# Integrating Low-Resolution Surveillance Camera and Smartphone Inertial Sensors for Indoor Positioning

Jiuxin Zhang, Pingqiang Zhou
School of Information Science and Technology
ShanghaiTech University
Shanghai, China
{zhangjx1, zhoupq}@shanghaitech.edu.cn

*Abstract*—**Indoor positioning techniques for pedestrians are the keys to the location-based services (LBSs) for human beings. Accuracy, overhead and ease of use are most noteworthy to evaluate an indoor positioning system. In this paper we present a novel indoor positioning system that integrates widely distributed low-resolution surveillance cameras and smartphone inertial sensors, which are perfect devices to build a system with low cost and high ease of use. Our proposed system receives video sequence from the surveillance camera and locates the pedestrians in the video view. The target pedestrian is recognized out of the crowd by matching the features from its smartphone inertial sensors with the pedestrian features extracted from the video sequence. We also apply a CNN-based visual object tracking algorithm to handle the situation that the target pedestrian is partly blocked in the video sequence. Our experimental results show that high positioning accuracy (at centimeter level) can be achieved. We also compare the other aspects of our technique against the state-of-art indoor positioning techniques, and show its superiority in many aspects.**

***Keywords—indoor positioning; visual tracking; surveillance camera; smartphone***

## I. INTRODUCTION

With the rapid development of internet and mobile devices, location based services (LBSs), especially the indoor navigation service, attract more and more attention. For example, in large buildings like malls and hospitals, navigation service is required for a guest to find the destinations of interest. Besides, the retailers in the mall may want to send online ads to the potential customers nearby. Despite the strong demand for LBS, indoor positioning remains a grand challenge and active research are desired to find an accurate, yet cheap and easy-to-use solution. In an indoor environment, due to signal shielding effect by the buildings, positioning techniques using global navigation satellite systems (GNSS) like GPS show very bad performance, thus various techniques have been developed for indoor positioning. Accuracy, overhead and ease of use are three important concerns for indoor positioning techniques.

Radio Frequency (RF)-based approaches have been widely used for indoor positioning. Wi-Fi-fingerprinting method takes full advantage of the already existing Wi-Fi routers. BLE and iBeacon positioning systems require additional Bluetooth equipment to achieve enough beacon density. Although RF-based systems are easy to set up and use, they have limited usage for many applications like retail navigation and shelf-level advertising due to limited accuracy (with errors up to meters) and not offering orientation information [1].

Smartphone Pedestrian Dead Reckoning (PDR) approaches do not need extra equipment. They use the inertial sensors in the smartphone to collect acceleration and orientation data, and then calculate the relative displacement information by a second integral of acceleration. As long as the initial position is specified, current earth-coordinate position can be reckoned. However, errors accumulate over time and increase rapidly, and the system relies on a proper technique to set initial position [2]. PDR systems are cheap and easy to use, but the accuracy is not guaranteed.

Visible Light Communication (VLC)-based systems utilize smartphone cameras and additional modulated LEDs [3], [4], [5]. Take Luxapose [5] as an example, smartphones take pictures of several LEDs and calculate the relative position of the smartphones with respect to the LEDs. Because of the straight propagation characteristic of visible light, the errors can be significantly reduced. But such system is not easy to use because the users have to hold the phone and take pictures of the LEDs at the ceiling or up on the wall. Another problem is that, the system needs densely distributed modified LEDs, which may lead to high cost and is not suitable for all buildings. Therefore, VLC-based systems are usually difficult to use and have high overhead.

Motivated by the VLC-based system, we realize that visual light is a promising media for signal transmission in LBS because of its high theoretical accuracy at the physical level. Considering ease of use, additional equipment for indoor positioning should be minimized, thus for most cases, smartphone is the best choice for positioning service, also the already existed surveillance cameras can be used for the visual light positioning. In this work, we propose an accurate, cheap and easy-to-use indoor positioning system based on surveillance cameras and smartphone. Our system uses surveillance cameras to collect video stream and then extracts the positioning information of pedestrians from the video. The sight line of camera is straight thus the positioning error can be reduced to decimeter level. Further, inertial sensors in PDR is applied in our system to collect the gait features of the users who need the positioning services, and to ensure a user-friendly solution. Surveillance cameras are already available and widely distributed in most of the buildings, also smartphones are widely

978-1-5386-1647-5/18/$31.00 ©2018 IEEE

used by pedestrians, so the overhead is very low. Unlike fingerprinting-based methods which demand frequently calibration, the surveillance cameras in our system only need to be calibrated once. Finally, to prevent the error or failure caused by obstructions before the pedestrians, we propose an object tracking algorithm using Convolutional Neural Network (CNN), which can track partly blocked or temporally completely blocked pedestrians.

