Machine Vision and Applications (2024) 35:46
https://doi.org/10.1007/s00138-024-01530-6

RESEARCH

Multi-person 3D pose estimation from unlabelled data

Daniel Rodriguez-Criado[1] · Pilar Bachiller-Burgos[2] · George Vogiatzis[3] · Luis J. Manso[1]

Received: 1 November 2023 / Revised: 13 February 2024 / Accepted: 11 March 2024 / Published online: 6 April 2024

Abstract

Its numerous applications make multi-human 3D pose estimation a remarkably impactful area of research. Nevertheless, it presents several challenges, especially when approached using multiple views and regular RGB cameras as the only input. First, each person must be uniquely identified in the different views. Secondly, it must be robust to noise, partial occlusions, and views where a person may not be detected. Thirdly, many pose estimation approaches rely on environment-specific annotated datasets that are frequently prohibitively expensive and/or require specialised hardware. Specifically, this is the first multi-camera, multi-person data-driven approach that does not require an annotated dataset. In this work, we address these three challenges with the help of self-supervised learning. In particular, we present a three-staged pipeline and a rigorous evaluation providing evidence that our approach performs faster than other state-of-the-art algorithms, with comparable accuracy, and most importantly, does not require annotated datasets. The pipeline is composed of a 2D skeleton detection step, followed by a Graph Neural Network to estimate cross-view correspondences of the people in the scenario, and a Multi-Layer Perceptron that transforms the 2D information into 3D pose estimations. Our proposal comprises the last two steps, and it is compatible with any 2D skeleton detector as input. These two models are trained in a self-supervised manner, thus avoiding the need for datasets annotated with 3D ground-truth poses.

Keywords 3D multi-pose estimation · Skeleton matching · Deep learning · Graph neural networks · Self-supervised learning

1 Introduction

Human detection and pose modelling have a plethora of applications, including video surveillance [50], assisted living [13], and autonomous vehicles [17]. In addition to any direct application, it is also the basis of trajectory prediction [41, 44], interaction detection, and gesture recognition [2]. The number and relevance of applications make it extremely impactful. Extensive research efforts have been made with different technologies such as LiDAR [45, 54], RGB cameras [47], and RGBD cameras [11, 56].

Multiple usability, cost and operational requirements can be expected in a Human Pose Estimator (HPE). First, most applications require support for more than one person. Secondly, except for very few niche cases, HPEs must work with occluded body parts. Most applications will also benefit from limiting sensors to RGB cameras, avoiding RGBD or other more expensive hardware. An ideal HPE would also exploit the available context to provide 3-dimensional data for all keypoints, even if not all of them are visible. Another desirable feature for any learning-based HPE would be not to require a labelled dataset to be implemented in a new space, as they are expensive to compile.

RGB-based multi-human and multi-view 3D pose estimation is usually done in three steps: a) detect humans and estimate their 2D poses on the images using, for example, a Convolutional Neural Network (CNN); b) search for correspondences in the different views of the people detected in the previous step; and c) estimate 3D poses for each person

✉ Pilar Bachiller-Burgos
pilarb@unex.es

Daniel Rodriguez-Criado
190229717@aston.ac.uk

George Vogiatzis
g.vogiatzis@lboro.ac.uk

Luis J. Manso
l.manso@aston.ac.uk

[1] Computer Science Dept., Aston University, Aston Triangle, Birmingham B4 7ET, UK

[2] Computer and Communication Technology Dept., Universidad de Extremadura, Avd. de la Universidad, 10001 Cáceres, Extremadura, Spain

[3] Computer Science Dept., Loughborough University, Epinal Way, Loughborough LE11 3TU, UK

based on the image coordinates of their keypoints for the different views. This work builds on top of publicly available pose detectors (e.g., [1, 12]) for the first step of the pipeline and presents a novel solution for the second and third steps. It is important to note that the system developed in this research can be integrated with any third-party 2D detector.

Regarding the second step, which consists of associating the 2D poses that correspond to the same person in the different images, the literature addresses the problem using both appearance and geometry cues. Examples of this are the use of epipolar geometry to assign a cost to each pose detected [10] or the embedding of appearance features using a pre-trained model to provide affinity scores between bounding boxes [15]. Due to the desired multi-person support and the irrelevance of the order in which people are detected, we chose to exploit Graph Neural Networks (GNNs) to match people's views, as they are order-invariant and can manage a variable number of input nodes.

Traditionally, the final step, 3D pose estimation, has been done using triangulation or pictorial structure models. The main limitation of these classic approaches stems from the inability to predict the occluded parts, as these methods are not capable of estimating positions for keypoints that are occluded in many or all views. To overcome these limitations, learning-based solutions have emerged. It can be argued that an artificial neural network can learn to *hallucinate* the occluded parts of the body even if they are not visible. This is based on the intuition that the network should be able to exploit contextual information from the rest of the keypoints and the existing views, if any. For instance, a network could learn to implicitly internalise the proportions of the human body and its bilateral symmetry. Therefore, if the keypoint for the left elbow cannot be seen from any camera, knowing the position of the wrist and the average proportions of a human forearm (or the length of the opposite forearm), the network could estimate the position of the elbow. Embedding these complex but helpful biases efficiently would be very challenging in non-data-driven approaches.

A significant limitation of most current data-driven solutions, and more importantly, all of those that provide multi-camera support for multi-person pose estimation, is the necessity of annotating the datasets to train the models in a supervised fashion. It is worth noting that multi-camera datasets are specific to the relative positions of the cameras, making the datasets scenario-specific. As a consequence, to use the corresponding approaches, an annotated dataset has to be compiled for every scenario, which is time-consuming and requires expensive tracking systems.

In addition, while it is feasible to utilise 3D data from an in-studio dataset to establish 2D-3D relationships for different camera configurations via 3D projections, a model trained on such a dataset would exhibit significant sensitivity to variations in the 2D detected keypoints. This sensitivity arises from training the model using ideal 2D coordinates, in contrast to the potentially noisy 2D coordinates used during inference. Even assuming that the 2D keypoint detection is noise-free, training using ground truth data will likely fail at inference time due to ground truth keypoint projection coordinates not matching the coordinates of the skeleton keypoints considered by the 2D detectors. Another limitation of this dataset generation approach is related to the variability of the dataset. Depending on the application, it could be necessary to have data on individuals with diverse complexion, heights, and even ages, which might not be readily available in an in-studio dataset. The process of gathering such diverse data would essentially take us back to the initial challenge: obtaining an annotated dataset.

To deal with these problems and avoid the need for annotated datasets, we propose a self-supervised learning-based solution, with two main contributions:

- An elegant solution for matching different 2D poses from several cameras using a GNN that allows having a variable number of people in the scenario.
- A model that infers the 3D keypoints of the detected humans using self-supervised learning by minimising the difference between the 2D detected keypoints' coordinates and those of the estimated poses' re-projections.

The following sections cover various aspects of our work. Section 2 reviews relevant 2D human pose detectors and presents the current state of 3D pose estimation. The proposed method is described in detail in Sect. 3. Section 4 presents experimental results, including a performance comparison with other state-of-the-art methods, using two distinct datasets. Additionally, we will show how the system can be applied to mobile robots without retraining for different scenarios as long as only on-board cameras are used. Finally, Sect. 5 summarises the main conclusions.

2 Related work

This section provides an overview of the leading literature on 3D Human Pose Estimation. We start with a brief discussion of popular 2D detectors, as they are leveraged in various 3D estimation models -including ours. We omit works that utilise RGB-D sensors, since our work focuses on RGB cameras, offering the advantage of significantly reduced equipment costs.

2D human pose estimators yield image coordinates of human anatomical keypoints in an image for every detected person. Recent advancements in deep learning have led to a significant improvement in the performance and accuracy of these models, surpassing the previous approaches that relied on probabilistic and hand-crafted features [14]. Most

of these learning-based models [1, 23, 26, 46] rely on Convolutional Neural Networks. There is a vast number of 2D pose estimators, with OpenPose [12] being one of the most popular. It leverages part affinity fields for human parts association using a bottom-up approach. A similar approach is followed by OpenPifPaf [27] and trt-pose.[1] Another widely known 2D pose detector is HRNet [43], which can maintain high-resolution representations through the detection process, claiming higher accuracy and spatial precision. One of the most popular datasets used for training and evaluating these 2D models is COCO [31], containing more than 100,000 annotated images.

In relation to the **3D pose estimation** problem, fuelled by the outstanding advances in 2D estimations, many works have tried to utilise these models for estimating 3D poses from the 2D points [49]. Many of them retrieve 3D human poses from monocular views [19, 29, 32–36, 39], although they suffer from the unavoidable fact that monocular depth estimation is an ill-posed problem, as multiple potential 3D poses are possible given a single 2D projection. Approaches like Park et al. [35] attempt to mitigate this by utilizing short video sequences for multiple perspectives, but limitations remain when camera motion or subject movement is minimal. **Multi-camera** systems offer significant advantages in terms of reducing ambiguity and enhancing robustness to occlusions and noise. However, multi-view Human Pose Estimation with **multiple people** introduces the challenge of matching each person's set of keypoints among the images of the different cameras. Previous works have addressed this problem with algorithms based on appearance and geometric information [7, 15]. [15] creates affinity matrices based on the appearance between two views and use them as input to their model to infer the correspondence matrix.

