Supplementary Material WorldPose: A World Cup Dataset for Global 3D Human Pose Estimation

Tianjian Jiang*[1], Johsan Billingham*[2], Sebastian Müksch[1], Juan Zarate[1], Nicolas Evans[2], Martin R. Oswald[1,3], Marc Polleyfeys[1,4], Otmar Hilliges[1], Manuel Kaufmann[1], and Jie Song†[1]

[1] ETH Zürich
[2] FIFA
[3] University of Amsterdam
[4] Microsoft

In this supplementary material, we provide additional information to complement the main text: 1) more implementation details about the data creation (see Sec. 1), 2) more details about the baselines (see Sec. 2), and 3) more statistics and sample images from WorldPose (see Sec. 3).

1 Data creation

1.1 Static Camera Calibration

For static camera calibrations, we developed a GUI program designed for the manual annotation of 2D points, as is shown in Fig. 1. This tool simplifies the process of adding and editing annotations. It also offers additional features, like 1) zooming, which is crucial for achieving pixel-level accuracy, and 2) previewing, where it dynamically updates the estimation of camera parameters using the annotated 2d points and generates preview results of the projection of the field markings.

Subsequently, we employ the Canny detector [3] to extract the field markings from OpenCV [2]. The calibration results from the previous stage are also utilized to remove uninteresting lines. The detected lines are then converted into a distance field matrix, where each element of the matrix represents the distance to the nearest field markings. With the distance field we can further refine the camera parameters by minimizing the distance of the projected point of the field markings to the closest line pixel. The detected field markings and distance matrix are visualized in Fig. 2.

However, there are still several problems remained: 1) by default, we only estimate the k_1 and k_2 distortion coefficients of the cameras. However, this may not be sufficient in some cases, especially for side cameras with wide angles. These cameras need to be handled separately, and usually, adding the k_3 coefficient is sufficient to achieve relatively good results. 2) another issue is that even when

* These authors contributed equally to this work
† Now at HKUST(GZ)&HKUST

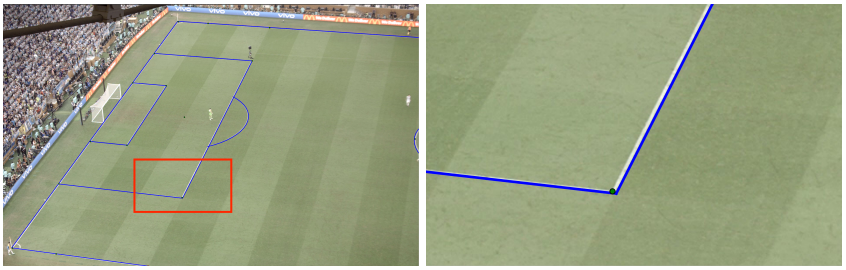


Fig. 1: Visualization of the annotation tools: manual annotation (left) and zoomed-in view (right). The manually selected point is indicated by the green marker. The blue lines show the preview results of the projection using the manually selected points.

the reprojection appears reasonable, it can be problematic for cameras looking at the penalty area, as shown in Fig. 3. Due to the lack of corresponding points in the right half of the image, there can be multiple sets of parameters that provide roughly the same reprojection for the field lines but very different results for the players. Therefore, the keypoints of players must be considered in the calibration process. This means it often takes multiple iterations of the entire camera calibration and keypoint estimation process to achieve desired accuracy.

