2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

When We First Met: Visual-Inertial Person Localization for Co-Robot Rendezvous

Xi Sun, Xinshuo Weng and Kris Kitani[1]

***Abstract*— We aim to enable robots to visually localize a target person through the aid of an additional sensing modality – the target person's 3D inertial measurements. The need for such technology may arise when a robot is to meet a person in a crowd for the first time or when an autonomous vehicle must rendezvous with a rider amongst a crowd without knowing the appearance of the person in advance. A person's inertial information can be measured with a wearable device such as a smart-phone and can be shared selectively with an autonomous system during the rendezvous. We propose a method to learn a visual-inertial feature space in which the motion of a person in video can be easily matched to the motion measured by a wearable inertial measurement unit (IMU). The transformation of the two modalities into the joint feature space is learned through the use of a triplet loss which forces inertial motion features and video motion features generated by the same person to lie close in the joint feature space. To validate our approach, we compose a dataset of over 3,000 video segments of moving people along with wearable IMU data. We show that our method is able to localize a target person with 80.7% accuracy averaged over testing data with various number of candidates using only 5 seconds of IMU data and video.**

I. Introduction

Person localization for a rendezvous is crucial in real-world applications such as assistive robots [1]–[3] and autonomous driving [4]–[19]. Consider the scenario where an autonomous vehicle rendezvous with its user for the first time. How does the autonomous vehicle localize the user without any information about what the user looks like? In this work, we consider the possibility of using the user's inertial measurement unit (IMU) data collected by her smartphone as a unique descriptor of the user's motion, which can be then used by the autonomous vehicle to localize the user with a dashboard camera.

Prior work on person localization often utilizes visual-visual feature matching, assuming that the target person's appearance information is known in advance. However, this assumption may not always hold as it requires a data capture process prior to the rendezvous. To deal with the situation where the target person's appearance information is not available, we must rely on other sensor that can capture target person's information in the wild. We choose to use the inertial sensor as the 3D inertial measurement that describes the user's motion and can be matched with the visual motion information collected by the dash camera for person localization. Also, the user's 3D inertial measurement can be easily obtained because modern smart wearable devices such as smart-phone and smart-watch are often equipped with an inertial sensor. Moreover, due to its low dimensionality compared to visual data, we can transmit the inertial measurement to the autonomous vehicle in real time at a low cost.

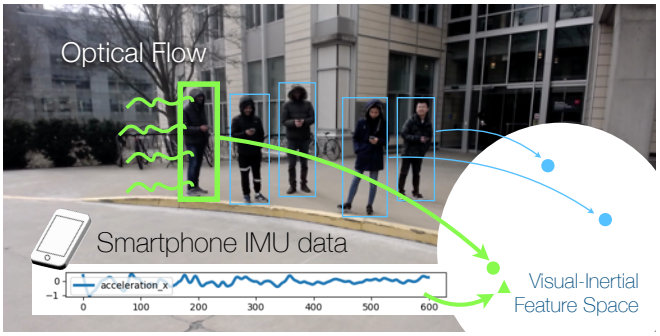

Fig. 1. Our visual-inertial feature transformer maps IMU motion and image motion from the same person to a similar location in the feature space.

Our approach is based on visual-inertial feature matching. Specifically, we first obtain the visual motion information from the dashboard camera by computing the optical flow [20] for a fixed time window. In the meantime, we obtain the motion information in 3D space measured by the IMU for the same time window. Since directly transforming the local 3D motion measurements and the 2D motion in the camera frame into same world coordinates is difficult and requires calibration of a fixed camera, we propose to learn a feature transformer based on the LSTM [21] and convolutional layers that can map the motion information from visual and inertial modalities into a joint feature space. The visual and inertial features are optimized using a triplet loss [22] so that the learned features of the same person lie close in the joint feature space.

As there is no existing dataset suitable for training our feature transformer for person localization, we collect a new visual-inertial dataset containing time-synchronized video and inertial data. Our dataset has over 3,000 video segments of moving people along with their corresponding IMU data. The IMU data is collected by the smartphones held in people's hands. Different from existing visual-inertial datasets which often rigidly attach the inertial sensor on people's back [23] or body limbs [24]–[26], we let people hold smartphones in their hands naturally to mimic the real-world scenarios. As a result, our dataset is more realistic but challenging as the location of the inertial sensor is more flexible and the motion of the inertial sensor might not always align with the motion of people's back or limbs.

To validate our approach, we evaluate it on the test split

[1]Xi Sun, Xinshuo Weng and Kris Kitani are with Robotics Institute, Carnegie Mellon University. xis@andrew.cmu.edu, {xinshuow, kkitani}@cs.cmu.edu.

of our visual-inertial dataset. Our experiments show that our approach is able to accurately identify a target person with 80.7% accuracy on average using only 5 seconds of IMU and video data. To summarize, our contributions are as follows:

1) **A new task, namely visual-inertial person localization**, which aims to localize the target without requiring the appearance information of the target in advance;
2) **A new large visual-inertial dataset**, which is collected in the wild with multiple persons without fixed attachment of the inertial sensor to each person's body;
3) **An effective approach for the proposed task**, also being the first learning-based approach for the task and outperforming competitive baselines we devised from state-of-the-art techniques.

II. Related Work

Visual-Inertial Person Localization. To the best of our knowledge, [23] is the only work that attempted matching between visual and inertial data for person localization. First, [23] employs a visual heading network to predict person's 3D orientation with respect to the camera from a single image. Then, they match the person's 3D orientation predicted from the image with the orientation integrated from angular velocity obtained from the inertial sensor to generate image-based person predictions. To rigidly align the orientation of the inertial sensor with the person's body orientation and make the orientation prediction problem easier, [23] attaches the inertial sensor to the back of the target person. This makes [23] not applicable in the real world scenarios where the inertial sensor can be flexible. Additionally, [23] employs velocity matching between inertial and visual data to formulate trajectories of the previously generated image-based predictions. Specifically, the 3D foot position of the person is estimated from an image, which is then used to compute the 3D velocity of the target person given a pair of images. Meanwhile, the 3D velocity can be also estimated by integrating the linear acceleration from the inertial data, which can be used to match with the 3D velocity computed from the visual data. Different from [23] which employs hand-crafted inertial features (*i.e.*, orientation and velocity obtained by integration) to match with the visual data, our proposed method learns to transform visual and inertial data into a joint feature space for matching. Also, our proposed method is more useful in real world scenarios as we do not restrict the placement of the inertial sensor.

