

# Sports Analysis and VR Viewing System Based on Player Tracking and Pose Estimation with Multimodal and Multiview Sensors

Wenxuan Guo\* , Zhiyu Pan\* , Ziheng Xi , Alapati Tuerxun , Jianjiang Feng†  and Jie Zhou 

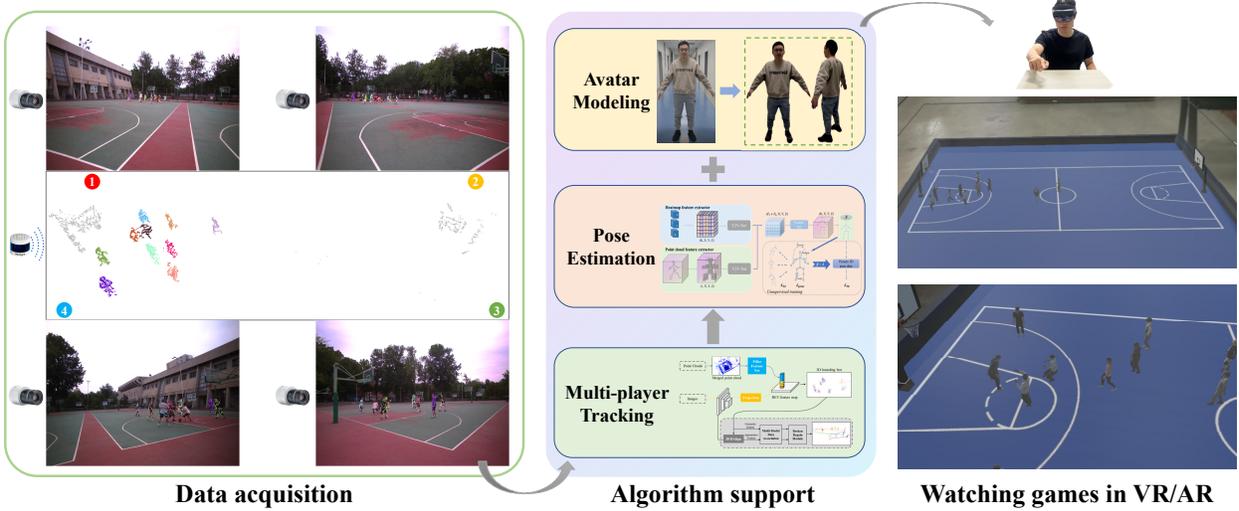


Fig. 1: The overview of our proposed sports analysis and VR viewing system. Firstly, game data is collected by multimodal and multiview sensors, including LiDAR and cameras. Subsequently, multi-object tracking and pose estimation algorithms are employed to perceive the positions and movements of players, while avatar modeling algorithm are used for creating 3D drivable models of the them. Finally, we drive these avatar models based on the position and pose results, visualizing them in virtual space. In this way, audiences can watch the game on VR/AR devices.

**Abstract**—Sports analysis and viewing play a pivotal role in the current sports domain, offering significant value not only to coaches and athletes but also to fans and the media. In recent years, the rapid development of virtual reality (VR) and augmented reality (AR) technologies have introduced a new platform for watching games. Visualization of sports competitions in VR/AR represents a revolutionary technology, providing audiences with a novel immersive viewing experience. However, there is still a lack of related research in this area. In this work, we present for the first time a comprehensive system for sports competition analysis and real-time visualization on VR/AR platforms. First, we utilize multiview LiDARs and cameras to collect multimodal game data. Subsequently, we propose a framework for multi-player tracking and pose estimation based on a limited amount of supervised data, which extracts precise player positions and movements from point clouds and images. Moreover, we perform avatar modeling of players to obtain their 3D models. Ultimately, using these 3D player data, we conduct competition analysis and real-time visualization on VR/AR. Extensive quantitative experiments demonstrate the accuracy and robustness of our multi-player tracking and pose estimation framework. The visualization results showcase the immense potential of our sports visualization system on the domain of watching games on VR/AR devices. The multimodal competition dataset we collected and all related code will be released soon.

**Index Terms**—Game visualization, virtual reality, multi-player tracking, pose estimation, avatar

## 1 INTRODUCTION

With the rapid development of technology, the sports industry is undergoing an unprecedented transformation. The close integration of technology and sports has opened up new areas of research and practice, demonstrating immense potential and value. This fusion leverages the latest technological advancements with the aim of comprehensively

enhancing the performance within the sports sector. This includes, but is not limited to, improving training methodologies, increasing the spectacle of competitions, and enhancing athlete safety. In this process, sports analysis and game visualization technologies play a crucial role.

Sports competition analysis, by collecting and analyzing vast amounts of game data, assists coaching teams and audiences in gaining an in-depth understanding of key moments in matches and athletes' performances. This enables the formulation of more scientific and precise training plans and competition strategies. Additionally, through biomechanical analysis and motion capture technology, coaches and scientists can meticulously analyze athletes' movement patterns. This helps identify inefficient or potentially injurious actions, thereby optimizing athletes' training regimes and techniques.

The integration of sports competitions with Virtual Reality (VR) or Augmented Reality (AR) technologies opens a new chapter in the spectator experience. VR/AR technologies enable spectators to feel as if they are on-site, watching the game from various angles and

• Wenxuan Guo, Zhiyu Pan, Ziheng Xi, Alapati Tuerxun, Jianjiang Feng and Jie Zhou are with Department of Automation, Tsinghua University.

E-mail: {gwx22, pzy20, xizh21, alpttx19}@mails.tsinghua.edu.cn, {jifeng, jzhou}@tsinghua.edu.cn

• \*: Equal contribution.

• †: Corresponding author.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

distances, and choosing to follow the perspective of a specific athlete, which significantly enhances the immersion of watching a game. For fans unable to attend the games, VR/AR viewing offers an excellent solution. Additionally, viewers can interact with other spectators or elements in the virtual environment, providing a more personalized and interactive viewing experience. With VR/AR products [2, 45, 46] becoming increasingly common and their performance improving, the foundation for VR/AR viewing has been established. However, current VR/AR viewing methods [2, 45] generally require a large array of cameras to collect game data, leading to high costs and making it difficult to apply to regular competitions. Furthermore, they require high computational costs for rendering, making it difficult to broadcast events in real-time, limiting their use to game replays. Therefore, there is a pressing need to explore new low-cost and real-time VR/AR viewing frameworks.

In this paper, we approach the challenge of watching games on VR/AR devices with a novel perspective, achieving analysis of game data meanwhile. Firstly, game data is collected by multimodal and multiview sensors, including LiDAR and cameras. Subsequently, multi-object tracking and pose estimation algorithms are employed to perceive the positions and movements of players, while avatar modeling algorithm are used for creating 3D drivable models of the them. Finally, we drive these avatar models based on the position and pose results, visualizing them in virtual space. Furthermore, we construct a new multimodal dataset for evaluating our method. The dataset comprises nearly 11,000 frames captured from two basketball games in various real-world scenarios.

3D multi-object detection and tracking is essential for visual surveillance and player tracking [13, 43]. Most of recent multi-object tracking methods [9, 62, 68] follow the tracking-by-detection paradigm, in which objects are first obtained by detection and then linked to previous trajectories by data association. The resulting trajectories may then be utilized for sports analysis like running speed and total distance covered [37]. Traditional player tracking systems typically employ multiview RGB cameras for observing the scene [6, 13, 52, 73]. However, poor lighting, limited ability to identify foreground, similar appearance, and inaccurate depth estimations have an impact on how well these methods perform. In recent years, LiDAR-based systems [9, 42, 60, 61] have developed rapidly and become increasingly cheaper, enabling precise 3D object detection and tracking in autonomous driving scenes. Since point clouds provide precise 3D information, robust tracking trajectories may be produced based on 3D detection results. However, due to the lack of appearance information, LiDAR-based systems are prone to tracking errors due to poor detection performance for extremely close objects, which is very common in sport games.

In this work, we propose an online tracking method that fuses multiview LiDARs and cameras. In order to improve the reliability and accuracy of dynamic object perception, multimodal sensors can be employed to provide richer clues and hence reduce detection and tracking errors. To our best knowledge, this is the first multi-player tracking research taking multiview RGB and point cloud as input.

Human pose estimation is a fundamental task in computer vision and has been widely applied in various fields, such as human-computer interaction, human-robot interaction, human activity recognition, etc. Specifically, multiview image datasets [5, 23, 27, 59] allow more precise 3D human pose estimation compared to single-view ones [1, 26, 36, 72], due to the ability of multiple views to capture 3D information from epipolar geometry. As technology advances, researchers [24, 35, 53, 57, 69] have achieved promising results on these multiview images datasets. However, practical scenarios are more challenging than existing public datasets, with diverse human motions, severe occlusions, and large scene. Combining different sensors such as LiDAR and RGB camera has the potential of boosting the accuracy of 3D human pose estimation performance in multiple situations. It is essential to consider integrating the information from sensors of various types and views, and fusing them into a cohesive representation which can significantly facilitate the accuracy of 3D human pose estimation.

