3) cameras positions and fields characteristics change during shots and across competitions.

WordCup Soccer benchmark [6] is the only public dataset that has been widely used in the literature, although some private datasets have been introduced for registration [3, 4]. However, a soccer field is relatively simple in appearance: a bi-axial symmetry with many unique visual local patterns. Thus we introduce a more challenging benchmark for Olympic swimming pool registration. Indeed, a swimming pool contains many repetitive patterns at different places in the pool (see Fig. 1) leading to ambiguities in the image and making the registration difficult. Therefore, we hope this will push forward the research on generic and robust sports field registration methods.

In summary, our contributions are:

- a new benchmark for swimming pool registration with new spatial and textural challenges,
- a new efficient sports field registration method that can be applied to any type of sports and reaches high execution speed and state-of-the-art precision.

2. RELATED WORK

The first sport fields registration methods [7, 8] relied on lines and circle detection using Hough Transforms [9]. The detected patterns were used as keypoints and, combined with RANSAC [10], enabled to compute a homography giving the absolute position of the camera view on the field. Other methods [11, 12] relied on sparse human video annotation (e.g. one frame per second of video) and used SIFT [13] to determine the camera shift between calibrated frames and the others.

Table 1. Statistics of the RegiSwim^{500}dataset. The races contain important lighting, textural, and spatial variations.

	#images	#races	images / s
Train Standard	226	6	1/3
Train Sequential	150	4	5
Train Merge	329	6	5 & 1/3
Test	174	3	5

Using more recent deep learning approaches, fully automated robust methods appeared. Homayounfar *et al.*[6] created a segmentation map and used a Markov Random Field and a SVM to compute the parameters of the cameras, which determine the homography. Other works [1, 2, 3] used a similar deep segmentation model approach using synthetic datasets. They generated a set of synthetic field views with varying camera angles, extracted features from them, and associated them to their homography (easy to obtain in a synthetic environment). At inference time, they generated similar features from real images, which they compared to their database, giving a good initial homography. Then they adjusted this homography by comparing their input image to their dataset template. In fact, the idea of refining an initial result is present in all recent works of the domain, with different methods for the initialisation. For instance, Jiang *et al.* [5] used a neural network to directly estimate the image homography. They used another model to refine the matrix by comparing the image and a template projected in the same point of view. Other approaches are based on field keypoint detection. Citraro *et al.*[14] used visual landmarks on the field (mostly line intersections). The main limitation of using visible elements is that the image may not show enough visual keypoints. Nie *et al.*[4] directly address this problem, creating a generic template made of equally distributed points across all the field, which is similar to our proposed approach. The key difference is that in [4] each point is disconnected from the others, despite spatial regularities.

3. A MORE CHALLENGING BENCHMARK

As CV techniques develop, the field registration task reaches excellent performance on existing benchmarks, which does not allow to compare the newest methods with significant margins. To adapt to this rapid evolution of registration techniques, we propose to study the unusual sport environment of a swimming pool. As explained in Fig. 1, it contains many challenges, namely positioning along the Y axis (A, B), positioning along the X axis (C, D) - both due to landmarks repetitions - and unstable background (wavelets, reflections, light problems etc.). The level of zoom and distance from the pool also change a lot depending on the competitions. Finally, swimmers occlude part of the landmarks. To articulate these challenges, we introduce the RegiSwim^{500}dataset, a swimming pool registration benchmark containing 503 manually annotated images of international events. The source videos are included to enable the use of temporal information. Numeric details of the dataset are summarized in Table 1. There are two train sets: standard and sequential. The first one has been created in a way similar to WorldCup Soccer and aims to be generic. The second one has a temporally dense annotation (5 frames per second), which can be used to train temporal models. These two can be merged to create a bigger, temporally heterogeneous dataset. Finally, the test set is also densely annotated, as this makes no difference on a standard benchmark perspective, but it allows also sequential models evaluation. The github page gives an open link to the dataset.