Once the cross-view correspondences are solved, there are several techniques to merge the information from the different views to extract the 3D pose. Most classical approaches rely on epipolar geometry by triangulating the 3D points from the 2D points [3, 7, 25]. The pictorial structure paradigm was extended to 3D to deal with multi-human pose detection in [6]. However, the model does not detect full skeletons in case of occlusions and they assume to know the number of people in the scene, which is not a realistic assumption [47]. Other works tackle the problem with prediction models based on **deep learning** and CNNs [38, 47, 55]. For example, VoxelPose [47] discretises the 3D space in small cubes called voxels. Using this representation, the 2D heatmaps detected from all the views are projected into a common 3D space and two 3D convolutional models are applied. The first model yields detection proposals for each person and the second estimates the positions of the keypoints for each proposal. This method avoids establishing cross-view correspondence based on poor-quality 2D poses. *Ye et al.* [55] present an accelerated version of VoxelPose which avoids the use of 3D convolutions, although the results are marginally worse. Firstly, they re-project the aggregated feature volume, which is acquired in the same way as in [47], to the ground plane (xy) by implementing max-pooling along the z-axis. Next, they employ a 2D-CNN network over the xy-plane to locate individuals and generate a 1D feature vector in the z-axis for each detection. Finally, they apply a 1D-CNN to that vector to get the final 3D pose estimation. These modifications enable their model to achieve results approximately 10 times faster without sacrificing precision. Another interesting work is presented by [30]. Their approach utilises a plane sweep stereo technique to simultaneously address the challenges of multiple-view fusion and 3D pose estimation. All these models use supervised learning, thus they require datasets with precise annotations. The number of these datasets is scarce mainly due to the costly equipment required as well as the need for a controlled environment to record the data. Some examples are the Human3.6M dataset [22], with more than 10 thousand annotations from 1 thousand images, and the CMU Panoptic dataset [24], with 5.5 h of video from different angles and 1.5 million of 3D annotated skeletons.

Besides CNN-based models, multiple works in the literature address the problem with the use of GNNs. For example, works such as [20, 53] obtain promising results from monocular views using GNNs. Wu et al. [52] propose a solution for multi-view and multi-person 3D estimation using GNNs with supervised learning for both, cross-view correspondence and final 3D pose estimation. They construct the graphs by transforming each detected keypoint into a graph node and use the natural connections in the body to generate the graph edges. Then the GNN applies a regression in the node features to obtain the 3D coordinates of the body joints. The main limitation is that the training of these networks requires datasets with accurate 3D ground truth annotations.

Due to the remarkable benefits of avoiding 3D annotations (i.e., datasets become much more affordable in terms of cost and effort, which facilitates collecting larger datasets), self or semi-supervised learning methods have been proposed in many works [5, 8, 9, 16, 18, 25, 28, 36, 37, 42]. These methods primarily focus on single-view 3D pose estimation or are constrained to single individuals. To the best of our knowledge, our proposal stands as the first multi-camera 3D Human Pose Estimation method that supports multiple individuals without requiring ground truth data. A qualitative comparison of recent works is presented in Table 1, showing that our method is the only one that meets these three criteria.

[1] https://github.com/NVIDIA-AI-IOT/trt_pose

Table 1 Qualitative comparison with literature

Reference	Multi-camera	Multi-person	Self-supervised
Tu et al. [47]	✓	✓	✗
Ye et al. [55]	✓	✓	✗
Wu et al. [52]	✓	✓	✗
Lin and Li [30]	✓	✓	✗
Liu et al. [32]	✗	✓	✗
Park et al. [35]	✗	✓	✗
Guan et al. [19]	✗	✗	✗
Biswas et al. [8]	✗	✗	✓
Kundu et al. [28]	✗	✗	✓
Srivastav et al. [42]	✗	✓	✓
Bouazizi et al. [9]	✓	✗	✓
Bartol et al. [5]	✓	✗	✓
Bala et al. [4]	✓	✗	✓
Gong et al. [18]	✓	✗	✓
Ours	✓	✓	✓

3 Method

The proposed system consists of a three-staged pipeline: a) a skeleton detector, b) a multi-view skeleton matching Graph Neural Network, and c) a pose estimation Multi-Layer Perceptron (MLP). Given that there are very efficient solutions for the first stage of the pipeline, no new alternative is proposed in this work. In fact, our system is independent of the skeleton detector used. The multi-view skeleton matching and the pose estimation network are our two main contributions. Figure 1 shows how these two stages of the pipeline work at test time, which take as input a set of detected skeletons per view that can be obtained using any skeleton detector. The code is available at https://github.com/gnns4hri/3D_multi_pose_estimator.

3.1 System calibration

Our proposal is not limited to a number or configuration of cameras but requires the camera configuration to be the same during the collection of the dataset and the final inference for pose estimation. The only exception to this is that our system allows the set of cameras used at inference time ($\mathbb{C}_i$) to be a subset of the cameras used during training ($\mathbb{C}_t$). In that case, only the cameras in $\mathbb{C}_i$ need to maintain the configuration they had at training time during inference time. If desired, the rest of the cameras in $\mathbb{C}_t$ can be removed from the system once it is trained. This is particularly helpful for two reasons: a) it allows to better *estimate* of the 3D positions of keypoints that are occluded or not detected at inference time by the cameras in $\mathbb{C}_i$; and b) it improves inference time accuracy when $\mathbb{C}_t$ can be higher than $\mathbb{C}_i$ (e.g., in mobile robots that need to use a small set of cameras at inference time, but can use additional cameras when compiling the training dataset).

Ideally, camera placement should cover the entire environment. While individual cameras do not need to cover the entire space, their combined fields of view should ensure complete coverage. Once the cameras are placed on site, the first step to set up the system is to calibrate the intrinsic parameters of all the cameras available, as well as their extrinsic parameters with respect to the desired global frame of reference. Using these parameters, the projection matrices of all the cameras ($T^c \ \forall c \in \mathbb{C}_t$) are created. These matrices are used during the training and inference phases of the two proposed neural networks, as described in sections 3.3 and 3.4.

3.2 Skeleton detection

For training purposes, once the system has been calibrated, a dataset specific to the camera configuration used at training time ($\mathbb{C}_t$) needs to be collected. The training works on the assumption that the dataset has been generated with a single person in the environment at a time. This is required to know unequivocally the correspondences among the different views, avoiding this way the process of manually labelling the dataset. Nevertheless, this requirement must only be held for the training data. At inference time, once the model is trained, it is fully applicable in multi-person environments, with a theoretically unbounded number of people.

The detected skeletons are represented as a list of keypoints, defined by their 2D image coordinates along an identifier for each skeleton keypoint and a certainty value for the detection. Datasets contain sequences of samples, each consisting of a list of the detected skeletons per camera.

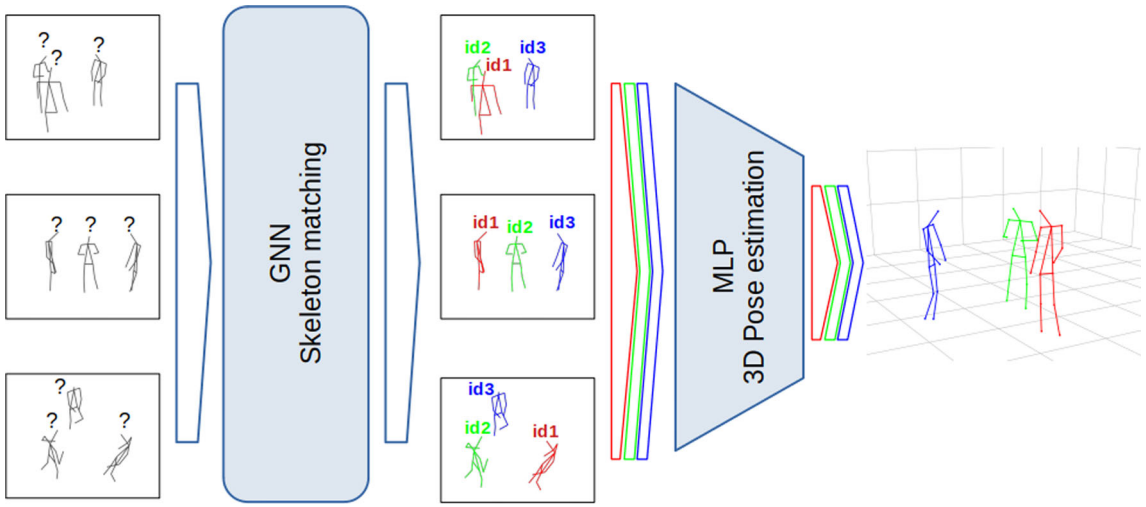


Fig. 1 Two last stages of the pipeline of the proposed system. The correspondences between the input skeletons in the different views are estimated by the GNN. This information is leveraged by the MLP to provide the final 3D poses

As aforementioned, our proposal can be used with any skeleton detector, regardless of the number of keypoints they provide for each skeleton. The number of keypoints only determines the size of the input features of each network, so it is a mere configuration parameter.

3.3 Skeleton matching

Once the skeletons are detected in the different views, the skeletons belonging to the same person are matched. Since the matching is expected to be order-invariant and the number of people is unknown at inference time, we train a GNN model to estimate the correspondence between all the views. This is because GNNs are order-invariant and allow for a variable number of input nodes.

One of the main limitations of learning-based approaches using supervised learning is the generation of the dataset, which needs to be annotated with ground truth. For this particular problem, if the dataset contained more than one skeleton at a time, it would be necessary to manually annotate their cross-view correspondence. To avoid this arduous process, our raw training dataset contains a set of sequences of single individuals moving around the environment -one at a time. These sequences can then be combined into a single processed dataset that contains ground truth labels built by aggregating the data of multiple individuals. We describe this process below.

