2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Robot Person Following Under Partial Occlusion

Hanjing Ye[1], Jieting Zhao[1], Yaling Pan[2], Weinan Chen[2], Li He[1] and Hong Zhang[1*]

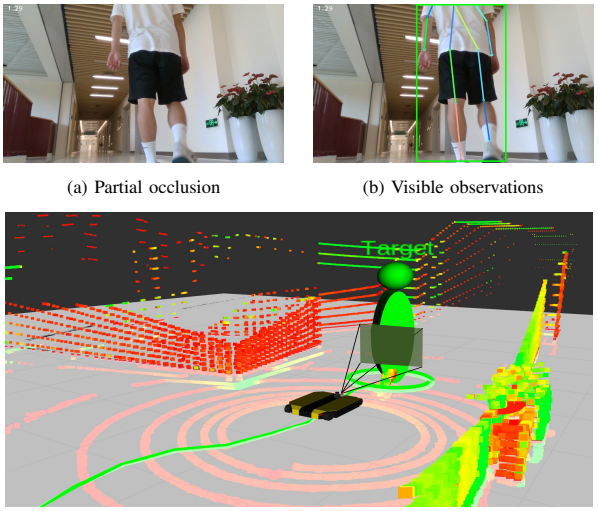
***Abstract*— Robot person following (RPF) is a capability that supports many useful human-robot-interaction (HRI) applications. However, existing solutions to person following often assume full observation of the tracked person. As a consequence, they cannot track the person reliably under partial occlusion where the assumption of full observation is not satisfied. In this paper, we focus on the problem of robot person following under partial occlusion caused by a limited field of view of a monocular camera. Based on the key insight that it is possible to locate the target person when one or more of his/her joints are visible, we propose a method in which each visible joint contributes a location estimate of the followed person. Experiments on a public person-following dataset show that, even under partial occlusion, the proposed method can still locate the person more reliably than the existing SOTA methods. As well, the application of our method is demonstrated in real experiments on a mobile robot.**

I. INTRODUCTION

Robot person following (RPF) [1] is a capability that supports many useful HRI [2] applications. Often, the person being followed can become partially occluded in various situations due to, for example, other objects or people in the robot working environment. Therefore, the ability of following a person under partial occlusion is essential.

RPF can be achieved with a distance measurement sensor such as a LiDAR and or an RGB-D camera. Methods in [3]–[8], for example, firstly track multiple people with the help of a distance measurement sensor. Once a target person is selected and tracked in the field of view of the robot sensor, the person can be followed on the basis of the tracked location. Such solutions, however, can be expensive due to the high cost of a distance measurement sensor. In addition, distance sensors lack textural information, and this prevents them from resolving data association effectively.

Alternatively, one can resort to vision to solve the problem of person following. [9] uses a vision-based single object tracker (SOT) [10]–[12] to track the person in the image space, and relies on visual servo to follow the person. Also using vision, [13] proposes a method under the assumption that the neck of the person is always visible. Conceptually, it can be considered as a *single-joint-based* method. However,



(a) Partial occlusion (b) Visible observations

(c) Target person location and following

Fig. 1. (a) An example of partial occlusion caused by a limited FoV of a camera while performing person following. Existing methods often fail to locate the person in such challenging scenarios. (b) The proposed method is based on partial observations, including the bounding box and 2D joints. (c) Person following under partial occlusion (the pointcloud is used as a reference for visualization only, and the body parts of the detected person (target) are marked in green solids in front of the robot).

with a camera of limited field-of-view (FoV), neck visibility cannot always be guaranteed even though the followed person may still be partially visible, as shown in the example in Fig. 1(a). In addition, many *deep-learning-based* methods have been developed for estimating a person's location through monocular depth prediction [14], monocular 3D bounding-box detection [15], etc. These methods, however, are known to experience poor generalization in terms of adapting to the case of partial observation.

One prior topic of research related to our study is 2D human joint detection [16][17], which has been well developed in recent years. It has been demonstrated that human joints can be detected reliably even under partial observation as depicted in Fig. 1(b). In this work, We exploit these well-detected joints of a partially occluded person in person location estimation. This idea of estimating the 3D pose of an object or a person from partial observation is not new [18]–[20]. To design a solution, one can first build a prior model describing the physical attributes of as well as constraints between the parts of the whole object/person, and this model can be created in the form of an implicit representation by a neural network [18], a parametric model like SMPL (skinned multi-person linear model) [19], or a CAD model [20]. Then given a partial observation, the 3D person/object poses can be inferred from the prior model.

