Pass Receiver Prediction in Soccer using Video and Players' Trajectories

A. Details of alignment between video and 2D trajectory

We used wide-angle videos that 20 players on the field were always visible. The expected role of these videos is to provide the model with appearance information such as the posture and facial orientation of the 20 players on the field in a match. However, since the bounding boxes that indicate the players' areas are not annotated in the video, we needed to develop an automatic annotation system using tracking data and basic computer vision. This is a general approach that can be applied to similar wide-angle videos and location information.

A.1. Obtaining videos of individual players

In order to obtain players' bounding boxes, we needed to obtain their coordinates from the videos: we tried to convert the tracking data described in the field coordinate system (x, y) into the video coordinate system (X, Y). A panoramic image showing the whole field was generated from each match video, and a homography matrix H_1 mapping between each frame of the match video and the panoramic image was calculated based on matching points obtained by SuperGlue which is a graph neural network for feature matching [5]. The homography matrix H_2 that maps the field image to the panoramic image was obtained by manually selecting corresponding feature points. Finally, the 2d trajectories were transformed from the field coordinate system into the video image coordinate system by Eq. (1).

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = H_1^{-1} H_2 \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{1}$$

However, due to camera distortion, the transformed coordinate points do not exactly match the player's area. We corrected this coordinate drift by using an object detector and point registration algorithms.

A.2. Using a deep object detection model

You Only Look Once (YOLO) is a high-performance object detection model that has been used for detecting persons and objects in sports video [4]. By applying YOLO version 5 (YOLOv5) [2] to the videos, we can obtain the position of any object and the exact bounding box. However, the necessary objects (20 players in our case) are not always detected, and since it is a detection model, it is difficult to track objects through time, as the detected objects are not ensured to be consistent frame by frame. On the other hand, the transformed coordinate points of each player from the tracking system are recorded for the 20 players, and individuals are tracked and identified, though the coordinate points do not perfectly match the detected positions by YOLOv5. Therefore, we assumed that the positions and bounding boxes detected by YOLOv5 are mostly accurate, and we tried to correct the miss detection and the positional shifts of the coordinates from the tracking system by solving the alignment problem between the two data.

A.3. Correction of missing points by ICP

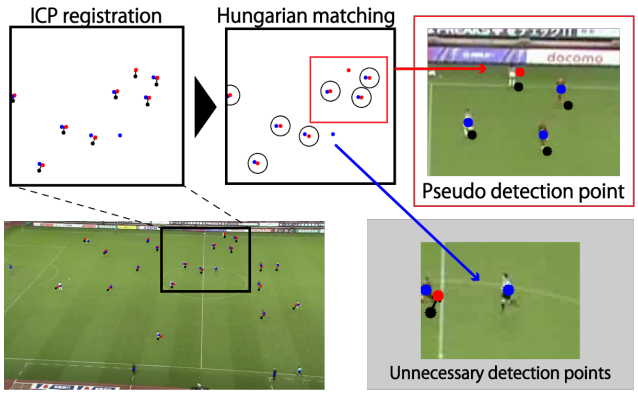


Figure 1. Addition pseudo-detected points by ICP (red squares) and removing unwanted points using Hungarian matching (gray squares).

To correct missing detected points by YOLOv5, we used iterative closest point (ICP) [6], which is a rigid point registration algorithm that is highly aware of the shape of a point cloud. In our process, this means that the formation of players is reflected in a rigid transformation matrix. Probable positions of undetected players can be estimated by spotting the players' coordinates that do not have the counterpart. Thus, we added these points to detected points by YOLOv5 as pseudo-detected points. However, unnecessary points such as referees are also moved by ICP. In order to

remove these points, we used Hungarian method. Also, ICP only estimates rigid transformation and it cannot address the uneven shift caused by camera distortion.

A.4. Removing unnecessary detection points using the Hungarian method.

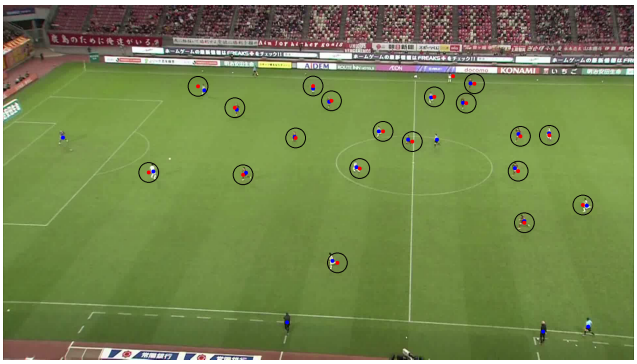
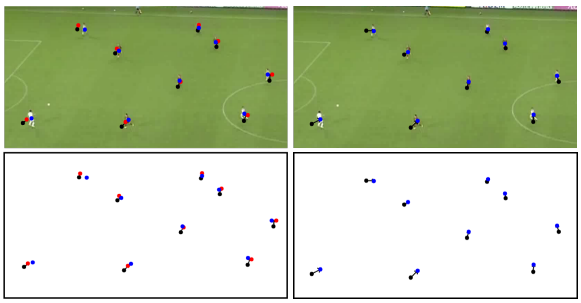


