

Combining Passive Visual Cameras and Active IMU Sensors for Persistent Pedestrian Tracking

Wenchao Jiang and Zhaozheng Yin

Missouri University of Science and Technology, USA

Abstract

Vision based pedestrian tracking becomes a hard problem when long-term/heavy occlusion happens or pedestrian temporarily moves out of the visual field. In this paper, a novel persistent pedestrian tracking system is presented which combines visual signals from surveillance cameras and sensor signals from Inertial Measurement Unit (IMU) carried by pedestrians themselves. IMU tracking performs Dead Reckoning (DR) approach utilizing accelerometer, gyroscope and magnetometer. IMU tracking has nothing to do with visual occlusion, so it keeps working even when pedestrians are visually occluded. Meanwhile, visual tracking assists in calibrating IMU to avoid the bias drift during DR. The experimental results show that the IMU and visual tracking are complementary to each other and their combination performs robust pedestrian tracking in many challenging scenarios.

Keywords: Cross-media Object Tracking; Persistent Pedestrian Tracking, Visual Object Tracking, IMU Tracking, Dead Reckoning.

1. Introduction

Automatic pedestrian tracking has a wide range of applications, such as security surveillance and behavior analysis. Typically, pedestrian tracking can be classified as “passive” and “active”[1]. Passive tracking utilizes devices that are not carried by pedestrians. The typical passive tracking is based on vision such as tracking targets in video streams from color or thermal cameras. Active tracking lets pedestrians carry electronic devices and sensor signals collected from pedestrians are used to locate the pedestrians. The most common active tracking devices include Global Positioning System (GPS), WiFi receiver, base station signal receiver and Inertial

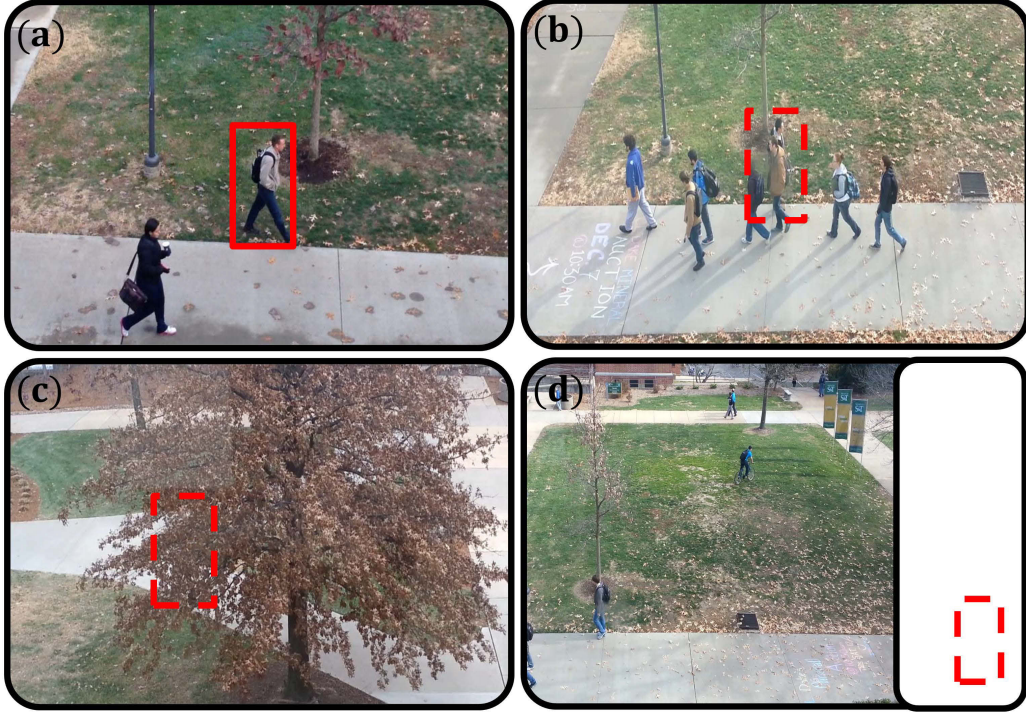


Figure 1: Visual pedestrian tracking and its challenges. (a) Successful pedestrian tracking; (b) The target pedestrian is occluded by other pedestrians; (c) The pedestrian is occluded by a tree over a long period; (d) The pedestrian moves out of the field of view temporarily.

Measurement Unit (IMU). This paper attacks the problem of persistently tracking cooperative targets (e.g., children, teens, the elderly, patients with autism/alzheimers/dementia) by combining passive and active tracking.

1.1. Related Work

1.1.1. Passive Pedestrian Tracking

Passive vision based pedestrian detection and tracking has been studied for several decades [2, 3, 4, 5, 6, 7, 8]. When there is heavy occlusion, large appearance change, nearby clutter or pedestrians temporarily moving out of the field of view as shown in Fig.1, it is still challenging for a vision based tracking algorithm to persistently track the pedestrian without any failure. Most of the previous efforts on pedestrian tracking in videos are made in the following two categories.

When the pedestrians are partially occluded, previous work mostly relies

on tracking the body parts that are visible. For example, Wu et al. [9] combine body part detectors which are learned by boosting edgelet feature based weak classifiers and whole pedestrian detectors. Merad et al. [10] classify pedestrian’s appearances into front and back poses in addition to segmenting detected individuals into several parts such as head and torso. However, part-based tracking algorithms can hardly overcome the challenges when the target is totally occluded or has disappeared for a long time.

Thus, more researchers focus on predicting the positions of occluded pedestrians. For realtime prediction, Li et al. [11] improves the traditional mean shift tracking algorithm by proposing “occlusion layers” to represent the pedestrian occlusion relation, and the non-occlusion part of the pedestrian is used for the mean shift tracking. Leykin and Riad [4] introduces a pedestrian tracker designed as a particle filter using a combined inputs from color and thermal cameras. These algorithms assume the pedestrians do not change their forwarding direction and speed when occluded, which may be not true in practice. Besides the aforementioned realtime tracking methods, Zhang et al. [12] proposes a unified algorithm that automatically learns the trajectory models from the local and global information to obtain an optimal assignment. Sherrah [13] tracks a pedestrian over short time-frames to form tracklets. Then an optimal path finding problem is posed in the generalized Hough space and can be solved using the Viterbi algorithm. Trajectory assignment depends on the similarity of pedestrians’ appearances. However, the appearance of the target may change a lot before and after the occlusion because of the illumination and viewpoint, which results in inaccurate matches.

1.1.2. Active Pedestrian Tracking

It is intuitive to track pedestrians with GPS since it is already widely used in vehicle navigation. However, the accuracy of a common GPS module is not high enough, for example, Garmin reports its GPS receivers are accurate to within 15 meters on average [14]. Compared with common moving vehicles, time delay on locating pedestrians slower than vehicles is highly possible when using GPS. When a pedestrian starts moving, the position obtained from GPS may maintain as a constant for 2~3 seconds. When the pedestrian stops, the position reported from GPS may still move for a while. Such asynchrony leads tracking failures or frequent mistakes when analyzing a pedestrian’s behaviour. Furthermore, GPS is based on the direct signal transmission between a GPS receiver and satellites. Obstructions

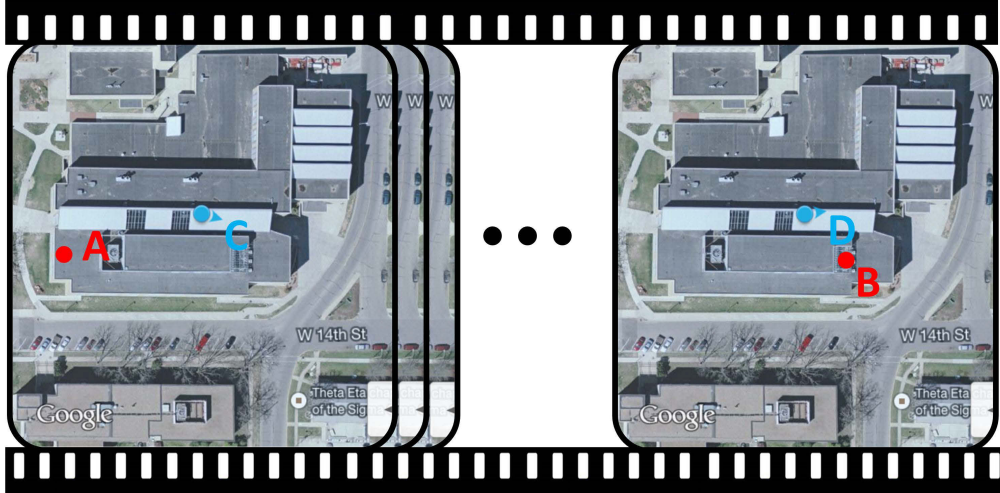


Figure 2: A video of a pedestrian’s position estimated by GPS. The pedestrian moves from red points A to B (ground truth). Blue points C and D are the positions estimated by GPS in the starting and end frames.

such as city canyons or tall trees outdoors and walls/ceilings indoors cut off or weaken the signals, making the GPS-based localization and tracking unreliable. The author performed a GPS-based pedestrian tracking using Google Map on a Samsung Galaxy s4, as shown in Fig.2, when the pedestrian walked along a straight line indoors from red points A to B (50 meters away from each other), the GPS-based trajectory is from blue points C to D which are almost stationary at the same location, deviating from the true pedestrian trajectory.

WiFi is another type of ubiquitous signal in daily lives [15, 16, 17]. However, WiFi hotspots are available mostly in indoor environments and the coverage area of most WiFi hotspots is less than 50 meters, limiting its application in pedestrian tracking outdoors. Base stations in mobile telephony are high power and can cover a radius as large as 35km, with which we can make a call and send messages. However, base station is only suitable for coarse localization and tracking [18].

Inertial Measurement Unit (IMU) is a good choice for active pedestrian tracking, which consists of Gyroscope, Magnetometer and Accelerometer [19, 20, 21]. In an ideal environment without any noise or movement, Magnetometer measures the direction of the earth’s geomagnetic north pole and

Accelerometer measures the gravity. The two signals can be used to set up a world coordinate system, which is available everywhere on the earth only except the north and south poles. So, unlike GPS that is inaccurate and unreliable indoors or WiFi that needs to artificially install WiFi hotspots, IMU-based active pedestrian tracking systems can be applied almost everywhere except the north and south poles.