Our GNN model receives as input an undirected graph $G = (V, E)$, where V is the set of nodes, and E is the set of edges. The set V is composed of two different types of nodes that we term *detection* nodes and *match* nodes. These two types of nodes are the elements of V_d and V_m respectively, such that $V = V_d \cup V_m$. Each *detection* node represents a 2D skeleton detected in one of the views used at inference time (i.e. a view $c \in \mathbb{C}_i$), while a *match* node represents a possible match between two different detections. It is worth noting that only the cameras in $\mathbb{C}_i$ are used because we are only interested in these cameras at inference time and including the rest of the cameras in $\mathbb{C}_t$ would not add any valuable information to this end. For each pair of detection nodes $v_i, v_j \in V_d$ such that v_i and v_j belong to different views, there is a match node $v_k \in V_m$. The edges in E connect the match nodes to their corresponding detection nodes. Thus, for each match node $v_k \in V_m$ linking two detection nodes $v_i, v_j \in V_d$, there are two edges (v_k, v_i) and (v_k, v_j). Therefore, the input graph G can be represented as follows:

$$G = (V_d \cup V_m, (v_k, v_i) \cup (v_k, v_j)) \tag{1}$$

where $v_i, v_j \in V_d$ and $v_k \in V_m$ corresponds to the match node between detection nodes v_i and v_j.

Each node (*detection* or *match*) is represented with a feature vector (x) with $N_k \times N_c \times 10 + 2$ elements, where N_k and N_c are the number of keypoints and cameras, respectively. Two of these elements denote a binary 1-hot encoding indicating if the node is a *detection* or a *match*. In the case of a *match* node, all other dimensions are fixed to zero. In the case of a *detection* node there is a 10-dimensional tuple for each camera-keypoint combination, each of which consists of:

- a flag indicating if the keypoint has been detected,
- the pixel coordinates if the keypoint is visible (2 zeros otherwise),
- a value within the range [0, 1] indicating the certainty of the detection of the keypoint (zero if the keypoint is not visible),

- six elements encoding the 3D line passing through the origin of the camera and the keypoint (image plane coordinates) in the global frame of reference (specified as a 3D point and a 3D direction vector).

Since there is a *detection* node per view of a person, occlusions in one view only affect the features of the corresponding node. Thus, each node contains the features of all visible keypoints of a skeleton from the associated view, regardless of the detections in other views.

Given the input graph $G = (V, E)$ and the feature vectors of the set of nodes ($x_i \in \mathbb{R}^d$, for $v_i \in V$), the GNN is trained to produce an output graph $G' = (V, E)$ with the same structure as G, but different feature vectors for each node ($y_i \in \mathbb{R}^{d'}$). In particular, it is trained to predict whether each *match* node $v_k \in V_m$ corresponds to a true match. Thus, we formulate the matching as a binary classification task, where the target labels are {0, 1} for non-matches and matches respectively. To ensure that each output of the GNN is within the range of [0, 1], we use the Sigmoid activation function in the output layer. Consequently, both the binary cross-entropy (BCE) loss and the mean squared error (MSE) loss are suitable for computing the loss during the training of the GNN. In our experimental results, the GNN trained with MSE loss yields slightly better performance than the GNN trained with BCE loss. Therefore, we define the GNN loss in terms of MSE loss:

$$\mathcal{L}_{SM} = \frac{1}{|V_m|} \sum_{v_k \in V_m} (y_k - \hat{y}_k)^2 \tag{2}$$

being $y_k \in \{0, 1\}$ the target label for node $v_k \in V_m$, and $\hat{y}_k \in [0, 1]$ the predicted probability that node v_k corresponds to a match.

As mentioned previously, to avoid manual labelling, we use footage of single individuals walking and moving through our system. Since each frame contains only one person, we can readily identify matching 2D detections. Using this data, we generate separate graphs for each person, where all *match* nodes are assigned a maximum score value (see Fig. 2a and 2b). We then combine the graphs of individual persons by adding *match* nodes with a score of 0 connecting pairs of detections of different persons, as depicted in figure 2c. By following this procedure, we generate the target label y_k of each *match* node $v_k \in V_m$ of the graphs composing the training set, allowing us to train the GNN in a pseudo-supervised manner. The number of individual graphs to be combined is randomly determined for each sample in the dataset, with a minimum of one and a maximum equal to the total number of sequences used to generate the dataset.

3.4 3D Pose estimation

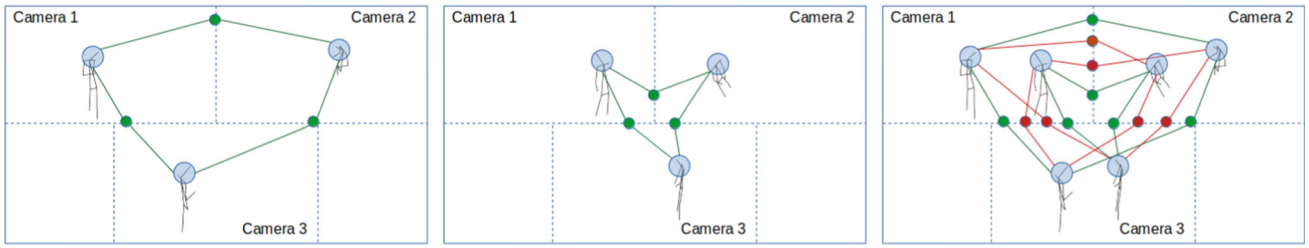
Having identified the different views of each person, an MLP is used to estimate the 3D coordinates of the keypoints for each of them. The input features of the model are the concatenation of 14 features per keypoint and camera. Therefore, if the skeleton detector detects up to 25 keypoints and the system uses 3 cameras at inference time ($|\mathbb{C}_i| = 3$), the dimension of the input feature vector would be $14 \times 3 \times 25$, 1050 dimensions in total. The 14 features per keypoint correspond to the 10 features described in the previous section for skeleton matching plus four additional features related to an initial estimation of the 3D. Specifically, if a keypoint of a person is detected by 2 or more cameras, its 3D coordinates are reconstructed by triangulation for every pair of cameras and an initial estimation is computed as the centroid of the obtained 3D points. This estimation is included as input using three of the four new features. The last feature is used to indicate the availability of the estimated 3D. It is set to 1 if there is more than one view of the keypoint and to 0 otherwise.

Using the aforementioned information per keypoint and camera, the network estimates the 3D positions of the keypoints in the global frame of reference, yielding x, y and z for each of them. Thus, assuming that the network predicts the position of 25 different keypoints, the output vector dimension is $3 \times 25 = 75$.

The training process, as explained in the introduction, follows a self-supervised learning approach, which represents the main advantage of this approach. This way, there is no need to use a ground truth to compare the output, since the loss function only uses the data from the skeleton detectors. However, calculating this loss is not trivial, since the network infers 3D poses from 2D positions. Our approach to solving this problem is to project the 3D coordinates of the keypoints predicted by the network into each camera used for training ($\mathbb{C}_t$). The transformation between global and image coordinates is done by using the projection matrices ($T^c \ \forall c \in \mathbb{C}_t$) obtained during the calibration process. Using the projected coordinates and the coordinates yielded by the skeleton detector, a measurement of the estimation error of the network is obtained. This error defines the loss function that the network is trained to minimise. More formally, assuming that the output of the network o is represented as a vector of 3D positions corresponding to the estimation of the subject's keypoints coordinates:

$$o := \left(o_0, o_1, ..., o_{N_k - 1}\right) \tag{3}$$

with N_k the number of keypoints, a vector p^c of image projected positions (p_i^c) can be obtained for each camera as



(a) Graph for subject 1. (b) Graph for subject 2. (c) Graph for subject 1 and 2.

Fig. 2 Generation of a sample of the dataset. Graphs of individual persons are generated first assigning a score of 1 to the *match* nodes connecting the views (green nodes). Then a final graph is generated from the individual ones adding *match* nodes with a score of 0 (red nodes)

follows[2]:

$$p_i^c = T^c \cdot o_i \quad \forall i \in [0, N_k) \tag{4}$$

Using p^c and the set of detected keypoints ($S^c = \{s_k^c\}$) for each camera c, the projection error e is computed as

$$e = \sum_{c \in \mathbb{C}_t} \sum_{s_k^c \in S^c} d(p_k^c, s_k^c) \tag{5}$$

being $d(\cdot)$ the Manhattan distance between the projected and detected points.

Applying equation 5 to each sample of the dataset D, the final loss is calculated using the mean squared error:

$$\mathcal{L}_{3D} = \frac{1}{|D|} \sum_{d \in D} e_d^2 \tag{6}$$

being e_d the result of equation 5 for the sample d.

Figure 3 depicts the process to compute the self-supervised loss, assuming 25 keypoints, 4 cameras in $\mathbb{C}_t$, and 3 cameras in $\mathbb{C}_i$. It can be observed that the loss computation utilises detections from all cameras in $\mathbb{C}_t$ but only the cameras in $\mathbb{C}_i$ are used for generating the network's input. Consequently, in the aforementioned example, the model receives detections from only three of the four cameras at inference time. However, even though the fourth camera would not be used at inference time, our HPE would still exploit what was learned from it at training time.

3.5 Data augmentation

Data augmentation is applied to extend the data used for training, to increase the variety of situations, and to increase the robustness against partial views. Specifically, for each original sample of the dataset, which we refer to as *seed samples*, several samples are generated by removing views. Given a sample s comprising data obtained from n different views ($s = d_1, d_2, ..., d_n$), m new samples can be generated by selecting subsets of the views. The subsets are chosen randomly from all possible combinations that can be obtained with the different number of views (from 1 to n). For example, suppose a seed sample s contains data from 5 views ($s = \{d_1, d_2, d_3, d_4, d_5\}$), and we want to generate 3 new samples from s, the following samples could be randomly selected from the possible view combinations and added to the dataset: $\{d_1, d_3, d_4\}$, $\{d_2, d_5\}$, $\{d_3\}$. The new data generated by this process are used as the input of the two networks. However, in the case of the pose estimation network, for each generated sample, the whole data of its seed sample is used in the computation of the loss (equation 6), as losing self-supervision information would not be beneficial.