In this paper, we propose PointVoxel, a 3D human pose estimation pipeline. Volumetric representation is an effective and reasonable ar-

chitecture to fuse RGB and point cloud. Researchers [24, 53, 57, 67] have shown the effectiveness of voxel-based method. Voxel-based architecture can naturally model a space's geometric characteristics. Besides, it is straightforward to either map the point cloud information or back-project the 2D information into the 3D volumetric representation. As to multimodal features fusion [15, 19], we refer to the work BEVFusion [34] and unify different modalities into a volumetric space that reserve each modality's geometric characteristics. Manually annotating or capturing 3D poses for multiple individuals in large scenes is challenging, expensive, and hard to ensure generalizability to new scenarios. To achieve better results on challenging scenarios without pose labels, we adopt the approach of pretraining on a synthetic dataset first and then unsupervised domain adaptation training on the target dataset.

In the film and gaming industries, the creation and animation of virtual characters is a complex and detailed process, with avatar modeling being one of the key components. Avatar modeling requires the models to be visually appealing and capable of simulating complex actions. This process involves several steps, including concept design, 3D modeling, texture mapping, rigging, and body animation. In recent years, to perform 3D modeling of real human, some works focus on using multiview photos or videos for real human avatar modeling. Our VR/AR viewing framework follows the modeling and driving processes in the animation production field. It involves pre-modeling the athletes in the competition using real human avatar modeling techniques, and driving the virtual athletes based on their positions and poses obtained through multi-player tracking and pose estimation. We utilize the software Unity to achieve 3D reconstruction and rendering of the competition, deploying it on VR/AR terminals to provide audiences with an immersive viewing experience.

To summarize, our contributions are as follows:

1. We propose a novel sports competition analysis and VR viewing system that boasts low cost and real-time performance. It utilizes multimodal and multiview data from LiDARs and cameras, and follows a process including data collection, multi-player tracking, pose estimation, player avatar modeling, and actuation.
2. We propose a method for multi-player tracking and pose estimation that integrates multimodal and multiview data, aiming to achieve precise 3D results. For pose estimation, we present a new training strategy without using high-cost manual annotations.
3. We construct a novel multimodal and multiview dataset of sport competitions to facilitate development in this area. Extensive experiments demonstrate the accuracy of our multi-player tracking and pose estimation method, as well as the significant potential of our VR/AR viewing system.

## 2 RELATED WORK

### 2.1 Sports Visualization in VR/AR

Virtual Reality (VR) and Augmented Reality (AR) technologies offer a new platform for viewing sports competitions, providing the global audiences with immersive and interactive participation experiences.

In the terms of VR, devices like Oculus Quest [45] and Pico Neo [49] allow users to enjoy a 360-degree view of the action, making them feel as if they are sitting in the stadium. AR devices [2, 46], on the other hand, enriches the live viewing experience by overlaying digital information onto the real-world environment. Applications on AR allow users to see player stats, game dynamics, and other relevant information in real-time while watching the game, either on-site or from a remote location. This not only makes the experience more engaging but also helps fans understand the game better. Moreover, the social aspect of watching sports could be enhanced through VR/AR technologies. Platforms such as Oculus Quest [45] and social features in apps like LiveLike [39] enable fans to watch games together virtually, regardless of their physical location. This can create a sense of community and shared experience that was previously only possible by attending the games in person.

Currently, the NBA and FIFA, as two of the world's leading sports event organizations, have adopted a series of innovative measures in

VR broadcasting and viewing to enhance the spectator experience and interactivity. For certain games, they offer VR viewing applications, allowing users to watch live broadcasts and replays of highlights through head-mounted devices. This viewing method offers the possibility to watch the game from different angles, including courtside and aerial views, bringing an unprecedented level of immersion and a sense of being at the venue to fans. However, existing solutions generally require a large array of cameras to collect game data, leading to high costs that make it difficult to apply to regular competitions. Additionally, they require significant computational resources for rendering, making it challenging to broadcast events in real time. Currently, they only support viewing from specific locations and within a limited range of angles, and do not allow for more complex viewing modes, such as choosing to follow the perspective of a specific athlete. Therefore, our work explores a low-cost, highly operable, and real-time framework for viewing sports competitions through VR/AR.

## 2.2 Multi-object Tracking

Recent Multi-object tracking (MOT) systems mostly follow the tracking-by-detection paradigm, in which objects are first obtained by a detector and then linked to previous trajectories by data association. DeepSort [63] uses Kalman filter to update the trajectories and solves data association as the bipartite matching problem using the Hungarian algorithm. CenterTrack [77], on the other hand, directly regresses the offset to obtain the data association across frames. These methods typically use deep features of images for cross-frame association, lacking accurate 3D geometry and location information.

For 3D MOT in the area of visual surveillance, multi-cameras [6, 13, 52, 73] are often used to calculate the 3D location of the objects to obtain 3D tracking results. KSP [6] first obtains the detection by background subtraction, projects it onto the ground to create a probability occupancy map, and finally uses the K-shortest paths optimization to obtain offline tracking results. Some methods [48, 71] followed it by using a 2D detector to improve the detection results. In the field of sports game analysis, Zhang *et al.* [71] propose a robust multi-camera player tracking framework by improving the KSP algorithm using player identification. In addition, the majority of player tracking datasets are captured by multiview cameras, such as APIDIS dataset [58] and STU dataset [71]. However, the 3D depth obtained by epipolar geometry is not accurate enough for robust association and tracking.

As for autonomous driving scenes, LiDAR-based approaches [9, 42, 55, 61, 68] are dominant. Weng *et al.* [61] propose a simple and robust baseline based on point clouds, using 3D Intersection over Union (IoU) as the data association. Chiu *et al.* [9] use the Mahalanobis distance instead of IoU to improve the association accuracy. CenterPoint [68] is an extension of CenterTrack [77], which solves the data association by regressing the offset of the object. Most of these methods rely on 3D point clouds and lack sufficient appearance information, which may lead to tracking errors for very close objects.

## 2.3 3D Human Pose Estimation

**Image-based.** Basic 3D pose estimation method is in two stage that estimates the 2D pose first and then lifts it into 3D space [11, 24, 35, 53, 57, 64]. As to multi-person setting, some methods [11, 35] match pedestrians from different views and then locate them through 2D pose similarity. But they are not robust to inaccurate 2D pose results. Zhang *et al.* [69] directly utilizes the 2D images as inputs and regresses the 3D pose by transformer architecture. However, their training process is time-consuming. Some voxel-based methods [53, 57] locate each person in a 3D volumetric space and estimate 3D poses. Such voxel-based methods greatly improve the precision of 3D human pose estimation. However, they are not suitable for the large scene because of large computational cost when detecting people.

**Point cloud-based.** 3D pose estimation methods focus on single-view depth map [14, 18, 20, 44] at the beginning. They treat the depth map as 2D information with depth value. On the other hand, a few of methods back-project depth map into 3D space as dense 3D point cloud and use Pointnet network to deal with [4, 74]. Moon *et al.* [47] propose a single-person pose estimation approach which treats depth map as point cloud

and fill it in volumetric space. Bekhtaoui *et al.* [4] use Pointnet related approach to detect and estimate 3D human pose. Recently, Li *et al.* [32] use sparse point cloud scanned from LiDAR to estimate single-person 3D human pose. However, sparse point cloud cannot provide enough information for accurate 3D pose estimation. Multimodal fusion is helpful for concise 3D human poses' perceiving.

**Multimodal based.** Many works have been introduced for RGBD human pose estimation [3, 15, 19, 75]. Multimodal information can not only help detect the person but also guarantee the accuracy of 3D human pose. Lately, some researchers [10, 32] introduce camera-LiDAR setting datasets for outdoors and large scene. As to the multimodal fusion methods, some approaches [4, 19, 75] fuse modalities in point-level by attaching extracted 2D features to every 3D point. And a few methods [33, 50] fuse modalities in feature-level strategy. They do not unify the 2D and 3D information in a unified interpretable space, and fuse them in an uninterpretable feature space instead.

Our approach utilizes multimodal information for human detection and 3D human pose estimation. Besides, we use volumetric representation representing real-world space to unify the 2D and 3D information.

## 3 METHOD

Our proposed sports analysis and VR viewing system utilizes multimodal and multiview point clouds and images as input, ultimately enabling watching games on VR/AR devices while providing accurate player statistics. In the following, we introduce the key technologies within the system. In Sec. 3.1, we present our proposed method for multi-player tracking. In Sec. 3.2, we introduce our pose estimation method along with an unsupervised training scheme. In Sec. 3.3, we describe avatar modeling and the VR/AR visualization methods.

### 3.1 Multiview Multimodal 3D Tracking

We propose a 3D online multi-object tracking approach, which improves reliability and accuracy via modality fusion. As shown in Fig. 2, our framework follows the widely employed tracking-by-detection paradigm. In the following, we first describe the multimodal detection stage in Sec. 3.1.1, which takes as input multiview images and merged multiview point clouds and outputs the 3D bounding boxes. The data association stage is presented in Sec. 3.1.2 which links the detection results to the previous trajectories.