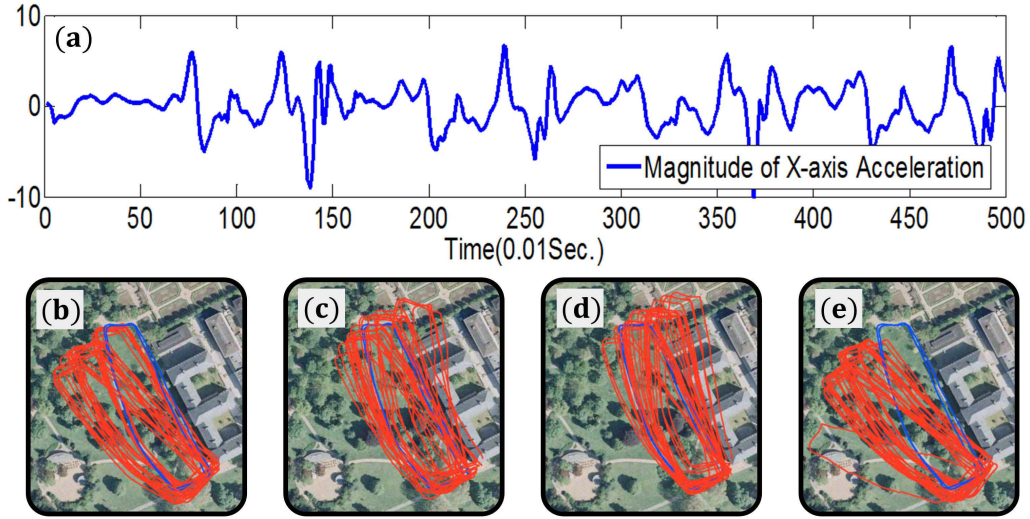


Figure 3: Challenges in IMU-based pedestrian tracking. (a) The magnitudes of the x-axis acceleration data collected in 5 seconds during a normal walking. (b)~(e) the experiment results of IMU-based pedestrian tracking by four different methods [22]. Blue curves are ground truth and red curves are tracking results.

Most of the previous IMU-based tracking methods depend on Dead Reckoning (DR) which adds the estimated current displacement vector to the previously estimated location [22, 23, 24, 25, 26]. DR is built upon three components: step detection, speed estimation and forward moving direction determination. However, all these three components have challenges for IMU-based DR: (1) **Step detection**. The human gait varies during walking, which results in irregular movement patterns. Fig.3(a) shows the magnitudes of the x-axis acceleration data collected in 5 seconds during a normal walking. The acceleration values are almost irregular, causing difficulties for step detection. Foxlin [23] solves this problem by binding IMU to the pedestrian's foot because the foot movement is more reliable for step detection compared with other body parts, but this method is not generally

applicable when the IMU is not bound to foot such as smartphones. Other step detection methods mostly rely on thresholds to detect steps [22, 24], but these thresholds are pedestrian-dependant and require the pedestrian’s speed to be stable over time. (2) **Speed estimation.** Theoretically speaking, acceleration can be used to calculate speed by integration over time. But it is unpractical because accumulated little noise leads to unacceptable large error. For example, an acceleration measurement noise of $0.1g$ sums up to 60 m/s speed error within one minute. Generally, speed is often estimated by the amplitude of acceleration [22, 24, 27]. However, the amplitude of acceleration only describes the intensity degree of movement. The length of legs also affects the speed which can not be measured by inertial sensors. (3) **Forward moving direction.** Acceleration is often used to determine the forward moving direction. However, when a pedestrian walks, the movement is a complicated combination rather than a merely forward direction. For example, if the sensor is in the lapel pocket, there is a moving direction which is perpendicular to the forward moving direction due to the swing of arms, so the moving direction calculated by acceleration is different from ground truth. Fig.3(b)~(e) show four IMU based pedestrian tracking results which are from the experiment section of [22]. Blue curves are ground truth and red curves are IMU-based tracking results over many times. The shape of the trajectories are mostly correct but the forward moving directions are inaccurate. Moreover, it is assumed in [22] that the speed is a constant which is not achievable practically.

1.2. Our Proposal

Passive visual based pedestrian tracking is intuitive and low cost but the occlusion problem is hard to solve. However, despite many limitations and challenges, almost all active pedestrian tracking methods have no problems of occlusion because they do not rely on vision. Thus, the occlusion problem of visual tracking can be compensated by active sensor tracking. In this paper, we propose a novel persistent pedestrian tracking system combining passive visual tracking and active sensor tracking. The visual signal is from stationary surveillance cameras and IMU devices are used for active tracking. IMU is chosen here because IMU can be applied almost everywhere merely except the earth’s north and south poles. No other infrastructure installations are needed other than the sensor module on a pedestrian’s body. The most significant contribution of this paper lies in the complementation of visual tracking and IMU-based active tracking. Not only can IMU as-

sist visual tracking when pedestrian is occluded, but also the challenges of IMU tracking are alleviated when visual signals are available. The drift of IMU-based speed and forward moving direction estimation can be rectified by visual tracking on videos. Thus, our persistent pedestrian tracking system solves two complementary problems simultaneously.

The persistent pedestrian tracking system can be applied to many areas. Coupled with IMU-based behaviour analysis [28, 29, 30], our system can comprehensively monitor patients' locations and daily actions in hospitals. Suspects carrying IMU's can be persistently tracked for security reasons. In addition, our system can be adopted to navigate robots in the WiFi- and GPS-denied environment. Considering the pervasiveness of smartphones which already embed with IMU modules, all our experiments are performed on the platform of a self-developed App on Android smartphones. There should be no problem to adopt professional IMU which collects more accurate signals to our system.

2. Passive Visual Tracking

Histogram of Oriented Gradient (HOG) features along with Support Vector Machine (SVM) have been popularly used to perform pedestrian detection in images [5, 6]. Visual pedestrian tracking in videos can be implemented by the classic tracking-by-detection algorithm [31]. A large dataset (both positive samples and negative samples) are usually needed to train a general pedestrian detector which is very time-consuming and the detector may not work well on scenarios different from the training dataset [32]. In common pedestrian detection, pedestrians are searched in different scales in every image to adapt to pedestrians' scale changes [33]. Considering a specific and fixed viewpoint, we propose a scene-specific pedestrian detector and an adaptive scale selection algorithm to improve the pedestrian detection performance and reduce its computational cost.

2.1. Training a Scene-specific Pedestrian Detector

In a fixed scene, the viewpoints from which pedestrians can be observed and the scales of pedestrians in images are limited. Moreover, the negative samples are limited (they are just the background in the scene!) We still use the HoG+SVM for the detector. The specific problem is how to classify those background samples that are very likely to be mistakenly classified from various pedestrian samples. If the detector can correctly classify those

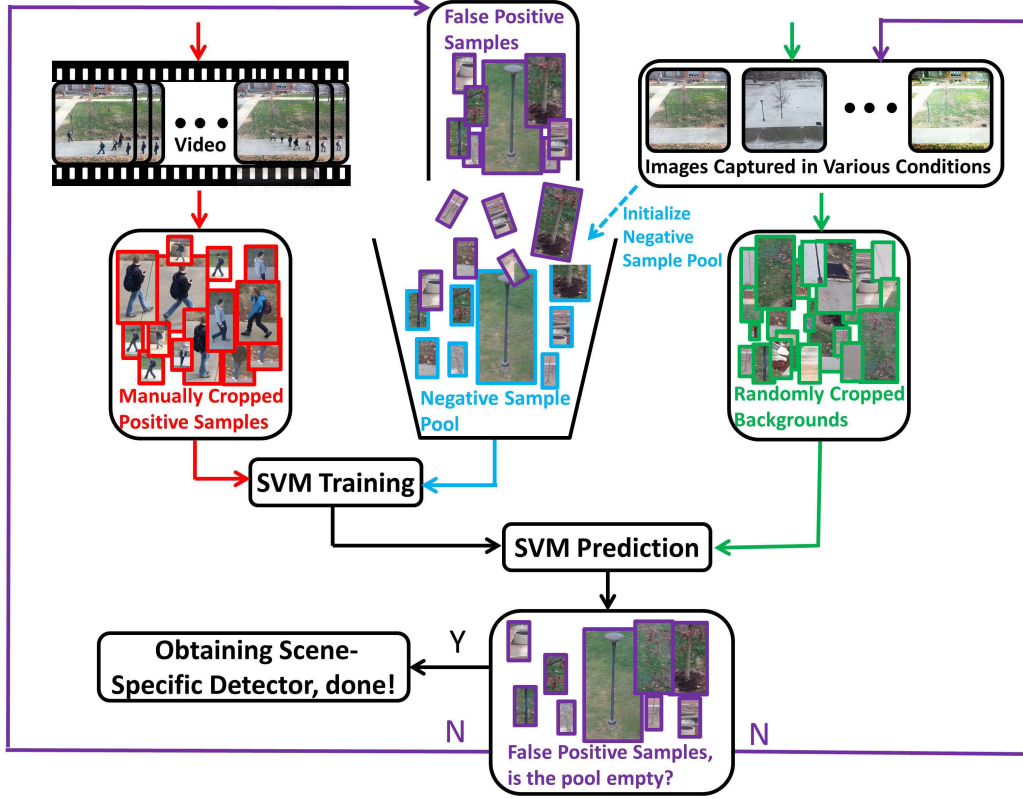


Figure 4: Flow chart of training a scene-specific pedestrian detector.

background samples whose feature descriptors are near the support vectors, it is sufficient to classify other background samples which are largely different from pedestrian samples.

In this paper, we propose a new scene-specific training algorithm to deal with this problem. As illustrated in Fig.4, the positive samples (images framed in red) are the manually cropped pedestrian images from videos taken on the specific scene. The positive samples include pedestrian images with different angles and scales that can be seen from the specific viewpoint. The pool of positive samples is not changed during the following iterative training.