To make use of the above idea in our study of RPF, without

*corresponding author (hzhang@sustech.edu.cn).
[1]Hanjing Ye, Jieting Zhao, Li He and Hong Zhang are with Shenzhen Key Laboratory of Robotics and Computer Vision, Southern University of Science and Technology (SUSTech), and the Department of Electronic and Electrical Engineering, SUSTech.
[2]Yaling Pan and Weinan Chen are with the Biomimetic and Intelligent Robotics Lab, Guangdong University of Technology.
This work was supported by the Pearl River Talent Recruitment Program under Grant No.2019QN01X761 and the National Natural Science Foundation of China (62103179).
Source Code: https://github.com/MedlarTea/Mono-RPF.

loss of generality, we build a prior model consisting of the heights of a person's joints relative to the ground plane. With the assumption of a standing person on the ground plane and a calibrated camera at a known height to the ground, the person's location with respect to the camera can be inferred from any joint detection given the prior model [21]. In fact, each visible joint contributes a location estimate and the concurrent observations of multiple joints can be fully utilized through any one of the many existing sensor fusion techniques such as the Kalman filter. Furthermore, to handle a crowded scene with multiple people, we perform people tracking and target re-identification by using the bounding boxes of detected people and an appearance model of the followed person respectively. Lastly, a robot motion control module is implemented to form a complete RPF solution, which we refer to as *visible-joint-based* RPF method. In experiments, our method can follow a target person reliably even under partial occlusion, as shown in Fig. 1(c).

II. RELATED WORK

A. Monocular Person Tracking in Robot Person Following

A person following robot usually involves four modules: a detection module, a tracking module, an identification module and a robot motion control module. The detection module detects the tracked person in the image. The tracking module locates the person in terms of a 3D pose with respect to the robot. The identification module usually maintains an appearance model useful for re-identification in case of a lost target. Given the person's estimated location, the robot motion control module computes a motion command for the robot to maintain a desired following position with respect to the target person.

Many existing works in people tracking [3]–[8] use distance measurement sensors, which can be expensive and have difficulty in dealing with cluttered indoor environments due to the lack of textural information that is critical for data association. Some works are based on monocular vision. [9] tracks a person in the image space, and achieves person following with visual servo by keeping the person in the center of the image plane . Image-based tracking is convenient to implement but ineffective compared to the position-based tracking in 3D as the person and the robot move physically in the Cartesian space, rather than the 2D image space [22].

For estimating a person's location in the robot coordinate frame, inspired by well-known techniques in video surveillance [21][22], [13] proposes a *single-joint-based* method to locate the person by assuming that the neck of the person is always visible. However, this assumption cannot always be satisfied. Given the fact that under partial occlusion, one or more joints (not necessarily the neck) are still visible, it should be possible to estimate a person's location from the observed joints. Therefore, we propose to use the observable joints to track the person instead of a specific joint to address the challenge of partial occlusion.

B. 2D Human Bounding-box Detection and Joint Detection

To develop the detection module of our RPF method, there are two related topics: 2D human bounding-box detection [23] and joint detection [24]. 2D human bounding-box detection [24] seeks to locate people as bounding boxes in an image, and the methods can be divided into two families: two-stage [25] and one-stage [26]. A two-stage [25] method usually first generates category-independent proposals and then utilizes category-specific classifiers to label the proposals, while a one-stage method [26] directly generates labeled bounding boxes without any proposal generation. In this paper, we use a one-stage method (YOLOX [26]) to detect people for its fast inference and stable performance.

2D human joint detection [24] has also been well developed in recent years. It aims at localizing human joints (keypoints) in an image. There are also two families of methods: top-down [17][27] and bottom-up [28][29]. A top-down method first detects bounding boxes, and then localizes 2D joints within the boxes, whereas a bottom-up method detects joints first and then groups them into full bodies. We adopt a top-down method (AlphaPose [17]) in the tracking module of our *visible-joint-based* RPF system, due to its good performance even under partial occlusion.

C. 3D Person Location Estimation

A RPF system must contain a tracking module for person location estimation, and the current leading methods in solving this problem are mostly deep-learning-based. Such methods employ a deep neural network to infer the location of a person from the observed image in an end-to-end fashion. MonoLoc [30] first uses a neural network to detect joints of a person in an image, and then utilizes these estimated 2D joint positions to locate the person in 3D by a multi-task neural network. EPro-PnP [31] describes the pose of a person in the form of a 3D bounding box by integrating learnable 2D-3D correspondences. RootNet [32] develops a top-down pose estimation solution that computes the 3D poses of multiple people with respect to the camera coordinate frame.