4 Experimental results

Our 3D multi-human pose estimation system has been tested using the CMU Panoptic Studio dataset [24] and a dataset generated at Aston University's Autonomous Robotics and Perception Laboratory for the purpose of this work. The experiments that are presented in this section provide empirical evidence that our approach performs faster than other state-of-the-art algorithms, with comparable accuracy, and most importantly, does not require annotated datasets. Finally, we provide evidence that the proposed HPE can successfully work on an autonomous robot.

4.1 Architecture details

To perform these experiments, the two neural models were trained for each dataset using a train/validation/test split to prevent data leakage. For the matching model, we use a Graph Attention Network (GAT) [48] with 4 hidden layers. The hidden layers are composed of [40, 40, 40, 30] hidden units and [10, 10, 8, 5] attention heads. LeakyReLU and Sigmoid are used as activation functions of the hidden and output layers, respectively. The MLP-based pose estimator has 7 hidden

[2] The conversions between homogeneous and standard coordinates are omitted for simplification.

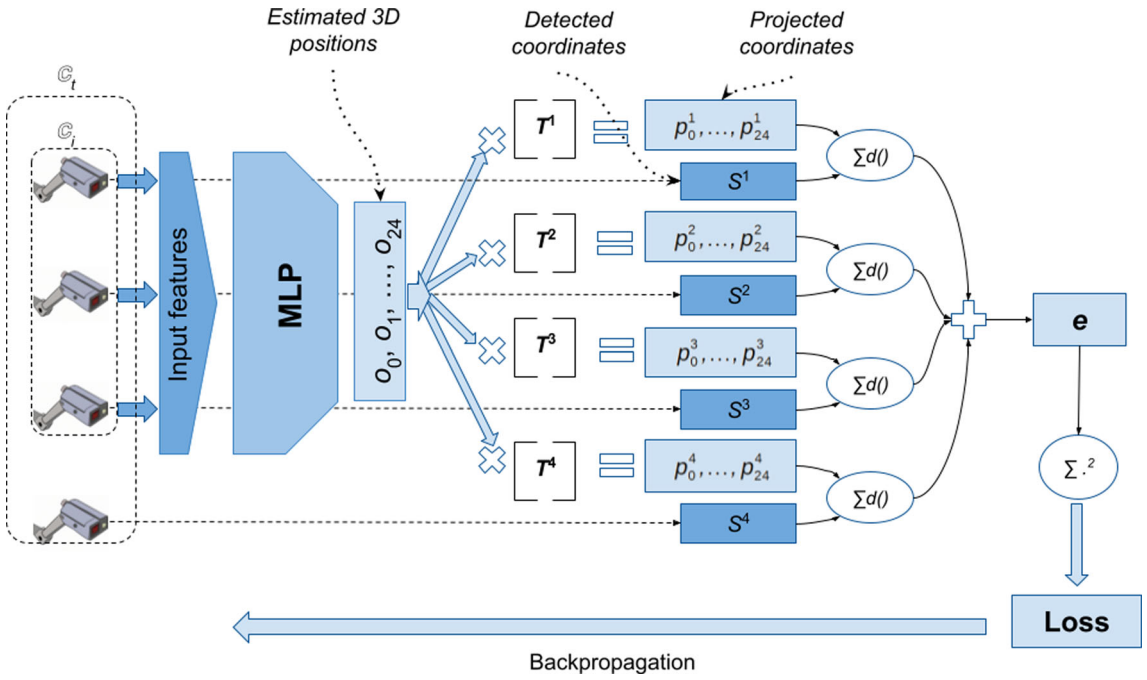


Fig. 3 Representation of how the 3D pose estimation network training loss is computed in a setup with 4 cameras in $\mathbb{C}_t$ and 3 cameras in $\mathbb{C}_i$

layers of [3072, 3072, 2048, 2048, 1024, 1024, 1024] hidden units, using LeakyReLU for the activation of the hidden layers and linear activation in the output layer.

4.2 Datasets

To test our proposal on the **CMU Panoptic dataset** and facilitate comparisons, we use the same sequences and views as VoxelPose [47]. Similarly, the four sequences used for testing VoxelPose were applied in our experiments. The data with the 2D skeletons' information were obtained using the backbone model provided in the VoxelPose project. Using this model, the 2D coordinates of the humans' keypoints were detected from the images. Nevertheless, our training strategy requires that each sample of the data includes the information of only one person, so it was necessary to organise the detection results to provide individual human data. To this end, the skeletons of the different views belonging to the same human were identified using the ground truth of the Panoptic sequences and grouped to obtain individual samples for each human.

As mentioned in section 3, our proposal assumes a fixed configuration of cameras for both the training and inference phases. All the cameras have to be calibrated according to a global frame of reference as a previous step of the training. Nevertheless, this requirement is not strictly met in the Panoptic dataset, which entails some limitations in the comparison with the ground truth. To illustrate the problem, table 2 shows the translation vector to the global frame of reference of the five selected cameras for two of the datasets. As can be observed, there are significant variations between the positions of the cameras in the two datasets, exceeding, in some cases, and for specific axes, 0.1m. This implies that each sequence considers a different global frame of reference. To overcome this limitation, we use the calibration file of one of the datasets (160224_haggling1) for training and, for testing, the ground truth of each test dataset is transformed from the global reference frame of that dataset to the global reference frame used for training. For applying that transformation, a specific camera is used as a common frame of reference for all the datasets. Thus, the ground truth is first transformed from the global frame of reference of the dataset to the camera frame of reference and then from the camera frame of reference to the global frame of reference used for training. Although this transformation partly solves the problem of having different calibration data for each dataset, a residual error remains due to small variations of the intrinsic and inter-camera extrinsic parameters in the Panoptic sequences. Even though this fact is detrimental to our approach in the comparison, the results are still comparable.

The **ARP Laboratory dataset** was generated from 4 cameras attached to the walls of the laboratory and 2 additional cameras mounted on a robot. These two cameras are part of a stereo system, implying they have a fixed relative position and orientation with respect to each other. The robot was static and located at a fixed position during the generation of the dataset. All the cameras were calibrated in relation to a global frame of reference. A total of 18 video sequences

Table 2 Translation (in millimeters) between each camera and the global frame of reference for two sequences of the CMU Panoptic dataset

	Sequence					
	160224_haggling1			160422_haggling1		
Camera	*X*	*Y*	*Z*	*X*	*Y*	*Z*
HD03	2087.71	−1510.89	1780.99	2015.19	−1512.49	1789.8
HD06	−677.9	−3394.66	−1704.22	−641.81	−3398.11	−1783.39
HD12	−76.23	−2392.45	2552.27	−173.19	−2395.36	2494.95
HD13	−1840.95	−3393.29	143.76	−1860.28	−3393.9	28.87
HD23	2343.16	−1526.22	−1433.97	2372.17	−1527.83	−1417.28

of single individuals moving were recorded. The sequences have variable lengths between 2′ and 39′. These sequences were used for training and testing separately the two models. Two additional sequences with groups of 2 and 4 people were recorded to test the whole system. These test sequences have a length of 3, 43′ and 2, 58′, respectively.

4.3 Evaluation of the skeleton-matching module

Since the goal of the skeleton-matching network is to group together the different views of a person, given an unknown number of people, it can be considered a clustering model. Thus, the proposed matching technique can be evaluated through a set of clustering metrics. Specifically, the following metrics have been used:

- **Adjusted rand index** (ARI) [21]: estimates the similarity between two clusterings according to the number of pairs belonging to the same or different clusters. It is adjusted using a random model as a baseline, ensuring a random clustering has a value close to 0. This score ranges between −0.5 (discordant clustering) and 1.0 (perfect clustering).
- **Homogeneity** (H) [40]: measures the homogeneity of the clusters. A cluster is considered homogeneous if it contains only members of the same class. It ranges between 0.0 and 1.0.
- **Completeness** (C) [40]: measures the completeness of the clusters. A cluster is considered complete if all the members of the same class are assigned to the same cluster. It ranges between 0.0 and 1.0.
- **V measure** (Vm) [40]: harmonic mean between homogeneity and completeness. This index quantifies the goodness of the clustering, considering both homogeneity and completeness. It ranges between 0.0 and 1.0.

These metrics have been applied to several skeleton matching networks trained for different numbers of views using the two datasets described in the previous section. Table 3 shows the results for two, three, and five views using the four test sequences of the CMU Panoptic dataset. For all the metrics, values close to 1 are obtained regardless of the number of views.

Table 3 Metrics of the skeleton matching network for the CMU Panoptic dataset

No. of views	ARI	H	C	Vm
2	0.9875	0.9968	0.9925	0.9943
3	0.9977	0.9993	0.9981	0.9986
5	0.9941	0.9978	0.9937	0.9956

Table 4 Metrics of the skeleton matching network for the ARP Laboratory dataset

No. of views	ARI	H	C	Vm
2	0.9770	0.9966	0.9886	0.9923
6	0.9842	0.9974	0.9847	0.9905

The effectiveness of the proposed skeleton-matching network was also evaluated using the ARP Laboratory dataset. Two models, one with two views and the other with six views, were trained using ten of the eighteen sequences of single individuals. The models were then tested on the remaining eight sequences, with a test dataset generated according to the multi-person dataset generation process detailed in section 3.3. This process provided the necessary ground truth to compute the evaluation metrics.