The negative sample pool initially consists of randomly cropped backgrounds from images taken on the specific scene with a fixed viewpoint in different weather and illumination conditions (images framed in blue). Then, the negative sample pool expands gradually during each iteration. The it-

erative training algorithm is performed in the following steps: when a new pedestrian detector is available after SVM training, it will be applied to classify background images randomly cropped from images taken on the specific scene (images framed in green); the false positive samples (images framed in purple) are put into the negative sample pool and the SVM will be updated for the next iteration; training stops when the number of false positive samples is zero. Every time the SVM is updated, the detector is more robust to classify those samples that are misclassified previously.

2.2. Adaptive Scale Selection

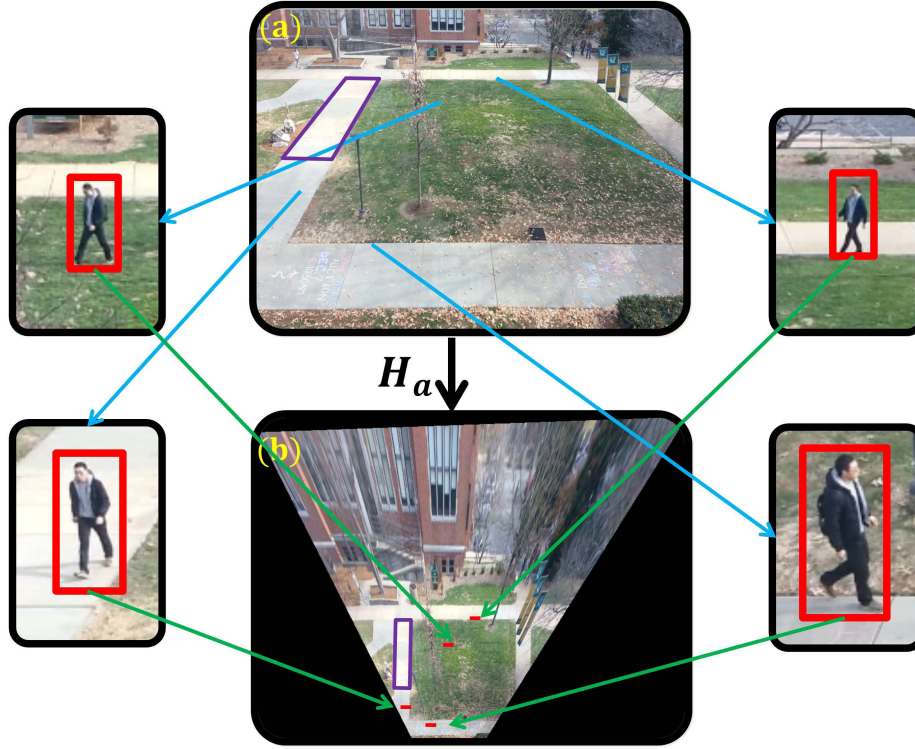


Figure 5: Adaptive scale selection.

In common pedestrian detection algorithms, for every frame, different scales defined by the height and width of rectangles need to be searched in the image exhaustively to detect all pedestrians. In a fixed scene, although the same person may display different scales at different locations in the

viewpoint. However, in the top-down viewpoint, the width of the pedestrian rectangle is a constant. Fig.5 shows the same pedestrian at four different locations in the fixed scene. After warping the original image to the top-down view, the red lines in Fig.5(b) are the warped rectangular bottoms from four detections. Although the pedestrian in four original detections are with different scales, their scales are the same in the top-down viewpoint. Thus, if we fix the ratio of the height and width of a detection rectangle and determine the standard scale S_{std} by the length of the bottom side of the warped rectangle, pedestrians' scales in every region of the specific scene can be estimated, i.e., we know which scale in the original image we should use to detect pedestrians.

We assume a homography transformation between the original image and the image with the top-down view. Let H_a be the homography transformation matrix. Four pairs of point correspondences are needed to estimate H_a . As shown in Fig.5, we choose the four corners of a rectangular region in the world as the correspondences. Assume points (x_i, y_i) are the four vertices of the trapezoid in Fig.5(a) and (x'_i, y'_i) are the corresponding vertices in Fig.5(b) where $i \in [1, 4]$. We have

$$\begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix} \sim H_a \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \quad (1)$$

where

$$H_a = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \quad (2)$$

The homogeneous coordinates can be transformed into two equations for each pair of correspondence:

$$x'_i = \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \quad (3)$$

$$y'_i = \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}}. \quad (4)$$

The degree of freedom of H_a is 8 because $\|H_a\|_2 = 1$ where $\|\cdot\|_2$ is the 2-norm. Four pairs of correspondences provide 8 linear equations for us to

solve the H_a . After we calculate H_a , the constant scale in the top-down view (Fig.5(b)) can be warped to every floor location in the original view (Fig.5(a)). Therefore, we know the suitable scale to search pedestrians at every location in the original image. H_a is constant for a fixed scene and it only needs to be updated when the viewpoint changes.

2.3. Tracking by Detection

With the pedestrian's location and scale in the previous frame $t - 1$, we apply our scene-specific pedestrian detector within a local region around the previous location to detect pedestrians in the current frame t . To estimate the pedestrian's scale in frame t , the bottom side of rectangle in frame $t - 1$ is warped to top-down viewpoint with H_a and the warped bottom side will be compared with S_{std} to decide whether the pedestrian's scale in frame t should be changed. Fig.6 shows the pedestrian images and their confidence maps corresponding to SVM scores of the scene-specific detector. The white in a confidence map denotes high score (confidence) of pedestrian detection. The pedestrian's location in frame t is determined by the position with the maximal confidence in the confidence map.

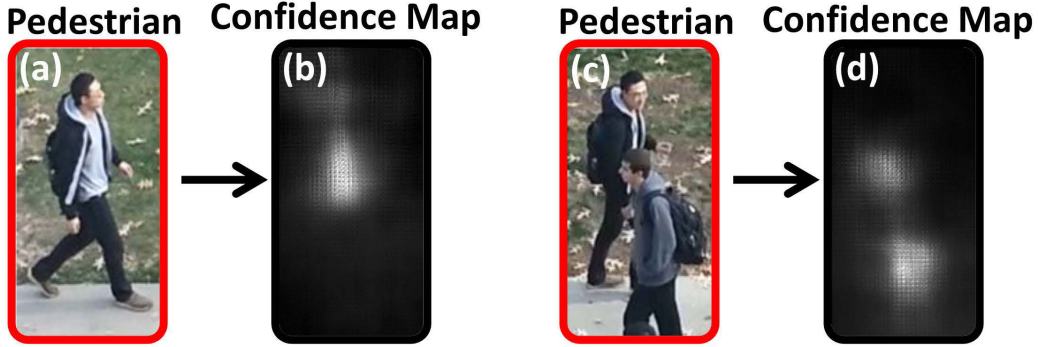


Figure 6: Confidence maps for the pedestrian tracking. (a) and (c) are the pedestrian images. (b) and (d) are their confidence maps, respectively.

If the pedestrian is not occluded (Fig.6(a)), there is a single global peak in the confidence map, thus the pedestrian is correctly tracked. However, as shown in Fig.6(c), when the target pedestrian is occluded by other pedestrians, there are multiple peaks in the corresponding confidence map. It is really possible that the non-target pedestrian is detected and tracked instead, causing drift on pedestrian tracking. Therefore, when occlusion and

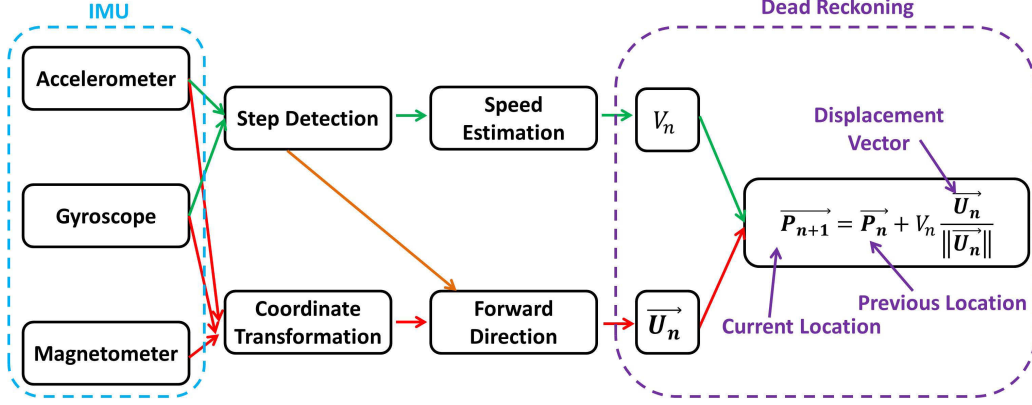


Figure 7: Flow chart of our IMU-based pedestrian tracking.

disappearance of the pedestrian happen, we refer to IMU-based tracking to correct the vision tracker and reidentify the lost pedestrian.

3. Active IMU-based Tracking

As shown in the blue part of Fig.7, IMU is made up of Accelerometer, Gyroscope and Magnetometer. Accelerometer measures tri-axis acceleration values. Magnetometer measures the strength of magnetic field. Gyroscope measures tri-axis angular velocities. The idea of our IMU tracking is based on Dead-Reckoning (DR) which adds the estimated current displacement vector, $V_n \frac{\vec{U}_n}{\|\vec{U}_n\|}$, to the previously estimated location $\vec{P}_n$. V_n and $\vec{U}_n$ are the speed and forward moving direction in step n , respectively. Our IMU tracking approach consists of four components: step detection, speed estimation, coordinate transformation and forward moving direction determination.

