# Visual-GPS: Ego-Downward and Ambient Video based Person Location Association

Liang Yang [1,4], Hao Jiang [2], Zhouyuan Huo[3], Jizhong Xiao [1,4]
[1] Robotics Lab, The City College of New York, City University, New York, USA
[2] Microsoft, Redmond, USA. [3] University of Pittsburgh, Pittsburgh, USA
[4] State Key Laboratory of Robotics, University of Chinese Academy of Sciences, China
lyang1,jxiao@ccny.cuny.edu, jiang.hao@microsoft.com,zhouyuan.huo@pitt.edu

## Abstract

*In a crowded and cluttered environment, identifying a particular person is a challenging problem. Current identification approaches are not able to handle the dynamic environment. In this paper, we tackle the problem of identifying and tracking a person of interest in the crowded environment using egocentric and third person view videos. We propose a novel method (Visual-GPS) to identify, track, and localize the person, who is capturing the egocentric video, using joint analysis of imagery from both videos. The output of our method is the bounding box of the target person detected in each frame of the third person view and the 3D metric trajectory. At glance, the views of the two cameras are quite different. This paper illustrates an insight into how they are correlated. Our proposed method uses several difference clues. In addition to using RGB images, we take advantage of both the body motion and action features to correlate the two views. We can track and localize the person by finding the most "correlated" individual in the third view. Furthermore, the target person's 3D trajectory is recovered based on the mapping of the 2d−3D body joints. Our experiment confirms the effectiveness of ETVIT network and shows* $18.32\%$ *improvement in detection accuracy against the baseline methods.*

## 1. Introduction

In recent years, wearable cameras are everywhere, e.g. on smartphones, portable cameras, and AR/VR devices. Egocentric video from these wearable cameras is a popular way to record a person's sports or daily activities. Meanwhile, the third view cameras, e.g surveillance cameras, have been deployed to many places and can be used for person localization and tracking. It is important to combine those two data sources for cross-view analysis. In this paper, we use the egocentric video to find the person in the third view for person tracking in a highly dynamic and large crowded environment. It is helpful to indoor assistive navigation [35, 17, 16] for visually impaired people, and urban street navigation for shopping and touring purpose [1].

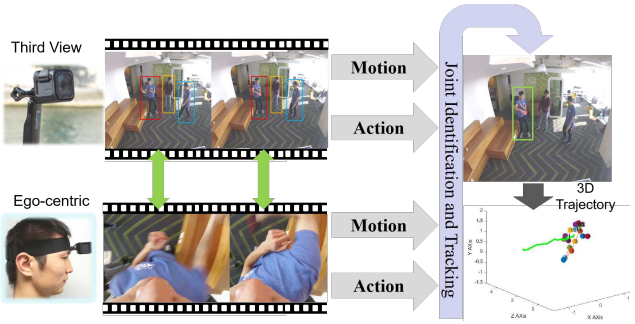



Figure 1. **Visual-GPS system**. Given an egocentric video and a third view video, visual-GPS system identifies and tracks the egocentric camera mounted person in the third view camera using a motion and action based model. Furthermore, the system is able to recover the 3D metric trajectory of the person.

In this paper, we propose a joint view approach as illustrated in Fig.1 to address this problem. We assume that the person wears a head-mount camera with a downward observation angle. We aim to not only identify and track the ego-camera mounted person in the third view, but also recover the 3D metric trajectory of the person.

There are some existing works on third and egocentric view joint identification. All of these approaches, however, use only color images. They use two-stream Siamese or triplet network architecture [13, 28, 5] to learn the correlation between third and ego views. In these models, a 3D convolutional neural network [31, 33, 23] and a segmental consensus for cross-domain verification [33, 13, 28] are commonly used. However, the forward view which only offers pure appearance features is not capable to model the association across views, especially when the illumi-

nation and the environment are dynamic. Thus, using an appearance-based siamese or triplet model to correlate these two views with temporal and spatial features would fail [13]. Moreover, the graph solution using relative view insight is not applicable under this situation [5].

Different from [5, 13], our method introduces both the motion and action factors, which are invariant to the environment appearance and light changes. We also argue that the clothes texture should not be used as a single feature for person identification in a large crowd because people may wear similar clothing. In order to train and validate our method, we designed 5 scenarios for data collection and collected 40 pairs of videos with more than 20 candidates. For each pair of an egocentric video and a third-person view video, a person is required to wear a GoPro camera with video recording and the third view GoPro camera is used to record the entire environment simultaneously.

In this paper, we build a visual-GPS system to detect and track a person in the third view camera. We formulate the problem as finding a person in the third view video whose motion and action are the most "correlated" with the egocentric view video. We construct a DNN (i.e. ETVIT) that learns the action and motion features for joint identification. Using motion features enables our method to be robust against the background and lighting changes. The ETVIT consists of four components i.e. ego action model (2D DNN), ego-motion model (2D CNN), third action model (3D DNN), and third motion model (2D DNN). We recover the 3D trajectory of the target person using a 3D-2D projection to estimate the real-time position and orientation of the person.