#### 3.1.1 Multimodal Detection

Our multimodal detection method consists of two streams: an image processing network and a point cloud processing network, designed to extract modal-specific feature maps. The multimodal feature map is obtained by combining the two modal-specific feature maps based on spatial position in bird's eye view (BEV). Finally, a detection head is employed to obtain accurate 3D object detection results.

**Point Cloud Processing Network.** The point clouds from all single views are spatio-temporally aligned and fused to obtain a multiview dense point cloud. The feature encoder and the backbone make up the two components of the point cloud processing network. Pillar Feature Net [31] is adopted by the feature encoder to transform point clouds into pseudo images, while backbone is used to extract high-level features of the pseudo images.

Firstly, the point cloud is assigned to multiple grids in the X-Y plane, so that a frame of point cloud data can be encoded into a dense tensor of dimension  $(D, P, N)$ , where  $D$  stands for the feature dimension,  $P$  for the number of grids, and  $N$  for the number of point clouds contained within each grid. We then utilized a simple PointNet [51] to create a tensor of dimension  $(C_p, P)$ . Finally, the pseudo image of  $(C, H_p, W_p)$  is generated by the scatter operator.

The backbone is made up of two sub-networks. A top-down sub-network is used to extract features on increasingly small spatial resolutions, and the other sub-network is responsible for up-sampling the features to the same dimensional size by a deconvolution operation. The final output is a combination of all features from different resolutions.

**Image Processing Network.** The input to the image processing network is images captured by cameras with different perspectives. The input images are processed simultaneously by a convolutional network

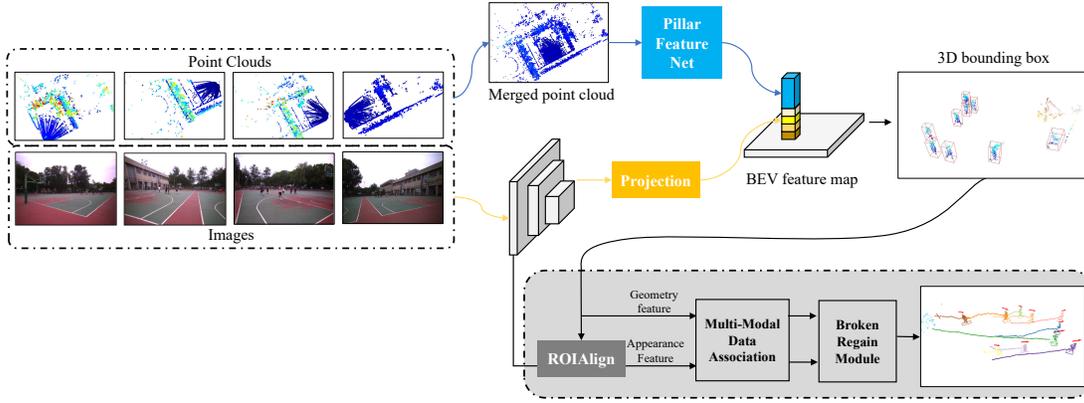


Fig. 2: Overview of our multi-player tracking method. We extract features in images and point clouds and fuse them in BEV for multimodal detection results. Both geometry information and appearance information are leveraged for long-term object tracking.

to extract features, which are then projected to BEV space for feature aggregation. We choose the lightweight ResNet18 [22] as the backbone because of its efficiency and capability of feature extraction. The convolutional network computes the  $C_i$ -channel feature maps of each of the  $M$  input images with shared weights. Before projection, the  $M$  feature maps are resized to a fixed size  $(H_i, W_i)$ , where  $H_i$  and  $W_i$  denote the height and width of the feature maps.

Multiview information aggregation is performed by the projection transformation. Assuming that the ground is horizontal, we can get the correspondence between the pixels of multiview images and the coordinates of ground point with the extrinsic and intrinsic parameters. To enhance the perception of spatial location, multiple projected feature maps and X-Y coordinate maps are concatenated to obtain the multiview image feature map of  $M \times C_i + 2$  channels.

**Detection Head and Loss.** The image and point cloud feature maps are concatenated according to spatial locations in BEV to obtain multimodal feature maps, and then a detection head is utilized to obtain 3D object detection results.

Considering the simplicity and computational complexity, we use Single Shot Detector (SSD) [38] as the detection head, which using 2D IoU to match the detection results with the ground truth during training. We apply the same loss functions introduced in SECOND [66].

### 3.1.2 Multimodal MOT

Given the 3D object detection results of the current frame, previous trajectories are associated by fusing geometry information and appearance information. As for the geometry distance metric, we employ the 3D distance intersection-over-union (3D DIoU) between the detection bounding boxes and the tracking bounding boxes. The 3D object detection boxes are projected onto the input multiview 2D camera planes to obtain the image patches, which are used to extract the appearance features of objects. And then we take into account the cosine distance of appearance features between the detections and the tracks as the appearance distance metric. We design a linear combination of the geometry distance and appearance feature distance to aggregate the two affinity matrices, which is passed to the Hungarian algorithm for data association and Kalman filter for prediction.

**Multimodal Data Association.** To associate the detections with the tracks, we calculate the 3D DIoU [76] between every detection-track pair. Assume that the detections  $D^t = \{d_i^t | i = 1, \dots, N\}$  generated by the multimodal detector, and the tracks  $G^t = \{g_i^t | i = 1, \dots, M\}$  predicted by the tracking module of  $t - 1$  frame, each  $d_i^t, g_i^t$  is a 3D bounding box denoted by centers, dimensions and the rotation angle. Considering the penalty term of center distance, we employ the 3D DIoU to avoid the collapse of the normal IoU for non-overlapping cases, which is defined as follows:

$$G(d_i^t, g_i^t) = 0.5 \times (IoU_{3D} - \frac{\rho^2(c_d, c_g)}{\rho^2(d, g)} + 1) \quad (1)$$

where  $c_d, c_g$  are the center of two bounding boxes ( $d_i^t, g_i^t$ ) and  $\rho(d, g)$  is the distance of their farthest corners. With the penalty term, two bounding boxes far apart will get a lower 3D DIoU score, which ranges from zero to one. We apply the 3D DIoU score for all detection-track pairs to obtain the geometry affinity matrix  $S_{ge} \in R^{M \times N}$ .

Despite obtained 3D bounding boxes have accurate 3D location information, using only geometry information is prone to errors since it's hard to distinguish the close objects from aggregated point clouds. Therefore, we also take into account the appearance information for association metric. We project the 3D detection bounding boxes onto the input multiview camera planes to obtain the image patches. It should be noted that the projection of the 3D bounding boxes onto the 2D images can cause occlusion problems. We determine the level of occlusions for each view separately, by calculating the 2D IoU score of each detection box. If the IoU of the two bounding boxes is greater than the threshold, we consider the object with larger depth to be occluded by the close object, which indicates that the 2D appearance feature of the occluded object is unreliable. For each valid projected bounding box, we extract a  $512 \times 25 \times 25$  image feature from the RoIAlign [21] feature of the image processing stream in Sec. 3.1.1, which is max-pooled to a 512 dimensional tensor as the appearance feature.

For each object  $d_i^t$ , our method extracts its appearance features  $F_i^t = \{f_{i,j}^t | j = 1, \dots, C\}$  from  $C$  valid views. Meanwhile, the appearance features of tracks  $g_m^t$  is obtained from *feature memory bank*, which consists of  $K$  feature tensors of all valid image patches in the past  $l$  frames. The cosine similarity of feature tensors is used as the appearance distance metric, which is defined as follows:

$$A(d_{i,j}^t, g_{m,n}^t) = \frac{f_{i,j}^t \cdot f_{m,n}^t}{\|f_{i,j}^t\| \|f_{m,n}^t\|} \quad (2)$$

where  $f_{i,j}^t$  are the appearance features of the  $i$ th detection in the  $j$ th view, and  $f_{m,n}^t$  are the  $n$ th appearance feature in *feature memory bank* of the  $m$ th track. For each detection-track pair, a cosine similarity matrix of size  $C \times K$  can be obtained, and then we take the top  $k$  largest cosine similarity values of each row as the valid matches for each view. The valid cosine similarity scores  $C \times k$  are eventually averaged as the final appearance similarity of this detection-track pair. If the valid appearance features of the track is less than  $k$  or there is no valid feature for this detection, which means that the appearance information is not reliable, we use the 3D DIoU score as the default value. Finally, we obtain the appearance affinity matrix as  $S_{ap} \in R^{M \times N}$ .

