Report for the FIFA Skeletal Tracking 'Light' Challenge

Artur Xarles[1,2] Sergio Escalera[1,2,3] Thomas B. Moeslund[3] Albert Clapés[1,2]
[1]Universitat de Barcelona, Barcelona, Spain
[2]Computer Vision Center, Cerdanyola del Vallès, Spain
[3]Aalborg University, Aalborg, Denmark
arturxe@gmail.com, sescalera@ub.edu, tbm@create.aau.dk, aclapes@ub.edu

We present here the method and results of our solution to the FIFA Skeletal Tracking 'Light' Challenge.

1. Task Definition

The objective of this challenge is to generate 3D skeletal tracking data from input video sequences lasting between 20 and 40 seconds. These videos correspond to broadcast footage of football matches. In addition to the videos, participants are provided with bounding boxes for each player and referee on the field, the camera's intrinsic parameters, and the initial extrinsic parameters (i.e., rotation and translation) at the start of the sequence.

The task can be divided into two main components:

- **Camera Calibration:** Estimating the camera's pose (extrinsic parameters) for each frame of the video. This step enables the transformation of detections from camera space into real-world coordinates, effectively mapping the scene onto a 3D model of the football field.
- **Pose Estimation:** Estimating the skeletal pose of each player and referee within the camera space.

By combining the results of these two components, we can translate the estimated poses into the real-world coordinate system and thereby generate the final 3D representation of the field, complete with skeletal tracking of players and referees.

The evaluation of the task is based on the combination of two metrics: Global MPJPE (Mean Per Joint Position Error) and Local MPJPE.

2. Method

Due to time constraints, our approach focused solely on improving the **Camera Calibration** component of the task, while relying on the provided baseline for pose estimation.

Below, we describe the sequential improvements made over the baseline method, resulting in our final score. Results on the evaluated subset (as reported on Kaggle) are summarized in Table 1.

Baseline. The baseline estimates the camera pose (i.e. the rotation matrix R and translation vector t) by starting from the initial pose and computing the relative rotation between consecutive frames. To estimate this relative rotation, a set of reference points assumed to be static in the real world is needed. The baseline uses the corners of player bounding boxes as these reference points. However, since players are not generally static, these points are unreliable and result in inaccurate camera pose estimation. Consequently, this leads to poor results as shown in Table 1.

After the initial estimation, a refinement step is applied: known real-world field lines are projected into the camera view, and a line detector is run on the frame. The refinement aims to minimize the distance between the projected lines and the detected image lines, thereby improving alignment and calibration accuracy. This refinement step is omitted in submissions 1-3.

Submission-1 The first improvement focused on identifying truly static keypoints in the real world. To this end, we used keypoints associated with the field lines. These were obtained using the keypoint detector from https://github.com/mguti97/No-Bells-Just-Whistles (NBJW), which detects line intersections and specific positions along the field lines. By leveraging these field-based keypoints rather than relying on player-related keypoints, we achieved more accurate point correspondences between consecutive frames. Since these points are static in the real world, they provide more reliable information for estimating relative camera rotation. In cases where an insufficient number of keypoints was detected to compute the relative rotation, we defaulted to using the previously estimated camera pose to maintain robustness.

Submission-2 We identified two main limitations in the previous approach: (1) the accumulation of errors due to continuously chaining relative rotations to estimate the camera pose, and (2) the presence of frames with too

Method	Key Improvement	Global MPJPE ↓
Baseline	Uses player bounding box corners as reference points; no robust static anchors.	5.23
Submission-1	Replaces player corners with field-line keypoints using NBJW detector.	2.75
Submission-2	Hybrid strategy: combines relative and global pose estimation to reduce drift.	2.40
Submission-3	Adds temporal smoothing using an averaging kernel to reduce jitter.	2.38
Submission-4	Reintroduces field-line refinement and stricter keypoint filtering (min. 6).	1.82
Submission-5	Updates 3D mapping using a reliable pose-based reference point.	1.62

Table 1. Ablation study showing the effect of successive improvements to the camera calibration method. Performance is reported using Global MPJPE (lower is better) on the evaluation subset.

few detected keypoints, which increases the drift and further degrades accuracy. To address this, our next submission followed a hybrid strategy that combined relative and global camera pose estimation. While the previous submission relied solely on relative rotation between frames, here we also directly estimated the global camera pose using detected keypoints and their known real-world correspondences. This hybrid method worked as follows: when enough reliable keypoints were available, we estimated the camera pose via relative rotation. However, during frames with insufficient keypoints, we deferred pose estimation until a subsequent frame with enough keypoints was available. At that point, we performed a global pose estimation, thus reducing drift and mitigating error accumulation caused by earlier frames with not enough keypoints. This modification led to a noticeable improvement in performance, as shown in Table 1.

Submission-3 In this submission, we addressed the jitteriness in the estimated camera poses by applying a smoothing filter using an averaging kernel. This temporal smoothing led to a slight performance improvement, as shown in Table 1.

Submission-4 After achieving a sufficiently accurate camera pose estimation, we reintroduced the refinement step from the baseline. This step minimizes the distance between the projected field lines and the detected lines in the image. Additionally, we increased the minimum number of keypoints required to estimate the relative rotation to six, ensuring more robust estimations. These modifications resulted in a further improvement in performance.

Submission-5 The final modification to our method involved updating the reference point used to translate the predicted pose into real-world coordinates. In the provided baseline, this was done by assuming that the bottom-right corner of each player's bounding box lies at $z = 0$, and corresponds to a specific foot of the player. However, this assumption is overly simplistic, as the bottom-right corner does not consistently align with a specific foot. Instead, we selected a more reliable reference point from the estimated player pose: the joint that is furthest to the right, lowest, and closest to the camera. This point better aligns with the bottom-right of the bounding box. This refinement led to further performance gains, resulting in our final submission.

3. Final Comments

Although the final submitted method includes a relatively robust camera pose estimation, we would have liked to dedicate more time to the pose estimation component. We believe that further improvements in this area could have further reduced the final error metric, contributing to a stronger overall solution for this challenge.