We make the following contributions in this paper: 1) We proposed a novel action and motion-based person identification and tracking model for cross views in Section.4; 2) We verified the mono 3D trajectory recover solution using 3D-2D pose estimation; 3) We build a comprehensive data-set and validate our proposed method in Section.5.

## 2. Related Works

**Egocentric and Third View Joint Modeling** - The problem of associating first (mobile) and third (static) view was firstly discussed in [3] to improve the object detection in the third view. [29] addressed the problem of using the egocentric and third view camera for action recognition. In [4], temporal and spatial graph matching is proposed to correlate the video from the first view and third view. In [13], persons are localized in the third view given both the third and ego camera frames using patial-domain semi-siamese, motion-domain semi-siamese, dual-domain semi-siamese, and dual-domain semi-triplet networks. Besides correlation method discussion, in [28] "Charades-Ego Dataset" of daily human activity is collecyed and the baseline of performing basic across-view frame-to-frame association is studied. These works mainly consider context features as the main clue, and they did not consider pose features and motion (odometry) feature. Besides, our work differs from other works that we perform an association of downward view and third static view, which could help to increase the robustness of tracking.

**Temporal and Spatial Model for action Learning** - Temporal information was first introduced to solve action recognition in [31], where a $3D$ convolutional operation with $3D$ max-pooling were first discussed which greatly improved the performance of learning temporal features. Then, a ResNet [14] based 3D convolutional neural network is proposed in [23] to achieve higher accuracy using a smaller model. Spatial information is commonly used in detection and correlation [18]. To match egocentric and third view, [4] uses a naive concatenation approach. $3D$ convolutional approach [13] has also been used. However, none of the above method learn the pose information in time or space. Current success in human pose detection [11] enables the learning of action in a graph convolution manner [36] in both temporal and spatial domain.

**Learning for Localization** - RGB-D method [27] is the first widely used localization approach. The first learning approach toward end-to-end localization is proposed in [20]. In order to address the sequence continuous constraints, authors in [12] proposed recurrent network to enable smooth localization. [32] demonstrated how to incorporate visual odometry prediction and global localization to relieve the requirement of a huge dataset while achieving higher localization accuracy at the same time. Recently, a similar approach [10] using both pose loss and velocity loss is proposed to increase the convergence of the model. Tracking is a traditional topic in both computer vision and robotic area [34], and later learning approach has been successfully demonstrated with real-time performance [15].

## 3. Visual-GPS:

In this section, we propose Visual-GPS for cross-view person identification and tracking in a crowded and dynamic environment. We first present an overview of the proposed Visual-GPS system and then discuss the approach of each part, especially the ETVIT network. Finally, we illustrate the recovery of the 3D trajectory of the person.

### 3.1. Approach Overview

Figure 1 depicts the framework of the proposed Visual-GPS system and figure 2 illustrates the cross-view identification and tracking network (ETVIT). The key idea of the Visual-GPS is to identify a person in the third view who is capturing the egocentric video using motion and action factors. Visual-GPS consists of three main steps. **First**, we track every person appear in the third view with the bounding box, person ID, and frame index output. Given the third

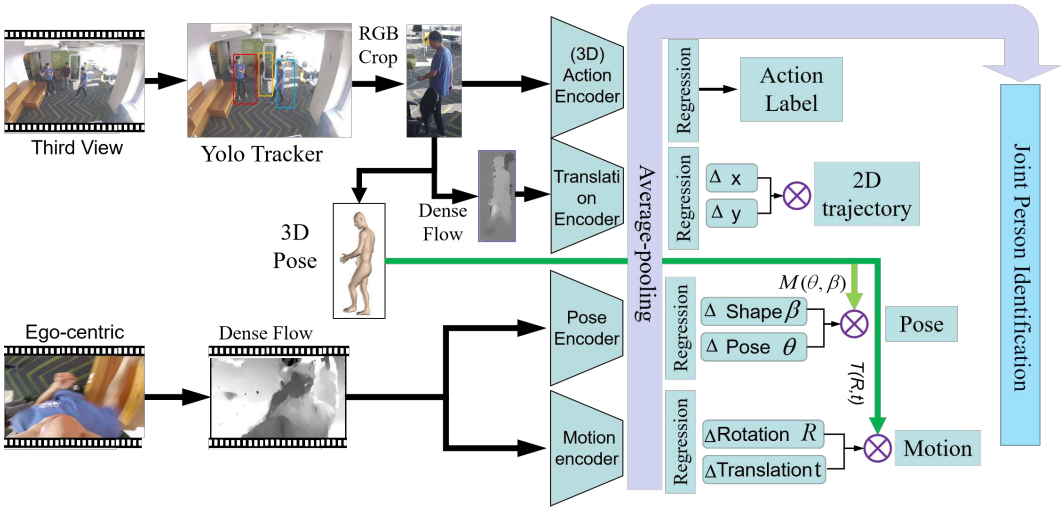



Figure 2. Pipe line of our Ego-downward and third view identification and tracking (ETVIT) model. The model is learning in a joint approach of ego-downward view and third view using motion and action feature.

and ego videos, we track every person in the third view video using a DNN tracker [25, 9] to obtain the frame ID and the corresponding person bounding box pair. Then, we crop each person out in the third view RGB image and the dense flow image based on the bounding box.