3.1. Step Detection

Speed requires to be estimated on a complete step period for DR, so the accurate beginning and end of each step is needed. In [22], step is detected in the time domain by finding the local maximum and minimum of filtered acceleration data and a threshold is set to rule out false positives. However, the threshold value depends on the speed and is pedestrian specific. When speed greatly changes, the rate of missed detection of step increases rapidly. Moreover, we need to set a new threshold each time when the target pedestrian changes. Therefore, a constant threshold poorly deals with speed

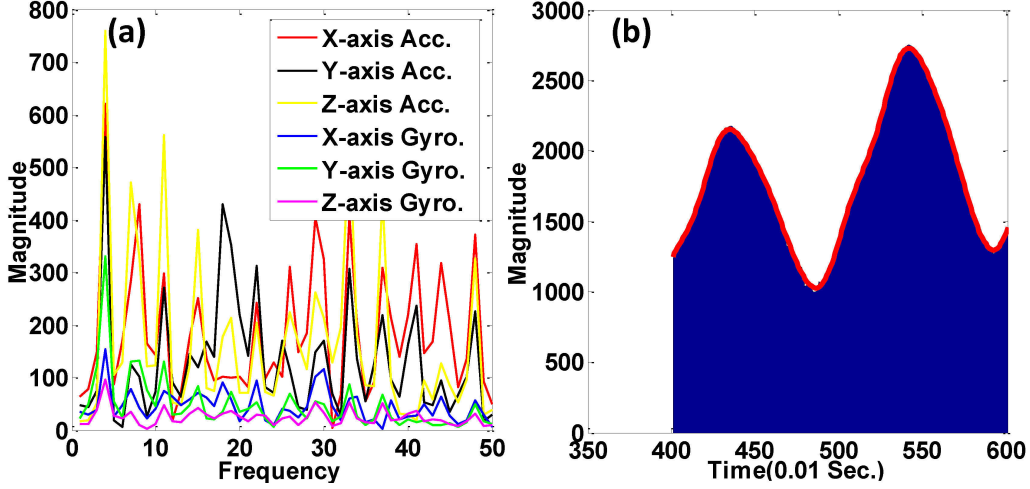


Figure 8: Step detection. (a) DFT results of 4-second signal with 400 samples. (b) The variance metric vs. signal length.

variation and this method can not be applied when we do not know any stride information of the target pedestrian in advance.

In this paper, a step detection algorithm based on adaptive sliding window and Discrete Fourier Transform (DFT) is introduced, which is inspired by the following observations: (1) When the pedestrian walks, the movement pattern is repetitive, so we can consider the IMU data as periodic signals. Therefore, DFT can be applied to find the number of periods in a certain sliding window, which is just the number of steps in that sliding window. (2) Magnitude field is sensitive to heading direction change, so it is not suitable for step detection. Instead, angular velocity and acceleration are ideal because they are independent of the forward moving direction; (3) Only one axis signal, no matter from Gyroscope or Accelerometer, is not reliable for step detection because the movement pattern is a combination of all three axes instead of one. Thus, all 6 axes of Gyroscope and Accelerometer are considered in our step detection by DFT.

Fig.8(a) shows the results after applying DFT to the six signals of Accelerometer and Gyroscope over a time sliding window L of 400 samples (in our IMU device, 100 samples of data are collected per second, so 400 samples of data imply data collected in 4 seconds). The horizontal axis in Fig.8(a) is the frequency related to the number of periods within the time sliding window. The vertical axis in Fig.8(a) is the corresponding magnitude. As

we know, the frequency component related to the step number should have higher magnitude, compared with the frequency component corresponding to noise. We compute the principle frequency of all six signals, f^* , by

$$f^* = \arg \max_f \sum_{i=1}^6 |F_i(f; L)| \quad (5)$$

where $|F_i(f; L)|$ denotes the magnitude of frequency component f of the i th signal within the time sliding window L . Note that f^* is an integer in DFT. In Fig.8(a), $f^* = 4$, but is there exactly f^* steps during the time sliding window L ? The answer is possibly NO. This is because DFT here only detects integral frequency. If there are 3.8 steps in the sliding window, the corresponding principle frequency of DFT is ceilinged to 4. If there are 4.3 steps in the sliding window, the corresponding principle frequency of DFT is floored to 4. Therefore, we need to search the accurate beginning and end sampling moments of complete steps in the signals to estimate the speed for DR. Otherwise, DR will deviate from the truth quickly due to the accumulated error. We observe that the principle frequency should have a large difference compared to its neighboring frequencies. For example, if one sliding window actually contains 4 steps, the magnitude of frequency 4 should be greatly larger than the magnitudes of frequency 3 and 5. Therefore, we propose a new metric M_L , the magnitude variance of the principle frequency compared with its neighboring frequencies, to search the accurate steps:

$$M_L = \sum_{i=1}^6 \text{var}([|F_i(f^*-1; L)|, |F_i(f^*; L)|, |F_i(f^*+1; L)|]). \quad (6)$$

We gradually increase the time sliding window L , compute f^* by Eq.5 and then compute M_L . Fig.8(b) shows the plot of M_L versus L . We can see the first peak is around 450 in $L = [1, 450]$ which means that there are 4 steps in 450 samples (4.5 seconds), i.e., each step period is about 113 samples. If we keep increasing L , we will find another peak around 565 in $L = [1, 565]$ which means that there are 5 steps in 565 samples. The peaks in Fig.8(b) indicate that at these points, the magnitude of the principle frequency has the largest difference compared to its neighboring frequencies. Thus we can detect the exact number of steps by adapting the time sliding window.

Fig.9 summarizes the workflow of our step detection based on adaptive sliding window and DFT. First, we coarsely define a time sliding window

as long as 400 samples (4 seconds). Secondly, we gradually increase the sliding window length until we find a peak of M_L , thus fine sliding window is determined. Then, the length of fine sliding window is divided by f^* (the number of steps in the sliding window) to get the step period, thus the beginning and end of every step in the time sliding window can be calculated. In video streams, steps are detected in non-overlapping time sliding windows (i.e., the starting time of next window $s_b = s_e + 1$ where s_e is the end time of the current k th sliding window L_k).

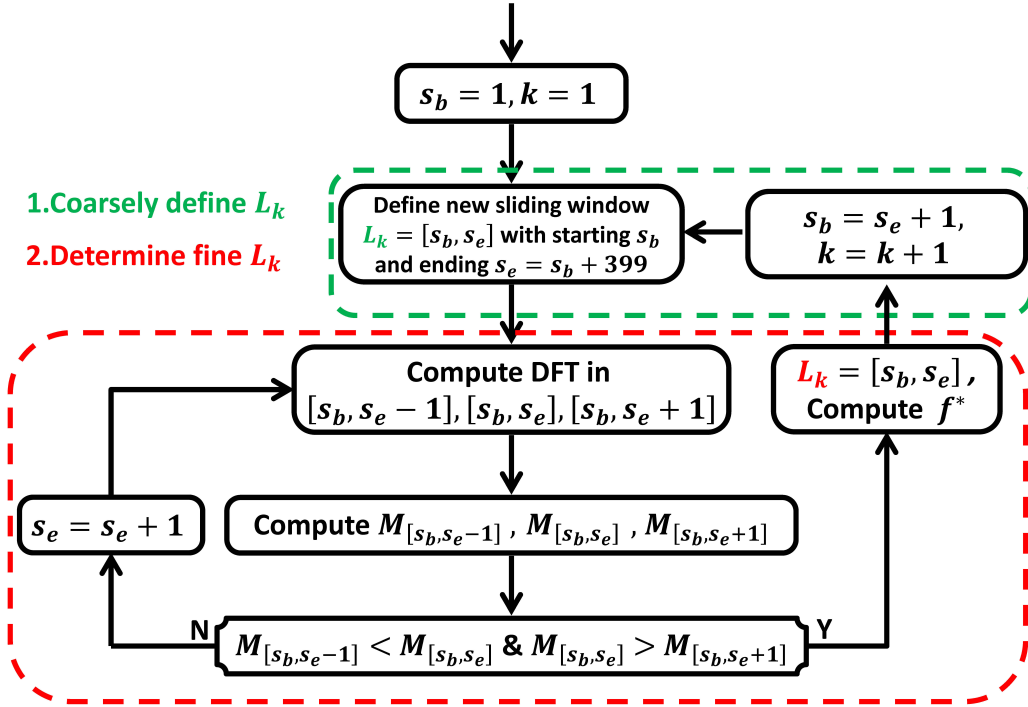


Figure 9: Flow chart of our step detection.

3.2. Speed Estimation

Practically, walking/running speed varies from pedestrian to pedestrian. Even for the same person, the speed is not constant. Integration on acceleration to obtain speed accumulates errors very fast, making it impractical for speed estimation. Observing that the intensity degree of movement is approximately proportional to speed, we propose to use the maximal difference

of angular velocity to measure the movement intensity. The measurement is only valid in complete movement pattern periods, which is at least one step. That is one of the reasons why we need accurate step detection and speed is calculated in the unit of step. The speed for step n is defined as

$$V_n = K(\max_{s \in [s_b^{(n)}, s_e^{(n)}]} \|\vec{e}_s\| - \min_{s \in [s_b^{(n)}, s_e^{(n)}]} \|\vec{e}_s\|) \quad (7)$$

where $s_b^{(n)}$ and $s_e^{(n)}$ denote the beginning and end time of the n th step and $\|\vec{e}_s\|$ computes the magnitude of angular velocity at sample s . K is the normalization factor depending on specific pedestrians. In merely IMU tracking, K is determined by calibration. When visual signal is considered, K is involved in a similarity warp matrix which will be introduced in Sec.4.

3.3. Coordinate Transformation

If there is no movement on IMU or noise, Accelerometer measures the gravity and Magnetometer measures the earth's magnetic field, so merely using Accelerometer and Magnetometer is enough to obtain the transformation between the IMU coordinate and the world coordinate. However, in practice, the estimation of gravity is influenced by IMU movement, and the measurement of the earth's magnetic field can be distorted by the presence of electrical components in surrounding environments. To mitigate these problems, Gyroscope can be added to compensate the bias from Accelerometer and Magnetometer [34], greatly improving the accuracy.