Figure 2. The result of hungarian matching between the detection points by YOLOv5 (blue) and moved points by ICP (red). Aligned points are surrounded with the black circles.

Applying hungarian matching between the detection points by YOLOv5 and moved points by ICP, there are two type of unaligned points: only detection points (blue points without black circle in Fig. 2) or only moved points (red points without black circle in Fig. 2). The first type is the part to be added as a pseudo-detection point as described in a previous section, and the second type can be judged as the points that are not necessary 20 players to be obtained from the tracking data, and by removing this unassigned points from the detection points of YOLOv5, all the detection points other than 20 players can be excluded. The correction process of missing points and unnecessary points is illustrated in Fig. 1.

A.5. Correction of uneven positional shift by CPD



(a) Result of ICP. (b) Result of CPD.

Figure 3. Each color dots means as same as those in Fig. 1. ICP cannot move players' coordinates (black dots) to detected points (blue dots), but CPD can.

Finally, we used coherent point drift (CPD) [3], which calculates the mobility of each point individually, to deal with non-uniform distortions. To obtain the bounding boxes of the newly detected players in this refinement, we referred to the size of the bounding of YOLOv5 detection that is closest to the player in interest.

B. Dataset statistic

We used the wide-angle videos of 25 matches in 3 stadiums. The number of teams and the number of matches included in 25 matches is Tab. 1. The first row indicates specific team and the second row shows the number of matches which the team was involved. Thus, our dataset has 16 teams. Each stadium has its own home team, which means that three specific teams played more games than the others. The names of the teams are given in alphabetical order so that they cannot be identified.

We used only the successful pass scenes.Each scene is a few seconds length, ranging from 1.0 to 7.0 seconds, and the pass occurs at the end of the scene. The number of scenes at each length is shown in Tab. 2.

C. The architecture of 3D CNN

The proposed model uses a 3D ResNet [1] to share the weights of the model when extracting the features of each player. However, since the network structure proposed in previous studies is redundant for the image size, we reduce the number of parameters by using up to the second Residual layer. The specific network structure of the 3D Resnet to be used is as follows: Tab. 3.

References

[1] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6546–6555, 2018. 2, 3

[2] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, Ayush Chaurasia, TaoXie, Liu Changyu, Abhiram V, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomammana, AlexWang1900, Jan Hajek, Laurentiu Diaconu, Marc, Yonghye Kwon, oleg, wanghaoyang0106, Yann Defretin, Aditya Lohia, ml5ah, Ben Milanko, Benjamin Fineran, Daniel Khromov, Ding Yiwei, Doug, Durgesh, and Francisco Ingham. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations, Apr. 2021. 1

[3] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), 2010. 2

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 1

team	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	total
match	10	2	10	2	3	2	2	2	1	9	1	1	2	1	1	1	25

Table 1. The number of teams included in the 25 games and the number of games played by each team. There are 16 teams (A-P) in our dataset.

	1sec	2sec	3sec	4sec	5sec	6sec	7sec	total
train	441	442	398	412	352	367	8499	10911
valid	54	58	59	54	37	57	1240	1559
test	136	98	112	94	97	95	2484	3116
total	631	598	569	560	486	519	12223	15586

Table 2. Number of scenes for each length.

Layer Name	Architecture
conv1	$7 \times 7 \times 7$, 64, stride 1(T), 2(XY)
2*conv2	$3 \times 3 \times 3$ max pool, stride 2
	$\begin{bmatrix} 3 \times 3 \times 3, 64, \text{stride 1(TXY)} \\ 3 \times 3 \times 3, 64, \text{stride 1(TXY)} \end{bmatrix} \times 2$
conv3	$\begin{bmatrix} 3 \times 3 \times 3, 128, \text{stride 2(TXY)} \\ 3 \times 3 \times 3, 128, \text{stride 1(TXY)} \end{bmatrix} \times 2$
	average pool, 64-d fc

Table 3. The architecture of 3D CNN used in our experiments, basing on 3D Resnet [1]

[5] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1

[6] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, oct 1994. 1