Visual-Inertial Dataset. Although visual-inertial person localization is under-explored in prior work, there are existing visual-inertial datasets collected for other vision tasks. The CMU Multi-Modal Activity Database [27] aims to understand cooking and food preparation activities. They rigidly attach multiple IMU sensors on person's body to collect the inertial data. In the meantime, video data is also collected from multiple viewpoints. The Total Capture Dataset [28] is designed for human pose estimation. Similarly, [28] contains synchronized multi-view video and IMU data with the inertial sensor attached to the human body. However, both [27] and [28] are not suitable for person localization as 1) they only collect data for one person at a time, 2) the location of the inertial sensor is fixed, and 3) the data is collected in the indoor setting. Different from existing datasets, we collect a new visual-inertial dataset with multiple persons outside and the location of the inertial sensor flexible, in order to mimic the real-world autonomous driving pick-up scenario.

Visual Person Localization. Depart from the visual-inertial person localization, prior work has investigated person localization using only visual data with the re-identification technique. The common approach is to first obtain the feature embedding from two sources of visual data (one from an unknown query person and the other from a pre-built database containing information of the target person), and then perform classification to identify if the query person is the target person. Once the target person is successfully identified, the localization is solved. To obtain effective visual embedding for identification and localization, prior work focuses on image-based [29]–[31] and video-based [32] methods for feature learning. However, visual person localization methods are only applicable when the pre-built database containing the information of the target person is available. In other words, if we do not have the target person's information in advance, we cannot solve the localization problem with only visual information but need the aid of an additional sensor. In this paper, we investigate the possibility of using the user's inertial data for localization.

III. Approach

Given a video with multiple people standing or walking, and the IMU readings from a smartphone carried by a person in the scene, our goal is to identify which person in the video the IMU data belongs to as shown in Fig. 1. As described above, we aim to learn a joint visual-inertial feature space in which the visual and inertial features from the same person lie close in that space.

Formally speaking, in a video segment (150 frames or 5 seconds), we denote each person in that video by an index $n \in [N]$, where N is the number of people in this video segment. For each person n, we extract a visual feature g_{VIS} to encode its motion in the video. Meanwhile, we extract an inertial feature g_{IMU} of the target person from the IMU data to encode its motion in 3D space. During training, we learn a visual feature embedding function $H_{\text{VIS}} : g_{\text{VIS}} \rightarrow f$ and an inertial feature embedding function $H_{\text{IMU}} : g_{\text{IMU}} \rightarrow f$ to map both features into the same joint visual-inertial feature space. Then a triplet loss is used to force the mapped features that belong to the same person to lie close in the joint feature space. At test time, once we find the visual embedding which is the closest to the inertial embedding in the joint feature space, the target person is localized in the video.

A. Visual Feature Extraction

In order to extract people's motion feature from a video segment, we first pre-process the video by performing person detection [33]–[36] using YOLOv3 [37] at all frames and

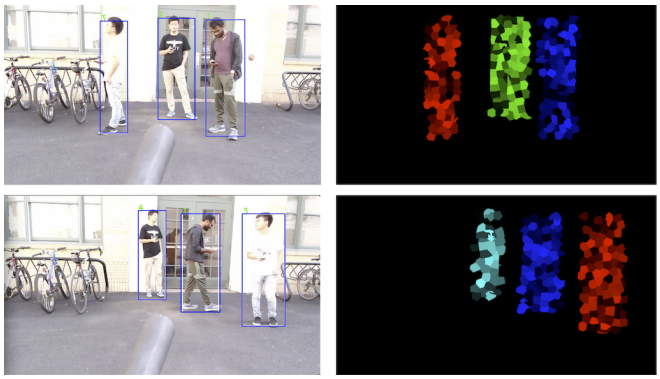

Fig. 2. **(Left)** YOLOv3 person detections. **(Right)** Temporal super-pixels (TSP) for each tracked person in the video. Average optical flow is computed as the motion feature for each TSP representing different body parts.

then associating the detections into trajectories using a multi-object tracker – DeepSORT [4]. Once we have obtained a trajectory of boxes for each person, we can now extract the motion feature. Specifically, we first extract the optical flow for each box trajectory, and then further decompose it into smaller temporal super-pixels using [20]. The reason for decomposition is that we believe the inertial data measured by the smartphone is only correlated with a part of the body where the smartphone is held, instead of the entire body. Without this decomposition, the optical flow representing the motion of the entire body might not be easily matched with the inertial feature representing the motion of a part of the body, thus leading to inferior localization performance.

Formally speaking, given a video segment $V_{t:t+T}$ with T frames, we denote the set of temporal super-pixels (TSPs) as $\boldsymbol{\xi} = \{\xi_1, \xi_2, \dots, \}$. We then filter out the TSPs that do not lie within the trajectories of boxes and obtain a subset of TSPs denoted as $\boldsymbol{\xi}^n \subset \boldsymbol{\xi}$ for each person. To obtain the motion feature for each TSP, we compute the average optical flow over all pixels for each temporal slice of a TSP:

$$\mathbf{v}_{\xi_i^n} = [(dx_t, dy_t), (dx_{t+1}, dy_{t+1}), ..., (dx_{t+T-1}, dy_{t+T-1})],$$

where each vector $\mathbf{v}_{\xi_i^n}$ represents the motion of a part of the human body as shown in Fig. 2.