We describe the transformation in the format of quaternion, a four-dimensional normalized vector [35]. The transformation quaternion from IMU to the world coordinate is defined as $\mathbf{Q}^{(p \rightarrow w)} = [q_1 \ q_2 \ q_3 \ q_4]$, where the superscripts p and w represent the IMU and world coordinates, respectively. $\|\mathbf{Q}^{(p \rightarrow w)}\|_2 = 1$. A general quaternion in the IMU coordinate, such as $\mathbf{v}^{(p)} = [v_1 \ v_2 \ v_3 \ v_4]$, is transformed to the world coordinate by

$$\mathbf{v}^{(w)} = \mathbf{Q}^{(p \rightarrow w)} \otimes \mathbf{v}^{(p)} \otimes (\mathbf{Q}^{(p \rightarrow w)})^* \quad (8)$$

where $\otimes$ is the quaternion multiplication defined as

$$\begin{aligned}
\mathbf{Q}^{(p \rightarrow w)} \otimes \mathbf{v}^{(p)} &= [q_1 \ q_2 \ q_3 \ q_4] \otimes [v_1 \ v_2 \ v_3 \ v_4] \\
&= \begin{pmatrix} q_1 v_1 - q_2 v_2 - q_3 v_3 - q_4 v_4 \\ q_1 v_2 + q_2 v_1 + q_3 v_4 - q_4 v_3 \\ q_1 v_3 - q_2 v_4 + q_3 v_1 + q_4 v_2 \\ q_1 v_4 + q_2 v_3 - q_3 v_2 + q_4 v_1 \end{pmatrix}^T
\end{aligned} \tag{9}$$

$(\mathbf{Q}^{(p \rightarrow w)})^*$ is the quaternion conjugate of $\mathbf{Q}^{(p \rightarrow w)}$ defined as

$$(\mathbf{Q}^{(p \rightarrow w)})^* = [q_1 \ -q_2 \ -q_3 \ -q_4] = \mathbf{Q}^{(w \rightarrow p)}. \tag{10}$$

In the format of quaternion, any point in the world or IMU coordinate is represented as a vector $[0 \ x \ y \ z]$, where x , y and z are the values of three axes, respectively. In [34], $\mathbf{Q}_{a\&m,s}^{(p \rightarrow w)}$ (the transformation quaternion merely computed by Accelerometer and Magnetometer at sample s where the subscript $a\&m$ denotes Accelerometer and Magnetometer) and $\mathbf{Q}_{gyr,s}^{(p \rightarrow w)}$ (the transformation quaternion calculated merely by Gyroscope at sample s where the subscript gyr denotes Gyroscope) are computed individually. $\mathbf{Q}_s^{(p \rightarrow w)}$ is then obtained by the combination of the above two transformation quaternions. $\mathbf{Q}_{a\&m,s}^{(p \rightarrow w)}$ is computed by solving the following optimization problem:

$$\begin{aligned}
\arg \min_{\mathbf{Q}_{a\&m,s}^{(p \rightarrow w)}} & (\mathbf{Q}_{a\&m,s}^{(p \rightarrow w)} \otimes \mathbf{a}_s^{(p)} \otimes (\mathbf{Q}_{a\&m,s}^{(p \rightarrow w)})^* - \mathbf{g}^{(w)})^2 + \\
& (\mathbf{Q}_{a\&m,s}^{(p \rightarrow w)} \otimes \mathbf{m}_s^{(p)} \otimes (\mathbf{Q}_{a\&m,s}^{(p \rightarrow w)})^* - \mathbf{m}_s^{(w)})^2
\end{aligned}$$

where $\mathbf{g}^{(w)} = [0 \ 0 \ 0 \ -1]$, a constant pointing to the gravity in the world coordinate. $\mathbf{a}_s^{(p)}$ is the acceleration in the IMU coordinate at sample s . $\mathbf{m}_s^{(w)}$ and $\mathbf{m}_s^{(p)}$ are the magnetic field at sample s in the world and IMU coordinates, respectively.

$\mathbf{Q}_{a\&m,s}^{(p \rightarrow w)}$ can be obtained by solving the optimization problem using gradient descent method. The component transformation quaternion $\mathbf{Q}_{gyr,s}^{(p \rightarrow w)}$ estimated by gyroscope at sample s , can be calculated by the ensemble transformation quaternion at $s-1$, $\mathbf{Q}_{s-1}^{(p \rightarrow w)}$, plus the angular change during the time interval Δs between sample s and $s-1$:

$$\mathbf{Q}_{gyr,s}^{(p \rightarrow w)} = \mathbf{Q}_{s-1}^{(p \rightarrow w)} + \frac{1}{2} \mathbf{Q}_{s-1}^{(p \rightarrow w)} \otimes \mathbf{e}_s^{(p)} \otimes \Delta s \tag{11}$$

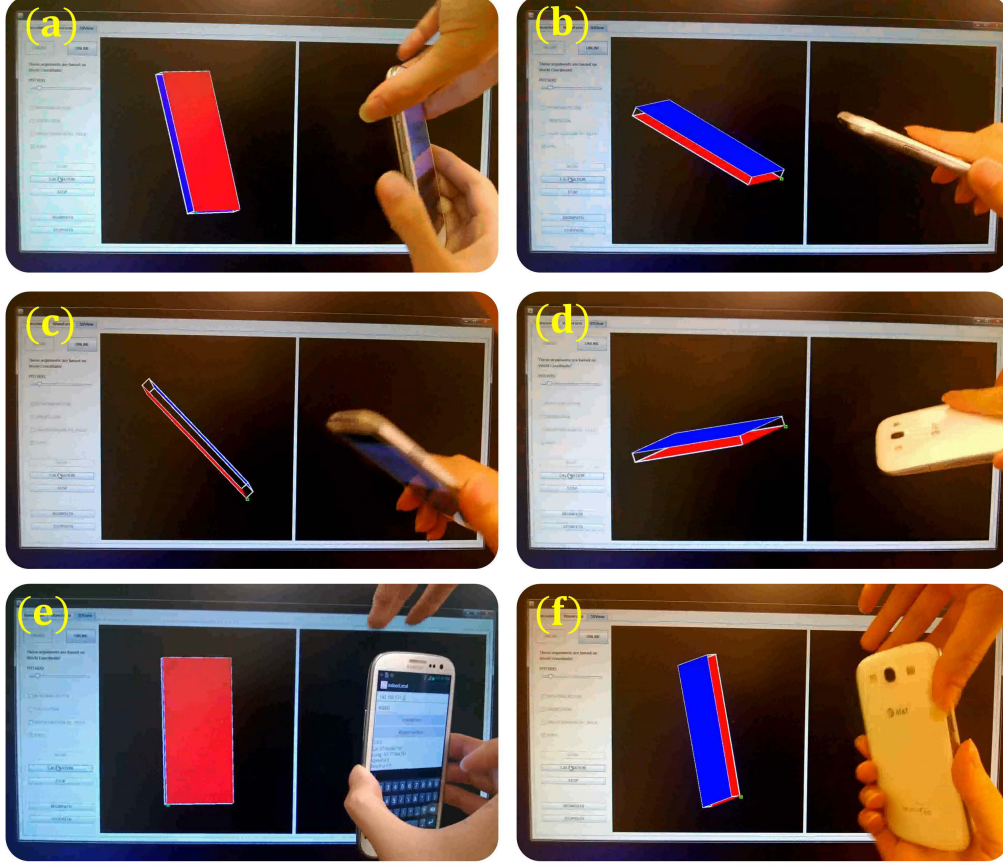


Figure 10: Test the coordinate transformation method by measuring a phone's posture.

where $\mathbf{e}_s^{(p)}$ is the angular velocity in the IMU coordinate. The ensemble $\mathbf{Q}_s^{(p \rightarrow w)}$ is defined by

$$\mathbf{Q}_s^{(p \rightarrow w)} = \gamma_s \mathbf{Q}_{gyr,s}^{(p \rightarrow w)} + (1 - \gamma_s) \mathbf{Q}_{a\&m,s}^{(p \rightarrow w)}, 0 \leq \gamma_s \leq 1 \quad (12)$$

where γ_s is a weighting coefficient related to the divergence of Gyroscope and convergence of Accelerometer and Magnetometer [34].

$\mathbf{Q}_s^{(p \rightarrow w)}$ can be applied to obtain the orientation of IMU in the world coordinate which is realized by transforming every point of IMU from the IMU coordinate to the world coordinate system. To test the effectiveness of this coordinate transformation method, we developed a smartphone orientation measurement program with the embedded IMU. As shown in Fig.10,

the smartphone was rotated by hands and the measured smartphone's orientation is shown on the software interface in real time. The precise results indicate that accurate $\mathbf{Q}_s^{(p \rightarrow w)}$ can be accurately estimated from the IMU sensors.

3.4. Forward Moving Direction Determination

The 3D acceleration vectors in the IMU coordinate during a short period (e.g., a few steps) can be projected to the horizontal plane in the world coordinate to infer the forward moving direction [22, 24]. This method works for professional IMU. But for low cost IMU such as the IMU module built in smartphones which is more likely to be influenced by noise, it performs poorly. Fig.11 shows the results when acceleration collected by a smartphone in 4 steps is projected to the world coordinate's horizontal plane. There is no obvious forward moving direction.

Considering the 3D acceleration vectors during a time sliding window as time-series signals, we transform them into the frequency domain. Since the principle frequency during the sliding window is already detected in the step detection process, we treat all non-principle frequency components as noise and zero out them. Then, we transform the filtered signal back to the time domain and project it to the world coordinate's horizontal plane. As shown in Fig.11(b), the moving direction is obvious. Ellipse-fitting (i.e., 2D Principle Component Analysis, PCA) is applied to the projected principle acceleration and the semi-major axis of the ellipse represents the forwarding direction $[u_x \ u_y]$.