All the methods mentioned above are entirely based on deep learning, although MonoLoc [30] consists of two separate networks, one for joint detection and the other person location, in a way that is similar to our solution. In addition, their training datasets usually involve a full observation of the objects of interest. However, in the person following scenario, a partial observation of the person often occurs, a situation that these methods cannot deal with effectively. In our work, we adopt a hybrid approach with a detection module and a tracking module where a learning-based detector [17] provides 2D information about the joints of the target person in an image to a subsequent model-based 3D person location tracker. The main advantages of this approach are: 1) 2D joint detectors, compared to the *deep-learning-based* methods, are shown to be more robust with respect to partial occlusion and 2) our proposed model-based location tracker is able to utilize a prior model of

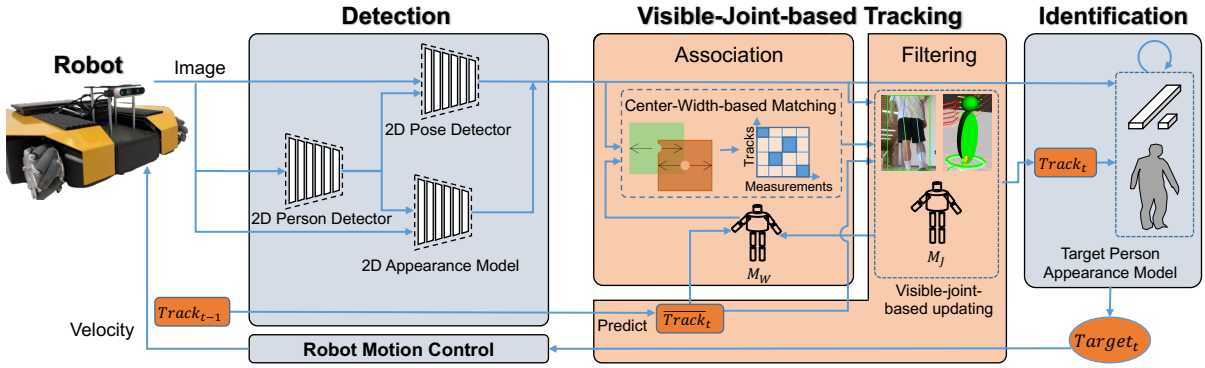


Fig. 2. Our proposed *visible-joint-based* RPF system is composed of a Detection module(Sec. III-A) including 2D person detector, 2D pose detector and appearance extraction model, a tracking module with track initialization (Sec. III-B), filtering (Sec. III-C), and data association (Sec. III-D), an identification module (not our focus in this paper) and a robot motion control module. $\mathcal{M}_J$ is the prior global model consisting of heights of joints and M_W represents the person's real width. This system can locate, track, and follow the target person even under partial occlusion.

the target person that easily computes the person's location analytically from these well-detected 2D joints.

III. METHODOLOGY

We introduce a novel *visible-joint-based* method by utilizing any visible joints to track the target person instead of a specific joint. Together with an identification module and a robot motion control module (which are not the focus in this paper), a complete RPF system is formed. The general overview of our system is shown in Fig. 2.

Typically, our tracking method follows the general process of the Kalman filter. In model construction (Sec. III-B), from detections of all joints in the first frame, we first build a prior model $\mathcal{M}_J$ consisting of the heights of the joints with respect to the ground plane. Then upon the detection of a target person, the target person's location can be initialized with the help of this prior model. In addition, in the filtering stage (Sec. III-C), tracks are updated with associated joint measurements where each visible joint contributes a location estimate, also with the help of the prior model. In the data association (Sec. III-D), we associate tracks and joint measurements in the bounding-box-like space due to the stability of the bounding-box detector. In case of target loss, our RPF system uses the identification module to re-identify a target person. Lastly, with the robot motion control module, person following is performed in the robot coordinate frame based on the estimated location of the target person.

As in existing works [13][22], the x-y plane of the robot coordinate frame is parallel to the ground plane. In addition, we use a calibrated camera whose origin has a known offset with respect to the robot coordinate frame and whose orientation is identical to the robot coordinate frame except for a known positive tilt angle (see Fig. 3).

A. Detection Module

First, we detect people to generate their bounding boxes and 2D joint positions in the image plane. We use the bounding boxes and joint positions for tracking the target person, and we use the appearance features within the bounding boxes for target identification. Specifically, we use YOLOX [26] for bounding-box detection and AlphaPose [17] for joint detection. We exploit all detected joints of a partially occluded person in his location estimation.