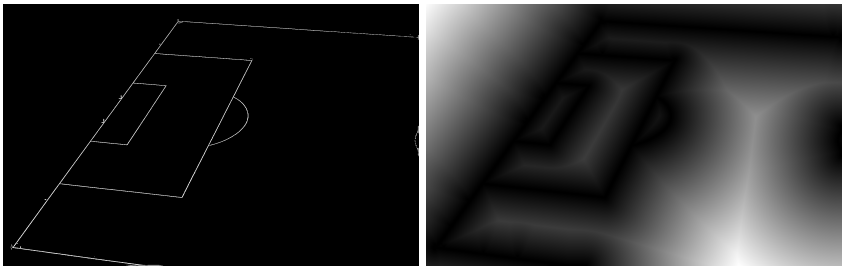
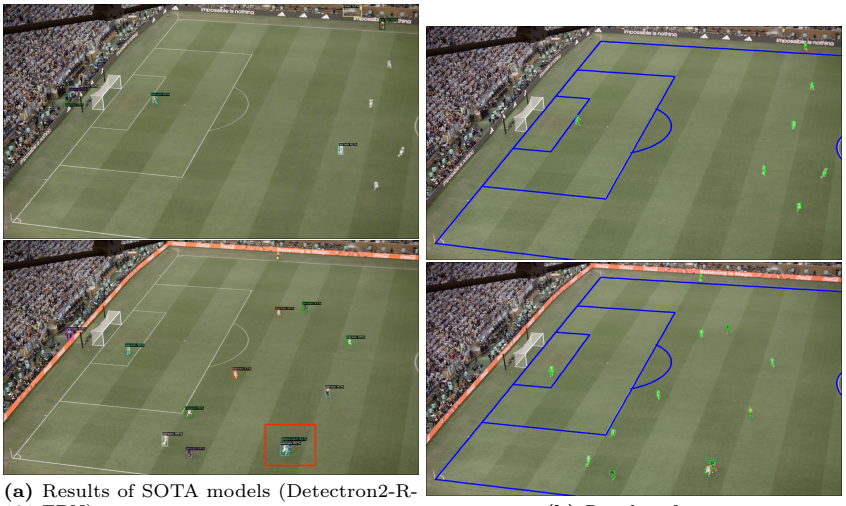


Fig. 2: Visualization of the photometric refinement process: extracted field markings (left) and distance field induced from the field markings (right). The brightness corresponds to the distance from the field markings.

1.2 Refinement of 2D detection results

Due to the distance between the camera and the players, the resolution of the players is rather low, which will negatively impact the accuracy of 2D detections. Consequently, even SOTA models may frequently miss the players or produce erroneous detections, as illustrated in the figure below.

To address these issues, we initially ensemble the predictions of multiple SOTA detection models by concatenating their detections and running Non-Maximum Suppression to eliminate duplicate detection boxes. However, the ensemble model is relatively slow and occasionally produces incorrect detections. Therefore, we apply this slower method to a subset of broadcasting images. We



(a) Results of SOTA models (Detectron2-R-101-FPN)

(b) Results of ours

Fig. 3: Comparison of SOTA models vs. ours. Note the missing players in the top row and erroneous detections highlighted in the red rectangle (it detects 3 players instead of 2). We get rid of the uninteresting detections outside the field (top row, top of the image) by leveraging the camera calibration.

manually inspect the results and remove incorrect detections. In this way, we semi-automatically annotate a small dataset and fine-tune the YOLO models with this dataset. Through this process, we achieve a 2D detection model with desired accuracy and speed.

1.3 SMPL Registration

For SMPL registration, we adopt the L-BFGS [7] optimizer with strong Wolfe line search and set the learning rate to 1. The hyperparameters of the loss functions are set to $\lambda_1 = 1$, $\lambda_2 = 0.1$, and $\lambda_3 = 0.1$ respectively.

1.4 Broadcasting Camera Calibration

For Broadcasting camera calibration we used Adam Optimizer [4] with a learning rate of 10^{-3} which empirically leads to slightly smoother camera parameters. Here, the hyperparameters are set to $\lambda_4 = 1$ and $\lambda_5 = 0.5$.

However, while this may be sufficient for achieving a low reprojection error, it often leads to less smooth distortion coefficients, as demonstrated by the blue curves in Fig. 4.

To address this, we have also incorporated additional smoothness regularizers, including a camera smoothness term and optical flow regularization (see Fig. 5). With these, we enforce that 1) the changes of focal length and distortion shall be smooth across frames, and 2) the reprojection of the keypoints and the