Although the TSP features are sufficient to represent the motion information of different body parts in the video, there is still a gap between the TSP features and the inertial 3D motion features as the TSP features are computed in the 2D image space, *i.e.*, the perspective projection of the person's 3D motion. To alleviate this issue and bridge the gap between the 2D and 3D space, we include extra information that is related to the 3D depth and orientation of the person, which can implicitly help the matching between the learned visual and inertial feature embeddings. Specifically, we use two types of information obtained from the video segment:

1) The height and width of the person as an indication of the distance between the person and the camera:

$$\mathbf{b}^n = [(h_t, w_t), (h_{t+1}, w_{t+1}), ..., (h_{t+T-1}, w_{t+T-1})].$$

2) The relative positions of the person's left and right shoulder keypoints to the bounding box center as an indication of the body orientation relative to the camera:

$$\mathbf{k}^n = [(\mathbf{ls}_t, \mathbf{rs}_t), (\mathbf{ls}_{t+1}, \mathbf{rs}_{t+1}), (\mathbf{ls}_{t+T-1}, \mathbf{rs}_{t+T-1})],$$

where $\mathbf{ls}$ and $\mathbf{rs}$ are tuples of the keypoint's x and y coordinates relative to the box center's coordinate in the image frame. We choose the shoulder keypoints because their positions are stable to the body orientation.

We use the bounding box trajectory outputs obtained by YOLOv3 and DeepSORT to extract the width and height information for each person. To obtain the positions of shoulder keypoints, we first use AlphaPose [38] to detect 17 keypoints of the full body, and then only select the two points representing the shoulder joints. If a person is temporarily occluded, we apply linear interpolation on the bounding box and keypoint trajectories. If a person exits out of the camera frame, we pad zeros to the trajectories.

B. Inertial Feature Extraction

To match with the visual motion feature, we also need to extract an inertial feature, which represents the 3D motion of the smartphone for the target person. Given the IMU data containing the 3D linear acceleration (pre-processed by removing the affect of gravity) $\mathbf{a} = [\boldsymbol{a}_x, \boldsymbol{a}_y, \boldsymbol{a}_z]$ and angular velocity $\boldsymbol{\omega} = [\boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z]$ in the smartphone's local coordinate frame, we construct the inertial feature for target person n denoted as $g^n_{\text{IMU}} = [\boldsymbol{a}_x, \boldsymbol{a}_y, \boldsymbol{a}_z, \boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z]^T$ by concatenating linear acceleration and angular velocity. As a result, the inertial feature g^n_{IMU} is a $6 \times T'$ matrix where T' is the number of frames temporally aligned with the video segment's time window. As the IMU frame rate is 100Hz, with a ratio of 3.33:1 to the video frame rate of 30Hz, we uniformly sample the inertial frames so that $T' = 3 \times T$, where T is the number of frames in a video segment.

C. Learning the Joint Visual-Inertial Feature Space

Although the raw visual and inertial feature contain sufficient information representing the person's 3D motion, they still lie in different feature spaces as they are obtained from different source of data and thus it is difficult to directly match them. To overcome this issue, we propose to learn a feature transformer that further transform the raw visual and inertial features into a joint feature space so that the matching for a same person is possible.

The proposed network for learning the joint feature space is shown in Fig. 3. To transform the raw visual feature into the joint space while model the temporal dependency, we first apply three LSTM networks for the TSP features (green), bounding box size data (red) and pose keypoints data (orange) respectively, each with different weights. Then, we combine these three features together as the visual feature. To transform the raw inertial feature into the joint space, we first use a 1D convolution layer to reduce the dimensionalities of the inertial feature to be the same as the visual feature. Then, we also apply an LSTM network (blue) to model the temporal dependency for the inertial feature. For both visual and inertial features, we use the hidden state from the LSTM