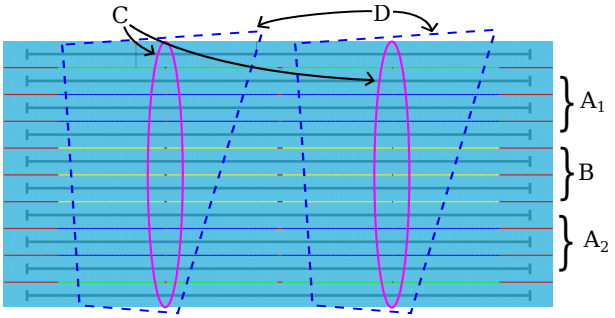

Fig. 1. Local appearance ambiguities of a swimming pool. $A_{1,2}$: 4 exact same lines at different places. B: 3 exact same lines in the middle. Both A and B create line mismatch problems. C: the 15m and 35m markers are identical. D: an example of 2 different camera view projections on the pool that display the exact same content, despite being at two completely separate places of the pool. Best viewed in color.

4. REGISTRATION METHOD

To find the homography from a camera view to a standard top-view, our method uses point associations: the model learns a mapping between keypoint positions in the input image and in a top-view template. The overall pipeline is explained in Fig. 2. The main emphases of this work is computational efficiency. Other methods [3] claim a fast inference speed but require powerful hardware which may not be accessible in practice. Our method uses a much smaller one-shot model (*i.e.*: without iterative refinement) such that real-time registration is possible with modest hardware (1080 GTX with 8GB).

4.1. Template Heatmap

This work proposes a model that, given an input image of a sports field, outputs a $(W \times H \times D)$ heatmap of keypoints, W and H being the width and height of the input image, D being the keypoints encoding dimension. The keypoints do not necessarily represent a visual landmark on the field: they are spread regularly, creating a grid (Fig. 2, "Grid Template"). One unique aspect of this method is the way it encodes the