**Second**, a four channel DNN (ETVIT), which incorporates the ego and third view motion and action feature, is used for cross-view identification. For the third view, it takes an RGB clip and a flow clip as input to learn action and motion separately using two DNNs. The action is learned based on the SMPL model using a 3D DNN model [23]. For the egocentric view, we predict the 3D body pose in the third view using SMPL model [19] to initialize the pose and motion of egocentric view (see Section.4 for details). Then, it predicts the relative ego-view motion and action between consecutive frames with dense flow images as input. Finally, the ETVIT regress the target person by concatenating the intermediate features from the four DNNs.

**Third**, we keep tracking of each person using a Bayesian filter considering motion constraints, and we recover the person 3D metric trajectory based on pinhole camera model based on the 3D-2D projection using the SMPL body mesh.

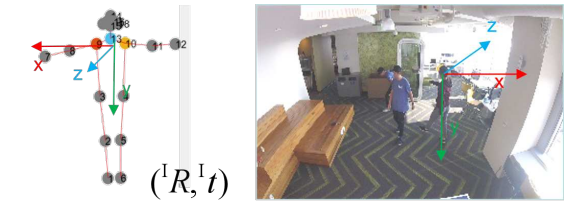
### 3.2. Front-End Tracking and Initialization

The person detection and tracking module is built upon prior works in person detection [25, 9], which predict the bounding box of a person as well the frame ID appeared across the time. In this paper, we use a Yolo-V3 [25] with Kalman motion filter to track a person with 3 second time tolerance. Then, we crop out each person in the third view as ${}^{cr}Y$ in RGB image and ${}^{cr}Y^{fl}$ in optical flow image. The cropped RGB images are directly used to predict the 3D body $\mathfrak{p}$ (i.e. 19 body joints) of each person [19]. There is one issue that the egocentric view does not have a sense of the global coordinate system (the third view). To solve this, we introduce a third view 3D body pose initialization method to transform the egocentric pose and motion to the world coordinate system.

**Initialize Egocentric Pose:** In this paper, We use a Skinned Multi-Person Linear (SMPL) [21] model to model the 3D body. We input a 2D image into a system that predicts the 3D 19 body joints which is similar to [19] to model a person. SMPL model factors the human body into shape $\beta$ - how individuals vary in height, weight, body proportion and poses $\theta$ - the 3D surface deforms with articulation.It forms a 3D mesh which is continuous quad structure, and represented as $M(\beta,\theta;\Phi): R^{|\theta|\times|\beta|} \mapsto R^{3N}$.

Given the 8 third view cropped RGB images, we use the first frame to predict the 3D pose of the person, that is, $M(\beta,\theta)$. Then, we define the corresponding egocentric first pose as $M(\beta,\theta)$. The egocentric pose model only needs to estimate the relative pose variance (i.e. $\Delta P_{smpl} = (\Delta\beta,\Delta\theta)$) for action learning.

**Initialize Egocentric Motion:** Ego-downward motion is highly related to the initial pose in the third view. Because, the same motion with different initialization would be totally different (in Section.4.1). In this paper, we define the ego-downward view coordinate system represented by joints $9: Rightshoulder, 10: Leftshoulder, 13: Neck$ as illustrated in Fig.3, where $x$ axis points from left shoulder to right shoulder, $z$ points out and perpendicular to the chest, and $y$ points downward which is perpendicular to $x$ and $z$ axis. Given 3D human body joints $\mathfrak{p} = \{(x_i, y_i, z_i)|i = 1, 2, ..., 19\}$, we can follow the above definition to obtain the initial transformation of the person, that is, ${}^{third}T_{init} = (\vec{r_p}, \mathfrak{p}_{center})$.




(a) Body cooridnate sytem (b) World cooridnate sytem

Figure 3. For Visual-GPS, we defined the world coordinate system as indicated in (b), where $x$ to the right, downward is $y$, and $z$ is the direction pointing to the inside of the image. For the egocentric view, the coordinate system is controlled by joint $9, 10, 13$, where the origin is the center node of the three and $z$ is upward and perpendicular to the surface formed by the three joints.

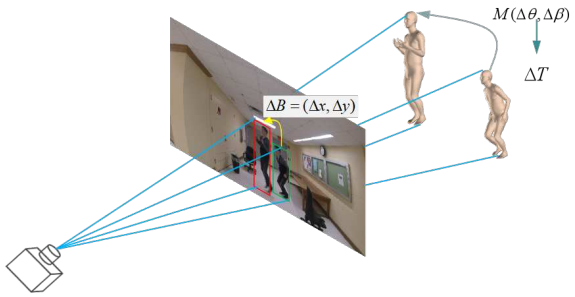



Figure 4. We can detect the 2D translation in the third-view image represented by $\Delta$Ḅ as in (a). Meanwhile, the third view human 3D pose $M(\theta, \beta)$ can also be used to obtain transformation $T$ and mapped to the 2D image.