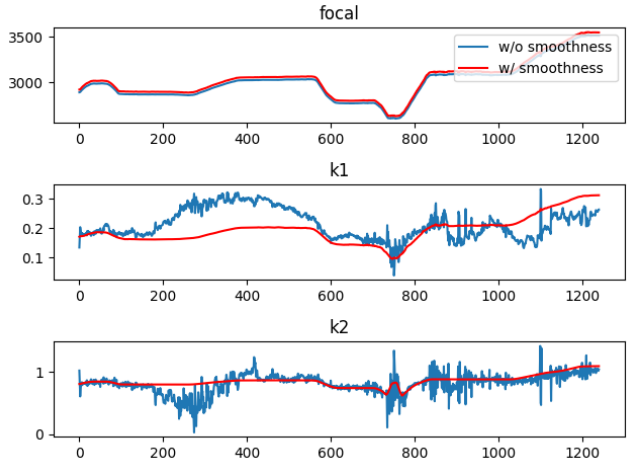


Fig. 4: Visualization of the camera parameters: The X-axis represents the frame index, and the Y-axes represent the focal length, k1, and k2 respectively. The smoothness of all camera parameters (including distortion coefficients) improved with the additional smoothness regularizations.

optical flow prediction shall be close to each other. In this way, we can achieve not only accurate but also visually smooth reprojection. The weights of these two regularizers are adjusted subject to the clip after a manual check.

2 Baseline Evaluation Details

2.1 GLAMR Baseline

Implementation Details For the GLAMR [11] baseline, we found that the official implementation was unable to detect many subjects within the frame. To address this we supply it with detection results generated by our preprocessing code that utilizes BYTETrack [12]. To achieve the best results, we run GLAMR on the entire video using a single A100 GPU.

Discussion In Fig. 6 we present the results of both HybrIK [6] and GLAMR. GLAMR utilizes HybrIK to initialize the SMPL estimation. With the provided detections, HybrIK generates accurate SMPL initializations for all subjects in the frame. However, despite the initialization provided by HybrIK, GLAMR struggles to predict plausible trajectories.

Unlike most other SLAM-based methods, GLAMR relies on its learning-based Global Trajectory Predictor to estimate the subjects' trajectories and infer the camera's extrinsic parameters based on these estimated trajectories. However, when the principal axes of the cameras are not parallel to the floor, as in our example, it has difficulty estimating the correct extrinsic parameters, leading to a tendency to place the players on a tilted plane.

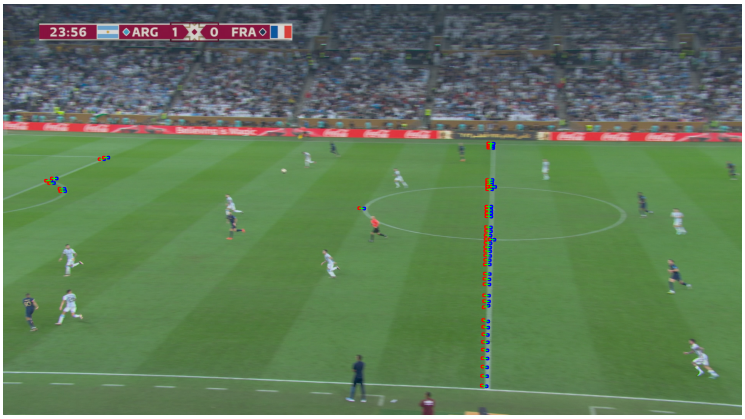


Fig. 5: Visualization of optical flow regularization: We employ the iterative Lucas-Kanade method with pyramids [1] to compute the optical flow for a sparse feature set consisting of points sampled from field markings. Here, the red points represent the sampled points from the previous frame, and the green points represent the predictions of the optical flow. Note outliers are removed with modified z-score.

(a) Hybrik (with our detections) **(b)** GLAMR

Fig. 6: Visualization of the GLAMR baseline: We present the results from both Hybrik (with our detections) and GLAMR (using Hybrik as initialization). Despite the good initialization, GLAMR struggles to place SMPL meshes in the correct locations

Additionally, we observed that in the implementation of GLAMR, it does not utilize the trajectories of multiple players to improve the estimation of extrinsic parameters (and consequently, the plane). Instead, it solely relies on the trajectory of the player with id=0. Therefore, while GLAMR is able to locate this player (the one annotated with the red rectangle in Fig. 6), it fails to accurately place other players, especially those that are far from the reference player.