Table 4 presents the results obtained from 2000 samples in the generated dataset, which contained varying numbers of persons ranging from 1 to 8. Similar to the CMU Panoptic dataset, the evaluation metrics demonstrated outstanding performance of the network for both the two and six views models. Furthermore, it is noteworthy that for both datasets, the homogeneity values are nearly 1, indicating that the skeleton groups are predominantly comprised of views from the same individual.

4.4 Evaluation of the multi-person 3D pose estimation system

The whole multi-person 3D pose estimation system has been evaluated for both, the CMU Panoptic and the ARP datasets. This section presents the results of this evaluation.

4.4.1 Evaluation on the CMU Panoptic dataset

The evaluation of the proposed 3D pose estimation system using CMU Panoptic has been carried out using the following metrics:

- **Mean per joint position error** (MPJPE): mean distance (mm) per keypoint between detected and ground truth poses.
- **Mean average precision** (mAP): mean of average precision over different distance thresholds (from 25mm to 150mm, taking steps of 25mm).
- **Mean recall** (mR): mean of recall over all the thresholds.
- **Time for persons' proposals** (t_{pp}): mean time required for generating persons' proposals. In our approach, this time corresponds to the skeleton matching stage.
- **Time for 3D pose estimation** (t_{3Dg}): mean time required for estimating the 3D poses.
- **Time for 3D pose estimation per human** (t_{3Di}): mean time required for estimating the 3D pose of one person.

To provide a comparison with other existing approaches, VoxelPose was trained using the same ten training Panoptic sequences. In addition, the results of our pose estimation model were compared with the 3D poses obtained by triangulation. Specifically, for each pair of views of a person identified by the skeleton matching model, the 3D position of each visible keypoint was estimated by triangulating the 3D of its 2D coordinates. If more than one estimation was obtained (i.e., the keypoint is visible from more than 2 cameras), the final 3D position for the keypoint was computed as the average of the individual estimations.

Regarding our proposal, we have used 3 different versions of the test dataset. In the first version (*D-detected*), the skeletons used as input contain the 2D coordinates of the keypoints detected from the backbone skeleton detector of VoxelPose. In the second version (*D-projected*), the 2D positions of the detected keypoints have been replaced with the projected coordinates of the ground truth 3D keypoints using the specific calibration data of each sequence. Finally, in the third version (*D-average*), the 2D positions of the keypoints have been computed as the average between the detected and projected 2D coordinates.

The main reason for using these 3 versions is the existing significant difference between the detected 2D keypoints and the ones obtained by projecting the ground truth 3D skeletons. This fact can be observed in table 5, where the reprojection error of the ground truth for the three datasets is shown. To create this table, the ground truth 3D, previously transformed into the training frame of reference as explained in Sect. 4.2, was projected into the images of the five cameras. This projection considered the calibration data employed during the model's training. Subsequently, the projected data was compared with the 2D keypoints of the three datasets to obtain the reprojection errors.

Table 5 Reprojection error of the ground truth 3D for the three versions of the test dataset using CMU Panoptic

Dataset	Camera				
	HD03	HD06	HD12	HD13	HD23
D-detected	7.01	10.73	7.63	10.71	6.37
D-projected	3.81	1.08	2.19	2.28	4.74
D-average	3.65	5.12	3.92	6.06	3.69

As can be observed, there are significant differences among the reprojection errors considering the three datasets. Thus, as expected, the largest reprojection error occurs on the *D-detected* dataset, which indicates some divergences in the positions of the keypoints of the human body between the skeleton detector model and the ground truth of the Panoptic datasets. In addition, certain differences are observed between the reprojected ground truth and the positions of the keypoints of the *D-projected* dataset. These differences are related to the different calibration data of each sequence of Panoptic. Specifically, as mentioned in section 4.2, the variations of the intrinsic and inter-camera extrinsic parameters of the sequences produce a remaining error that is reflected in the second row of table 5. In fact, compared with the *D-avarage* dataset, cameras *HD03* and *HD23* present higher reprojection errors for the *D-projected* dataset.

Table 6 shows a summary of the accuracy and time metrics obtained for VoxelPose, triangulation, and our proposed method across the four test sequences of the CMU Panoptic dataset. The last three rows of the table correspond to our model's performance on the three dataset variations (*D-detected*, *D-projected*, and *D-average*).

Regarding accuracy, VoxelPose and our model for the *D-projected* dataset have similar performance, even though our model was trained with detected data. Additionally, our model's results on the *D-average* dataset for *MPJPE*, *mAP*, and *mR* are quite comparable to those of VoxelPose. The highest value of the *MPJPE* is produced for our model with the *D-detected* variation of the test dataset. As mentioned earlier, this is due to the divergences between 2D detected and ground truth projected coordinates. Nevertheless, triangulation yields a similar mean position error despite its computation only considering the keypoints for which triangulation can be applied, that is, the keypoints visible from

Table 6 Accuracy and time metrics of VoxelPose, triangulation, and our proposal using the CMU Panoptic dataset

Method	MPJPE	mAP	mR	t_{pp}	t_{3Dg}	t_{3Di}
VoxelPose	17.97	96.61	97.41	135.92	169.99	50.53
Triangulation	22.63	76.99	85.10	32.56	10.06	2.99
Ours-detected	26.06	89.25	92.63	31.67	19.65	5.83
Ours-projected	17.84	96.23	97.76	31.96	19.94	5.89
Ours-average	19.77	95.67	97.39	32.22	19.81	5.85

two or more cameras. Furthermore, triangulation performs the worst in terms of *mAP* and *mAR*. This is due to the fact that triangulation does not always yield complete pose estimations. Figures 4 and 5 provide examples of complete and incomplete results using triangulation. Figure 4 showcases some samples where all the keypoints for every person in the scene can be estimated by triangulation. In these cases, the estimated poses provided by our model (images on the left) and triangulation (images on the right) are very close to the ground truth poses (shown in gray). However, in Fig. 5, some poses cannot be entirely determined by triangulation, as there are keypoints that are not visible from two or more cameras. In such scenarios, our model provides complete estimates for all poses, with minimal differences from the ground truth. Interestingly, the second scenario of Fig. 5 presents a situation where the ground truth is incomplete (green skeleton). It can be seen how our model provides a realistic pose despite the lack of information.

In terms of computational time, VoxelPose takes an average of 305.91 ms for the whole estimation process, which is almost 6 times longer than the time required by our proposal. The metrics of table 6 do not include the time required for skeleton detection, which may vary depending on the specific detector. However, efficient solutions for detection do exist, such as *trt-pose* which can perform at 251 FPS on Jetson Xavier [51]. Furthermore, skeleton detection for all the views can be run in parallel, making the timing roughly independent of the number of views. Thus, assuming skeleton detection can be achieved at 30 FPS, our proposal still runs more than 3 times faster than VoxelPose.

Besides the aforementioned benefits regarding real-time execution in comparison with VoxelPose, our self-supervised proposal can be more easily implemented in new environments than the existing alternatives. The fact that **no ground truth is required** to train the two models makes our proposal easily replicable, regardless of the space, organization and extension.

To complete the evaluation of our proposal on the CMU Panoptic dataset, we present an experiment that aims to assess the impact of the number of views. Specifically, we trained two new models using different combinations of inference and training views: one model was trained using 3 cameras for both inference and training and the other one used 3 cameras for inference and the whole set of cameras (5) for training.

Table 7 Accuracy results for different combinations of inference and training views using the CMU Panoptic dataset

$\#\mathbb{C}_i$	$\#\mathbb{C}_t$	AP_{25}	AP_{50}	AP_{100}	AP_{150}	MPJPE
5	5	87.25	95.51	98.75	99.50	17.84
3	3	29.77	81.92	94.24	98.48	32.86
3	5	46.48	90.16	97.36	98.76	26.25

Table 7 shows the results of these experiments focusing on the average precision (AP) measured at thresholds of $25\,mm$, $50\,mm$, $100\,mm$, and $150\,mm$, along with the MPJPE. Results of the model using the whole set of views for training and inference are included for comparison purposes.

The results indicate that reducing the number of views leads to a decrease in the model's accuracy. However, even with fewer views, the models still produce reasonably good results, particularly when trained with a larger number of views. Such a model provides comparable results to those reported in [47] and [55], improving all the metrics in comparison with the model trained with 3 cameras.

The next section extends this evaluation of the influence of using different combinations of inference and training views on our model's performance for the ARP Laboratory dataset.

4.4.2 Evaluation on the ARP Laboratory dataset

The proposed multi-person 3D pose estimation system has also been evaluated using the ARP Laboratory dataset. Since this dataset does not include a ground truth, the accuracy metrics employed for the CMU Panoptic dataset can not be applied. Instead, we use the reprojection error of the estimated 3D for all the cameras. We trained four different models using different sets of cameras at training and inference times: $M_{6/6}$ which uses the six cameras for training and inference; $M_{2/6}$, which uses the six cameras for training and the two ones mounted on the robot for inference; $M_{1/6}$, that was trained with the six cameras but only uses one of the cameras of the robot at inference time; and $M_{2/2}$ that uses only the two cameras of the robot for training and inference.