With the geometry affinity matrix  $S_{ge}$  and appearance affinity matrix  $S_{ap}$ , we design a linear combination of the two affinity matrices to aggregate the two affinity matrices, which is passed to the Hungarian algorithm for data association and Kalman filter for prediction. The final affinity matrix  $S$  is defined as follows:

$$S = \alpha S_{ge} + (1 - \alpha) S_{ap} \quad (3)$$

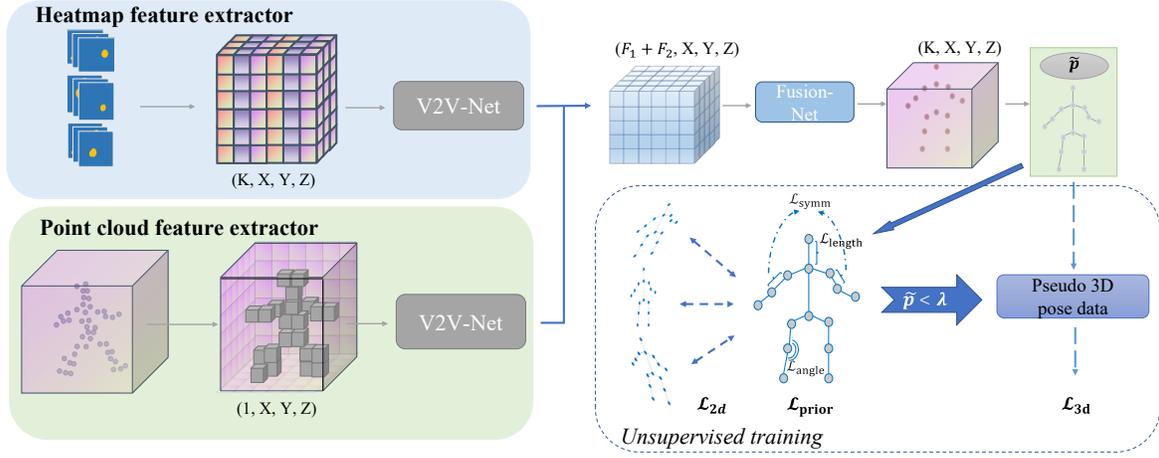


Fig. 3: The detailed structure of PointVoxel in 3D human pose estimation and its unsupervised training strategy. Dashed arrows indicate calculating corresponding loss.  $\rightarrow$  represents conditional flow which means the pose will be the pseudo 3D label for the next epoch if required condition is met.

where  $\alpha$  is the proportion of images and point clouds contribution to the data association. After the data association at  $t$  frame, the obtained appearance features of the object  $d_i^t$  are added to the *feature memory bank* of  $g_m^t$  for the next frame.

**Trajectory Regain Module.** This module is designed to regain the broken trajectories, which are caused when objects are extremely close or out of the field. After the objects are re-detected, the broken trajectories should be regained for a long-term tracking. However, traditional tracking systems [61, 68] tend to treat them as new objects lack of appearance information.

To regain the broken trajectories, we manage all unmatched detections and unmatched trajectories using appearance features under geometry constraints. On the one hand, we consider all unmatched trajectories as probable existed objects which are missed due to extremely close or out of the field. Instead of deleting the potential broken trajectories, we keep tracking them and retain their appearance features in the memory bank. The location of the potential broken trajectories is predicted by Kalman filter until the objects leave the field.

On the other hand, we consider all unmatched detections as the potential objects of the broken trajectories before. We calculate the regain score between each unmatched detection  $d_i^t$  and each broken trajectory  $g_j^t$  under the geometry constraints. In sports competitions, like basketball and football games, the 3D location of the player does not change abruptly, therefore players won't suddenly appear or disappear from inside the court. We construct geometry constraints by assuming the number of objects in the local area is constant, that is to say, the number of objects in the local area is the same as the number of objects in the historical frames. For the unmatched trajectory  $g_j^t$ , if the object is moving out of the area by the edge, we consider the unmatched detection enters from the same edge to meet the geometry constraints. If the broken trajectory is in the area, we consider a neighboring area as the local area, whose radius is improving by the time since the trajectory has been broken, and the corresponding unmatched detection should be in the local area to meet the geometry constraints.

The regain score of inner tracks is defined by exploiting both geometry and appearance information:  $s = \alpha G(d, g) + (1 - \alpha)A(d, g)$ . As for the objects moving out of the field by the edge, the regain score is considered by the appearance similarity  $A(d, g)$ . The score is set to  $-1$  if the constraints are not met. If the maximum score  $S(d_i^t, g_j^t)$  is greater than the threshold, the detection and the trajectory are considered to belong to the same object, and thus the broken trajectory is maintained and interpolated to improve the continuity of the long-term object tracking.

### 3.2 Multi-player Pose Estimation

The proposed method is composed of two parts: 3D human pose estimation and unsupervised domain adaption training, as shown in Fig. 3.

Section 3.2.1 introduces the approach of the top-down manner of pose estimation. We introduce the details about voxel-based multimodal fusion. Section 3.2.2 illustrates how we generate the synthetic dataset to support the training of multimodal pose estimation model, and how to combine entropy with the human prior loss to realize domain shift.

#### 3.2.1 PointVoxel

Similar to [11, 53, 57], we adopt the top-down manner to estimate the 3D pose. Through the multi-player tracking method described in Sec. 3.1, we obtain long-term 3D and 2D bounding boxes for each player. At first, we can define a volumetric space centered at the pointcloud's centroid whose size is consistent with detected bounding boxes', and discretize it into an  $X \times Y \times Z$  resolution. So that we can fill each voxel according to each point cloud's coordinate. In our case, we set value 1 to the voxel containing pointcloud. Then we can get the point cloud-related feature  $(F_1 \times X \times Y \times Z)$  via 3D convolution backbone V2V-Net [47].

Thanks to the development of 2D pose estimation methods [12, 65], relatively accurate corresponding 2D pose can be predicted without extra training. Hence, we can get each view's 2D pose heatmap via Gaussian blur. We apply the same projecting way as [57] to calculate each voxel's information, and apply V2V-Net to extract the rgb-related features  $(F_2 \times X \times Y \times Z)$  in 3D space. Then, we concatenate the two modal features in the same 3D space for modality fusion and get the final 3D human pose heatmap via a 3D convolution head Fusion-Net. In order to mitigate quantization error, we adopt *Soft-argmax* to calculate each joints' 3D coordinate  $J^k$  through 3D heatmaps and minimize the  $L_1$  loss with the ground truth  $J_*^k$ :

$$\mathcal{L}_{\text{pose}} = \sum_{k=1}^K \left\| J^k - J_*^k \right\|_1 \quad (4)$$

where  $K$  is the number of joints.

#### 3.2.2 Unsupervised Domain Adaption

Due to the lack of annotated multiview multimodal 3D human pose dataset, we choose to generate a synthetic dataset to assist training. SyncHuman we designed can create 3D dynamic scenes of multiple persons with accurate labels. Besides, we develop an efficient unsupervised domain adaption means by designing a loss function to transfer the pretrained model from synthetic dataset to real-world dataset.

**SyncHuman.** We develop the synthetic system based on Unity Engine. Due to its flexibility and productivity, we can formulate a scene with a specific size and background as per our specifications. Concerning sensors' aspect, we implement Unity's built-in camera to obtain RGB images, and extract the depth information from the GPU's depth buffer with custom shaders used in the rendering pipeline. Furthermore, we

can attain the colored point cloud by sampling the depth and RGB images. For the sake of imitating the scanning process of LiDAR, we need to sample points along to a scanning function of time. Currently, we consider Livox Mid-40 LiDAR only, since its price is sufficiently low for sports or surveillance applications. We use the following function to simulate its scanning process:

$$r = \alpha \times \cos(3.825 \times (\theta_0 + 0.0017 \times n)) \quad (5)$$

where  $n \in [0, t \times 1e5]$  and  $t$  is in second.  $\alpha$  is the scanning radius in pixel, and  $\theta_0$  is a random initial angle. This equation is defined in polar coordinate. Final sampling points can be acquired by transforming it from polar coordinates to Cartesian coordinates and translating it to the center of the image space.

As for the avatars, we download various human 3D models from Adobe Mixamo<sup>1</sup>. With a view to guarantee the diversity of the generated actions, we can drive these avatars by either ready-made action files or other public datasets' keypoint annotations. Currently, we have developed the driving APIs for COCO17, COCO19 [36] and SMPL [40] standard keypoint annotated inputs. Regarding the groundtruth matter, we can obtain the 3D human pose from the avatars' humanoid skeleton; 2D pose label can be acquired via projecting 3D pose into 2D view. Additionally, we can fetch the mesh vertex of each avatar and then compute the semantic segmentation label for each point cloud by considering the pose label meanwhile.