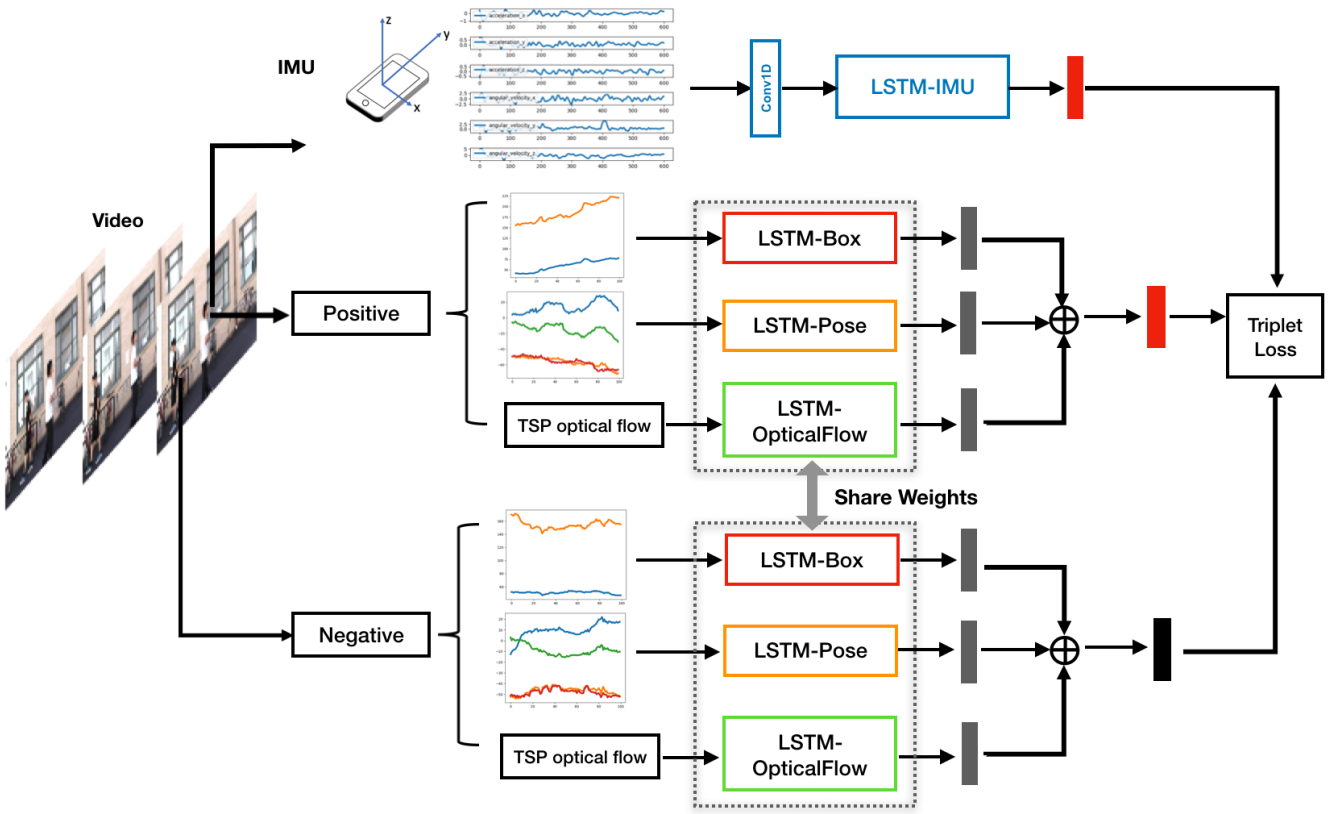

Fig. 3. **Proposed Network**. Our network has one branch to extract the inertial feature of the target person and two branches to extract the visual features from one positive and one negative sample. At each iteration of training, the positive visual feature is extracted from the target person while the negative visual feature is from a randomly picked different person. Once the raw inertial and visual features are extracted, they are fed into our visual-inertial feature transformer so that the transformed feature embeddings lie in a same feature space. A triplet loss is then applied to minimize the L2 distance between the inertial embedding and the positive visual embedding while maximize the L2 distance between the inertial embedding and the negative visual embedding. At test time, we compute the visual embeddings for all persons in the video and also compute the inertial embedding of the target person. The predicted target person in the video is then the person whose visual embedding has minimum distance to the target person's inertial embedding.

at each timestep as the final output embedding, which are formally defined as follows:

$$H_{\text{VIS}}(\mathbf{v}_{\xi_i^n}, \mathbf{b}^n, \mathbf{k}^n) = f_{\text{OF}}(\mathbf{v}_{\xi_i^n}) + \alpha f_{\text{Pose}}(\mathbf{k}^n) + \beta f_{\text{Box}}(\mathbf{b}^n),$$

$$H_{\text{IMU}}(g^n_{\text{IMU}}) = f_{\text{IMU}}(g^n_{\text{IMU}}),$$

where the final visual embedding is computed by summing over three different input features and α and β are two hyper-parameters defining the weights. As each person n has a set of TSPs $\boldsymbol{\xi}^n$ and thus we have $|\boldsymbol{\xi}^n|$ final visual embeddings, we duplicate the number of final inertial embeddings so that we have the same number of visual and inertial embeddings for each person in a time window with T frames. During training, we use every pair of the inertial and visual embeddings and minimizing the L2 distance between them if they belong to the same person:

$$\mathcal{L}(g^n_{\text{IMU}}, g^n_{\text{VIS}}(\xi_i)) = ||H_{\text{VIS}}(g^n_{\text{VIS}}(\xi_i)) - H_{\text{IMU}}(g^n_{\text{IMU}})||_2,$$

where $g^n_{\text{VIS}}(\xi_i)$ denotes the tuple $(\mathbf{v}_{\xi_i^n}, \mathbf{b}^n, \mathbf{k}^n)$. Additionally, we use the triplet loss as in [39], [40]. Specifically, for each target person n with the inertial embedding, we use the visual embedding obtained from the same target person as a positive example and use the visual embedding obtained from a randomly sampled different person as a negative example. The positive and negative samples share the same weights in the LSTM networks (*i.e.*, LSTM-OpticalFlow, LSTM-Pose, LSTM-Box). Then, the triplet loss is applied to minimize the L2 distance between the inertial and positive visual embedding and maximize the L2 distance between the inertial and negative visual embedding:

$$\mathcal{L}(g^n_{\text{IMU}}, g^+_{\text{VIS}}(\xi_i), g^-_{\text{VIS}}(\xi_j)) = \max(||H_{\text{VIS}}(g^+_{\text{VIS}}(\xi_i)) - H_{\text{IMU}}(g^n_{\text{IMU}})||_2 - ||H_{\text{VIS}}(g^-_{\text{VIS}}(\xi_j)) - H_{\text{IMU}}(g^n_{\text{IMU}})||_2 + \kappa, 0),$$

where $g^+_{\text{VIS}}(\xi_i)$ is the extracted visual feature given a TSP from the positive person and $g^-_{\text{VIS}}(\xi_j)$ is the extracted visual feature given a randomly selected TSP from a non-target person. κ is the margin separating the positive and negative feature space. At test time, given a video segment $V_{t:t+T}$ with N people in the scene, we choose one person as the target person at a time and compute its inertial embedding. Meanwhile, we compute the visual embedding for all persons in the video. Then, the predicted target person is the person whose visual embedding averaged over all TSPs has the minimum distance to the target person's inertial embedding:

$$\hat{n} = \arg\min_{n' \in [N]} \frac{1}{|\boldsymbol{\xi}^{n'}|} \sum_{i=1}^{|\boldsymbol{\xi}^{n'}|} ||H_{\text{VIS}}(g^{n'}_{\text{VIS}}(\xi_i)) - H_{\text{IMU}}(g^n_{\text{IMU}})||_2,$$

where $|\boldsymbol{\xi}^{n'}|$ is the number of TSP's for person n'.

IV. A New Visual-Inertial Dataset

To train our proposed method for visual-inertial person localization in the wild, we need a dataset with synchronized

TABLE I
STATISTICS OF THE VIDEO DATA COLLECTED IN OUR DATASET.