### 3.3. ETVIT Person Identification

One of the main goal of this paper is to provide a robust person identification and tracking system even in a dynamic and crowded environment, especially regardless of the changes in the environment. Given the cropped third view RGB and flow images, and the egocentric view flow images, we use four individual encoders and regressors to learn the motion and action features. The overall network pipeline is outlined in Fig.2.

The third view has a motion encoder and an action encoder to learn the 2D translation and action feature for joint identification. For the action encoder, it is a 3D network inspired by P3D network [23] and we adopt using 18 layers. It takes consecutive 8 RGB images as input and predicts the action label (the label is generated using K-means as discussed in Section.4.1). The motion model predicts relative translation in the 2D image frame taking flow image as input, that is, it predicts $(\Delta x, \Delta y)$. Thus, we can predict the trajectory of the person as discussed in Section.4.1.

For the egocentric view, we also have a motion encoder and an action encoder to learn the motion and action. Since we have the third view to initialize the motion (i.e. $T_{init} = ({}^{T}R, {}^{T}t)$) and the action $M(\beta, \theta)$, the egocentric motion model and action model take the flow image as input and learn the relative pose variance and transformation for integration.

Finally, we concatenate the intermediate features from the four sub-models to perform regression to identify the person from cross views (shown in Fig.2).

### 3.4. Tracking and 3D Trajectory Prediction

After obtaining the identification of the person who provides the egocentric video, we introduce to use a Bayesian filter to further smooth the prediction of the target person. It follows the fact that the motion of the person should be continuous. Following [30], the tracking prediction will be updated and normalized.

In order to recover the 3D trajectory of the person, we assume that the human is with a 1.8 meters height as proposed in [19]. Through HMR [19], we have 3D joints to 2D joints mapping, thus we can predict the person position, i.e.$(x, y, z)$, in the world coordinate system using the pinhole camera model [24] given cameras intrinsic parameters $K^c$.

## 4. ETVIT Network Model

In this section, we discuss the proposed ETVIT network for cross view person identification. We first present the detail for each module, and then illustrated the regression of the overall network.

### 4.1. Learning Action Feature by Applying 3D Pose

**Learning Third View Action** To predict the action, we follow the two-step: 1)first, we use K-means to classify the action for semi-supervised learning. 2) then, we train a deep 3D-ResNet 18 to predict the action. We first predict 3D poses using HMR [19] on our training data, over 80000 images, and then we use K-means to classify every 8 consecutive pose into 400 clusters with label ${}^{trd}L = \{0, 1, 2, ..., 399\}$. For a snippet (i.e. 8 consecutive RGB cropped images, ${}^{cr}Y = \{{}^{cr}I_i | i = 0, 1, ..., 7\}$) and its corresponding 3D action cluster label ${}^{trd}L \mapsto K - means(\{{}^{trd}\mathfrak{p}_i | i = 0, 1, ..., 7\})$. Each third view clip has a dimension of $8 \times W \times H \times C$, with $C$ Channels, $W$ width, $H$ height, and 8 frames. We introduce a 3D ResNet-18 to learn the action which is inspired by [23]. 3D ResNet-18 has a total of 4 blocks. The first three blocks are with a max-pooling of $2 \times 2 \times 2$ in both spatial and temporal channels, and there is no temporal pooling with the four blocks. It also doubles the depth while the dimension decreased from 64 for the first block to 512 for the fourth block. The final output is a 512 dimensional feature vector. Finally, we construct a 3 layered fully-connected network for action regression.

**Learning Ego View Pose Variation** As discussed in

Section.3.2 that we initialize the Ego-view pose using the first frame of third view, we only need to predict the pose variance between two frames. In this paper, we propose to learn the pose variance based on flow image ($x$ and $y$ direction, respectively) using a ResNet-50. For a clip of RGB images, we have 7 flow images. The input is $W \times H \times C$ image with channel $C = 2$, width and height $W = H = 112$ as the original model. The output of the average-pooling layer is a 2048 dimensional vector. Finally, we iterative optimize shape and pose using a 3 layers fully-connected network to obtain the pose variance, $\Delta\beta = \Delta\beta + \Delta\Delta\beta$ and $\Delta\theta = \Delta\theta + \Delta\Delta\theta$, where $\Delta\Delta$ is the variation of the iterative difference.

### 4.2. Learning Motion for Correlation

**Learning Third View Translation** To learn third view motion, we introduce 2D $ResNet-50$ and followed by two fully connected layers architecture to predict the relative translation. The input are 7 consecutive third view cropped flow images ${}^{cr}Y = \{{}^{trd}I_i^{flow} | i = 1, ..., 7\}$, and the expectation is the tracked bounding box centers sequence $\mathbb{B}^{trj}$.

For each flow frame, the motion model predicts relative translation, i.e. $(\Delta x, \Delta y)$. For a flow clip, the model predicts the relative translation as $V = \{(\Delta x_i, \Delta y_i) | i = 1, 2..., 7\}$. Thus, the predicted trajectory of the video clip is, $[0, 0; \Delta x_1, \Delta y_1; ...; \sum_{i=1}^{7} \Delta x_i, \sum_{i=1}^{7} \Delta y_i]$.