We can apply the above procedure (filtering, projection and PCA) repetitively to determine the forward direction for every step, but the method can be further improved. In Fig.12, if an IMU follows the forward directions $A \rightarrow B \rightarrow C \rightarrow D$, the black vector that is constant in the IMU coordinate always points to the forward direction after projecting to the world coordinate. Thus, it is possible to find a constant vector $\mathbf{r}^{(p)}$ in the IMU coordinate system and then project it to the world coordinate to represent the forward moving direction. $\mathbf{r}^{(p)}$ can be found by formulating the following optimization problem

$$\arg \min_{\mathbf{r}^{(p)}} \sum_{s \in [s_b^{(1)}, s_e^{(1)}]} \| \mathbf{Q}_s^{(w \rightarrow p)} \otimes \mathbf{u}^{(w)} \otimes (\mathbf{Q}_s^{(w \rightarrow p)})^* - \mathbf{r}^{(p)} \| \quad (13)$$

where $\mathbf{u}^{(w)} = [0 \ u_x \ u_y \ 0]$ is the forward direction in the world coordinate

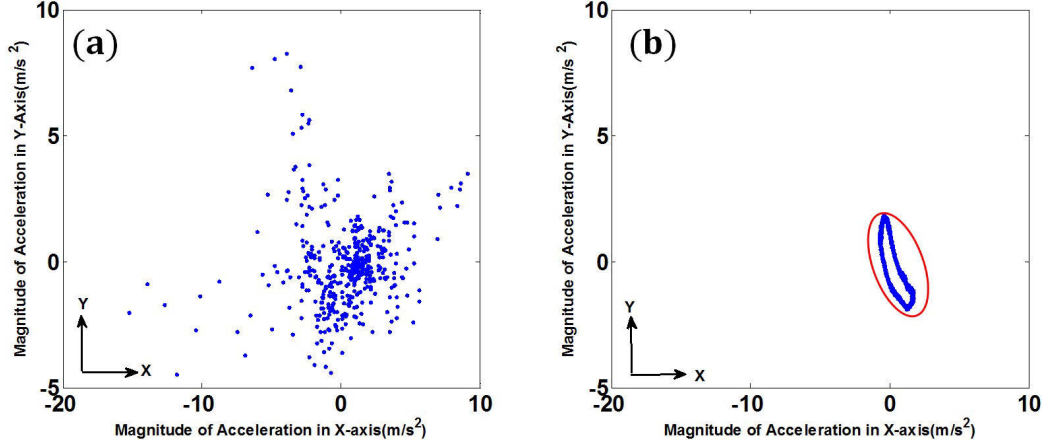


Figure 11: Determine the forward moving direction. (a) The acceleration in a short period (4 steps) is projected to the horizontal plane in the world coordinate. (b) The acceleration corresponding to the principle frequency in a short period is projected. The semi-major axis of the ellipse represents the forward moving direction.

computed by the filtering, projection and PCA approach in the first step. $s_b^{(1)}$ and $s_e^{(1)}$ denote the beginning and end sample of the first step. $\mathbf{r}^{(p)}$ is obtained by solving the optimization problem using the least square method.

$\mathbf{r}^{(p)}$ is fixed as long as the IMU's position on a human body is not changed. Once $\mathbf{r}^{(p)}$ is computed by some samples in the first step, the forward direction in the following sensor samples can be easily computed as $\mathbf{Q}_s^{(p \rightarrow w)} \otimes \mathbf{r}^p \otimes (\mathbf{Q}_s^{(p \rightarrow w)})^*$ where the coordinate transformation quaternion $\mathbf{Q}_s^{(p \rightarrow w)}$ is computed in Eq.12. Note that, $\mathbf{r}^{(p)}$ only needs to be computed once and it is only updated when the IMU position on a human body is changed.

4. Integration of Visual and IMU Tracking

There are two differences between the visual and IMU tracking methods which we should consider when developing a persistent pedestrian tracking system: (1) Visual tracking algorithm tracks pedestrians in surveillance videos frame-by-frame, but IMU tracking algorithm tracks pedestrians in every adaptive time sliding window (SW); (2) The trajectories generated by visual tracking is in the scene-specific coordinate while the trajectories from IMU tracking is in the horizontal plane of the world coordinate. To overcome the two problems, we update visual and IMU trajectories every sliding

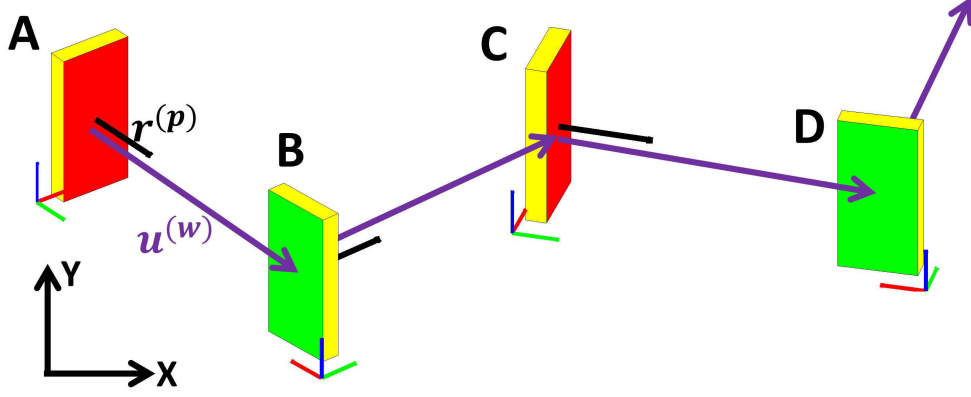


Figure 12: A constant vector in the IMU coordinate always represents the forward direction after projecting to the world coordinate.

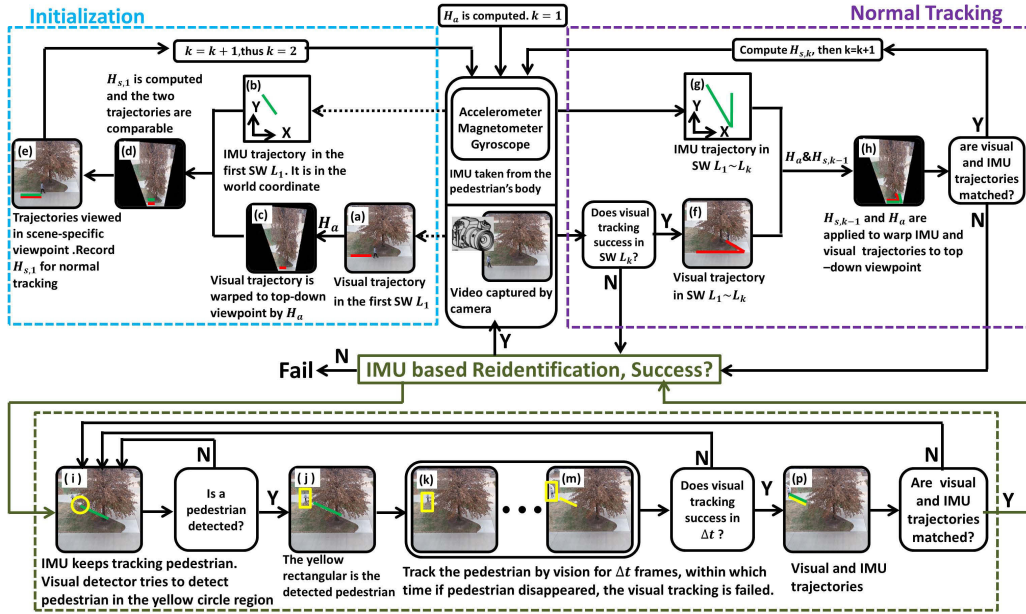


Figure 13: The flow chart of the persistent tracking system.

window. The IMU trajectories in the world coordinate are warped to the image coordinate to match with visual trajectories. As Fig.13 shows, the persistent pedestrian tracking system can be divided into three parts: Initialization (blue), normal tracking (purple) and IMU-based reidentification

(atrovirens).

4.1. Initialization

In this step, the initial relationship between visual and IMU trajectories is built. Fig.13(a) and Fig.13(b) show the visual trajectory (red) and IMU trajectory (green) in the first sliding window L_1 , respectively. The trajectory generated by IMU tracking is in the world coordinate, so it is a 2D curve in the horizontal plane viewed from top to down. Unlike IMU trajectory, visual trajectory is in the image coordinate depending on the specific camera viewpoint, thus they are not comparable. As illustrated in Fig.13(c), we warp the visual trajectory from scene-specific viewpoint to top-down viewpoint by H_a . Since the transformation between the warped visual trajectory (Fig.13(c)) and IMU trajectory (Fig.13(b)) is just rotation, translation and scaling (i.e., similarity transformation), we match the two trajectory curves by computing the similarity transformation matrix $\mathbf{H}_{s,k}$ in sliding window L_k using the least square procedure:

$$\arg \min_{\mathbf{H}_{s,k}} \sum_t (\mathbf{H}_{s,k} \mathbf{T}_t^{(v,k)} - \mathbf{T}_t^{(s,k)})^2 \quad (14)$$

where $\mathbf{T}_t^{(v,k)}$ and $\mathbf{T}_t^{(s,k)}$ denote the uniformly sampled points on the warped visual trajectory and sensor trajectory in sliding window L_k , respectively. The initialization step is performed in the first sliding window, so $k = 1$. Fig.13(d) shows the result of IMU trajectories matched to visual trajectories. For better visualization, we can warp the top-down viewpoint to the scene-specific viewpoint by the inverse of H_a . Therefore, two matrices, H_a (homography transformation) and $H_{s,k}$ (similarity transformation), make visual and IMU trajectories compatible. H_a does not change unless the scene-specific viewpoint changes. $H_{s,k}$ keeps being updated during each sliding window of the persistent tracking.

4.2. Normal Tracking

The initialization step only needs to be performed once, then our system goes to the normal tracking. Fig.13(f) and (g) show the trajectories based on visual and IMU tracking, respectively, in sliding windows $L_1 \sim L_k$. Then, $H_{s,k-1}$ and H_a are applied to warp IMU and visual trajectories to top-down viewpoint. The average distance d between trajectories in Fig.13(h) is calculated. If $d < d_{thr}$, visual and IMU trajectories are matched, then new

$H_{s,k}$ is computed using Eq.14 and we go to the next sliding window. In our persistent pedestrian tracking system, we set $d_{thr} = 2\text{m}$ which is the highest error that the current system can tolerate.