Number of people	2	3	4	5	6
Number of videos	17	15	11	7	8
Number of total frames	12,900	19,600	21,400	10,084	5,000

video and inertial data that include multiple people acting freely outside, each carrying a smartphone in their hand. However, existing visual-inertial datasets [23], [27], [28] do not satisfy these requirements and often have three limitations: 1) they rigidly attach the inertial sensor to person's body (*e.g.*, limb or back) so that the motion of the inertial sensor tightly aligns with the body part; 2) they often record the data in the indoor setting; 3) only one person is recorded at one time. As a result, prior datasets are not applicable to our challenging visual-inertial person localization task, and we have to collect a new dataset to satisfy the task conditions, which we plan to make public to encourage future research on similar tasks.

A. Video Recording

We set up a static HD webcam with a resolution of 1920×1080 on a tripod about one meter above the ground for video recording, similar to the setting of a dashboard camera in a car. We choose to record the video outside of public buildings in the daytime, in order to mimic real-world autonomous vehicle pickup scenarios. At each time of the recording, we hire 2 to 6 different volunteers, assign a smartphone to each of the volunteer, and ask the volunteers to perform casual random motion (*e.g.*, walking or standing still while holding the smartphones at hands). Each video recording is about half to two minutes long with a frame rate of 30Hz. In total, we have recorded 58 videos with a total of 68984 frames. We summarize the statistics of our data recording in Table I. Our dataset contains common types of pedestrian motion such as standing, walking and turning, recorded at four different backgrounds to increase the diversity of the dataset. Also, we do not provide and allow to use the calibration parameters of the camera in our dataset, as in the real world the calibration parameters of the dashboard camera might vary across vehicles and not available to our approach for person localization. Each video frame is time-stamped with the UTC time for synchronization with IMU.

B. IMU Recording

We use Apple iPhone (model 7 and 8) as the smartphone device to collect the inertial data. To that end, we have developed an iOS application with the iOS Core Motion Framework to obtain the linear acceleration and angular velocity data from the onboard accelerometer and gyroscope. For linear acceleration, we use the processed data by the device that only reflects the user-generated acceleration after removing the gravity. The IMU data is recorded at 100Hz with UTC timestamps. At each time of the recording, we ask the volunteers to start the iOS application on their smartphones so that the data can be saved to the device. As the data synchronization is handled by matching the timestamp, volunteers do not need to start the application exactly at the same time.

C. Data Pre-Processing

As optical flow is needed to obtain the visual embedding, we pre-compute the flow for all videos so that the online training can be faster. However, computing optical flow on the raw images with a resolution of 1920×1080 is very expensive, we thus downsample the raw images to a resolution of 691×389 to speed up the pre-processing step. Also, as our network can only process a short video segment at a time, we convert the raw video and inertial data into short segments using a sliding window approach. Specifically, we use a window size of 150 frames (the best experimental setting) and a step size of 20 frames, which results in over $3,000$ synchronized video and inertial data segments.

As we have the inertial data for all persons in each data segment, we can iteratively mark each person in the data segment as the target person. This means that each video segment can serve as m data segment samples during training and evaluation where m equals to the number of persons in the video. This data augmentation technique further increases the number of our data segment samples about four times. Additionally, As our proposed method relies on the motion feature matching between the inertial and visual modalities, it is difficult to perform the matching if the target person has nearly no motion. As a result, we filter out data segment samples (about 25% of the data) during training and evaluation where the target person's IMU acceleration magnitude has a standard deviation less than $0.02m/s^2$. In the future, we will deal with this limitation with additional features that are more sensitive to small motions and achieve person localization even the target person has no motion.

V. EXPERIMENTS

A. Evaluation Details

Since our visual-inertial person localization is formalized as a matching problem, we use the classification rate as our evaluation metric, namely the probability that our method can output a correct match for the target person. We split our collected data into train (35 videos), validation (2 videos) and test set (4 videos), where each set contains videos with different number of people. Note that some challenging videos are not used, *e.g.*, videos with moving cameras, as our current method is limited to deal with static video. The evaluation of our method and baselines is only conducted on the test set, while the validation set is used for parameter tuning. Usually, when there are more people in the scene, it is more likely that people will have similar motion (*e.g.*, walking in the same direction), which makes the data more difficult for matching and localization. Naturally, we expect that our visual-inertial person localization task to become harder when there are more people in the scene. We show quantitative results with $\{2, 3, 4, 5\}$ people in the scene.