In each image, we define as our observation of a person–detected bounding box and joint positions including shoulder, hip, knee and ankle in the image plane. They are represented as $\mathcal{D} = \{\mathcal{B}, \mathcal{P}\}$, where $\mathcal{B} = \{[u, v], w, h\}$ defines the center, the width and the height of the bounding box, and $\mathcal{P}$ is the set of visible joints–a subset of $\{\mathbf{p}_{neck}, \mathbf{p}_{hip}, \mathbf{p}_{knee}, \mathbf{p}_{ankle}\}$ where $\mathbf{p}_i \in \mathbb{R}^2$, describes the pixel coordinates of a visible joint. Note that in our study, we describe each of the three pairs of hip, knee, and ankle joints by a single 2D point $\mathbf{p}_i$ in the image plane. The horizontal coordinate of $\mathbf{p}_i$ is that of the bounding box of the detected person; the vertical coordinate of $\mathbf{p}_i$ is the average height of the both joints or that of a single joint, in the pair, depending upon if one or two joints of the pair are visible.

B. Prior Model Construction and Track Initialization

With above detected observation $\mathcal{D}$ in the first frame, through resolving a well-designed over-constrained minimization problem, we construct a *joint-height-based* prior model of the target person, including his position with respect to the camera, which can be used to initialize the Kalman filter tracker. The prior model is represented as:

$$\mathcal{M}_J = \{h_{neck}, h_{hip}, h_{knee}\}, \tag{1}$$

which corresponds to the heights of the person's neck, hip and knee relative to the ankle, which is assumed to be on the ground at height 0. Further, we define the person's location as his ankle location, $\mathbf{X} \in \mathbb{R}^3$, in the camera coordinate frame with the ground plane constraint–$\mathbf{N}^T\mathbf{X} + \gamma = 0$ [33]. This constraint means that the ground plane is defined with a known normal $\mathbf{N} \in \mathbb{R}^3$ at a distance $\gamma > 0$ with respect to the optical center (see Fig. 3).

Based on the above definition, the locations of the other joints can be defined with respect to the ankle location as $\mathbf{X}_j \approx \mathbf{X} + h_j \cdot \mathbf{N}$ [34] where $h_j \in \mathcal{M}_J$. In this work, for model construction, we assume that a full body can be observed in the first frame and define 3D joint positions by the set $\mathcal{X}_{all} = \{\mathbf{X}_{neck}, \mathbf{X}_{hip}, \mathbf{X}_{knee}, \mathbf{X}\}$. We can then obtain the person's location $\mathbf{X}$ and the prior model $\mathcal{M}_J$ by the

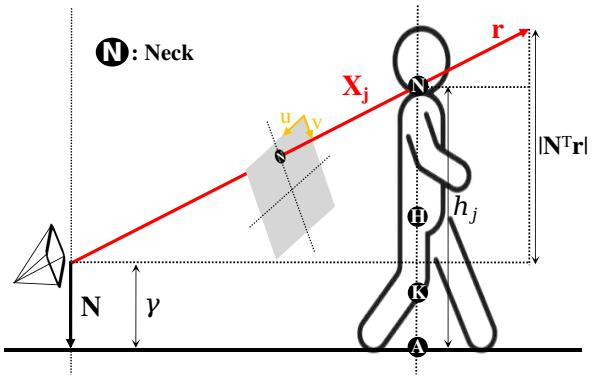


Fig. 3. Example of person's location estimation with a neck observation. Any visible joint can be utilized to initialize the person's location. The definition of the parameters and the estimation process are shown in Eq. 3.

minimization of the reprojection error:

$$\{\mathbf{X}, \mathcal{M}_J\} = \underset{\mathbf{X}, \mathcal{M}_J}{\operatorname{argmin}} \sum_i ||\mathbf{p}_i - g(\mathbf{X}_i)||_2^2, \tag{2}$$

where $g(\mathbf{X}_i)$ denotes the camera projection function of a 3D point in the camera coordinate frame to the image plane, i.e., $g : \mathbb{R}^3 \rightarrow \mathbb{R}^2$. After minimization, the prior model $\mathcal{M}_J$ is obtained, as is the person's location $\mathbf{X}$, which serves as the initial state of the Kalman filter tracker (see next section).

With $\mathcal{M}_J$ constructed, in any of the following frames, the person's location $\mathbf{X}$ can be re-initialized from any one visible joint by Eq. 3 [21] upon target loss and re-identification, without the need to observe the full body. Given the homogeneous pixel coordinates $\bar{\mathbf{p}}_j = [u, v, 1]^T$ of a joint, camera intrinsics $\mathbf{K}$, normal vector with $\mathbf{N}$ and its length γ, and the observed height of a joint h_j, we can compute the person's location as:

$$\begin{aligned} \mathbf{r} &= \mathbf{K}^{-1}\bar{\mathbf{p}}_j \\ \mathbf{X}_j &= \frac{|\gamma - h_j|}{|\mathbf{N}^T\mathbf{r}|} \cdot \mathbf{r} \\ \mathbf{X} &\approx \mathbf{X}_j - h_j \cdot \mathbf{N} \end{aligned} \tag{3}$$

Theoretically, we can initialize the person's location by any joint using Eq. 3. However, in practice, because the noise level in the observed joints varies, we prefer to initialize with Eq. 3 from the most reliable joint. Empirically, when using AlphaPose [17], we observe 1) the upper body is more stable than the lower body for a walking person [35]; 2) the shoulder is more distinguishable than the hip because the shoulder is near the background; and 3) the movement range of the knee is smaller than the movement range of the ankle. Therefore, we use Eq. 3 to initialize the person's location with only one joint, in the order of neck, hip, knee, and ankle.