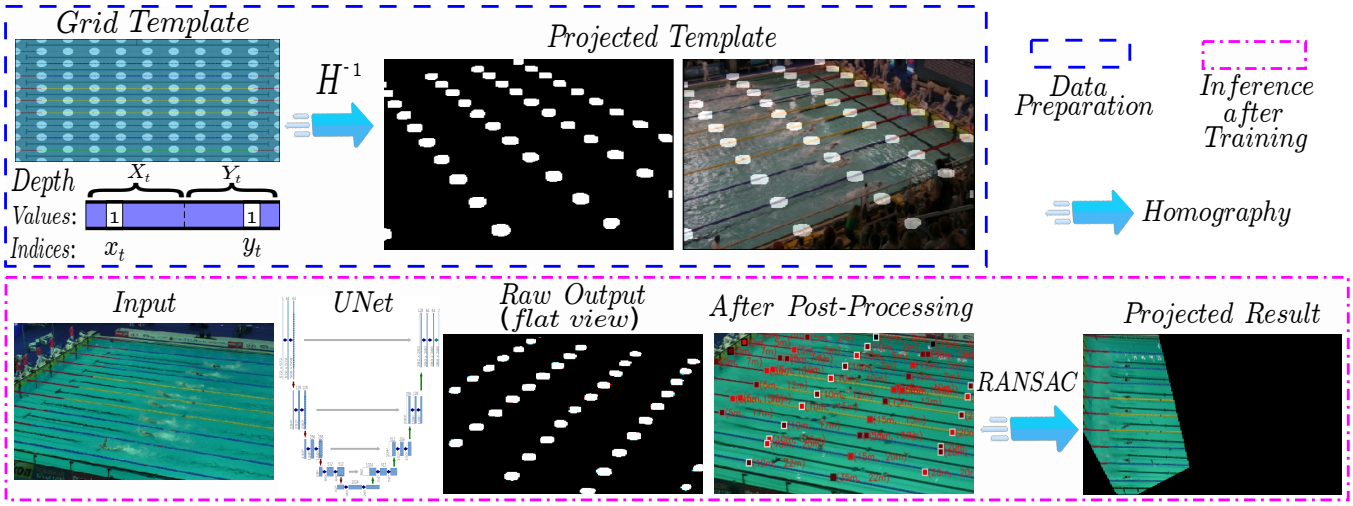

Fig. 2. Top: data preparation. A generic template with regularly spaced keypoints is created. The template's depth encodes the keypoints' position in the top-view frame. For each image in the dataset, a corresponding projection of the template is created. Bottom: inference. The model generates a heatmap of keypoints. These keypoints have a known position in the image as well as a known position in the absolute pool coordinate system, encoded in their depth. Using the RANSAC algorithm, they enable the homography matrix estimation, giving the final projection of the input image in the top-view frame. Best viewed in color.

points. The depth vector is composed of two subsets: X_t and Y_t. They are one-hot vectors whose maxima index (x_t, y_t) encode one line/column along the grid axis: a combination of any value of x_t and y_t gives a node position in the top-view frame (Fig. 2, "Depth").

Compared to having C channels for the C keypoints in the template, as in [4], this method has speed benefits: it avoids the depth to increase geometrically with the number of keypoints. A pair of one-hot vectors only linearly increases the output depth, for the same level of encoding. This improves the speed and scaling of the solution. For instance, a grid of (15×7) contains 105 channels in [4] but only 22 in ours. In addition, as each channel does not only represent one point, but one line/column in the field, their semantic meaning is more interesting and enables a better scene understanding.

4.2. Data Generation and Model Training

Once the top-view template is created, the data generation can start using a dataset that contains images with their corresponding homography matrix. The matrix is used to project the template into the point of view of its image (Fig. 2, "Projected Template"). With such projection only semantic information has to be inferred.

Our approach relies on a UNet architecture [15], which is widely used for image segmentation. The cross-entropy loss is used to train the pixel-wise keypoints one-hot classification. As there is no "background" class (which would be over-representated in the data), this loss is only applied at the ground truth keypoints location, using a mask. To ensures that the keypoints are at the correct place, the binary cross-entropy loss (BCE) is used. To do so, the ground truth (Truth) and output (Out) heatmaps are flattened with a depth-wise *MAX* operation. The 2D resulting heatmaps are compared, in order to align the estimated "blobs" with the expected ones. Formally:

$$
\begin{aligned}
L_{class}^{axis} &= CrossEntropy(Out, Truth) * Mask_{keypoints}^{truth} \ , \\
L_{pos} &= BCE(Max_{depth}(Out), Max_{depth}(Truth)) \ , \\
L_{total} &= L_{class}^{x} + L_{class}^{y} + \lambda \cdot L_{pos} \ ,
\end{aligned}
$$

with $\lambda \in \mathbb{R}$ being a weighting coefficient.

4.3. Post-Processing

To extract the keypoints' absolute position from the heatmap, one could study each pair of (X,Y) channels to verify if each (x, y) point is represented. This results in a $X_G \times Y_G \times K$ complexity (X_G and Y_G being the template grid resolution, and K the number of keypoints to be found). We propose a much faster algorithm whose complexity is in $(X_G + Y_G) \times K$ (the K operations are parallelizable). A depth-wise *MAX* operation is applied to Out, the whole output, resulting in Out_{flat}, a 2D heatmap (the *Max* operation is extremely well optimized in processors and insignificant compared to the rest). Its M local maxima are identified and if they exceed a certain threshold, their (x^m, y^m) positions are kept. On Out, the depth vectors at these (x^m, y^m) positions are isolated. Their one-hot vectors return the index of their most activated dimension, (x_t^m, y_t^m), the position on the top-view template. Based on these $((x^m, y^m), (x_t^m, y_t^m))$ pairs, RANSAC [10] can be used to compute the homography matrix. This is formally described in the Algorithm 1.