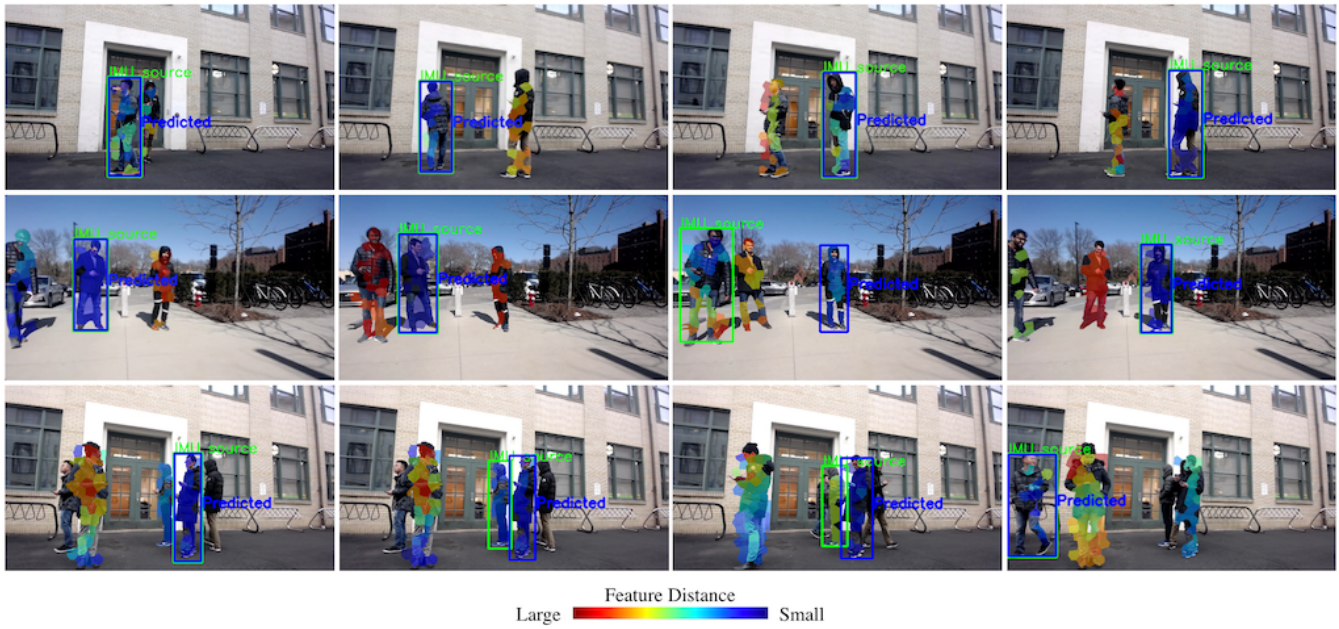

Fig. 4. We show qualitative results of our method for visual-inertial person localization on three test videos with different number of people in the scene. The green box indicated as the IMU source is the target person while the blue box is the predicted target person by our method. When the green and blue boxes fall on a same person, it is a correct match. We show both successful and failure cases in the results. Also, we visualize the distance of the visual feature for each TSP to the inertial feature of the true target person.

B. Comparison to Baseline Methods

As there is not open-source code released for [23], we try our best to re-implement the module proposed in [23] for visual-inertial person localization. Besides [23], there is no other baseline in prior work to compare against, we thus devise several competitive baselines based on the existing techniques. For fair comparison, we use the same time window of $K = 150$ frames for our method. All the baselines are listed below (1-4 are direct feature matching using cosine distance; 5-6 are supervised learning for transforming one modality with the other being the ground-truth label. At test time, the predicted person has the minimum distance between her visual feature and the query inertial feature):

1) **Velocity Magnitude.** For the visual feature, we compute a sequence of magnitude of the optical flow for each TSP feature: $\{\sqrt{v_x^2 + v_y^2}\}_{k=1}^{K}$. For the IMU feature, we compute the 3D velocity integrated from the 3D linear acceleration: $\boldsymbol{v}_t = \boldsymbol{v}_{t-1} + \boldsymbol{a}_t \nabla t$. Then, we also compute a sequence of velocity magnitude: $\{||\boldsymbol{v}_t||_2\}_{k=1}^{K}$, which is used to match with the visual velocity magnitude.
2) **Acceleration Magnitude.** We compute a sequence of magnitude of the optical flow gradients for each TSP feature: $\{\sqrt{a_x^2 + a_y^2}\}_{k=1}^{K}$, representing visual acceleration magnitude. For the IMU feature, we compute a sequence of magnitude of the linear acceleration: $\{||\boldsymbol{a}_t||_2\}_{k=1}^{K}$. Then, we match two computed acceleration magnitudes for visual-inertial person localization.
3) **Velocity Magnitude Histogram.** We first use the same method as 1) to compute the velocity magnitude. Then, the sequence of velocity magnitude is binned to create a velocity magnitude histogram, where we use 150 bins.
4) **Acceleration Magnitude Histogram.** We first use the same method as 2) to compute the acceleration magnitude. Then, the sequence of acceleration magnitude is binned to create a histogram, where we use 150 bins.
5) **3D Orientation.** We re-implement the image-based 3D orientation estimation technique in [23] where the person's 3D orientation is predicted from a VGG16 network with RGB image input. Following [23], we add two fully connected layers to the VGG16 backbone and learn the mapping from image to the person's 3D orientation. The network employs two adjacent images of a person in the tracklet as the input and regresses the 3D orientation change. We train the network using the angular velocity obtained from the IMU as ground truth. At test time, we can obtain a sequence of 3D orientation change for the person from the image, in order to match with the 3D angular velocity obtained from the inertial data.
6) **2D Optical Flow.** For the visual feature, we use the optical flow for each TSP feature. Meanwhile, we learn to map a sequence of 3D acceleration $\boldsymbol{a}$ and 3D angular velocity $\boldsymbol{w}$ to a sequence of velocity in the 2D space. The mapping function is learned by supervised learning where the input is $(\boldsymbol{a}, \boldsymbol{\omega})$ and the ground truth is the 2D optical flow. We use the same conv-1D and LSTM-IMU network in our method as the mapping function. At test time, we match the 2D optical flow from the visual feature with the estimated 2D velocity from the inertial feature.

We show quantitative comparison of our method and above baselines in Table II. We can see that baseline methods 1 to 4 with hand-designed features often perform poorly as the motion features from the visual and inertial modalities are in different feature spaces, and it is challenging to directly

TABLE II

QUANTITATIVE COMPARISON OF ACCURACY ON TEST VIDEOS WITH DIFFERENT NUMBER OF PEOPLE.

Method	N=2	N=3	N=4	N=5
Random Guess	0.500	0.333	0.250	0.200
1) Velocity Magnitude	0.500	0.379	0.429	0.456
2) Velocity Mag. Histogram	0.500	0.379	0.464	0.474
3) Acceleration Magnitude	0.500	0.379	0.536	0.456
4) Accel. Mag. Histogram	0.500	0.379	0.429	0.456
5) 3D Orientation [23]	0.502	0.344	0.306	0.194
6) 2D Optical Flow	0.682	0.402	0.392	0.439
Ours	**0.906**	**0.840**	**0.667**	**0.816**

TABLE III

PERFORMANCE OF OUR METHOD ON TEST VIDEOS WITH RESPECT TO DIFFERENT WINDOW LENGTHS.