C. Filtering

After the construction of $\mathcal{M}_J$ and the initialization of the target location, we adopt the unscented Kalman filter (UKF) as our filtering framework due to its advantages in dealing with non-Gaussian noise and non-linear observation function. In our UKF-based filtering, the state to be estimated only includes the person's location and velocity on the ground plane, defined as $\mathbf{s}_t = [x_t, y_t, \dot{x}_t, \dot{y}_t]^T$, which is enough to satisfy the need for person following. Further, a constant velocity motion model is assumed, i.e., for motion prediction,

$$\mathbf{s}_{t+1} = \mathbf{s}_t + \Delta t \cdot [\dot{x}_t, \dot{y}_t, 0, 0]^T, \tag{4}$$

where Δt is the time interval between two consecutive frames. Then we update the state with all visible joints so that each visible joint can contribute a location estimate. In other words, the update step of the Kalman filter depends on the joints that are visible. As defined in Sec. III-A, on the image plane, 2D visible joint positions are a subset of $\{\mathbf{p}_{neck,t}, \mathbf{p}_{hip,t}, \mathbf{p}_{knee,t}, \mathbf{p}_{ankle,t}\}$. Let $\mathcal{X}_{vis,t}$ be the corresponding 3D joint locations in the camera coordinate frame. Then in the image plane, our observation model used by the UKF is defined by:

$$\mathbf{z}_t = [g(\mathbf{X}_{j,t})]^T, \mathbf{X}_{j,t} \in \mathcal{X}_{vis,t}. \tag{5}$$

D. Data Association

In this filtering framework, we perform data association based on the bounding box information $\mathcal{B}$ instead of the joint information $\mathcal{P}$. Because 2D joint detection, compared to the bounding-box detection, is noisy due to occlusion and motion blur, leading to incorrect data association. Therefore, by assuming the person's shape is a cylinder with width M_W, we can associate the person's location $\mathbf{X}$ to the bounding box measurement $[u, w]^T$ by calculating the following expected observation:

$$\bar{u} = g(\mathbf{X})|_x, \quad \bar{w} = f_x \cdot M_W / \mathbf{X}_z, \tag{6}$$

where $\bar{u}$ and $\bar{v}$ represents expected the horizontal component of the bounding-box center and the bounding-box width respectively; $g(\mathbf{X})|_x$ represents the horizontal component of the pixel; f_x is the focal length of the camera. Then our distance metric is defined as:

$$d = ((\bar{u} - u)^2 + (\bar{w} - w)^2)^{1/2}, \tag{7}$$

where $\bar{u}, \bar{w}$ are from Eq. 6 and u, w are from the bounding-box information $\mathcal{B}$. Finally a global nearest neighbor method is utilized to match the measurements to the tracks.

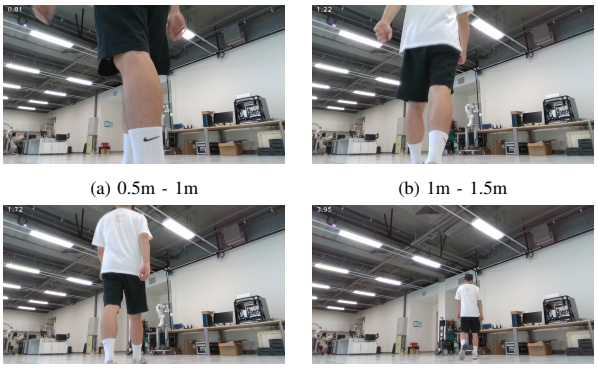
IV. EXPERIMENTS

To verify the performance of our proposed *visible-joint-based* method and the whole RPF system, we conduct experiments on different datasets. In this section, we first introduce the datasets, the baselines and implementation details of our experiments. Secondly, the effectiveness of our *visible-joint-based* method is demonstrated by a comparison with the *single-joint-based* and *deep-learning-based* methods on a custom-built dataset. Lastly, we show the superiority of our whole RPF system on a public person following dataset.