The rest of the paper has the following structure. In Section II we briefly introduce the background. In Section III we discuss the details of the proposed indoor positioning system. We present experimental results in Section IV and make the conclusions in Section V.

## II. Background

### A. *Foreground Extraction*

In positioning applications, we usually focus on the moving objects in a video, instead of the stable environment. Gaussian mixture model (GMM) is widely used in foreground detection and extraction [6], which can extract the pixels of moving objects as the foreground. GMM models the values of each pixel in the video as a mixture of Gaussians. Pixels that do not fit corresponding Gaussian contributions are considered to be foreground. In real environment, the background is changing, thus the Gaussians actually model the relatively static background pixels, and the background models are updated frame by frame using online expectation maximization (EM) algorithm. In other words, the environmental changes, such as illumination change, chair moving and door opening, only have short-term impact on the extracted foreground for several frames, before Gaussian models are updated to the new background.

### B. *CNN-Based Visual Object Tracking*

Visual object tracking is one of the classical computer vision problems and has been the hot topic of research for many years. Generally speaking, visual object tracking algorithms receive the pixel region of the tracking target and output the pixel region of target object in each of following frames. Correlation filters show good performance and computational efficiency in visual object tracking [7], however, they are weak in handling complex situations such as occlusion and illumination changes.

With the rapid improvement of computing power, Convolutional Neural Network (CNN) presents extraordinary performance in a wide range of computer vision tasks [8]. CNN algorithms can be highly parallel implemented with GPU and obtain better efficiency. To get satisfactory predict accuracy, CNN has to be trained on a large dataset, but the object detecting and tracking tasks are usually insufficient in training data. Region-based CNN (R-CNN) is pre-trained offline on a large-scale dataset and fine-tuned online on the target dataset, thus it performs well in object tracking.

CNN-based visual object tracking algorithms have great accuracy and strong performance in handling illumination change and occlusion, which may lead to fatal error in tracking tasks.

### C. *Inertial Sensors in Smartphones*

Smartphone built-in sensors have attracted great attention in indoor positioning research. In most cases, smartphone sensors are Micro-Electro-Mechanical Systems (MEMS) including accelerometers and magnetometers which deliver motion data. Accelerometers output accelerated speeds along the three axes of a three-dimensional cubic coordinate system, which can be integrated to estimate relative displacement in PDR algorithms [2]. But the integral of accelerated velocity show big error. Magnetometer detects the direction of the geomagnetic field, from which we can figure out the heading azimuth of smartphone.

In our work we process the motion data from the MEMS accelerometers and magnetometers, and extract the features of the smartphone owner. The proposed system uses these features to identify smartphone owner out of the crowd, which is detailed in section III(B)-4.

## III. Positioning System

### A. *System Overview*

The main framework of the proposed indoor positioning system is shown in Fig. 1. The whole system includes two main parts. The first part is surveillance video processing. The surveillance camera collects video stream and sends real-time video image to the back-end server. GMM algorithm extracts