Window Length / frames	N=2	N=3	N=4	N=5
100	0.820	0.747	0.643	0.456
150 (**Ours**)	**0.906**	**0.840**	**0.667**	**0.816**
180	0.667	0.447	0.333	0.429
200	0.556	0.631	0.605	0.480

match them. Also, learning to transform one modality to the other (*i.e.*, baseline methods 5 and 6) does not achieve superior performance. This proves again the significant gap between the two modalities. We show that, only when we transform the features from both two modalities into a joint feature space in our method, significant improvement can be achieved across videos with different number of people in the scene. Moreover, we noticed that our method performs better when N=5 than N=4, which is counter-intuitive. To find out the reasons, we visualized the results and found that the test videos with 5 people happen to have more distinct motions among the candidates compared to the test videos with 4 people, which proves that motion diversity in the data is a key factor to our method's performance.

Additionally, we show qualitative results of our method on the test set in Fig. 4. The results show that our method can predict a correct match in most of the frames, while in the failure cases the true target is often confused with the false predicted target with similar motion (best viewed in video).

C. Ablation Study: Length of the Time Window

As more discriminative motion feature can be found in a longer time window, we believe the length of time window is an important factor to the performance of our method and run ablation experiments with respect to it. Specifically, we run experiments with a window length of $100, 150, 180, 200$ frames (*i.e.*, $3.33, 5, 6, 6.67$ seconds). We use the same step size of 20 frames (0.67 seconds) as the sliding time window for all experiments. We found that the highest accuracy is achieved with a window length of 150 frames. Also, we observed a performance drop when the window length goes beyond 150 frames. It turns out that when the window length increases beyond 150 frames, the number of data samples drops significantly as most of the person trajectories in our dataset are short due to heavy occlusion by other persons. As a result, due to limited data samples, training process of our network becomes unstable and evaluation is not trustable. Additionally, a longer time window means a larger latency of our method. Therefore, we did not further investigate longer time window but use the window of 150 frames in our model.

TABLE IV

PERFORMANCE OF OUR METHOD WITH RESPECT TO DIFFERENT VARIATIONS OF THE INERTIAL FEATURE REPRESENTATION.

Inertial Feature Representation	N=2	N=3	N=4	N=5
$(\hat{\mathbf{v}}, \mathbf{a}, \boldsymbol{\omega})$	0.680	0.793	0.357	0.509
$(\mathbf{a}, \boldsymbol{\omega})$	0.820	0.747	0.643	0.403
$(\hat{\mathbf{v}}, \boldsymbol{\omega})$	0.600	0.632	0.321	0.491
$(\mathbf{a}_{\text{LPF}}, \boldsymbol{\omega}_{\text{LPF}})$ (**Ours**)	**0.906**	**0.840**	**0.667**	**0.816**

TABLE V

PERFORMANCE OF OUR METHOD WITH RESPECT TO THE LOSS WEIGHTS ON THE KEYPOINT AND BOUNDING BOX SIZE FEATURES.

Loss Weight α	0.0	0.2	0.5	0.8	1.0
N=2	0.875	0.813	**0.906**	**0.906**	0.750
N=3	0.671	0.780	**0.840**	0.758	0.597
Loss Weight β	0.0	0.2	0.5	0.8	1.0
N=2	0.700	**0.906**	0.760	0.860	0.667
N=3	0.563	**0.840**	0.598	0.701	0.632

D. Ablation Study: Inertial Feature Representation

The use of a different feature representation can result in significant differences in performance. Here, we first investigate different variations of the inertial feature representation. In addition to the linear acceleration and angular velocity, we believe the linear velocity might be also an informative feature for matching with the visual motion feature. To that end, we integrate the linear acceleration from the IMU to estimate the linear velocity $\hat{\mathbf{v}} = [\hat{v}_x, \hat{v}_y, \hat{v}_z]$ as an additional 3D motion information. As we ask the volunteers to stand still at the beginning of each video recording and then to start moving freely, we can use an initial velocity of 0 for the integration. Results in Table IV first row ($\hat{\mathbf{v}}$, $\mathbf{a}$, $\boldsymbol{\omega}$) show that concatenating the estimated linear velocity with the linear acceleration and angular velocity unfortunately performs slightly worse than without adding the linear velocity as shown in the second row of Table IV. Also, we experiment a variant that concatenates the estimated linear velocity and angular velocity in the third row of Table IV, which has a even lower performance than both the first and second row. These results demonstrate that the estimated linear velocity through integration might not be accurate enough due to the error accumulation from the IMU drift and thus we do not use the linear velocity in our final model.

Additionally, as the inertial data obtained from the IMU sensor often has high-frequency noise, we experiment the effect of a low-pass filter to our method. Specifically, we apply the filter to both the linear acceleration and angular velocity and obtain a smoother version of the inertial features $(\mathbf{a}_{\text{LPF}}, \boldsymbol{\omega}_{\text{LPF}})$, which turns out improving overall performance by 11.5% across settings with different number of people.

E. Ablation Study: Visual Feature Representation

To verify whether adding the relative positions of person's shoulder keypoints and the bounding box size to the visual

feature is useful in our model, we run experiments with different values of the hyper-parameters α and β, which controls how much we use the information of the shoulder keypoints and bounding box size during training. For example, when α or β are 0, we turn off the branch for learning shoulder keypoint and bounding box size features. From the results in Table V, we observed that the shoulder keypoint and bounding box size features are indeed useful with proper weights and improve the performance of our method on test videos with different number of people.

VI. CONCLUSIONS

We explored the possibility of using the inertial data to localize the target person in the video, in the case where we do not have access to the target person's appearance information in advance. We term this proposed task as the visual-inertial person localization. To solve this task, we first collected a new large visual-inertial dataset, which is significantly different from existing datasets in that our new dataset contains multiple people in the wild and does not have strict constraint on the attached location of the inertial sensor. Additionally, we proposed an effective approach that learns a transformer and maps the visual and inertial features into a joint feature space for matching. Through extensive experiments, we showed effectiveness of each component of our method and demonstrated that the proposed method outperforms competitive baselines on our challenging dataset.