A. Datasets

RPF needs accurate person location estimation and continuous target person tracking. To test these two aspects

of our method, we evaluate it on two datasets named as person location (PL) dataset and person tracking (PT) dataset respectively. The PL dataset is a custom-built dataset involving four sequences where the 3D location of a partially occluded person is recorded by a motion capture system or a LiDAR sensor. The distance between the person and the robot varies between 0.5m - 6.0m. Some examples are shown in Fig. 4 where an occluded body is often observed, a difficult condition for location estimation. For evaluation of tracking continuity in the 2D image space, the PT dataset, a public dataset [36] with eleven sequences, is used where the target person's ground truth position is represented by a bounding box. This dataset involves challenging situations including illumination change, appearance change and occlusion due to people crossing, which are hard for continuous target person tracking.



(a) 0.5m - 1m (b) 1m - 1.5m

(c) 1.5m - 3m (d) 3m - 6m

Fig. 4. Examples of the custom-built dataset for evaluating the location accuracy of existing methods. From (d) to (a), with the distance decreasing, more joints are occluded. Our method can accurately locate the person under such partial occlusion.

B. Baselines

For evaluating the accuracy of 3D location estimation of our method on the PL dataset, we conduct a comparison with *single-joint-based* and *deep-learning-based* methods introduced in Sec. II-C. All compared methods locate the person in the camera coordinate frame. They are named as follows:

- **Single-joint-based** [13] tracks the person only when the full body is observed including his neck.
- **MonoLoco** [30] could locate the person directly by a neural network with the 2D pose of the person as input.
- **Mono3DBox** predicts the 3D bounding box of the person based on EPro-PnP [31] of which the bottom center is utilized as the estimated location.
- **Mono3DPose** performs person location by predicting his 3D root location through RootNet [32].
- **MonoDepth**, based on MiDas [37], first obtains the scene depth map and the person's bounding box, then the distance to the person is achieved by averaging the reduced region of the person's depth map. Subsequently, the location of the person can be obtained with the camera reprojection.

All estimated locations are transformed to the ground plane and smoothened by the UKF in our evaluation. The main purpose of this comparative experiment is to establish that our solution to location estimation is superior to these SOTA methods in the case of partial occlusion due mostly to the assumption of full-body observation by these SOTA methods during their training.

In addition, we evaluate the person tracking ability of the whole RPF system (involving *visible-joint-based* method and the identification module) as previous RPF works [36] [13], which regard RPF as a special case of the object tracking. For comparison with object tracking methods, our RPF system, although assuming a calibrated camera to the ground plane to track the person in 3D space, is evaluated in the image space by the associated bounding boxes. Specifically, we compare our RPF system with popular MOT (multiple object tracking) and SOT baselines, including QDTrack [38], Bytetrack [39], SiamRPN++ [40] and STARK [41]. These methods initialize the target person by specifying the ID of the desired person (for MOT) or annotating his bounding box in the first frame (for SOT).

C. Implementation Details

In our method, bounding-box detection model relies on YOLOX [26] and 2D joint detection on AlphaPose [17]. The feature extraction model used for appearance description is a deep re-identification model based on CNN [42]. The minimization problem in model construction (Eq. 2) is solved by the Levenberg-Marquardt algorithm.

All evaluations are run on a computer with Intel® Core™ i7-10700F CPU @ 2.90GHz and NVIDIA GeForce RTX 2060. For real robot experiments, a Clearpath Dingo-O and a laptop with Intel(R) Core(TM) i5-10200H CPU @ 2.40GHz and NVIDIA GeForce RTX 1650 are used. A RealSense D435i with 1280×720 resolution and 30Hz frequency is mounted on the robot.

D. Experimental Results

1) Evaluation of visible-joint-based person location estimation: This experiment is conducted on the PL dataset, and three metrics are used for evaluation: 1) average location error (ALE) is calculated in the Euclidean space of the ground plane, 2) recall is the ratio of the number of recognized frames to that of all frames, and 3) weighted location error (WLE) is the proportion of ALE to recall, which can evaluate the overall effectiveness of a location estimation method considering both accuracy and robustness. Results are shown in Table I. We can observe that, compared to other methods, our method achieves the lowest WLE in all sequences at 0.11m, 0.09m, 0.10m and 0.08m respectively.