Table 2. Quantitative results on Soccer World Cup and RegiSwim^{500}datasets. Best in bold. Real-time methods underlined.

Method	Benchmark	IOU^{avg}_{part}	IOU^{med}_{part}	IOU^{avg}_{whole}	IOU^{med}_{whole}	FPS	Memory - GPU
Citraro *et al.*[14]	WorldCup	93.9	95.5	-	-	9	NA - Titan RTX
Sha *et al.*[3]	WorldCup	94.2	95.4	83.2	84.6	<u>250</u>	**48GB - Titan RTX**
Chen *et al.*[2]	WorldCup	94.5	96.1	89.4	**93.8**	2	16GB - NA
Jiang *et al.*[5]	WorldCup	95.1	96.7	89.8	92.9	0.74	8GB - 1080 GTX
Nie *et al.*[4]	WorldCup	**95.9**	**97.1**	**91.6**	93.4	2	8GB - 1080 GTX
Ours, soccer field	WorldCup	94.6	95.9	81.2	86.0	<u>50</u>	8GB - 1080 GTX
Ours, swimming pool	RegiSwim^{500}	83.3	94.7	72.6	91.5	<u>50</u>	8GB - 1080 GTX

Algorithm 1 Fast identification of keypoints on a heatmap. Det returns the position of the local maxima in the heatmap. The correspondence table Tab associates to each channel an absolute position in the field template.

Require: Model Output Out, Threshold T, maxima detector Det, Correspondence Table Tab
  $Pairs \leftarrow \emptyset$
  $Out_{flat} \leftarrow Max_{depth}(Out)$
  $Max\_List \leftarrow Det(Out_{flat})$
  for (x^m, y^m) in Max_List **do**
    if $Out_{flat}[x^m, y^m] < T$: SKIP
    $depth\_vector \leftarrow Out[x^m, y^m]$
    $X_t, Y_t \leftarrow depth\_vector$
    $x^m_t \leftarrow Tab(argmax(X_t))$
    $y^m_t \leftarrow Tab(argmax(Y_t))$
    $Pairs \leftarrow Pairs \cup ((x^m, y^m), (x^m_t, y^m_t))$
  end for
  $Homography\ Matrix \leftarrow RANSAC(Pairs)$
  return $Homography\ Matrix$

5. RESULTS

The model was trained for 150 epochs with Adam optimizer [16]. The learning rate started at $1e{-}3$ for 50 epochs and then decreased to $1e{-}4$ for the remaining 100 epochs, with a batch size of 16. Coefficient λ is set to 2. For soccer, the grid size chosen is $(15{\times}7)$ and for swimming it is $(11{\times}11)$. Our metric is the Intersection Over Union (IOU) between binary masks of the ground truth top view and the estimated homography. This is either done with only the visible field (IOU_{part}) or using the whole field (IOU_{whole}). The average and median of these metrics are computed on the test dataset. Results are shown in Table 2.

Although our approach does not quite attain the top results from the literature (see Table 2), it is still among the best ones. This is remarkable, considering it contains no refinement process while all the other methods do. However, this impacts the IOU_{whole} metric, where the slightest shift on the visible side of the field has big repercussions on the other side. Nonetheless, this second metric can be considered less interesting for real-world applications, such as placing the players on a field, as they must be visible on image to be detected in the first place. These results might be improved using methods such as self-training on unlabelled data.

Regarding speed, our model is one of the only two exceeding real time ($>$ 25 FPS), although it has been tested on the least powerful hardware according to benchmarks [17, 18]. Looking in the details, one can even argue that our model is faster than Sha *et al.*[3] on the same hardware. Indeed, our architecture is a subset of theirs, to which they add 2 more CNNs, a Spatial Transformer Network, and an exhaustive search among field templates. All these additional steps have a significant time cost and our method might be faster by up to this amount. The model's speed could be increased even more using distillation [19] to train a more condensed, shallower and faster version of UNet.