ACKNOWLEDGMENT

This work is sponsored in part by Highmark.

REFERENCES

[1] S. Kayukawa, K. Higuchi, J. Guerreiro, S. Morishima, Y. Sato, K. Kitani, and C. Asakawa, "BBeep: A Sonic Collision Avoidance System for Blind Travellers and Nearby Pedestrians," *CHI*, 2019.

[2] A. Manglik, X. Weng, E. Ohn-bar, and K. M. Kitani, "Forecasting Time-to-Collision from Monocular Video: Feasibility, Dataset, and Challenges," *IROS*, 2019.

[3] J. Guerreiro, D. Sato, S. Asakawa, H. Dong, K. M. Kitani, and C. Asakawa, "CaBot: Designing and Evaluating an Autonomous Navigation Robot for Blind People," *ASSETS*, 2019.

[4] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," *ICIP*, 2017.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," *ICIP*, 2016.

[6] X. Weng, J. Wang, D. Held, and K. Kitani, "3D Multi-Object Tracking: A Baseline and New Evaluation Metrics," *IROS*, 2020.

[7] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep Reinforcement Learning for Autonomous Driving: A Survey," *arXiv:2002.00444*, 2020.

[8] S. Wang, D. Jia, and X. Weng, "Deep Reinforcement Learning for Autonomous Driving," *arXiv:1811.11329*, 2018.

[9] X. Weng, Y. Yuan, and K. Kitani, "Joint 3D Tracking and Forecasting with Graph Neural Network and Diversity Sampling," *arXiv:2003.07847*, 2020.

[10] F. Leon and M. Gavrilescu, "A Review of Tracking, Prediction and Decision Making Methods for Autonomous Driving," *arXiv:1909.07707*, 2019.

[11] Y. Man, X. Weng, X. Li, and K. Kitani, "GroundNet: Monocular Ground Plane Normal Estimation with Geometric Consistency," *ACMMM*, 2019.

[12] W. Luo, B. Yang, and R. Urtasun, "Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net," *CVPR*, 2018.

[13] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-End Interpretable Neural Motion Planner," *CVPR*, 2019.

[14] X. Weng, J. Wang, S. Levine, K. Kitani, and N. Rhinehart, "Sequential Forecasting of 100,000 Points," *arXiv:2003.08376*, 2020.

[15] A. Rangesh and M. M. Trivedi, "No Blind Spots: Full-Surround Multi-Object Tracking for Autonomous Vehicles using Cameras & LiDARs," *IV*, 2019.

[16] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *arXiv:1906.05113*, 2019.

[17] X. Weng, Y. Wang, Y. Man, and K. Kitani, "GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with 2D-3D Multi-Feature Learning," *CVPR*, 2020.

[18] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. Paixão, F. Mutz, L. Veronese, T. Oliveira-Santos, and A. F. De Souza, "Self-Driving Cars: A Survey," *arXiv:1901.04407*, 2019.

[19] X. Weng and K. Kitani, "Monocular 3D Object Detection with Pseudo-LiDAR Point Cloud," *ICCVW*, 2019.

[20] J. Chang, D. Wei, and J. W. Fisher, III, "A Video Representation Using Temporal Superpixels," *CVPR*, 2013.

[21] S. Hochreiter and J. Urgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 1997.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," *CVPR*, 2015.

[23] R. Henschel, T. von Marcard, and B. Rosenhahn, "Simultaneous Identification and Tracking of Multiple People Using Video and IMUs," *CVPRW*, 2019.

[24] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering Accurate 3D Human Pose in the Wild Using IMUs and a Moving Camera," *ECCV*, 2018.

[25] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll, "Deep Inertial Poser: Learning to Reconstruct Human Pose from Sparse Inertial Measurements in Real Time," *SIGGRAPH Asia*, 2018.

[26] T. von Marcard, B. Rosenhahn, M. Black, and G. Pons-Moll, "Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs," *Eurographics*, 2017.

[27] F. de la Torre, J. K. Hodgins, J. Montano, and S. Valcarcel, "Detailed Human Data Acquisition of Kitchen Activities: the CMU-Multimodal Activity Database (CMU-MMAC)," *CHIW*, 2009.

[28] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse, "Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors," *BMVC*, 2017.

[29] T. Chen, S. Ding, J. Xie, Y. Yuan, W. Chen, Y. Yang, Z. Ren, and Z. Wang, "ABD-Net: Attentive but Diverse Person Re-Identification," *ICCV*, 2019.

[30] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-Scale Feature Learning for Person Re-Identification," *ICCV*, 2019.

[31] Y.-J. Li, Z. Luo, X. Weng, and K. M. Kitani, "Learning Shape Representations for Clothing Variations in Person Re-Identification," *arXiv:2003.07340*, 2020.

[32] J. Li, J. Wang, Q. Tian, W. Gao, and S. Zhang, "Global-Local Temporal Representations For Video Person Re-Identification," *ICCV*, 2019.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *NIPS*, 2015.

[34] N. Lee, X. Weng, V. N. Boddeti, Y. Zhang, F. Beainy, K. Kitani, and T. Kanade, "Visual Compiler: Synthesizing a Scene-Specific Pedestrian Detector and Pose Estimator," *arXiv:1612.05234*, 2016.

[35] X. Weng, S. Wu, F. Beainy, and K. Kitani, "Rotational Rectification Network: Enabling Pedestrian Detection for Mobile Vision," *WACV*, 2018.

[36] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila, "The EuroCity Persons Dataset: A Novel Benchmark for Object Detection," *TPAMI*, 2019.

[37] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767*, 2018.

[38] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose Flow: Efficient Online Pose Tracking," *BMVC*, 2018.

[39] E. Hoffer and N. Ailon, "Deep Metric Learning Using Triplet Network," *International Workshop on Similarity-Based Pattern Recognition*, 2015.

[40] A. Hermans, L. Beyer, and B. Leibe, "In Defense of the Triplet Loss for Person Re-Identification," *arXiv:1703.07737*, 2017.