4.3. IMU-based Reidentification

The atrovirens part of Fig.13 illustrates the IMU-based reidentification. Two cases lead to the reidentification: (1) The pedestrian disappears in visual tracking. The pedestrian may move out of the visual field or be occluded by obstructions such as trees; (2) Visual and IMU trajectories do not match each other. This may be caused by visual tracking drift (i.e., track a non-target pedestrian).

As shown in Fig.13(i), IMU keeps tracking the pedestrian even the target is occluded by a tree. The green curve is the IMU trajectory. Meanwhile, visual pedestrian detector tries to detect pedestrians in a search region estimated by IMU (yellow circle in Fig.13(i)). If detected, the pedestrian will be tracked by visual tracking for Δt frames (Fig.13(k)~(m)). If any visual tracking failure happens within the Δt frames, we go back to the IMU-tracking (Fig.13(i)). If the tracking within the Δt frames successes, the average distance d between IMU and visual trajectories during Δt is computed to judge if they match. If $d < d_{thr}$, the target pedestrian is reidentified and we go back to the normal tracking again. Otherwise, we go back to the IMU-tracking (Fig.13(i)) for reidentification.

The above persistent pedestrian tracking system elucidates why visual tracking and IMU tracking are “complementary”. First, when visual tracking fails, IMU tracking keeps working and offers the clue where the pedestrians should be, helping visual tracking reidentify the pedestrian. Secondly, $H_{s,k}$ contains rotation, translation and scale relationship between IMU and visual trajectories. The visual trajectory corrects the bias of speed and forward direction estimation in IMU tracking by the similarity matrix $H_{s,k}$. The value of K in Eq.7 is involved in $H_{s,k}$. As we keep updating $H_{s,k}$, visual tracking rebuilds the relationship with IMU tracking and rectifies the deviation of IMU-based tracking trajectory.

5. Experiments

To validate the effectiveness of our persistent tracking system, we apply it for pedestrian tracking in daily environments. Smartphones embedded with IMU modules are selected as the IMU signal collector. The IMU module in a

smartphone is low cost and more likely to be influenced by noise. If our system works well in smartphones, it is reasonable to justify that it should work using expensive and professional IMU devices. In addition, the popularity of smartphones offers more possibilities of application of our tracking system. We developed an App to collect IMU signals when the target pedestrian is walking. The IMU signals are transmitted back to a groundstation by GSM. Meanwhile, a stationary surveillance camera collects visual signals of the target pedestrian. The visual signal is taken at 30 frames per second and the sampling frequency of IMU signal is 100 samples per second. To synchronize the two signals, for every frame of the video, the nearest IMU signal is found according to the timestamp provided by the smartphone system. We design the experiments in two aspects. (1) IMU helps visual tracking in challenging cases such as long term occlusion, nearby clutter and suddenly changes on movement patterns. (2) Visual tracking assists in calibrating IMU tracking by rectifying the bias during IMU tracking.

5.1. IMU Tracking Assists Visual Tracking

Visual tracking challenges include long term and heavy occlusions, sudden speed/appearance change and temporarily disappearance. We record four videos in different conditions to test the performance of our persistent pedestrian tracking system. Fig.14 shows the visual and IMU trajectories from our persistent pedestrian tracking system (video demos are provided in the supplemental materials). Video 1 is taken in an occlusion environment with a small slope. The target pedestrian is occluded twice for 9 and 16 seconds, respectively. This video aims to validate the proposed algorithm in heavy and long-term occlusions. Fig.14(a)(b) show that the pedestrian is successfully tracked in such a scenario. In video 2 (Fig.14(c)(d)), the pedestrian changes his speed from walking to sudden run and then stop when hidden by the tree. Ten seconds later, the pedestrian begins to walk toward the starting point. The pedestrian changes his movement pattern (i.e., speed and forward direction) when occluded, which is very hard to visual-based tracking algorithms. Fig.14(e)(f) show the case where there are multiple pedestrians. The pedestrian is occluded by other walking pedestrians, when two nearby people depart towards different directions, visual tracking fails because of the clutter of similar appearance. However, IMU tracking tracks and reidentifies the pedestrian successfully. Fig.14(g)(h) shows a comprehensive case, where the pedestrian is occluded by moving pedestrians at location A and moves out of the visual field from location B. At locations C and D, the pedestrian is

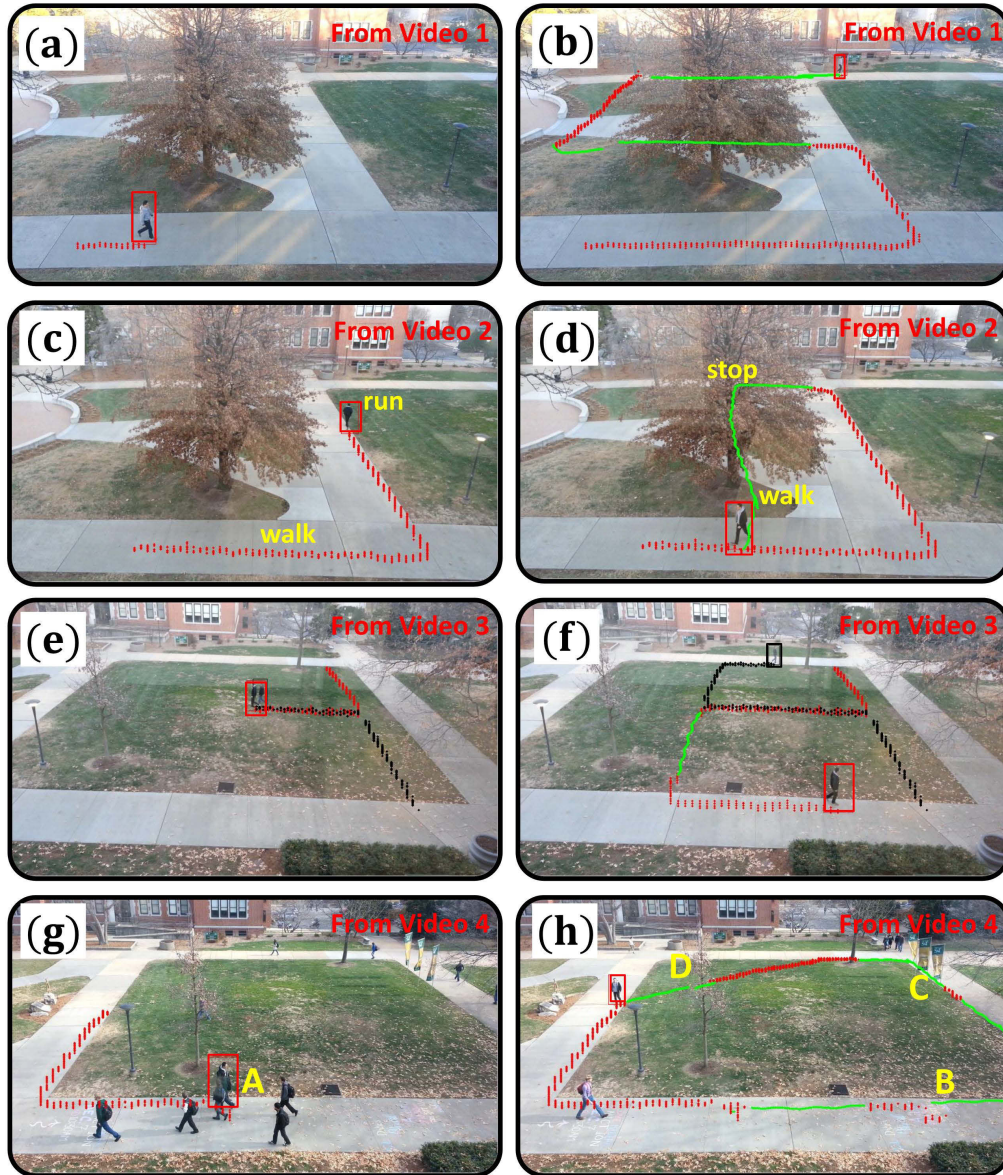


Figure 14: Trajectories of the target pedestrian. Red curves are visual trajectories and green curves are IMU trajectories when visual tracking fails. (a)(b) Screenshots from video 1. (c)(d) Screenshots from video 2. (e)(f) Screenshots from video 3. (g)(h) Screenshots from video 4.

occluded by background objects frequently. Despite the complex scenarios, our persistent tracking system successfully tracks the pedestrian.

Table 1 summarizes the quantitative evaluation about the persistent tracking system. The ground truth is labelled by a human annotator in each frame of the videos. A frame is considered to be successfully tracked when the distance between the trajectory in that frame and the ground truth is no larger than 100 inches. The average trajectory error is estimated by the difference between the successfully tracked trajectories and ground truth. The standard deviation shows the stability of these methods. Visual tracking fails when the pedestrian is occluded or disappears at the first time in each video while our proposed tracking system can persistently track the pedestrian in all videos. Note that the performance of Only-IMU tracking is largely influenced by parameters (e.g., K in Eq.7). The results of Only-IMU tracking in Table 1 are obtained by carefully predefined parameters, but it is infeasible to manually choose parameters for different people in practice. The experiments validate that visual tracking combined with IMU tracking can achieve both persistency and accuracy.

Table 1: [Persistent Pedestrian Tracking Results](#). FS: number of Frames Successfully tracked. AVG: Average Error (inch). STD: Standard deviation (inch).