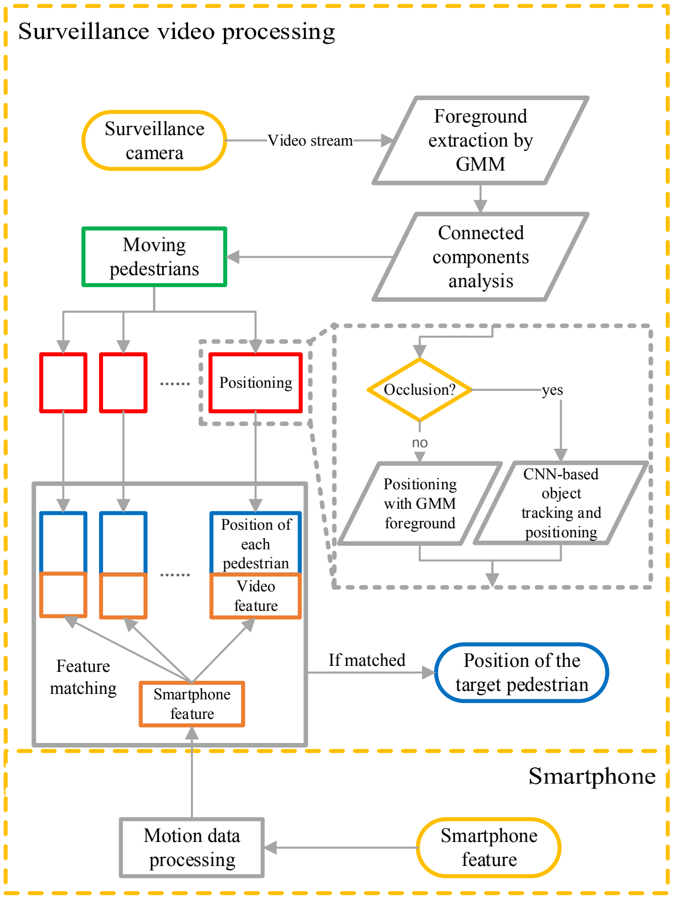

Surveillance video processing
Surveillance camera
Video stream
Foreground extraction by GMM
Connected components analysis
Moving pedestrians
......
Positioning
Occlusion?
yes
no
Positioning with GMM foreground
CNN-based object tracking and positioning
Position of each pedestrian
Video feature
......
Feature matching
Smartphone feature
If matched
Position of the target pedestrian
Smartphone
Motion data processing
Smartphone feature

Fig. 1. Positioning system framework.

the foreground mask of the whole video image, then the foreground mask of each moving pedestrian in the video is separated by the connected components analysis approach [9]. The system then runs positioning algorithm on each moving pedestrian. If a pedestrian is under occlusion, the pedestrian is tracked and positioned by a CNN-based algorithm; if not, the pedestrian is positioned by the foreground mask directly. Meanwhile, the video feature of each pedestrian is extracted together with the positioning information. The second part is smartphone feature extraction using smartphone built-in sensors. Finally the system matches the smartphone feature with the features of all the pedestrians in the video, and figures out which one is the target pedestrian, then outputs the corresponding position.

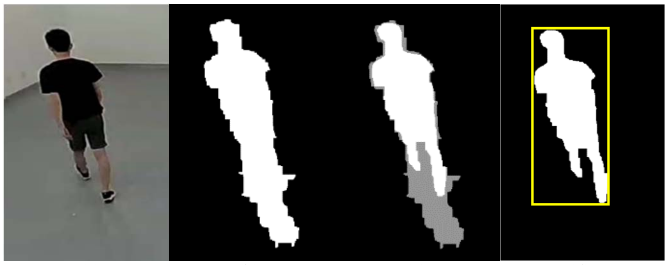
In most real-life scenarios, the resolution of the surveillance cameras are fairly low. To mimic the worst case of equipment condition in real-world, we use a fairly low resolution camera with 480p resolution. The camera is installed on the wall with 3-meter height above the floor, it collects video stream of the walking pedestrians in the field view of fixed angle. The methodology is detailed in the following section.

### B. Technical Details

#### 1) Moving Object Extraction

Surveillance cameras have intrinsic distortion that result in fish-eye effect in the video – A straight line in the real word can be crooked in the video. The fish-eye effect may make the subsequent coordinate transformation much more difficult. To eliminate intrinsic distortion, the camera is pre-calibrated using a checkerboard, which is widely used in camera calibration. After pre-calibration, the un-distortion algorithm can remove fish-eye effect from each video frame using the calibration parameters [10].

To track the pedestrians and extract their positional information, we firstly extract the pixel region of moving objects in the video. GMM method introduced in section II is applied to extract the foreground mask that represents moving objects. Each pixel of the image holds $K$ Gaussian models. In our experiments, we have observed that the tracking performance is the best when $K = 6$. The Gaussian parameters are initialized offline but trained online. Pixels that do not match any Gaussian model are regarded as foreground [6]. Afterwards we use a shadow suppression algorithm to obtain the clear outlines of pedestrians [11]. Fig. 2 shows the foreground extraction pipeline. Fig. 2(a) shows one video frame, and Fig. 2(b) presents the foreground mask extracted by GMM. We can see that the extracted foreground is not clear because the dynamic shadow is also considered as foreground. A robust shadow suppression algorithm is applied to remove the dynamic shadow. In an RGB image, pixels that have same R: G: B ratio as the background are considered as dynamic shadow. The gray region in Fig. 2(c) indicates the dynamic shadow. Fig. 2(d) shows the clear foreground mask of a moving pedestrian in surveillance video after shadow suppression. Afterwards, the local masks of different pedestrians in the video space are separated with bounding boxes by a connected-component-analysis algorithm.