**Learning ego-downward View Translation** Similar to PosNet [20], ego-translation model predicts the 6-DoF transformation between two frames. The rotation is represented using quaternion $q$, the integration of rotation uses an error quaternion [6], that is, $q_{t_{k+1}} = q_{t_{k+1}|t_k} \otimes q_{t_k}$. Where, $q_{t_{k+1}|t_k}$ is called the error quaternion,

$$
\begin{aligned}
q_{t_{k+1}|t_k} &= exp(\tfrac{\Delta\theta}{2}) \\
&= \begin{cases} \begin{bmatrix} cos(||\frac{\Delta\theta}{2}||) \\ sin(||\frac{\Delta\theta}{2}||)\frac{\Delta\theta}{||\Delta\theta||} \end{bmatrix} & ||\Delta\theta|| \neq 0 \\ [1\ 0\ 0\ 0] & ||\Delta\theta|| = 0. \end{cases}
\end{aligned} \tag{1}
$$

The model only needs to predict the error quaternion $\Delta q \in R^3$ (which is only 3 parameters) and relative translation $\Delta t \in R^3$ using 6 parameters.

### 4.3. Training and Regression Details

As we do have the label and expectation to directly supervise the training for both ego and third view action and motion, we can use these two errors as non-local losses to optimize the estimation for both views. As shown in Fig.5, we show how the four modules contribute to action and motion loss.

**Action Regression** To learn the action, the third view directly takes the clip as input for the 3D-ResNet, and the ego view predicts the pose variance and integrates the pose

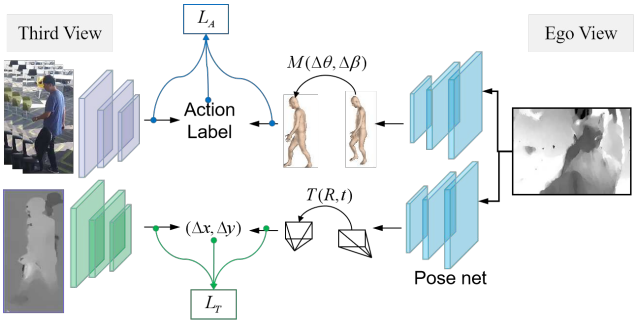



Figure 5. Our loss module consists of four parts: third view action loss, third view motion loss, ego view action loss, ego view motion loss.

for regress. We introduce cross entropy loss to optimize action learning,

$$
L(X^a) = \sum_{i=0}^{399} y_{o,i} log(P_{o,i}), \tag{2}
$$

where $y_{o,i}$ is the binary indicator if the class label $i$ is the correct prediction of current observation and $P_{o,i}$ denotes the corresponding probability.

**Motion Regression** For motion, the ground truth is the 2D trajectory that tracked by the Yolo tracker. The third view directly predicts the translation in third view image, and the ego view predicts the $error\ quaternion$ $\Delta q$ and $relative\ translation$ $\Delta t$. We define the motion as a $L_1$ norm between the expectation and the prediction,

$$
L(X^t) = ||\mathbb{B}^{trj} - t^{clip}||_{L1}, \tag{3}
$$

where $||||_{L1}$ denotes $L1$ norm, $\mathbb{B}^{trj}$ is bounding box center trajectory, $t^{clip}$ is the predicted trajectory.

## 5. Experiments

### 5.1. Dataset Collection

The dataset collection considers the following challenges: 1) same color dressing or close color; 2) background difference as context inference for verification; 3) the number of people related with accuracy; 4) similar motion situation. All the data collected are listed in Table.1, which contains a total number of 40 videos. For the training and validation purpose, we collected 30 single person ego-downward and third view videos under 5 different backgrounds. For each pair, it contains an ego-downward video and a third view static video. For all the video pairs, we generate clips which contain 8 raw images and 7 flow images as training and testing purpose. We highlight the challenge of verification if the person in the third view have the same dressing and collect extra data on this. The testing data con-


Figure 6. Represented illustration of our collected data. The training and validation data are collected under different backgrounds. For test data, we consider the same-dressing in the third view (SDT) and also a large area with over dozens of people. SDE denotes same dressing ego view.

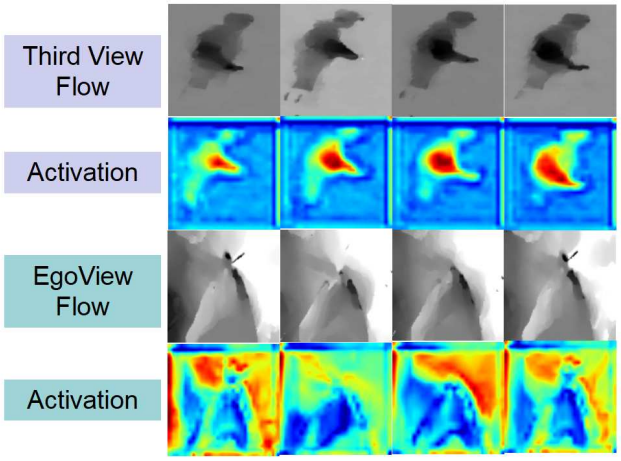



Figure 7. The motion model Block 3 activations. The colors range from blue to red, denoting low to high activations.