Video No.	Number of Frames in the Video	Only Vision		Only IMU		The Proposed	
		FS	AVG $\pm$ STD	FS	AVG $\pm$ STD	FS	AVG $\pm$ STD
1	2009	580	50.3 $\pm$ 6.1	2009	66.0 $\pm$ 9.5	2009	51.2 $\pm$ 6.3
2	1940	717	35.2 $\pm$ 5.0	1940	45.2 $\pm$ 8.7	1940	37.0 $\pm$ 5.4
3	1063	625	22.4 $\pm$ 4.8	1063	15.8 $\pm$ 6.6	1063	19.6 $\pm$ 5.2
4	1857	392	33.1 $\pm$ 4.9	1857	36.1 $\pm$ 8.1	1857	34.5 $\pm$ 5.4

5.2. Visual Tracking Assists IMU Tracking

We use video 1 as an example to compare different IMU tracking methods and shows the benefit of visual tracking to help IMU tracking. Fig.15(a) is the ground truth of the pedestrian trajectory, which is obtained by warping the pedestrian’s trajectories in the scene-specific viewpoint to the top-down viewpoint by H_a . All trajectories in Fig.15 are in the horizontal plane of the world coordinate. Fig.15(b) is based on the PCA2D method introduced in [22], which detects step in the time domain. There are many misdetections

on step and direction by this approach and the tracked trajectory drifts away from the ground truth largely. Fig.15(c) is the result by our IMU tracking method without any assistance from the visual tracking. Step is detected by DFT and forward direction is calculated by the constant vector obtained in the first step. Fig.15(d) shows the trajectory results of IMU tracking assisted by the visual tracking. Our IMU tracking is more reliable than other step detection algorithms, but the IMU trajectory in Fig.15(c) still drift from the ground truth a little. When visual tracking is combined, visual trajectories constantly adjust the orientation and scale of IMU trajectories with $H_{s,k}$, the IMU trajectory in Fig.15(d) is very close to the ground truth.

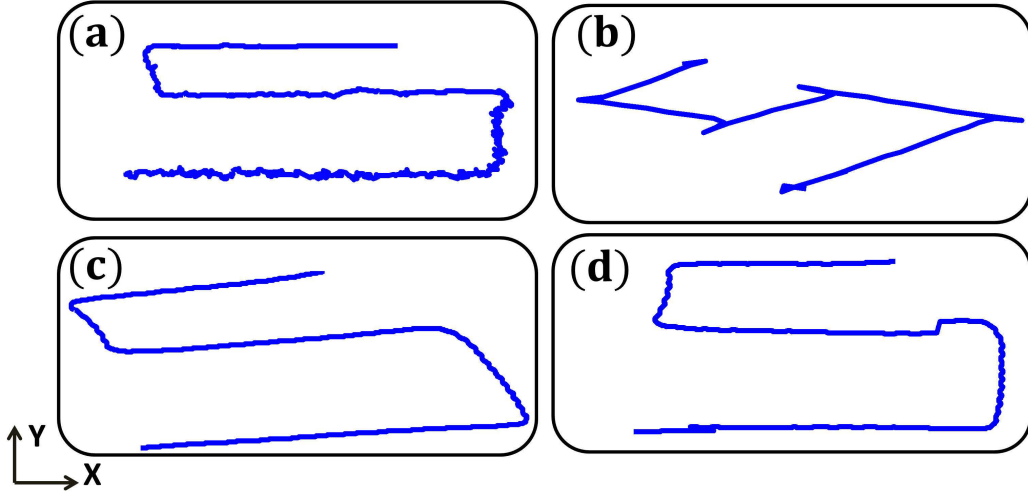


Figure 15: IMU trajectories processed by three different approaches. (a) Ground truth trajectory; (b) Trajectory by the time-domain step detection [6]; (c) Trajectory by our DFT approach; (d) Trajectory by our DFT approach assisted by the visual signal.

6. Conclusion

To persistently track pedestrians in a specific scene with occlusion, we present a novel tracking system combining the visual and Inertial Measurement Unit (IMU) signals, obtained from surveillance cameras and IMU devices carried by the targets themselves, respectively. Not only can IMU assist visual tracking when the target is occluded, but also the challenges of IMU tracking (calibration and drift) are alleviated when visual signals are

available. Experimental results show that visual and IMU tracking are complementary to each other and their integration achieves good performance on persistent people tracking under challenging daily environments.

7. Acknowledgements

This work was supported by Intelligent Systems Center (ISC) at Missouri University of Science and Technology, and the National Science Foundation (NSF) grant CMMI-1646162.

References

- [1] W. Jiang, Z. Yin, Combining passive visual cameras and active imu sensors to track cooperative people, in: IEEE International Conference on Information Fusion, 2015, pp. 1338–1345.
- [2] M. Danelljan, G. Häger, F. S. Khan, M. Felsberg, Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking, in: Computer Vision and Pattern Recognition, 2016.
- [3] F. Xiao, Y. J. Lee, Track and segment: An iterative unsupervised approach for video object proposals, in: Computer Vision and Pattern Recognition, 2016.
- [4] A. Leykin, R. Hammoud, Robust multi-pedestrian tracking in thermal-visible surveillance videos, in: IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2006, pp. 136–136.
- [5] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, IEEE transactions on pattern analysis and machine intelligence 34 (4) (2012) 743–761.
- [6] M. Enzweiler, D. M. Gavrila, Monocular pedestrian detection: Survey and experiments, IEEE transactions on pattern analysis and machine intelligence 31 (12) (2009) 2179–2195.
- [7] D. Geronimo, A. M. Lopez, A. D. Sappa, T. Graf, Survey of pedestrian detection for advanced driver assistance systems, IEEE transactions on pattern analysis and machine intelligence 32 (7) (2010) 1239–1258.

- [8] A. Roy, P. Chattopadhyay, S. Sural, J. Mukherjee, G. Rigoll, Modelling, synthesis and characterisation of occlusion in videos, *IET Computer Vision* 9 (6) (2015) 821–830.
- [9] B. Wu, R. Nevatia, Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors, *International Journal of Computer Vision* 75 (2) (2007) 247–266.
- [10] D. Merad, K.-E. Aziz, R. Iguernaissi, B. Fertil, P. Drap, Tracking multiple persons under partial and global occlusions: Application to customers behavior analysis, *Pattern Recognition Letters*, 2016.
- [11] Z. Li, Q. L. Tang, N. Sang, Improved mean shift algorithm for occlusion pedestrian tracking, *Electronics Letters* 44 (10) (2008) 622–623.
- [12] S. Zhang, J. Wang, Z. Wang, Y. Gong, Y. Liu, Multi-target tracking by learning local-to-global trajectory models, *Pattern Recognition* 48 (2) (2015) 580–590.
- [13] J. Sherrah, Occluded pedestrian tracking using body-part tracklets, in: *International Conference on Digital Image Computing: Techniques and Applications*, 2010, pp. 314–319.
- [14] <http://www8.garmin.com/aboutGPS/>.
- [15] H. Wang, H. Lenz, A. Szabo, J. Bamberger, U. D. Hanebeck, Wlan-based pedestrian tracking using particle filters and low-cost mems sensors, in: *Workshop on Positioning, Navigation and Communication*, 2007, pp. 1–7.
- [16] S. Woo, S. Jeong, E. Mok, L. Xia, C. Choi, M. Pyeon, J. Heo, Application of wifi-based indoor positioning system for labor tracking at construction sites: A case study in guangzhou mtr, *Automation in Construction* 20 (1) (2011) 3–13.
- [17] P. Falcone, F. Colone, A. Macera, P. Lombardo, Localization and tracking of moving targets with wifi-based passive radar, in: *IEEE Radar Conference*, 2012, pp. 0705–0709.
- [18] J.-P. Dubois, J. S. Daba, M. Nader, C. El Ferkh, Gsm position tracking using a kalman filter, *World Academy of Science, Engineering and Technology* 68 (2012) 1610–1619.

- [19] D. B. Ahmed, E. M. Diaz, S. Kaiser, Performance comparison of foot- and pocket-mounted inertial navigation systems, in: *Indoor Positioning and Indoor Navigation*, 2016, pp. 1–7.
- [20] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, Activity sequence-based indoor pedestrian localization using smartphones, *IEEE Transactions on Human-Machine Systems* 45 (5) (2015) 562–574.
- [21] M. Hardegger, D. Roggen, G. Tröster, 3d actionslam: wearable person tracking in multi-floor environments, *Personal and Ubiquitous Computing* 19 (1) (2015) 123–141.
- [22] U. Steinhoff, B. Schiele, Dead reckoning from the pocket-an experimental study, in: *IEEE International Conference on Pervasive Computing and Communications*, 2010, pp. 162–170.
- [23] E. Foxlin, Pedestrian tracking with shoe-mounted inertial sensors, *IEEE Computer graphics and applications* 25 (6) (2005) 38–46.
- [24] Y. Jin, H.-S. Toh, W. S. Soh, W. C. Wong, A robust dead-reckoning pedestrian tracking system with low cost sensors, in: *IEEE International Conference on Pervasive Computing and Communications*, 2011, pp. 222–230.
- [25] T. N. Hung, Y. S. Suh, Inertial sensor based two feet motion tracking for gait analysis, *Sensors* 13 (5) (2013) 5614–5629.
- [26] R. Feliz Alonso, E. Zalama Casanova, J. Gmez Garca-Bermejo, Pedestrian tracking using inertial sensors, *Journal of Physical Agents* 3 (1) (2009) 35–43.
- [27] M. Kourogi, T. Kurata, Personal positioning based on walking locomotion analysis with self-contained sensors and a wearable camera, in: *International Symposium on Mixed and Augmented Reality*, 2003, p. 103.
- [28] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M. Srivastava, Using mobile phones to determine transportation modes, *ACM Transactions on Sensor Networks* 6 (2) (2010) 13.

- [29] G. Hache, E. D. Lemaire, N. Baddour, Mobility change-of-state detection using a smartphone-based approach, in: IEEE International Workshop on Medical Measurements and Applications Proceedings, 2010, pp. 43–46.
- [30] J. Yang, H. Lu, Z. Liu, et al., Physical activity recognition with mobile phones: challenges, methods, and applications, in: Multimedia Interaction and Intelligent User Interfaces, 2010, pp. 185–213.
- [31] M. Andriluka, S. Roth, B. Schiele, People-tracking-by-detection and people-detection-by-tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
- [32] Y. Mao, Z. Yin, Training a scene-specific pedestrian detector using tracklets, in: IEEE Winter Conference on Applications of Computer Vision, 2015, pp. 170–176.
- [33] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2000, pp. 142–149.
- [34] S. O. H. Madgwick, A. J. L. Harrison, R. Vaidyanathan, Estimation of imu and marg orientation using a gradient descent algorithm, in: IEEE International Conference on Rehabilitation Robotics, 2011, pp. 1–7.
- [35] <http://en.wikipedia.org/wiki/Quaternion>.