(a) (b) (c) (d)

Fig. 2. Foreground extraction and shadow suppression.

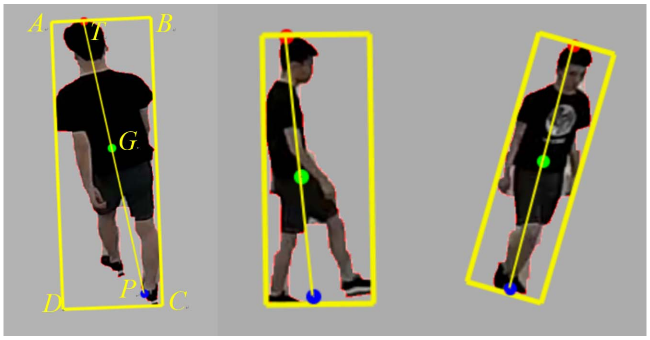


Fig. 3. Positioning auxiliary points and lines.

#### 2) Positioning without Occlusion

Since we have the foreground mask of the target pedestrian, we can figure out the position coordinates on the video space. The system only reports two-dimensional positions, so we just focus on the ground point where the target pedestrian stands. The exact ground point below the pedestrian's physical gravity center is difficult to locate by just analyzing video images, but we can make an approximation utilizing the foreground mask. The positioning algorithm extracts top point and the gravity center of the pedestrian, and the ground position point is considered to be on the extension line of these two points. Fig. 3 illustrates the auxiliary points and lines that help to locate the position coordinates. Firstly, we draw a minimal bounding rectangle ($ABCD$ in Fig. 3) of the foreground mask. The bounding rectangle with minimal area is calculated by an iterative improvement method. The height of pedestrian $h$ is considered as a scaling of the length of the longer side ($L_{AD}$) of minimal bounding rectangle, where $\lambda$ is a scaling factor that is found to be 0.95 in our work.

$$h = \lambda L_{AD} \tag{1}$$

Point $T$ in Fig. 3 is the top point of the foreground mask, while $G$ is the center of gravity of the foreground pixels. The ground point $P$ is on the extension cord of $\overrightarrow{TG}$ and $L_{TP} = h$. Therefore, we make an approximation of the ground point by extracting the top point $T$, gravity center $G$ and the dynamic height $h$ of pedestrian's foreground mask, and get point $P$ on the video space.

The next step is to compute the corresponding coordinate of point $P$ in the real world. The transformation of coordinates can be modeled as a rigid transform problem [12]:

$$\begin{bmatrix} X_r \\ Y_r \\ 1 \end{bmatrix} = R \begin{bmatrix} X_v \\ Y_v \\ 1 \end{bmatrix} + \delta \qquad (2)$$

where the video space coordinate is $(X_v, Y_v)$ and $(X_r, Y_r)$is the corresponding coordinate in the real word. Matrix $R$ is a rotation matrix and $\delta$ is a translation vector. Generally, we collect at least 4 point pairs to calibrate the coordinate transformation function. After substituting the coordinates of all point pairs, we can use least squares to get the regression of $R$ and $\delta$.

### 3) *Occlusion Handling*

When a pedestrian is blocked by another object, extracting clean and complete foreground image becomes an impossible task. Fig. 4 shows two main cases of occlusion. In Fig. 4(a), the pedestrian is partly-blocked by a pillar. Analogously, static obstacles can hide pedestrians in the back, thus the extracted foreground masks are incomplete. Fig. 4(b) shows another situation, one pedestrian is blocked by another. In this case, the foreground masks of two pedestrians roll into one. Once the camera is not able to catch a clear image of the pedestrian, we cannot obtain its accurate location information.

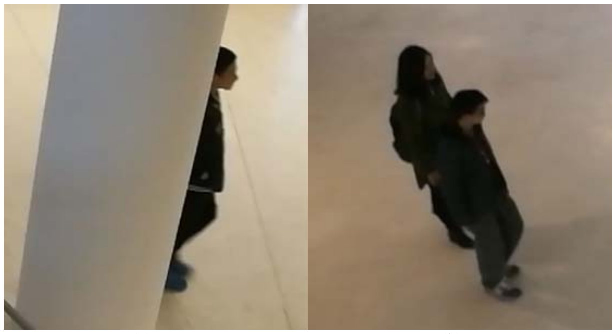
In this case, we run a CNN-based object tracking algorithm to continue tracking. In our implementation, the CNN net is based on MDNet [13], which has 3 convolutional layers and 3 FC (Fully Connected) layers. The network is not very deep because deeper CNN may dilute spatial information, which is significant in localization tasks.