The *single-joint-based* method [13] achieves a 0.10m ALE and a 64.0% recall in Sequence I where a full body can be observed occasionally. While in other sequences where only a partially occluded body is observed, the *single-joint-based* method always fails to locate the target person as expected. MonoLoco [30] and Mono3DBox [31] also perform poorly with a low recall and a high ALE, especially on Sequence II to IV. Mono3DPose [32] and MonoDepth [37] can work on

TABLE I. Comparison of location performance between our method and esisting baselines on the PL dataset. All sequences are captured within a distance range of 0.5m-2m, except sequence I is within 0.5m-6m. † indicates deep-learning-based methods. **ALE** (m) represents average location error, **Recall** is the ratio of the number of recognized frames and that of all frames, and **WLE** (m) is the proportion of ALE to recall. ✗means the algorithm fails to locate the target person in case of failed recognition or the ALE is larger than 5 meters. Our method achieves the lowest WLE in all sequences at 0.11m, 0.09m, 0.10m and 0.08m respectively.

Methods	**I**			**II**			**III**			**IV**		
	ALE ↓	***Recall*** ↑	***WLE*** ↓	***ALE***	***Recall***	***WLE***	***ALE***	***Recall***	***WLE***	***ALE***	***Recall***	***WLE***
Single-joint-based [13]	**0.10**	0.64	0.17	0.12	0.04	3.28	✗	✗	✗	✗	✗	✗
MonoLoco† [30]	1.07	0.48	2.22	✗	✗	✗	✗	✗	✗	✗	✗	✗
Mono3DBox† [31]	0.25	0.66	0.38	0.59	0.21	2.76	2.10	0.09	22.25	1.59	0.39	4.12
Mono3DPose† [32]	0.36	0.95	0.38	0.51	**1.00**	0.51	0.95	**1.00**	0.95	1.11	**1.00**	1.11
MonoDepth† [37]	0.57	**1.00**	0.57	0.32	**1.00**	0.32	0.36	**1.00**	0.36	0.22	**1.00**	0.22
Ours	0.11	**1.00**	**0.11**	**0.09**	0.98	**0.09**	**0.10**	**1.00**	**0.10**	**0.08**	0.92	**0.08**

TABLE II. Evaluation of 2D target person tracking between our method and other object tracking baselines on the PT dataset. Our *visible-joint-based* RPF system achieves the best performance with a 97.5% accuracy.

Methods	**Type**	**Accuracy (%)**
QDTrack [38]	MOT	48.0
Bytetrack [39]	MOT	88.6
SiamRPN++ [40]	SOT	93.6
STARK [41]	SOT	96.5
Single-joint-based	RPF	92.0
Visible-joint-based	RPF	**97.5**

almost all frames with near 100% recall but at the expense of a higher ALE compared to our method.

2) Evaluation of our RPF system: The evaluation of our *visible-joint-based* RPF system is conducted independently from considering robot control as a special case of object tracking. Thus we compare the RPF system with other popular object tracking baselines on the PT dataset based on the accuracy metric. Accurate tracking is defined by considering a recognized target person's bounding box as a true positive if the distance between estimated and ground-truth centers of the bounding boxes is less than 50 pixels.

Results are shown in Table II. STARK [41] achieves good performance in MOT and SOT baselines with a 96.5% accuracy, while our method achieves the best accuracy at 97.5%. This indicates that in person following, our method can track the target person as well as the popular object tracking methods. Our RPF system, despite the power of our person identification module (which is not the focus in this paper), achieves continuous target person tracking due largely to the ability of tracking under partial occlusion. This can be further verified by the comparison of *single-joint-based* and our *visible-joint-based* RPF systems where only the location estimation method is different. We can observe that if the method changes to *single-joint-based*, the accuracy drops to 92.0%.

E. Discussion

As is shown in Table I, the proposed method achieves the best location estimation performance with a 0.10m ALE, a 98% recall and a 0.10m WLE on average. Compared to the *single-joint-based* method, it can accurately locate the person even under partial occlusion. Such results indicate that: 1) 2D learning-based pose estimator (AlphaPose [17]) can perform well under partial occlusion; and 2) our model-based method is able to utilize these well-detected joints to locate the partially occluded person. Compared to the *deep-learning-based* baselines, our method is also superior because the learning-based methods are limited in terms of generalization, and are therefore sensitive to environmental changes. On the other hand, our hybrid approach is not dependent on labeled 3D training data, and yet outperforms the baselines. In conclusion, due to the combination of the robust 2D learning-based joint detector and our model-based location estimator, our *visible-joint-based* method is able to locate the person accurately in robot person following, especially under partial occlusion.

From Table II, we can observe that our method achieves the highest accuracy at 97.5% which is higher than that of the MOT and SOT baselines. Such a result indicates that, in person following, our method is reliable for not only locating the person accurately but also tracking the person as persistently as these well-designed object tracking methods. The accuracy of our RPF system would drop by 5.5% if the location method changes to a *single-joint-based* one. This is because, on the PT dataset, there are many situations of partial occlusion, caused by people crossing, corner walls and limited FoV. In above cases, the *single-joint-based* method would lose the person while our method can persistently locate the person and correctly handle these situations of partial occlusion.