2.2 SLAHMR Baseline

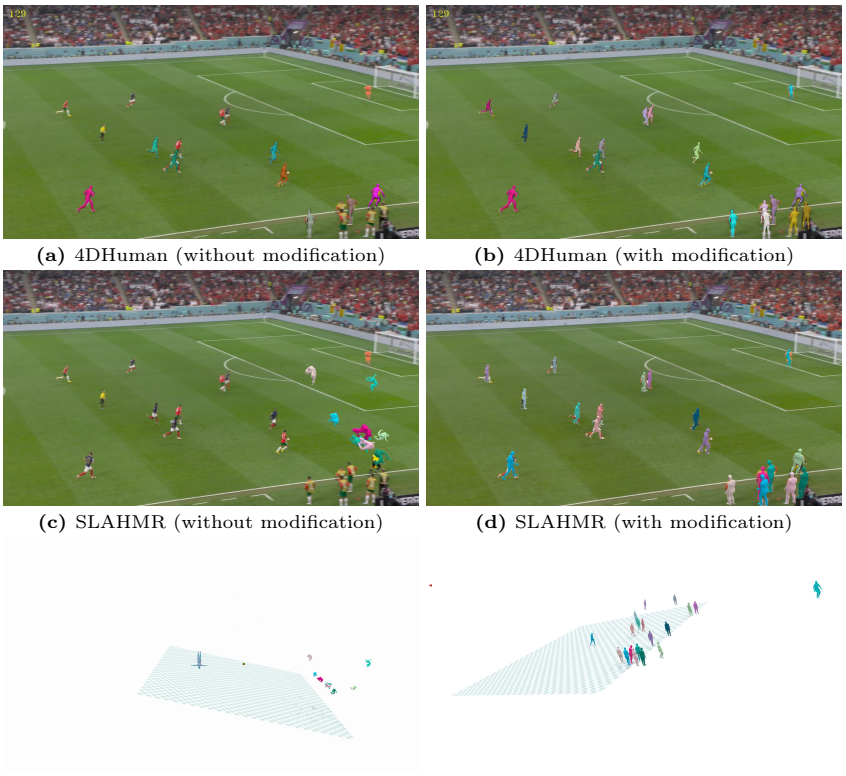
Implementation Details For fair comparison with GLAMR, we supply the same detection results used in the GLAMR to SLAHMR [10]. Additionally, we made a few changes to the official SLAHMR Implementation:

1. We notice that SLAHMR tends to overlook a few subjects when the number of subjects is relatively large. We made the following changes to the official implementation to address this:
   (a) We increased the constant MAX_NUM_TRACKS in the preprocessing code of SLAHMR from 12 to 30. This change allows SLAHMR to keep track of all subjects.
   (b) For 4DHuman, we lowered the confidence threshold to 0.5. These changes were made to ensure that all potential players are correctly recognized.
2. We observed that during the motion chunk stage, the optimization failed to converge due to an incorrect floor estimation. Therefore, we specified in the configuration to use a shared floor for all players. In this way, the model will try to align all players to the same floor and yield slightly improved results. Additionally, we enabled the "est_floor" parameter in the configuration file, allowing the model to estimate the floor normal rather than assuming it is parallel to the xy-plane. We found that this approach improves the performance, particularly when the camera is slightly tilted, as in our case.

Following the original paper, we first run DROID-SLAM [9] over the entire video, partition the video into chunks of 100 frames each and optimize each chunk separately. This is because the motion prior model of SLAHMR, HuMoR [8], is trained on short motion clips and it is recommended by the official HuMoR repository that it should be applied to short clips of 2-3 seconds.

Discussion In Fig. 7, we ablate the impacts of our modifications on SLAHMR. With our modification, SLAHMR is able to generate relatively accurate and feasible trajectories in our data. However, while able to produce a reasonable trajectory for individual subjects, SLAHMR struggles to identify the correct relative positioning between different subjects (see subfigure Fig. 7 f).