Table 1. A summary of collected videos in our dataset.

| | | |
|---|---|---|
| Single Person | Three backgrounds | A total 30 pair of videos containing over $100,000$ image pairs |
| Multi-person | Two Person: No Crossing | 1 pair of videos |
| | Two Person: Crossing | 1 pair of videos |
| | Three Person: No Crossing | 1 pair of videos |
| | Three Person: Crossing | 1 pair of videos |
| | Group Crossing: | 4 pair of videos |
| Same Dressing | Two Person: No Crossing | 1 pair of videos |
| | Group : Crossing | 1 pair of videos |

tains 2 to 3+ person in view cases, and the synchronization is performed using a GoPro camera remote controller.

### 5.2. Implementation Details

**Dataset Preparation** For each pair of videos, we perform the following operations which can be repeated in a step by step manner: 1) parse the videos into images; 2) Generate dense optical flow and represent in $x$ and $y$ directional separate images [8]; 3) For third view frames, first we perform person detection and tracking to obtain the bounding boxes [9] for cropping. Then 3D pose estimation of generating the 3D joints is performed for each cropped image using HMR [19]; 4) The 3D poses set of each clip is then clustered using K-means algorithm [7], with $K = 400$ in this paper. Then, we can obtain the action label for each frame. We also tried 300, and 500. It should be advised that a bigger $K$ should be more accurate for verification considering a more general application purpose.

Following the above procedures, we can obtain: 1) raw image, flow images, and action label for ego-downward view; 2) raw image, flow images, bounding box, and action label of each person, and the corresponding 3D pose indicated by 19 joints for third view (it is used to calculate the initial transformation $T = (R, t)$ for motion model). For all the 30 single person videos, we choose 24 for training and 6 for testing.

**Training Details** We choose to initialize each model using a pre-trained ResNet [14] which is trained on ImageNet-ILSVRC [26]. All the models are implemented in Pytorch [22], with a learning rate as $0.01$ and weight decay $0.001$ for 200 epochs using two Nvidia 1080 GPUS. For our network, we trained each sub-model independently. Then we perform joint optimization for final verification.

### 5.3. Results and Comparison

**Baselines** We first implement multiple baselines to compare the performance considering inputs, and mod-

Table 2. Verification accuracy (in %) baselines on our dataset, and higher is better. Where SW denotes share weight.

| | Resnet-18 | Resnet-34 | Resnet-50 | Resnet-101 |
|---|---|---|---|---|
| Siamese Image | 50.39 | 51.03 | 50.55 | 50.42 |
| Siamese Flow | 52.53 | 50.75 | 51.63 | **52.06** |
| Semi-siamese SW | 53.34 | 52.41 | 52.78 | 51.35 |
| Semi-siamese | 52.1 | 51.89 | 51.29 | 50.91 |
| Temporal-Siamese Image | 52.21 | 51.6 | 51.43 | - |
| Temporal-Siamese Flow | **54.77** | **55.9** | **55.10** | - |
| Temporal Semi-siamese | 51.74 | 53.96 | 50.89 | - |
| Triplet [28] | 52.80 | 51.28 | 51.63 | 51.49 |

els. These baseline method are proposed in peer researches [13, 28, 23] including spatial-domain siamese network [13], motion-domain siamese network[13], two-stream semi-siamese network [13], triplet network [28], and temporal domain image and flow network [13, 23]. We also demonstrate the weight share performance for siamese-network. We deploy 2D and 3D Resnets [23] to learning spatial and temporal features.

For feature, we performed the training and testing using image data and flow data in independent networks, while we also performed learning using both information in a semi-siamese approach. Table.2 summaries the accuracies of the above models. In this paper, we use accuracy as a metric to evaluate the models as [28]. It shows in the table that temporal models are significantly much better for our tracking problem, and also flow information is more accurate. Since we ask the person to move frequently and fast, thus it is harder to verify using pure context feature. The maximum accuracy according to these methods is $55.9\%$ which is 3D temporal Resnet-34 model using optical flow as input. However, the semi-temporal model does not show any improvement, which may be caused by limited data of a color feature of our dataset.

In this table, we can also see that a share weight siamese-model is more effective than the none-share models with an average $1\%$ percent higher. For Semi-siamese model, in the spatial domain, it is a four channel network takes both flow and image as input. The triplet model is implemented as proposed in paper [28], where a none-corresponding image has used as input of the model. The resulting accuracy indicates that the triplet structure can achieve similar performance compared to the temporal flow model, and it does not require a huge amount of parameter to train.

For the baseline implementation, we did not implement semi-triplet as proposed in [13] since we regard the tracking is performed in a large crowd. Thus, the semi-triplet model will have to perform exponential times due to the requirement of input. However, the above data tells: 1) flow information is more important for identification; 2) complex model may not help if simply using spatial and temporal information.