**Unsupervised Domain Adaptation.** Similar to the 2D projection supervision utilized by other unsupervised or weakly supervised methods [16,29], we directly employ off-the-shelf 2D human pose estimation model to get the pseudo 2D pose label  $\tilde{J}_{2D}$ .  $\mathcal{L}_{2D}$  is obtained by calculating the  $L_2$  norm between 3D pose projection results and pseudo 2D label from each view.

$$\mathcal{L}_{2D} = \sum_{v=1}^V \sum_{k=1}^K \left\| \mathcal{P}_v(J^{v,k}) - \tilde{J}_{2D}^{v,k} \right\|_2 \quad (6)$$

where  $\mathcal{P}$  is the projection function.  $V$  is the number of views.  $\mathcal{L}_{2D}$  is the fundamental loss for unsupervised training. In order to attain pseudo 3D label, we choose to use information entropy as one uncertainty index. The entropy of one keypoint prediction heatmap (a spatial probability distribution)  $h^k$  is defined as:

$$\mathcal{H}(h^k) = - \sum_i h_i^k \times \log h_i^k \quad (7)$$

where  $i$  represents the voxel index. The higher the entropy is, the more uncertain the keypoint location is. To measure a person's uncertainty, we take the maximum entropy value of all keypoints. Different from Kundu et al. [30] who train the uncertainty value, we consider it as an artificial index. Our experiments demonstrate that the magnitude of entropy values can serve as an indicator of pose estimation quality for a specific network. Therefore, an entropy threshold  $\lambda$  can be set to sort out the reasonable predicted 3D human poses. In other words, we select the predicted pose  $\tilde{p}$  whose entropy value is less than  $\lambda$  as the pseudo 3D pose label for the next epoch's training, and it can be updated after each training epoch. Therefore, the 3D pseudo pose loss can be obtained by:

$$\mathcal{L}_{3D} = \sum_{k=1}^K \left\| J^k - \tilde{J}^k \right\|_1 \quad (8)$$

where  $\tilde{J}^k$  is the 3D joint of the selected pseudo 3D pose label  $\tilde{p}$ .

To further ensure the anatomical plausibility of the pose, we introduce a human prior loss  $\mathcal{L}_{prior}$ <sup>2</sup> adapted from Bigalke et al. [8]. Specifically, we formulate three losses to penalize asymmetric limb lengths  $\mathcal{L}_{symm}$ , implausible joint angles  $\mathcal{L}_{angle}$ , and implausible bone lengths  $\mathcal{L}_{length}$ . To sum up, the final loss function is defined as:

$$\mathcal{L}_{unsup} = \omega_1 \mathcal{L}_{2D} + \omega_2 \mathbb{1}(\tilde{p} < \lambda) \mathcal{L}_{3D} + \omega_3 \mathcal{L}_{prior} \quad (9)$$

<sup>1</sup><https://mixamo.com>

<sup>2</sup>Please refer to Appendix for details.

where  $\omega_1, \omega_2$  and  $\omega_3$  are the weights of each loss.

### 3.3 Avatar Modeling and Visualization in VR/AR

#### 3.3.1 Avatar Modeling

In our work, we employ an self-supervised 3D digital avatar modeling method SelfRecon [25], and focus on the high-precision reconstruction of players during competitions. The core of our method is the integration of implicit and explicit representations to recover space-time coherent geometries of clothed human bodies from monocular videos.

Specifically, our modeling process starts with a monocular video that captures a self-rotating player. Through the analysis of video frames, we use explicit representations to capture the player's overall shape, while implicit representations refine the geometric details of clothing and posture. The explicit representation recovers the overall shape through differentiable mask loss, and the implicit details are refined through sampled neural rendering loss and predicted normals. Consistency loss is introduced to maintain coherence between the two geometric representations.

Following SelfRecon, we apply forward deformation fields to generate space-time coherent explicit meshes, which are then updated through non-rigid ray casting and an implicit rendering network to refine the shape of the implicit neural representation. This method allows us to reconstruct high-fidelity, space-time coherent 3D models of players from video captured from a fixed perspective, maintaining high accuracy even when modeling various clothing types.

#### 3.3.2 Visualization in VR/AR

In previous sections, we present acquiring player locations and poses, as well as creating digital avatars for each player. In the following, we introduce how these elements are brought to life in virtual environments, allowing audiences to watch the game through VR/AR devices.

Utilizing Unity software, a leading game development platform known for its versatility in creating both 2D and 3D interactive content, we construct a realistic simulation of the sports venue. This digital environment serves as the backdrop for the reconstructed match, offering a virtual stage where the digital avatars can be showcased. The initial step involves importing the digital avatars of the players into the Unity scene. These models are then rigged with skeletal systems, enabling them to mimic the movements and poses of the real players based on the positional and postural data captured during the actual game.

This process of skeletal rigging is crucial, as it translates the athletes' dynamic actions into the digital realm, allowing for a detailed 3D reconstruction of the entire match. By doing so, we create a bridge between the physical performances on the field and their digital counterparts, ensuring that every sprint, tackle, and goal is reflected with precision in the virtual environment.

Furthermore, leveraging Unity's extensive toolset, we implement various interactive features to enhance the spectator experience. These include the ability to rotate the view 360 degrees, enabling viewers to observe the action from any angle. Spectators can also choose their viewing distance and position freely, offering a level of control that goes beyond traditional broadcast views. For a more personalized experience, we introduce a follow-cam feature that allows viewers to adopt the perspective of specific players, providing insight into their movements and strategies throughout the game.

The integration of VR/AR technology with sports visualizations offers a more engaging and immersive experience. Through the use of Unity software and the detailed recreation of player avatars, we are able to bring fans closer to the action than ever before, allowing them to explore the game from within a virtual space that replicates the intensity and excitement of the live event.

## 4 EXPERIMENTS

### 4.1 Basketball Dataset

In this paper, we introduce a *large-scale player tracking dataset*, Basketball, acquired with a flexible multiview multimodal imaging system, which consists of multiple statically positioned nodes with overlapping fields of view. Each acquisition node is equipped with a Livox Mid-100

Table 1: Tracking results are evaluated on Basketball test set using 3D MOT evaluation tool. Regain module and geometry constraints are denoted as “R” and “G” respectively.

	Point cloud	Image	Regain	Geometry	HOTA (%)	AssA (%)	MOTA (%)	IDS
3D	✓				32.86	14.69	95.24	169
2D		✓			29.42	13.08	86.71	56
2D+3D	✓	✓			50.56	34.90	95.28	64
2D+3D R	✓	✓	✓		52.91	38.03	96.14	26
2D+3D R+G (Ours)	✓	✓	✓	✓	<b>55.48</b>	<b>41.82</b>	<b>96.16</b>	<b>24</b>

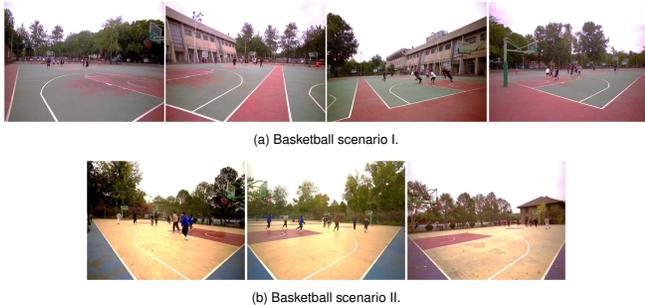


Fig. 4: Acquisition scenarios of our dataset

LiDAR sensor and a FLIR camera. We synchronize all sensors with  $\sim 5$  ms accuracy and provide an accurate extrinsic and intrinsic calibration following the method in [70]. All the multimodal sequences are recorded with a frame rate of 10 Hz. The images are captured of a resolution of  $2048 \times 1536$  pixels and the density of LiDAR points is about 30,000 points per frame for each node.

We collect data from two basketball games in different real-world scenarios. A total of nearly 11,000 frames are recorded, which are divided into 3 sequences. The first and second sequences are collected in basketball scenario I by four multimodal nodes, with each part consisting of 2,000 frames. The third sequence comprises 7,000 frames and is collected in basketball scenario II by three nodes. As for annotations, we provide 3D bounding boxes and 2D bounding boxes with tracking IDs for 4,000 frames of the first two sequences, consisting of 39,798 3D person detections in total.

## 4.2 Multi-player Tracking

We evaluate our method on Basketball dataset, using the first sequence as the training set and the second one as the evaluation set. The 7,000 unlabeled frames are used to verify the reliability of our system. We follow the evaluation setting of [61] and report the results of MOTA and IDS in CLEAR-MOT metrics [7]. To evaluate association accuracy, we provide the results of HOTA metric [41], which is a more comprehensive metric than CLEAR-MOT to balance the effect of performing accurate detection, association and localization.

**Multimodal Fusion Results.** We use the same 3D detection results provided by our multimodal multiview detector for a fair comparison. Utilizing the same detection results, we compare the tracking results of different modalities. We set  $\alpha = 0.5$  as a general proportion to fuse two modalities. As shown in Tab. 1, our multimodal multiview tracking method outperforms the single-modal tracking method by more than 22.62% in HOTA. Due to high detection performance, we achieve a bit better performance compared to the single-modal tracking method in CLEAR metrics which overemphasize the detection accuracy. It should be noted that the results in Tab. 1 demonstrate the MOTA metric is not suitable for evaluating long-term object tracking. Our multimodal multiview tracking method achieves a significantly better association accuracy, which is more important for the long-term tracking. This is because our method can better employ the 3D geometry information of point clouds and the appearance information of images to improve the association accuracy. We can observe that tracking only by 3D geometry information performs better than tracking only by appearance information, because the appearance information of images is less robust than the geometry information in 3D space.