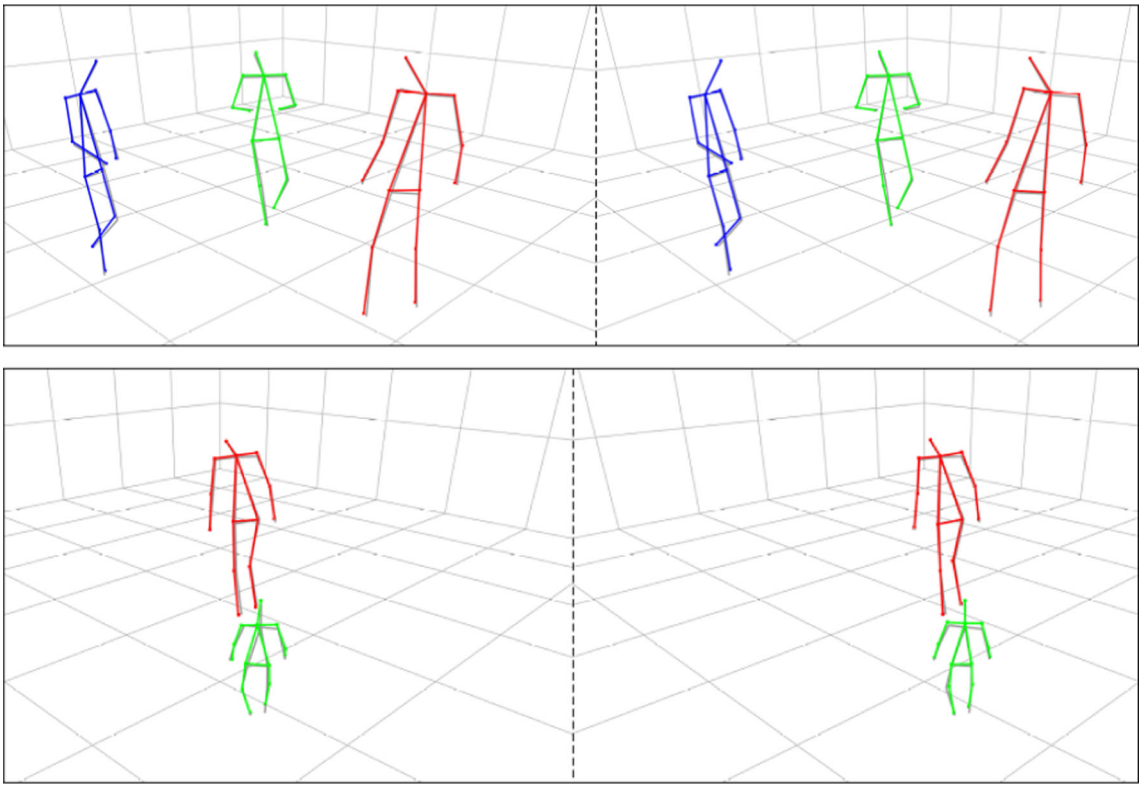


Fig. 4 Pose estimation results for 2 samples of the test sequences using our model (left images) and triangulation (right images). The ground truth is shown in gray. Triangulation provides complete poses in the 2 samples

Mean and median reprojection error for the four models and the six cameras (wall cameras: W0, W1, W2, and W3; robot cameras: R0 and R1) using the two ARP Laboratory sequences with 2 and 4 people are depicted in table 8. The lowest reprojection error for the wall cameras is given by the model $M_{6/6}$. This model is the most reliable one since it uses data from all the views. Despite models $M_{2/6}$ and $M_{2/2}$ using the same cameras at inference time, there are very significant differences in their behavior, which is reflected in the reprojection errors of the cameras of the walls. In particular, it can be observed a very high error of $M_{2/2}$ for the camera $W1$. Such a large error is produced when there is limited visibility of a person from the cameras used by the model. This is the case with the second sample of Fig. 6, where the model places the red skeleton far away from its actual position. Besides this specific case, in general, the keyponts' positions estimated by model $M_{2/2}$ differ from the estimates of model $M_{6/6}$ as observed in that figure. In contrast, the model $M_{2/6}$ can estimate the pose of the person correctly, even though the input of both models is common. Generally, the poses provided by the model $M_{2/6}$ are very similar to those provided by the model $M_{6/6}$, as demonstrated by both Fig. 6 and the reprojection errors in table 8. Finally, the model $M_{1/6}$ shows outstanding results considering it only receives information from one of the cameras of the robot (*R0*). The model is capable of predicting complete 3D poses with similar accuracy to model $M_{6/6}$, producing comparable reprojection errors to model $M_{2/6}$[3].

The results of this experiment, along with the ones presented in the previous section, demonstrate that our system is capable of providing good estimations with a reduced number of cameras, by simply considering the information of an extended set of cameras during training. This is a significant advantage for its application in autonomous robots, which is the focus of the next section.

4.5 Evaluation in a mobile robot

This experiment aims to show the application of the proposed system in a mobile robot equipped with only two RGB cameras. The main goal is to endow the robot with the ability to estimate the complete 3D human poses with enough accuracy to enhance the interaction between them.

Since the robot does not stay in a fixed location, the only visual information it can use is that provided by its two cameras. In the case of a mobile robot, triangulation is less

[3] Bear in mind that, even though the reprojection errors are lower for $M_{1/6}$ than for $M_{2/6}$ in four of the six cameras, non-visible people from camera *R0* (see the third sample of Fig. 6) are not considered in the error computation of model $M_{1/6}$.

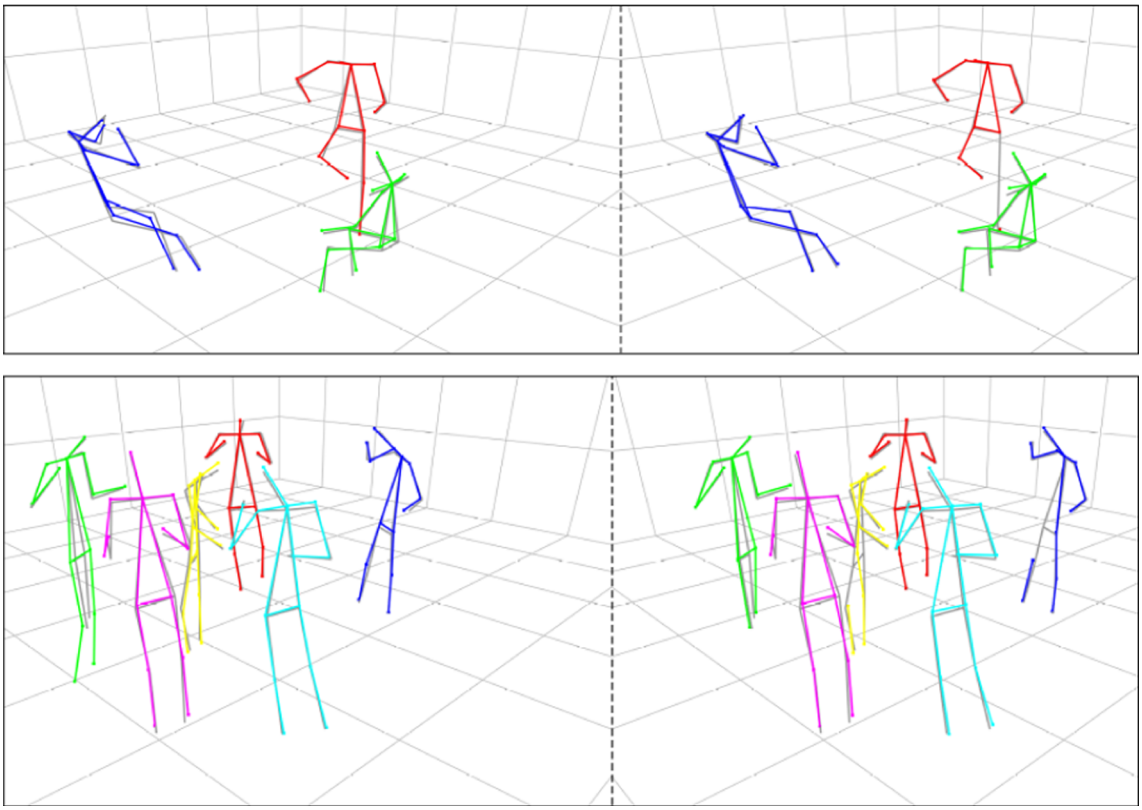


Fig. 5 Pose estimation results for 2 samples of the test sequences using our model (left image) and triangulation (right image). The ground truth is shown in gray. In these samples, triangulation cannot provide complete poses due to an insufficient number of views for some keypoints

Table 8 Mean and median reprojection error in the 6 cameras of the ARP Laboratory for 4 models trained with different numbers of train and inference cameras

Camera	Model			
	$M_{1/6}$	$M_{2/6}$	$M_{6/6}$	$M_{2/2}$
W0	14.65 / 11.26	14.35 / 11.00	10.28 / 8.23	45.73 / 26.85
W1	11.84 / 9.28	12.02 / 9.49	8.28 / 6.80	290.63 / 15.62
W2	14.02 / 10.68	14.04 / 10.55	11.12 / 8.41	29.78 / 21.86
W3	12.18 / 9.29	12.36 / 9.40	7.88 / 6.40	31.94 / 19.24
R0	6.69 / 5.27	7.06 / 5.34	8.98 / 6.97	4.50 / 3.37
R1	9.05 / 6.83	7.79 / 6.07	9.49 / 7.51	4.50 / 3.38

informative. The short baselines of robots' stereo systems rarely provide complete poses, and more importantly, they can cause small deviations in keypoints' image positions to produce large 3D errors. Nevertheless, using only the data captured by the two cameras to train the pose estimation model does not provide reliable results, as shown in the previous section. For this reason, we use the model $M_{2/6}$, which only requires the information of the two cameras on the robot at inference time, but uses the data from the four additional wall cameras during training.