Table 3. Verification performance of proposed model. AP(%): Average Precision, and AR%: Average Recall

| Model | Accurcy % | AP | AR |
|---|---|---|---|
| Action Model | 72.50 | 68.92 | 42.32 |
| Translation Model | 70.03 | 64.38 | 38.74 |
| ETVIT Model | 74.22 | 69.78 | 47.93 |

**ETVIT Model Testing**

**1) Performance and Analysis** We also test our proposed model on a single person dataset. The results are summarized in Table.3, where we also test the action model and motion model separately. We can obtain that the proposed method outperforms the best baseline by $18.32\%$. The independent action model can achieve $72.5\%$ in accuracy and translation model can achieve a $70.03\%$.

**2) Action VS Motion Model** The result shows that Action model has a $2.47\%$ higher accuracy than Motion model, and $4.54\%$ higher average precision. It is because the motion model does not tell any difference when human is static or just move the part of the body. We also visualize the activations and the overlay to the image of the motion model as illustrated in Fig.7. It can be seen that the third view translation highly attend to the center of the flow, while, the ego-motion model attend to the outer body region for translation estimation. For action sub-model, the activations of each model the third block is Fig.8. We observe the action model attending to joints to perceive pose information both in RGB-image and flow images.

**3) Ego Odometry VS Third View Odometry** We also compare the importance of ego-view translation and third view translation. We directly introduce to add the translation as an independent channel into the temporal semi-siamese model, in a fully connected layer (Appear In appendix). The result shows that third view translation can increase the validation accuracy ($20\%$ of the training data) from $79.05\%$ to $81.80\%$. It can be explained according to Fig.7 that our ego view has a limited view of the world, also the head motion introduces error.

**4) Test On Multi-person Videos** Then, we test the proposed model in our multi-moving people cases with results illustrated in Table.4. For the ground truth, it is obtained using a tracker and manual label. It is can be seen in Table.4 that ETVIT can achieve an average accuracy $67.77\%$ for all the test cases. For society cross, the filtering fails due to much crossing happens. For implementation, we perform the prediction of all the detected person and conclude based on the maximum score.

ETVIT model has lower accuracy when the ego-camera

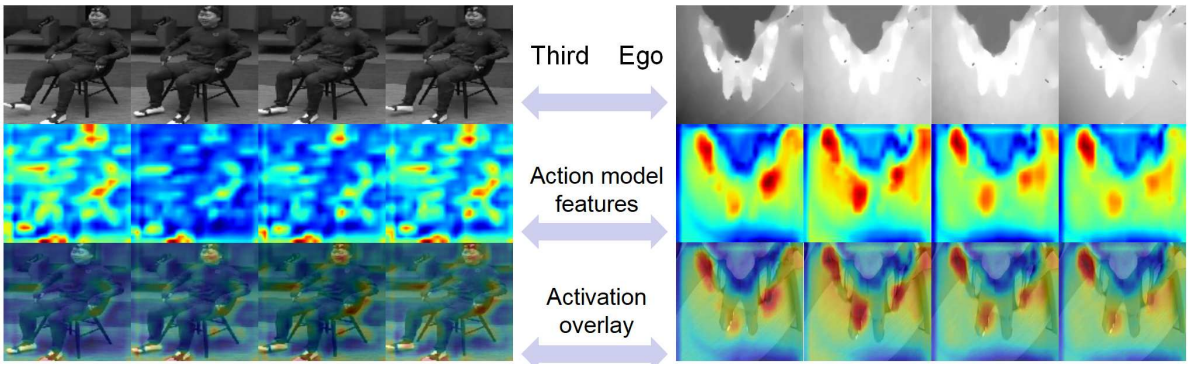



Figure 8. Block 3 activations of the action model. The colors range from blue to red, denoting low to high activations.

Table 4. The verification accuracy % on multi-people testing data.

| | Test Case | Accuracy | Bayes Filter |
|---|---|---|---|
| Multi-person | Two Person : No Crossing | 72.26 | 96.17 |
| | Two Person : Crossing | 62.18 | 80.76 |
| | Three Person : No Crossing | 72.25 | 92.27 |
| | Three Person : Crossing | 65.39 | 91.52 |
| | Group Crossing : | 57.26 | - |
| Same Dressing | Two Person : No Crossing | 72.26 | 96.17 |
| | Three Person : Crossing | 65.39 | 91.52 |

mounted person crossed with other pedestrians. It is due to partial observable of the body, the 3D pose estimation would fail. In this paper, we also introduce a Bayes filter with motion prediction to filter the verification results[2]. The filtered results are illustrated in Table.4, which shows promising if given a few people in view.

**5) Adaptivity Analysis** From all tests, we conclude that our model achieves a high identification accuracy by at least 10%. We also find several limitations of our model at the current stage. First, if all the persons are static or with a similar pose in view, our algorithm would fail. Second, if all person with the same action and motion, it also fails.