Fig. 5: Qualitative results of our tracking method on two special cases: top is the objects leaving the field and bottom is the objects aggregated together. Our regain module can effectively recover the broken trajectories using the appearance features.

**Broken Trajectory Regain Module.** To evaluate the effectiveness of our trajectory regain module, we conduct an ablation study for the data association. We set a plain multimodal tracking method without the regain module and geometry constraints as the baseline. We compare the tracking results of the baseline and our multiview multimodal method with the regain module, evaluated on the Basketball dataset. As shown in Tab. 1, the regain module without geometry constraints reduce the ID switches because the regain module can effectively recover the broken trajectories using the appearance feature, which can improve the tracking continuity. Furthermore, our method with geometry constraints outperforms the baseline by 4.92% in HOTA. This is because the regain module can effectively recover the broken trajectories using the appearance features, and geometry constraints can ensure reasonable association results.

We show our qualitative results on two sequences of special cases in Fig. 5. The top sequence is the case where two players leave from the field and reappear, while our broken trajectory regain module is able to retrack the missing objects after they are detected. In the bottom sequence, two players are too close to split them up by detection. The regain module associates the broken trajectory with the detected object again as expected.

## 4.3 Pose Estimation

In this section, we analyze the performance of our pose estimation pipeline in supervised and unsupervised manner on different datasets. Since the Basketball dataset does not include annotations for pose estimation, we incorporate CMU Panoptic Studio [27] to quantitatively evaluate our method. Besides, we also conduct ablation studies to verify the effectiveness of each strategy or loss in unsupervised domain adaption.

### 4.3.1 Implementation Details

We use V2V-Net [47] as the point cloud and heatmap voxel’s feature extractor. The detailed V2Vnet design is the same with PRN [57]. The resolution of the voxel grid is  $64 \times 64 \times 64$  defined in a  $2m \times 2m \times 2m$  space. We set  $\omega_1 = 0.02$ ,  $\omega_2 = 1$ ,  $\omega_3 = 10$ , and  $\lambda = 6$ . We train the voxel-based pose estimation network on an NVIDIA GeForce RTX-3090 (24G) with a batch size of eight. The learning rate is set to 0.001 and the optimizer is Adam [28].

### 4.3.2 Datasets and Metrics

We use the following datasets in our experiments.

**I. CMU Panoptic Studio [27].** It is indoors with a valid scene range around  $5m \times 5m$ . In order to realize the multimodal inputs, we select

Table 2: Comparison of different 3D pose estimation methods on Panoptic and Player-Sync. As to unsupervised manner, we pretrain PointVoxel on Player-Sync in Panoptic testing and on Panoptic in Player-Sync testing.

Methods	Supervision	Panoptic		Player-Sync	
		MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
MvP [69]	sup.	25.78	25.02	291.11	240.32
PlanePose [35]	sup.	18.54	11.92	119.55	65.80
VoxelPose(PRN) [57]	sup.	14.91	11.88	40.80	34.34
<i>Ours</i>	sup.	<b>14.44</b>	<b>11.61</b>	<b>31.84</b>	<b>27.36</b>
DLT [54]	no sup.	38.26	44.77	101.49	62.77
<i>Ours</i>	no sup.	<b>22.00</b>	<b>15.96</b>	<b>72.92</b>	<b>62.72</b>

Table 3: Comparison of different input modalities for PointVoxel on Panoptic and Player-Sync. Model tested on Player-Sync is pretrained on Panoptic, and is pretrained on Player-Sync for Panoptic testing.

Input modality	supervision	Panoptic		Player-Sync	
		MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
p.c.	sup.	164.30	148.78	136.01	123.22
h.m.	sup.	14.91	11.88	40.80	34.34
h.m.+p.c.	sup.	<b>14.44</b>	<b>11.61</b>	<b>31.84</b>	<b>27.36</b>
h.m.	no sup.	25.72	17.24	263.05	183.85
h.m.+p.c.	no sup.	<b>22.00</b>	<b>15.96</b>	<b>72.92</b>	<b>62.72</b>

several subsets<sup>3</sup> that are evaluated in [35] and have depth information. We unify the depth information from Kinect 1 to 5 and process these depth maps by Eq. (5) to get the sparse point cloud which is similar to Mid-40 Livox LiDAR’s scanning. About the 2D pose estimation results, we use the results provided by [35] which are claimed to be predicted from HRNet [56].

**II. Basketball.** As described in Sec. 4.1, it has labels for multi-player detection and tracking, while lacks of the 3D keypoints groundtruth. Therefore, it is mainly used for qualitative analysis.

**III. Player-Sync.** It is a synthetic dataset generated by SyncHuman with the same sensors’ configuration and setting as Basketball. The point cloud is obtained by simulating the scan pattern of Mid-100 Livox LiDARs. It contains ten avatars acting randomly with the first eight for training and the rest two for testing among all frames (3336 frames in 10Hz). The predicted 2D human poses are extracted by VitPose [65].

**Evaluation metrics.** For evaluating the 3D human pose, we calculate the standard mean per-joint position error before and after Procrustes Alignment [17] as MPJPE and PA-MPJPE in millimeter respectively.

#### 4.4 3D Pose Estimation Analysis

In this section, we evaluate the 3D human pose estimation results of different approaches. For PlaneSweepPose [35], VoxelPose [57] and PointVoxel in supervised manner, we utilize the groundtruth location of each person into the process of training and testing. As to MvP [69], it is difficult to add the groundtruth detection information into the network due to its architecture design, which also means that it cannot utilize the depth or point cloud information directly. For fairness, we just evaluate the MPJPE@500 metrics (only considering per joint error less than 500 mm) for the matched person as to MvP.

We evaluate these approaches on Panoptic Studio and Player-Sync as small scene and large scene respectively. Table 2 shows that our approach outperforms them in terms of MPJPE and PA-MPJPE especially in large scene’s setting. It is worth noting that our approach with unsupervised learning manner can even outperform supervised RGB inputs multiview approaches.

Additionally, we analyze the performance with inputs of different modalities (point cloud, RGB, and both of them) in volumetric archi-

<sup>3</sup>“160422\_ultimatum1”, “160224\_haggling1”, “160226\_haggling1”, “161202\_haggling1”, “160906\_ian1”, “160906\_ian2”, “160906\_ian3”, and “160906\_band1” for training; “160906\_pizza1”, “160422\_haggling1”, “160906\_ian5”, “160906\_band2” for testing.

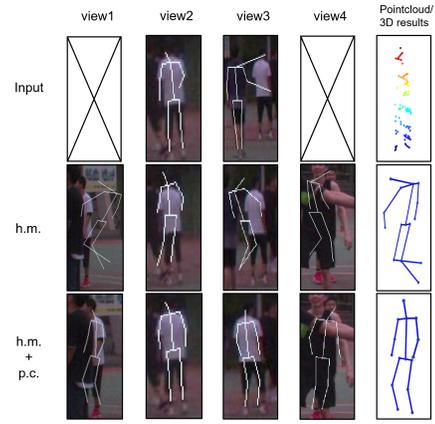


Fig. 6: Comparison of different input modalities training on Basketball. The first row is the 2D pose and point cloud for model inputs, the second row’s results are from the model trained by 2D pose heatmap (h.m.) inputs, and the third row’s results are from the model trained by 2D pose heatmap and point cloud (h.m.+p.c.) inputs. Both models utilize the complete unsupervised training strategy.

Table 4: Comparison of different unsupervised training strategies. The model is pretrained on Player-Sync for Panoptic testing, and on Panoptic for Player-Sync testing. “Entropy” means adding entropy selected pseudo 3D pose supervision, and “Prior” means adding human prior loss.

	Unsup. loss		Panoptic		Player-Sync	
	Entropy	Prior	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
			29.14	16.48	84.49	75.55
✓			28.26	25.85	77.38	68.84
		✓	22.88	24.69	76.45	65.54
✓	✓		<b>22.00</b>	<b>15.96</b>	<b>72.92</b>	<b>62.72</b>

ture. Table 3 illustrates that the Livox point cloud information alone is unable to accurately extract 3D human skeleton due to its sparsity, and cannot be adopted in unsupervised manner without 2D pseudo labels from heatmaps. However, when combined with RGB information, the performance is greatly enhanced. Figure 6 demonstrates that multimodal inputs increase the model’s tolerance to 2D pose estimation error, including severe situations like miss-prediction or predicting on a wrong person, during unsupervised domain adaption training.