We have conducted two different experiments to validate the effectiveness of the proposal. In the first experiment, the robot remains static facing the location of a person at 2.75 meters from the front part of the robot. The 3D pose of the person is recorded as they walk towards the robot in a straight line, covering a distance of approximately 1.5 meters. Figure 7a illustrates the displacement of the person's two ankles (magenta lines), along with the position of the robot (dark blue cross). The green cross in the map represents the initial position of the person and the red one the position of the global frame of reference. Figure 7b displays the displacement in meters of the coordinates (projected on the floor plane) of some representative keypoints from the initial position. As can be seen, the traveled distance for all the keypoints goes from nearly 0 to roughly 1.5 meters. In addition, the distance between symmetric keypoints presents only small variations along the entire route (e.g. the standard

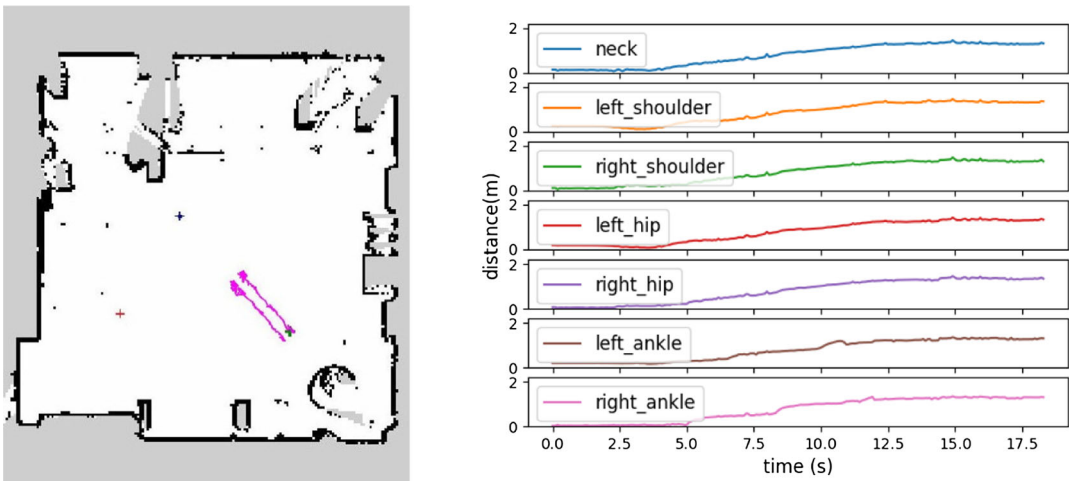
Fig. 6 Pose estimation results from our proposal of four samples of the ARP Laboratory multi-person sequences. From left to right, the results correspond to the models $M_{1/6}$, $M_{2/6}$, $M_{6/6}$, and $M_{2/2}$

deviation is 2.3cm for the hips and 0.64cm for the eyes), which is indicative of the stability of the estimations.

The second experiment employs the same setup, with the person remaining stationary while the robot approaches them following a straight line covering a distance of 1 meter. Figure 8a shows the robot's path (thick blue line), extracted from its localisation system, and the position of the person's ankles, with the initial point situated between them. Figure 8b displays the graphs depicting the distances from the initial position of the person, at every instant, of the projection onto the floor of the same representative keypoints as in the previous experiment. As shown in this figure, the distances remain almost constant, which is the expected result. As in the previous experiment, the variations in the distances between symmetric keypoints are insignificant, with values from 1.8cm for the ankles to 0.8cm for the hips, which demonstrate the robustness and good accuracy of the model.

5 Conclusions

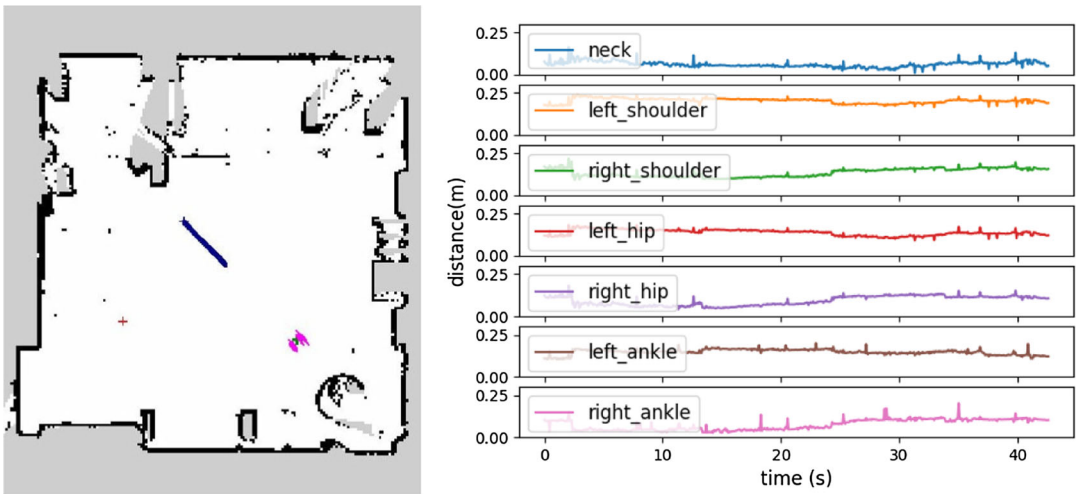
Multi-person 3D pose estimation is an important research field with multiple applications. Deep learning is a powerful tool to learn human physiological priors. Nevertheless, conventional deep learning solutions require large amounts of labelled data. We propose a GNN to identify the views of the different people in the scene and an MLP to estimate the complete 3D pose of each person. Both networks are trained



(a) Trajectory of the ankles of the subject.

(b) Distance from the initial position of the person to a subset of keypoints.

Fig. 7 Experiment maintaining the robot in a fixed position while a person walks 1.5 meters towards the robot. Only two RGB cameras are used at inference time



(a) Trajectory of the robot.

(b) Distance from the initial position of the person to a subset of keypoints.

Fig. 8 Experiment where the person remains still while the robot moves 1 meter towards the person. Only two RGB cameras are used at test time

using completely unannotated data. The unique requirement for the training of each network is that each element in the dataset corresponds to an individual person. Besides, both networks use information that can be directly obtained from the RGB images, so our approach only requires regular RGB cameras.

Experimental results over our skeleton matching model for the CMU Panoptic and the ARP Laboratory datasets show outstanding model performance, which, as indicated in section 4.3, yields nearly perfect values for all the clustering metrics.

Regarding the accuracy of the 3D pose estimation model, in comparison to VoxelPose, our proposal shows slightly lower accuracy values over detected coordinates, with a mean per joint precision error of 26.06mm. Nevertheless, it is important to note, as investigated in Sect. 4.4, that there is a significant difference between the detected 2D and the projection of the 3D ground truth in the CMU Panoptic dataset. Using the projected 3D as a test set, without retraining the network, we obtain a mean per joint precision error of 17.84mm (slightly better than VoxelPose error). This disparity suggests that our 3D pose estimator's true accuracy might be higher than the one obtained in the initial comparison. Furthermore, the computational complexity of our system is significantly lower than VoxelPose, making it an effective solution for real-time applications. Despite real-time solutions already

existing (such as [55]), from a qualitative perspective, the fact that our approach does not rely on annotated data constitutes a remarkable benefit over VoxelPose and other multi-person and multi-view 3D pose estimation methods.

The experiments conducted in Sect. 4.5 illustrate the advantages of training our models with a subset of cameras for inference time. This approach enables the use of the system in a mobile robot with only two RGB cameras in real-time applications. As demonstrated in these experiments, the system has sufficient precision to be used in social robotic applications.

In future work, we aim to enhance the accuracy of our estimator model by refining the training with hyperparameter tuning. Additionally, we consider developing models trained with data including intrinsic and extrinsic camera parameters to pave the way to remove the need for scenario-specific training. With this latter project, the model would be robot-agnostic, allowing users to simply mount the cameras on the robot, calibrate them, and run the system without any training.

Supplementary information

The data and models that support the findings of this paper have been made publicly available at https://www.dropbox.com/sh/6cn6ajddrfkb332/AACg_UpK22BlytWrP19w_VaNa?dl=0. The link contains both the preprocessed datasets and pretrained models. The code is available in a public GitHub repository at https://github.com/gnns4hri/3D_multi_pose_estimator. Additionally, the experimental results utilize the CMU Panoptic dataset [24] and a dataset compiled specifically for this research work. We deleted all information that identifies individuals in compliance with the conditions set by the ethics committee of Aston University.