While the modification improves the performance of SLAHMR on some sequences, we note that the motion chunk stage remains very fragile and could easily diverge, especially during fast camera movements, which are quite common in broadcasting scenarios. The core issue lies in SLAHMR's need to estimate the floor before introducing the motion prior model. However, the only



(a) 4DHuman (without modification)

(b) 4DHuman (with modification)

(c) SLAHMR (without modification)

(d) SLAHMR (with modification)

(e) SLAHMR (without modification) in World Coordinate Frame

(f) SLAHMR (with modification) in World Coordinate Frame

Fig. 7: Visualization of SLAHMR and 4DHuman: the left and right columns show the results before and after the modification. While SLAHMR appears to produce seemingly reasonable results (as shown in subfigure f), it does not have the correct scale due to an incorrect focal length. Specifically, the distance between the players should be much larger, as the stadium is approximately 70 meters wide.

loss that aligns the players to the floor comes from the motion prior model. This creates a chicken-egg situation: if the players are already roughly on the same plane without the motion prior model, SLAHMR can converge to reasonable results. However, if this is not the case, the motion prior model will not provide any meaningful gradient, leading to complete divergence, as is shown in Fig. 8. Specifically, we observed that for some sequences, SLAHMR fails to converge on as many as half of the chunks, even with ground-truth camera parameters. This can be confirmed from the higher PA-MPJPE loss compared to 4DHuman, as shown in Table 3 of the main paper (which is related to the divergence).

Fig. 8: Visualization of typical failure cases of SLAHMR: When the distances between the players are relatively large, SLAHMR struggles to locate the floor, resulting in complete divergence.

Similar to GLAMR, our evaluation of SLAHMR on WorldPose reveals several limitations: 1) SLAHMR has a tendency to generate overly smooth motions which poses challenges in capturing fast-paced movements. 2) the motion chunk stage can be quite unstable with large focal length and when players are not standing close to each other, 3) We noticed that although the camera trajectory remains smooth across the boundary of each chunk, there is a visible gap in the predicted SMPL meshes between the chunks. This issue could potentially be mitigated if SLAHMR divides the sequence into overlapping clips, thereby enforcing smoothness regularization across the chunks, similar to the approach employed in PACE [5]. 4) the optimization is time-consuming: 40 minutes per 100 frames with 4 subjects as reported in the original paper (which aligns with our observations).

2.3 Evaluation

To align the predicted SMPL poses with the ground-truth, we employed a greedy matching algorithm based on Intersection-over-Union (IoU), comparing

2D bounding boxes of the ground truth with 2D predictions. We found sometimes baselines split trajectories in case of re-entries or lose track, so we merge tracks corresponding to the same ground truth subject during post-processing. For evaluation, we only consider the subjects and frames when they are both available in the prediction and the ground-truth, and the MPJPE is calculated with selected SMPL keypoints (including the nose, neck, shoulders, wrists, elbows, hips, knees, and ankles) which are generally more reliable.

3 Additional Statistics on WorldPose

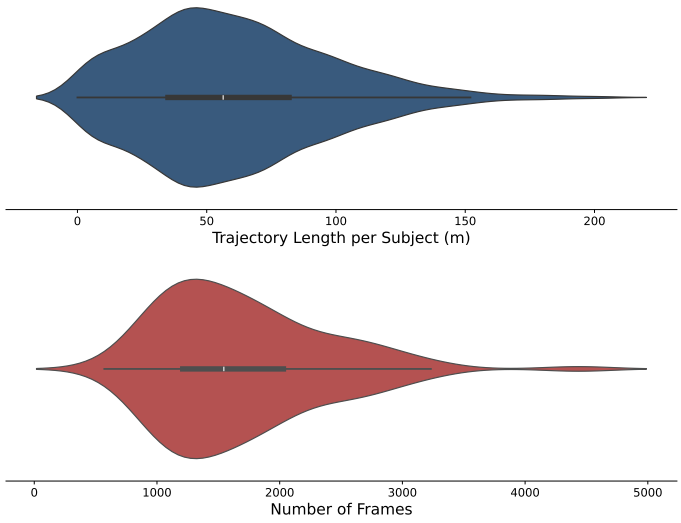


Fig. 9: Distribution of player trajectory lengths (top) and clip lengths (bottom) in WorldPose. The availability of long trajectories up to 200 m sets WorldPose apart from existing datasets.

We plot the distribution of sequence lengths and per-player trajectories appearing in WorldPose in Fig. 9. For additional sample images, please refer to Fig. 10.