#### 4.5 Unsupervised Domain Adaption

For unsupervised domain adaption, pseudo 2D pose supervision is necessary, and it is the baseline. Besides, we adopt an interpretable human prior loss and a pseudo 3D pose (selected by entropy value) loss to assist the learning. Therefore, we conduct an ablation study to analyze the impact of these two losses. From Tab. 4, entropy selected pseudo 3D pose loss improves the performance, but it does not limit the rationality which means it may generate low-entropy invalid 3D poses. Prior loss controls the results in reasonable action range. Therefore, we choose both losses as our efficient training strategy.

Table 5: The data statistics of one player in the basketball game.

Player ID	playing time 3min 18sec	running distance/m 329.3	sprint distance/m 3.5
10	sprint time/s 0.5	jogging time/s 53.5	top speed/m/s 9.6

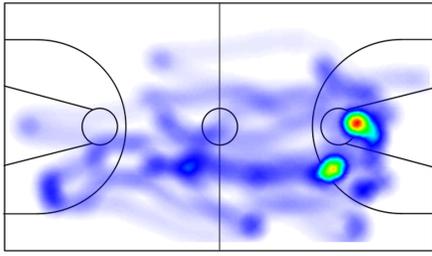


Fig. 7: The position heatmap of one player in the game.

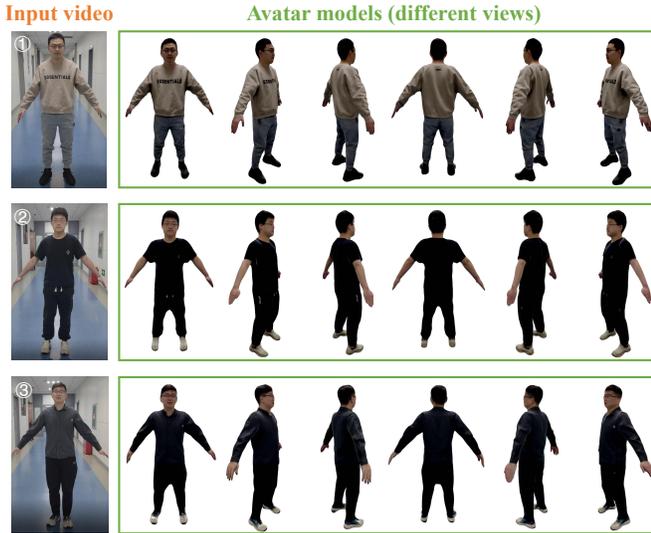


Fig. 8: Several examples of our avatar modeling. The left side shows the first frame of the input monocular video with the player rotating in place. The right side displays the results of 3D modeling observed from different viewpoints.

#### 4.6 Sports Analysis

The player tracking approach described in this paper can be used to build an entire data analysis pipeline. Each player’s statistics and analysis can be obtained according to the tracking results, like playing time, position distance, top speed, sprint distance, jogging time, etc. The results of one player are shown in Tab. 5. In this context, a sprint is defined as a speed of more than 6 m/s and a jog as a speed of less than 1 m/s. Additionally, as shown in Fig. 7, we plot a player’s position heatmap. The position heatmap helps to analyze the player’s movement as well as propensity to assault and defend in certain regions. Besides these, the tracking results can be adopted to complete a wide range of statistics, which is very useful for applications involving game analysis.

#### 4.7 Avatar Modeling

With monocular videos, we perform 3D avatar modeling for each player in the match to facilitate visualization on VR/AR platforms. The avatar modeling process results in 3D meshes of the players, with colored textures. These models are drivable, meaning they can be animated or manipulated to replicate the real movements of players during a game. The avatar models of several participants are shown in Fig. 8. We can observe that these meshes and textures capture the physical appearance, clothing details, and even subtle expressions, contributing to a lifelike representation. This realism enhances the immersive experience for audiences, making them feel like they’re witnessing the game firsthand or interacting with real-life athletes in the virtual space.

#### 4.8 Watching Games in VR/AR

We drive the avatar models of players based on the results of tracking and pose estimation, visualizing them in virtual space. In this way, au-



Fig. 9: Various perspectives and distances when watching the game with the HoloLens. The six viewpoints are selected from actual game-viewing videos, with more details available in the supplementary material. When watching the game, users can choose any angle and distance they prefer, even opting to follow a specific player.

diences can watch the game on VR/AR devices. In Fig. 9, we showcase the effects of watching the game in AR device HoloLens [46] from different positions and perspectives. The corresponding demonstration video is included in the supplementary materials, presenting an audience using the HoloLens to watch a game. This visualization illustrates the versatility and immersive potential of AR technology in sports viewing, allowing users to experience the game as if they were present in various locations within the stadium or even on the field itself. By adjusting their viewpoint, users can choose their preferred perspective, whether it’s a bird’s eye view, a sideline perspective, or a position right behind the goalposts. This flexibility enriches the viewing experience, offering a personalized approach to engaging with live sports in a way that traditional broadcasting cannot match. The integration of accurately driven avatar models into these immersive environments further enhances the realism, making the virtual experience more compelling and engaging for fans worldwide.

### 5 CONCLUSION AND FUTURE WORK

In conclusion, this work presents a new approach in sports competition analysis and VR/AR visualization. By leveraging multimodal and multiview LiDARs and cameras for data collection, coupled with a novel framework for multi-player tracking and pose estimation, we enhance the accuracy of sports visualization and analysis. Furthermore, the integration of avatar modeling and actuation of players provides audiences with a more immersive and engaging experience, while also enhancing usability and interactivity. Our system stands out for its real-time performance and cost-efficiency. The extensive experiments underscore the potential of our system to transform the sports domain by providing a more detailed and immersive viewing experience.

While we efficiently complete the 3D modeling and actuation of players in the VR/AR visualization process, we have not accounted for certain props specific to games, such as basketballs and soccer balls in basketball and soccer matches, or racquets in tennis and badminton matches. These props are also important for the audiences’ viewing experience. In our future work, we will focus on the modeling and tracking of props used in games. This inclusion aims to enhance the realism and completeness of the virtual viewing experience by accurately representing not only the players but also the essential elements they interact with during a game.

## REFERENCES

- [1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3686–3693, 2014. 2
- [2] Apple Inc. Apple vision pro. <https://www.apple.com/vision-pro/>, 2024. Accessed: 2024-03-30. 2
- [3] R. Bashirov, A. Ianina, K. Iskakov, Y. Kononenko, V. Strizhkova, V. Lempitsky, and A. Vakhtov. Real-time rgbd-based extended body pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2807–2816, 2021. 3
- [4] W. Bekhtaoui, R. Sa, B. Teixeira, V. Singh, K. Kirchberg, Y.-j. Chang, and A. Kapoor. View invariant human body detection and pose estimation from multiple depth sensors. *arXiv preprint arXiv:2005.04258*, 2020. 3
- [5] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3D pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1669–1676, 2014. 2
- [6] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011. 2, 3
- [7] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP J. Image Video Process.*, 2008. 7
- [8] A. Bigalke, L. Hansen, J. Diesel, and M. P. Heinrich. Domain adaptation through anatomical constraints for 3D human pose estimation under the cover. In *International Conference on Medical Imaging with Deep Learning*, pp. 173–187, 2022. 6
- [9] H. Chiu, J. Li, R. Ambrus, and J. Bohg. Probabilistic 3D multi-modal, multi-object tracking for autonomous driving. In *ICRA*, pp. 14227–14233, 2021. 2, 3
- [10] P. Cong, Y. Xu, Y. Ren, J. Zhang, L. Xu, J. Wang, J. Yu, and Y. Ma. Weakly supervised 3D multi-person pose estimation for large-scale scenes based on monocular camera and single LiDAR. *arXiv preprint arXiv:2211.16951*, 2022. 3
- [11] J. Dong, W. Jiang, Q. Huang, H. Bao, and X. Zhou. Fast and robust multi-person 3D pose estimation from multiple views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7792–7801, 2019. 3, 5
- [12] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 5
- [13] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2007. 2, 3
- [14] N. Garau, N. Bisagno, P. Bródka, and N. Conci. DECA: Deep viewpoint-equivariant human pose estimation using capsule autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11677–11686, 2021. 3
- [15] B. G. Gerats, J. M. Wolterink, and I. A. Broeders. 3D human pose estimation in multi-view operating room videos using differentiable camera projections. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp. 1–9, 2022. 2, 3
- [16] M. Gholami, A. Rezaei, H. Rhodin, R. Ward, and Z. J. Wang. Self-supervised 3D human pose estimation from video. *Neurocomputing*, 488:97–106, 2022. 6
- [17] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975. 8
- [18] H. Guo, G. Wang, X. Chen, and C. Zhang. Towards good practices for deep 3D hand pose estimation. *arXiv preprint arXiv:1707.07248*, 2017. 3
- [19] L. Hansen, M. Siebert, J. Diesel, and M. P. Heinrich. Fusing information from multiple 2D depth cameras for 3D human pose estimation in the operating room. *International Journal of Computer Assisted Radiology and Surgery*, 14:1871–1879, 2019. 2, 3
- [20] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei. Towards viewpoint invariant 3D human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pp. 160–177, 2016. 3
- [21] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *ICCV*, pp. 2980–2988, 2017. 4
- [22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016. 4
- [23] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2013. 2
- [24] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7718–7727, 2019. 2, 3
- [25] B. Jiang, Y. Hong, H. Bao, and J. Zhang. Selfrecon: Self reconstruction your digital avatar from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5605–5615, 2022. 6
- [26] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, and P. Luo. Whole-body human pose estimation in the wild. In *Proceedings of the European Conference on Computer Vision*, pp. 196–214, 2020. 2
- [27] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3334–3342, 2015. 2, 7
- [28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7
- [29] M. Kocabas, S. Karagoz, and E. Akbas. Self-supervised learning of 3D human pose using multi-view geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1077–1086, 2019. 6
- [30] J. N. Kundu, S. Seth, P. YM, V. Jampani, A. Chakraborty, and R. V. Babu. Uncertainty-aware adaptation for self-supervised 3D human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20448–20459, 2022. 6
- [31] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *CVPR*, pp. 12697–12705, 2019. 3
- [32] J. Li, J. Zhang, Z. Wang, S. Shen, C. Wen, Y. Ma, L. Xu, J. Yu, and C. Wang. Lidarcap: Long-range marker-less 3d human motion capture with lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20502–20512, 2022. 3
- [33] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3D object detection. In *Proceedings of the European Conference on Computer Vision*, pp. 641–656, 2018. 3
- [34] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang. BEVFusion: A simple and robust lidar-camera fusion framework. *CoRR*, abs/2205.13790, 2022. 2
- [35] J. Lin and G. H. Lee. Multi-view multi-person 3D pose estimation with Plane Sweep Stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11886–11895, 2021. 2, 3, 8
- [36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pp. 740–755, 2014. 2, 6
- [37] D. Linke, D. Link, and M. Lames. Football-specific validity of TRACAB’s optical video tracking systems. *PLoS one*, 15(3):e0230179, 2020. 2
- [38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV (1)*, vol. 9905, pp. 21–37. Springer, 2016. 4
- [39] Live Like, Inc. Live like: Transforming fan engagement with interactive experiences. <https://livelike.com/>, 2024. Accessed: 2024-03-30. 2
- [40] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):1–16, 2015. 6
- [41] J. Luiten, A. Osep, P. Dendorfer, P. H. S. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vis.*, 129(2):548–578, 2021. 7
- [42] W. Luo, B. Yang, and R. Urtasun. Fast and Furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, 2018. 2, 3
- [43] M. Manaffard, H. Ebadi, and H. A. Moghaddam. A survey on player tracking in soccer videos. *Comput. Vis. Image Underst.*, 159:19–46, 2017. 2
- [44] A. Martínez-González, M. Villamizar, O. Canévet, and J.-M. Odobez. Residual pose: A decoupled approach for depth-based 3D human pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 10313–10318, 2020. 3
- [45] Meta Platforms, Inc. Oculus quest: All-in-one vr gaming system. <https://www.oculus.com/quest/>, 2024. Accessed: 2024-03-30. 2