Acknowledgements Experiments were run on Aston EPS Machine Learning Server, funded by the EPSRC Core Equipment Fund, Grant EP/V036106/1. This work was also supported by the Spanish Government under Grants PID2022-137344OB-C31, TED2021-131739B-C22, and PDC2022-133597-C41.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

1. Abdulla, W.: Mask R-CNN for object detection and instance segmentation on Kerasand TensorFlow. https://github.com/matterport/Mask_RCNN, gitHub repository (2017)
2. Aggarwal, J.K., Xia, L.: Human activity recognition from 3d data: a review. Pattern Recogn. Lett. **48**, 70–80 (2014)
3. Amin, S., Andriluka, M., Rohrbach, M., et al.: Multi-view pictorial structures for 3d human pose estimation. In: Bmvc (2013)
4. Bala, P., Zimmermann, J., Park, H., et al.: Self-supervised secondary landmark detection via 3d representation learning. Int. J. Comput. Vision **131**(8), 1980–1994 (2023). https://doi.org/10.1007/s11263-023-01804-y
5. Bartol, K., Bojanić, D., Petković, T., et al.: Generalizable human pose triangulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 11028–11037 (2022)
6. Belagiannis, V., Amin, S., Andriluka, M., et al.: 3d pictorial structures for multiple human pose estimation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp 1669–1676, https://doi.org/10.1109/CVPR.2014.216 (2014)
7. Belagiannis, V., Amin, S., Andriluka, M., et al.: 3D pictorial structures revisited: multiple human pose estimation. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 1929–1942 (2016). https://doi.org/10.1109/TPAMI.2015.2509986
8. Biswas, S., Sinha, S., Gupta, K., et al.: Lifting 2d human pose to 3d: a weakly supervised approach. In: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, pp 1–9 (2019)
9. Bouazizi, A., Wiederer, J., Kressel, U., et al.: Self-supervised 3d human pose estimation with multiple-view geometry. In: 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), pp 1–8, https://doi.org/10.1109/FG52635.2021.9667074 (2021)
10. Bridgeman, L., Volino, M., Guillemaut, J.Y., et al.: Multi-person 3d pose estimation and tracking in sports. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 2487–2496, https://doi.org/10.1109/CVPRW.2019.00304 (2019)
11. Camplani, M., Paiement, A., Mirmehdi, M., et al.: Multiple human tracking in rgb-depth data: a survey. IET Comput. Vision **11**(4), 265–285 (2017). https://doi.org/10.1049/iet-cvi.2016.0178
12. Cao, Z., Hidalgo Martinez G., Simon, T., et al.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
13. Cardinaux, F., Bhowmik, D., Abhayaratne, C., et al.: Video based technology for ambient assisted living: a review of the literature. JAISE **3**, 253–269 (2011). https://doi.org/10.3233/AIS-2011-0110
14. Chen, H., Feng, R., Wu, S., et al.: 2D Human pose estimation: a survey. Multimed. Syst. **29**, 3115–3138 (2023). https://doi.org/10.1007/s00530-022-01019-0
15. Dong, J., Fang, Q., Jiang, W., et al.: Fast and robust multi-person 3d pose estimation and tracking from multiple views. IEEE Trans. Pattern Anal. Mach. Intell. (2021). https://doi.org/10.1109/TPAMI.2021.3098052
16. Drover, D., VR, M., Chen, C.H., et al.: Can 3d pose be learned from 2d projections alone? In: Leal-Taixé, L., Roth, S. (eds.) Computer Vision - ECCV 2018 Workshops, pp. 78–94. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11018-5_7
17. Gerónimo, D., López, A.M., Sappa, A.D., et al.: Survey of pedestrian detection for advanced driver assistance systems. IEEE Trans.

Pattern Anal. Mach. Intell. **32**(7), 1239–1258 (2010). https://doi.org/10.1109/TPAMI.2009.122
18. Gong, X., Song, L., Zheng, M., et al.: Progressive multi-view human mesh recovery with self-supervision. In: 0001 BW, 0001 YC, Neville J (eds) Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023. AAAI Press, pp 676–684, https://ojs.aaai.org/index.php/AAAI/article/view/25144 (2023)
19. Guan, S., Lu, H., Zhu, L., et al.: Posegu: 3d human pose estimation with novel human pose generator and unbiased learning. Comput. Vis. Image Underst. **233**, 103715 (2023). https://doi.org/10.1016/j.cviu.2023.103715
20. Hu, W., Zhang, C., Zhan, F., et al.: Conditional directed graph convolution for 3d human pose estimation. In: Proceedings of the 29th ACM International Conference on Multimedia, pp 602–611 (2021)
21. Hubert, L., Arabie, P.: Comparing partitions. J. classif. **2**(1), 193–218 (1985)
22. Ionescu, C., Papava, D., Olaru, V., et al.: Human3.6m: large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2014)
23. Jain, A., Tompson, J., Andriluka, M., et al.: Learning human pose estimation features with convolutional networks. 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings pp 1–11. arXiv:1312.7302 (2014)
24. Joo, H., Liu, H., Tan, L., et al.: Panoptic studio: A massively multiview system for social motion capture. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. IEEE Computer Society, pp 3334–3342 (2015)
25. Kocabas, M., Karagoz, S., Akbas, E.: 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, https://doi.org/10.1109/CVPR.2019.00117 (2019)
26. Kreiss, S., Bertoni, L., Alahi, A.: Pifpaf: Composite fields for human pose estimation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, Los Alamitos, CA, USA, pp 11969–11978, https://doi.org/10.1109/CVPR.2019.01225 (2019)
27. Kreiss, S., Bertoni, L., Alahi, A.: Openpifpaf: composite fields for semantic keypoint detection and spatio-temporal association. IEEE Trans. Intell. Transp. Syst. (2021). https://doi.org/10.1109/TITS.2021.3124981
28. Kundu, J.N., Seth, S., Jampani, V., et al.: Self-supervised 3d human pose estimation via part guided novel image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6152–6162 (2020)
29. Li, S., Chan, A.B.: 3d human pose estimation from monocular images with deep convolutional neural network. In: Asian Conference on Computer Vision, Springer, pp 332–347 (2014)
30. Lin, J., Lee, G.H.: 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)
31. Lin, T.Y., Maire, M., Belongie, S., et al.: Microsoft coco: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., et al. (eds.) Computer Vision - ECCV 2014, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
32. Liu, S., Shuai, J., Li, Y., et al.: Mmda: Multi-person marginal distribution awareness for monocular 3d pose estimation. IET Image Proc. **17**(7), 2182–2191 (2023). https://doi.org/10.1049/ipr2.12783
33. Mehta, D., Sotnychenko, O., Mueller, F., et al.: Xnect: real-time multi-person 3d motion capture with a single rgb camera. ACM Trans. Graph. (2020). https://doi.org/10.1145/3386569.3392410
34. Moreno-Noguer, F.: 3d human pose estimation from a single image via distance matrix regression. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2823–2832, https://doi.org/10.1109/CVPR.2017.170 (2017)
35. Park, S., You, E., Lee, I., et al.: Towards robust and smooth 3d multi-person pose estimation from monocular videos in the wild. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, Paris, France, p 14726-14736, https://doi.org/10.1109/ICCV51070.2023.01357, https://ieeexplore.ieee.org/document/10377078/ (2023)
36. Pavlakos, G., Zhou, X., Derpanis, K.G., et al.: Coarse-to-fine volumetric prediction for single-image 3d human pose. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, Los Alamitos, CA, USA, pp 1263–1272, https://doi.org/10.1109/CVPR.2017.139 (2017)
37. Rhodin, H., Spörri, J., Katircioglu, I., et al.: Learning monocular 3d human pose estimation from multi-view images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8437–8446, https://doi.org/10.1109/CVPR.2018.00880 (2018)
38. Rodriguez-Criado, D., Bachiller, P., Bustos, P., et al.: Multi-camera torso pose estimation using graph neural networks. In: 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), IEEE, pp 827–832 (2020)
39. Rogez, G., Weinzaepfel, P., Schmid, C.: Lcr-net++: multi-person 2d and 3d pose detection in natural images. IEEE Trans. Pattern Anal. Mach. Intell. **42**(5), 1146–1161 (2019)
40. Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). Association for Computational Linguistics, Prague, Czech Republic, pp 410–420 (2007)
41. Shafiee, N., Padir, T., Elhamifar, E.: Introvert: Human trajectory prediction via conditional 3d attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 16815–16825 (2021)
42. Srivastav, V., Gangi, A., Padoy, N.: Self-supervision on unlabelled or data for multi-person 2d/3d human pose estimation. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23, Springer, pp 761–771 (2020)
43. Sun, K., Xiao, B., Liu, D., et al.: 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, https://doi.org/10.1109/CVPR.2019.00584 (2019)
44. Sun, L., Yan, Z., Mellado, S.M., et al.: 3dof pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 5942–5948 (2018)
45. Taipalus, T., Ahtiainen, J.: Human detection and tracking with knee-high mobile 2d lidar. In: 2011 IEEE International Conference on Robotics and Biomimetics, pp 1672–1677, https://doi.org/10.1109/ROBIO.2011.6181529 (2011)
46. Tompson, J.J., Jain, A., LeCun, Y., et al.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Ghahramani, Z., Welling, M., Cortes, C., et al. (eds.) Advances in Neural Information Processing Systems, vol. 27. Curran Associates Inc (2014)
47. Tu, H., Wang, C., Zeng, W.: Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In: European Conference on Computer Vision, Springer, pp 197–212 (2020)

48. Veliković, P., Cucurull, G., Casanova, A., et al.: Graph attention networks. In: International Conference on Learning Representations (2018)
49. Wang, J., Tan, S., Zhen, X., et al.: Deep 3D human pose estimation: a review. Computer Vision and Image Understanding 210(August 2020):103225. https://doi.org/10.1016/j.cviu.2021.103225 (2021)
50. Wang, X.: Intelligent multi-camera video surveillance: a review. Pattern Recognit. Lett. **34**, 3–19 (2013)
51. Welsh, J.: trt_pose. https://github.com/NVIDIA-AI-IOT/trt_pose, accessed: 2022-06-09 (2012)
52. Wu, S., Jin, S., Liu, W., et al.: 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)
53. Xu, C., Chen, S., Li, M., et al.: Invariant teacher and equivariant student for unsupervised 3d human pose estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 3013–3021 (2021)
54. Yan, Z., Duckett, T., Bellotto, N.: Online learning for human classification in 3d lidar-based tracking. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 864–871, https://doi.org/10.1109/IROS.2017.8202247 (2017)
55. Ye, H., Zhu, W., Wang, C., et al.: Faster voxelpose: real-time 3d human pose estimation by orthographic projection. In: Part, V.I. (ed.) Computer Vision-ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, pp. 142–159. Springer (2022)
56. Zhang, J., Li, W., Ogunbona, P.O., et al.: RGB-D-based action recognition datasets: a survey. Pattern Recogn. **60**, 86–105 (2016). https://doi.org/10.1016/j.patcog.2016.05.019