**Visual-GPS Demonstration and Comparison**

Visual-GPS 3D trajectory is shown in Fig.9. The results are competitive compared to pure visual odometry (VO). We found that VO fails tacking after 13.4 seconds, and we can see that it drifts so quick compare to our Visual-GPS. The Visual-GPS performs the trajectory predict via 2D-3D pose association and the 3D trajectory is illustrated in Fig.9(b). Besides, the start and end position is given in

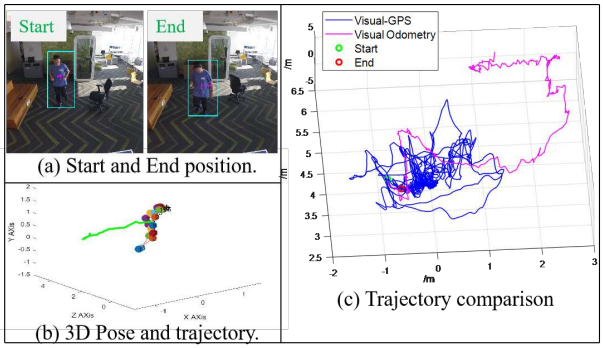



Figure 9. Visual-GPS 3D trajectory recover demonstration and comparison with visual odometry.

Fig.9(a), and we can visualize in Fig.9(c) that Visual-GPS accurately predict the 3D position of the target person.

## 6. Conclusion

We present a Visual-GPS system which incorporates a downward egocentric camera and a third view camera to identify, track, and localize a target person in the crowded and dynamic environment. The Visual-GPS proposes an action and motion learning model for cross-view identification. It is motivated by the observation that the ego view is not able to sense the third view' coordinate information. Our experimental results show that our method outperforms the state-of-art verification model on cross view verification, even with the same dressing. It delivers a competitive generalization of cross-view verification on semi-supervised learning for localization and tracking using action and motion clue.

**Acknowledgements**: This work is partially done by Liang Yang interned at Microsoft Research & AI. We thank Dr. Wei Li for help with polishing the paper, and all the anonymous participants involved in data collection.

## References


[1] Google maps ar. In *https://insights.dice.com/2018/03/19/google-opens-its-maps-api-to-augmented-reality-development/*. 1

[2] F. Ababsa and M. Mallem. Robust camera pose tracking for augmented reality using particle filtering framework. *Machine Vision and applications*, 22(1):181–195, 2011. 8

[3] A. Alahi, M. Bierlaire, and M. Kunt. Object detection and matching with mobile cameras collaborating with fixed cameras. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008*, 2008. 2

[4] S. Ardeshir and A. Borji. Ego2top: Matching viewers in egocentric and top-view videos. In *European Conference on Computer Vision*, pages 253–268. Springer, 2016. 2

[5] S. Ardeshir and A. Borji. Integrating egocentric videos in top-view surveillance videos: Joint identification and temporal alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 285–300, 2018. 1, 2

[6] L. Armesto, J. Tornero, and M. Vincze. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *The International Journal of Robotics Research*, 26(6):577–589, 2007. 5

[7] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii. Sklearn k-means. volume 5, pages 622–633. VLDB Endowment, 2012. 6

[8] S. Baker and I. Matthews. Opencv dense optical flow. volume 56, pages 221–255. Springer, 2004. 6

[9] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 3, 6

[10] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[11] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 2

[12] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2017. 2

[13] C. Fan, J. Lee, M. Xu, K. K. Singh, Y. J. Lee, D. J. Crandall, and M. S. Ryoo. Identifying first-person camera wearers in third-person videos. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4734–4742. IEEE, 2017. 1, 2, 7

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 6

[15] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765. Springer, 2016. 2

[16] F. Hu. *Vision-based Assistive Indoor Localization*. PhD thesis, The Graduate Center, City University of New York, Feb 2018. 1

[17] F. Hu, Z. Zhu, and J. Zhang. Mobile panoramic vision for assisting the blind via indexing and localization. In *Computer Vision-ECCV 2014 Workshops*, pages 600–614. Springer, 2014. 1

[18] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015. 2

[19] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 4, 6

[20] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 2, 5

[21] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 3

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 6

[23] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 1, 2, 3, 4, 7

[24] L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Transactions on pattern analysis and machine intelligence*, 21(8):774–780, 1999. 4

[25] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 3

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 6

[27] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013. 2

[28] G. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari. Actor and observer joint modeling of first and third-person videos. In *CVPR-IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 7

[29] B. Soran, A. Farhadi, and L. Shapiro. Action recognition in the presence of one egocentric and multiple static cameras. In *Asian Conference on Computer Vision*, pages 178–193. Springer, 2014. 2

[30] L. Taycher, G. Shakhnarovich, D. Demirdjian, and T. Darrell. Conditional random people: Tracking humans with crfs and grid filters. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE COMPUTER SCIENCE AND ARTIFICIAL . . . , 2005. 4

[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 1, 2

[32] A. Valada, N. Radwan, and W. Burgard. Deep auxiliary learning for visual localization and odometry. *arXiv preprint arXiv:1803.03642*, 2018. 2

[33] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016. 1

[34] J. Watada, Z. Musa, L. C. Jain, and J. Fulcher. Human tracking: A state-of-art survey. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 454–463. Springer, 2010. 2

[35] J. Xiao, S. L. Joseph, X. Zhang, B. Li, X. Li, and J. Zhang. An assistive navigation framework for the visually impaired. *IEEE transactions on human-machine systems*, 45(5):635–640, 2015. 1

[36] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018. 2