V. CONCLUSION

In this paper, for performing robot person following under partial occlusion, we propose a practical *visible-joint-based* method to locate the person with the observation of any of his neck, hip, knee and ankle joints. Our method takes advantage of a prior model of the tracked person and uses one or more of observed joints for locating the person. The key benefit of our method is that it can locate the person accurately and persistently even under partial occlusion. Compared to baselines, our RPF system achieves SOTA target person tracking performance across multiple evaluation metrics.

REFERENCES

[1] M. J. Islam, J. Hong, and J. Sattar, "Person-following by autonomous robots: A categorical overview," *The International Journal of Robotics Research*, vol. 38, no. 14, pp. 1581–1618, 2019.

[2] M. A. Goodrich and A. C. Schultz, *Human-robot interaction: a survey*. Now Publishers Inc, 2008.

[3] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2d laser scanners," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 726–733.

[4] Y. Sung and W. Chung, "Hierarchical sample-based joint probabilistic data association filter for following human legs using a mobile robot in a cluttered environment," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 340–349, 2015.

[5] J. Yuan, S. Zhang, Q. Sun, G. Liu, and J. Cai, "Laser-based intersection-aware human following with a mobile robot in indoor environments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 354–369, 2018.

[6] M. Wang, D. Su, L. Shi, Y. Liu, and J. V. Miro, "Real-time 3d human tracking for mobile robots with multisensors," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5081–5087.

[7] K. Koide and J. Miura, "Identification of a specific person using color, height, and gait features for a person following robot," *Robotics and Autonomous Systems*, vol. 84, pp. 76–87, 2016.

[8] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 5512–5519.

[9] M. Zhang, X. Liu, D. Xu, Z. Cao, and J. Yu, "Vision-based target-following guider for mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9360–9371, 2019.

[10] L. Zheng, M. Tang, Y. Chen, G. Zhu, J. Wang, and H. Lu, "Improving multiple object tracking with single object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2453–2462.

[11] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[12] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4660–4669.

[13] K. Koide, J. Miura, and E. Menegatti, "Monocular person tracking and identification with on-line deep feature selection for person following robots," *Robotics and Autonomous Systems*, vol. 124, p. 103348, 2020.

[14] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth prediction," in *The International Conference on Computer Vision (ICCV)*, October 2019.

[15] D. Park, R. Ambrus, V. Guizilini, J. Li, and A. Gaidon, "Is pseudo-lidar needed for monocular 3d object detection?" in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[16] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[17] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2353–2362.

[18] Y. Cheng, B. Yang, B. Wang, W. Yan, and R. T. Tan, "Occlusion-aware networks for 3d human pose estimation in video," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 723–732.

[19] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image," in *European conference on computer vision*. Springer, 2016, pp. 561–578.

[20] V. N. Nguyen, Y. Hu, Y. Xiao, M. Salzmann, and V. Lepetit, "Templates for 3d object pose estimation revisited: Generalization to new objects and robustness to occlusions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6771–6780.

[21] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 3–15, 2008.

[22] W. Choi and S. Savarese, "Multiple target tracking in world coordinate with single, minimally calibrated camera," in *European Conference on Computer Vision*. Springer, 2010, pp. 553–567.

[23] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, pp. 261–318, 2020.

[24] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[25] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[26] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[27] 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*, 2019, pp. 5693–5703.

[28] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," *Advances in neural information processing systems*, vol. 30, 2017.

[29] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[30] B. Lorenzo, K. Sven, and A. Alexandre, "Perceiving humans: From monocular 3d localization to social distancing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7401–7418, 2022.

[31] H. Chen, P. Wang, F. Wang, W. Tian, L. Xiong, and H. Li, "Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[32] G. Moon, J. Y. Chang, and K. M. Lee, "Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 10 132–10 141.

[33] Y. Ma, S. Soatto, J. Košecká, and S. Sastry, *An invitation to 3-d vision: from images to geometric models*. Springer, 2004, vol. 26.

[34] X. Fei, H. Wang, L. L. Cheong, X. Zeng, M. Wang, and J. Tighe, "Single view physical distance estimation using human pose," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 406–12 416.

[35] L. Bertoni, S. Kreiss, and A. Alahi, "Monoloco: Monocular 3d pedestrian localization and uncertainty estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6861–6871.

[36] B. X. Chen, R. Sahdev, and J. K. Tsotsos, "Integrating stereo vision with a cnn tracker for a person-following robot," in *International Conference on Computer Vision Systems*. Springer, 2017, pp. 300–313.

[37] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, 2022.

[38] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2021.

[39] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[40] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[41] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 448–10 457.

[42] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.