Naturally, for our more challenging RegiSwim^{500}dataset, the performance is lower. Our model handles correctly Y-axis challenges (A and B in Fig 1) and lighting problems, mostly because of the grid density and distribution, which prevents focusing on a single part of the image. The big difference between the mean and median result is due to multiple left-right inversions: images with an IOU score of 0, reducing the mean but not the median as they are a minority. These are quite difficult to prevent in a pool (challenges C and D in Fig. 1). This first baseline clearly shows the challenges and limitations raised by this new benchmark.

6. CONCLUSION

This work introduces an efficient and precise method for automatic sports fields registration, which reaches very good performance and real-time inference speed. The RegiSwim^{500} dataset has been introduced and made publicly available in order to improve the registration challenge. Future works will include ways to optimize even more the model's inference speed, and new methods to increase its precision.

7. ACKNOWLEDGEMENTS

We thanks the members of the Neptune project for their interesting advices regarding swimming. This work was funded by the CNRS.

8. REFERENCES

[1] Rahul Anand Sharma, Bharath Bhat, Vineet Gandhi, and C. V. Jawahar, "Automated top view registration of broadcast football videos," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 305–313.

[2] Jianhui Chen and James J. Little, "Sports camera calibration via synthetic data," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pp. 2497–2504.

[3] Long Sha, Jennifer Hobbs, Panna Felsen, Xinyu Wei, Patrick Lucey, and Sujoy Ganguly, "End-to-end camera calibration for broadcast videos," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13624–13633.

[4] Xiaohan Nie, Shixing Chen, and Raffay Hamid, "A robust and efficient framework for sports-field registration," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1935–1943.

[5] Wei Jiang, Juan Camilo Gamboa Higuera, Baptiste Angles, Weiwei Sun, Mehrsan Javan, and Kwang Moo Yi, "Optimizing through learned errors for accurate sports field registration," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 201–210.

[6] Namdar Homayounfar, Sanja Fidler, and Raquel Urtasun, "Sports field localization via deep structured models," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4012–4020.

[7] Fei Wang, Lifeng Sun, Bo Yang, and Shiqiang Yang, "Fast arc detection algorithm for play field registration in soccer video mining," in *2006 IEEE International Conference on Systems, Man and Cybernetics*, 2006, vol. 6, pp. 4932–4936.

[8] Hyunwoo Kim and Ki Sang Hong, "Soccer video mosaicing using self-calibration and line tracking," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, 2000, vol. 1, pp. 592–595 vol.1.

[9] Richard O Duda and Peter E Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[10] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, jun 1981.

[11] Elan Dubrofsky and Robert J. Woodham, "Combining line and point correspondences for homography estimation," in *Advances in Visual Computing*, Berlin, Heidelberg, 2008, pp. 202–213, Springer Berlin Heidelberg.

[12] Ankur Gupta, James J. Little, and Robert J. Woodham, "Using line and ellipse features for rectification of broadcast hockey video," in *2011 Canadian Conference on Computer and Robot Vision*, 2011, pp. 32–39.

[13] David G Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*. IEEE, 1999, vol. 2, pp. 1150–1157.

[14] Leonardo Citraro, Pablo Márquez-Neila, Stefano Savarè, Vivek Jayaram, Charles Dubout, Félix Renaut, Andrés Hasfura, Horesh Shitrit, and Pascal Fua, "Real-time camera pose estimation for sports fields," *Machine Vision and Applications*, vol. 31, no. 16, 03 2020.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234–241, Springer International Publishing.

[16] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2015.

[17] "1080 ti vs rtx 2080 ti vs titan rtx deep learning benchmarks with tensorflow - 2018 2019 2020," `https://bizon-tech.com/blog/gtx1080ti-titan-rtx-2080-ti-deep-learning-benchmarks`, Accessed: June 2022.

[18] "Deep learning gpu benchmarks 2021," `https://www.aime.info/en/blog/deep-learning-gpu-benchmarks-2021/`, Accessed: June 2022.

[19] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015.