- [46] Microsoft Corporation. Hololens: Mixed reality technology for professionals and developers. <https://www.microsoft.com/en-us/hololens>, 2024. Accessed: 2024-03-30. 2, 9
- [47] G. Moon, J. Y. Chang, and K. M. Lee. V2V-Posenet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 5079–5088, 2018. 3, 5, 7
- [48] D. M. H. Nguyen, R. Henschel, B. Rosenhahn, D. Sonntag, and P. Swo-boda. LMGP: lifted multicut meets geometry projections for multi-camera multi-object tracking. In *CVPR*, pp. 8856–8865, 2022. 3
- [49] Pico Interactive, Inc. Pico neo 3: Standalone vr headset for business. <https://www.picoxr.com/cn/neo3/>, 2024. Accessed: 2024-03-30. 2
- [50] A. Piergiovanni, V. Casser, M. S. Ryoo, and A. Angelova. 4D-net for learned multi-modal alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15435–15445, 2021. 3
- [51] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, pp. 77–85, 2017. 3
- [52] K. G. Quach, P. Nguyen, H. Le, T.-D. Truong, C. N. Duong, M.-T. Tran, and K. Luu. DyGLIP: A dynamic graph model with link prediction for accurate multi-camera multiple object tracking. In *CVPR*, pp. 13784–13793, June 2021. 2, 3
- [53] N. D. Reddy, L. Guigues, L. Pishchulin, J. Eledath, and S. G. Narasimhan. Tesseract: End-to-end learnable multi-person articulated 3D pose tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15190–15200, 2021. 2, 3, 5
- [54] E. Remelli, S. Han, S. Honari, P. Fua, and R. Wang. Lightweight multi-view 3D pose estimation through camera-disentangled representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6040–6049, 2020. 8
- [55] M. Simon, K. Amende, A. Kraus, J. Honer, T. Sämman, H. Kaulbersch, S. Milz, and H. Gross. Complexer-YOLO: Real-time 3D object detection and tracking on semantic point clouds. In *CVPR Workshops*, pp. 1190–1199, 2019. 3
- [56] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703, 2019. 8
- [57] H. Tu, C. Wang, and W. Zeng. VoxelPose: Towards multi-camera 3D human pose estimation in wild environment. In *Proceedings of the European Conference on Computer Vision*, pp. 197–212, 2020. 2, 3, 5, 7, 8
- [58] C. D. Vleeschouwer and D. Delannay. Basket ball dataset from the european project apidis, 2009. 3
- [59] T. Von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll. Recovering accurate 3D human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision*, pp. 601–617, 2018. 2
- [60] S. Wang, Y. Sun, C. Liu, and M. Liu. PointTrackNet: An end-to-end network for 3-D object detection and tracking from point clouds. *IEEE Robotics Autom. Lett.*, 2020. 2
- [61] X. Weng, J. Wang, D. Held, and K. Kitani. 3D multi-object tracking: A baseline and new evaluation metrics. In *IROS*, pp. 10359–10366, 2020. 2, 3, 5, 7
- [62] X. Weng, Y. Wang, Y. Man, and K. M. Kitani. GNN3DMOT: graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning. In *CVPR*, 2020. 2
- [63] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pp. 3645–3649, 2017. 3
- [64] S. Wu, S. Jin, W. Liu, L. Bai, C. Qian, D. Liu, and W. Ouyang. Graph-based 3D multi-person pose estimation using multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11148–11157, 2021. 3
- [65] Y. Xu, J. Zhang, Q. Zhang, and D. Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*, 2022. 5, 8
- [66] Y. Yan, Y. Mao, and B. Li. SECOND: sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 4
- [67] H. Ye, W. Zhu, C. Wang, R. Wu, and Y. Wang. Faster VoxelPose: Real-time 3d human pose estimation by orthographic projection. In *Proceedings of the European Conference on Computer Vision*, pp. 142–159, 2022. 2
- [68] T. Yin, X. Zhou, and P. Krähenbühl. Center-based 3D object detection and tracking. In *CVPR*, pp. 11784–11793, 2021. 2, 3, 5
- [69] J. Zhang, Y. Cai, S. Yan, J. Feng, et al. Direct multi-view multi-person 3D pose estimation. *Advances in Neural Information Processing Systems*, 34:13153–13164, 2021. 2, 3, 8
- [70] M. Zhang, W. Guo, B. Fan, Y. Chen, J. Feng, and J. Zhou. A flexible multi-view multi-modal imaging system for large-scale outdoor scenes. In *3DV*, 2022. 7
- [71] R. Zhang, L. Wu, Y. Yang, W. Wu, Y. Chen, and M. Xu. Multi-camera multi-player tracking with deep player identification in sports video. *Pattern Recognit.*, 102:107260, 2020. 3
- [72] S.-H. Zhang, R. Li, X. Dong, P. Rosin, Z. Cai, X. Han, D. Yang, H. Huang, and S.-M. Hu. Pose2seg: Detection free human instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 889–898, 2019. 2
- [73] Y. Zhang, C. Wang, X. Wang, W. Liu, and W. Zeng. Voxeltrack: Multi-person 3D human pose estimation and tracking in the wild. *CoRR*, abs/2108.02452, 2021. 2, 3
- [74] Z. Zhang, L. Hu, X. Deng, and S. Xia. Sequential 3D human pose estimation using adaptive point cloud sampling strategy. In *International Joint Conferences on Artificial Intelligence Organization*, pp. 1330–1337, 2021. 3
- [75] J. Zheng, X. Shi, A. Gorban, J. Mao, Y. Song, C. R. Qi, T. Liu, V. Chari, A. Cornman, Y. Zhou, et al. Multi-modal 3D human pose estimation with 2D weak supervision in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4478–4487, 2022. 3
- [76] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. Distance-IoU loss: Faster and better learning for bounding box regression. In *AAAI*, pp. 12993–13000, 2020. 4
- [77] X. Zhou, V. Koltun, and P. Krähenbühl. Tracking objects as points. In *ECCV (4)*, vol. 12349, pp. 474–490, 2020. 3