References

1. Bouguet, J.Y., et al.: Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel corporation **5**(1-10), 4 (2001)
2. Bradski, G.: The opencv library. Dr. Dobb's Journal: Software Tools for the Professional Programmer **25**(11), 120–123 (2000)

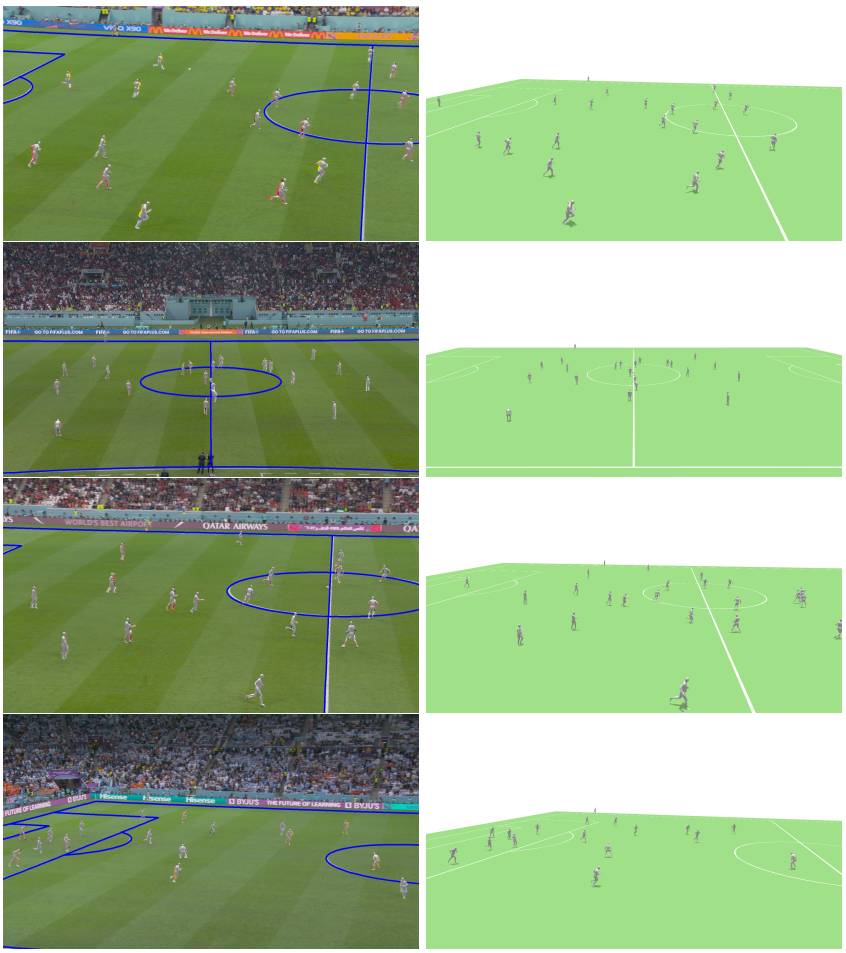


Fig. 10: Additional sample images: We include more sample images from our dataset. These images demonstrate that our dataset can provide accurate SMPL meshes and camera parameters, even when the camera zooms in, and fewer corresponding points of field markings are available.

3. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence (6), 679–698 (1986)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Kocabas, M., Yuan, Y., Molchanov, P., Guo, Y., Black, M.J., Hilliges, O., Kautz, J., Iqbal, U.: Pace: Human and motion estimation from in-the-wild videos. In: 3DV (2024)
6. Li, J., Xu, C., Chen, Z., Bian, S., Yang, L., Lu, C.: Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3383–3393 (2021)
7. Liu, D.C., Nocedal, J.: On the limited memory bfgs method for large scale optimization. Math. Program. **45**(1–3), 503–528 (Aug 1989)
8. Rempe, D., Birdal, T., Hertzmann, A., Yang, J., Sridhar, S., Guibas, L.J.: Humor: 3d human motion model for robust pose estimation. In: International Conference on Computer Vision (ICCV) (2021)
9. Teed, Z., Deng, J.: DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. Advances in neural information processing systems (2021)
10. Ye, V., Pavlakos, G., Malik, J., Kanazawa, A.: Decoupling human and camera motion from videos in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21222–21232 (2023)
11. Yuan, Y., Iqbal, U., Molchanov, P., Kitani, K., Kautz, J.: Glamr: Global occlusion-aware human mesh recovery with dynamic cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
12. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: European Conference on Computer Vision. pp. 1–21. Springer (2022)