Initial input of CNN is the bounding box of the target pedestrian's foreground image in the last not-blocked frame. The input bounding box is a minimal plumb rectangular region that covers the pedestrian's foreground mask. In our work, a pedestrian is considered to be blocked when the camera cannot see both of his feet simultaneously for several frames. We can see in Fig. 3 that when a pedestrian is not blocked, the camera can always capture both legs in one of several consecutive frames. Note that there could be other ways to judge whether a pedestrian is blocked. Once a pedestrian is judged to be blocked, the system traces back to the last frame when the pedestrian is not blocked and runs the CNN algorithm to continue tracking, until the image of the pedestrian becomes clear again.

Then CNN reads the subsequent frames, tracks the pedestrian and outputs the predicted bounding box of the target in the following frames. The parameters of convolutional layers is pre-trained offline while the parameters of the FC layer are updated online.

The experimental results show the tracking performance of CNN algorithm is good, but the positioning accuracy is lower than the situation in Section III(C). This is because with occlusion, the available foreground information to the CNN tracking algorithm is incomplete, and we can just predict the position point as the midpoint of the rectangle's lower border (see Fig. 2(d)).

### 4) *Gait Feature Extraction and Pedestrian Identification*



(a) (b)

Fig. 4. Occlusion circumstances.

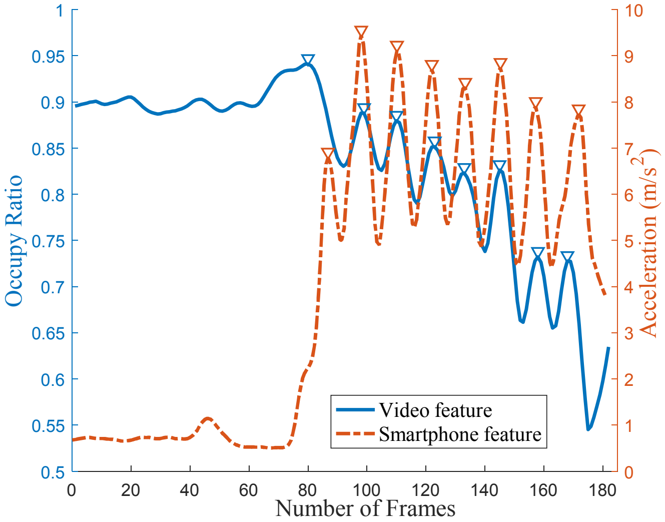


Fig. 5. Gait feature patterns from video and smartphone.

In a surveillance camera video, there are always more than one pedestrians in most cases. The problem is how to recognize the pedestrian of interested out of the crowd. Face-recognition systems don't fit this situation because the surveillance cameras in real-life usually have low resolution, thus the pedestrians' faces in the video are too blur to be recognized. What's more, one surveillance camera can catch the face of a pedestrian only when it is facing the camera. However, we can extract a lot of features from a surveillance video sequence which are critical in identifying pedestrians. Gait feature pattern and heading azimuth are two of the most notable features [14], [15]. There exists consistency in features between video and smartphone of the same pedestrian, because the two kinds of features can faithfully present the motion information of the same pedestrian.

- **Gait feature pattern:** Gait features show the walking state of a pedestrian. It can tell us its walking frequency and whether the pedestrian is walking or not. Our system extracts gait feature patterns from both video sequence and smartphone inertial sensors in different forms. Fig. 5 illustrates the normalized and smoothed gait feature patterns.

    a) Video gait feature pattern is a curve of the occupation ratio $R$ that denotes the relative area of the video pixels on pedestrian's legs. The occupation ratio $R$ can also be expressed as follows

$$R = \frac{N_{lower}}{h_{bb} w_{bb}} \qquad (3)$$

where $N_{lower}$ is the number of pixels of the foreground mask in the lower half of the bounding box (see the yellow rectangle in Fig. 2(d)), while $h_{bb}$ and $w_{bb}$ are respectively the pixel-wise height and width of the bounding box. We have observed that when a pedestrian is walking, the occupation ratio $R$ has a periodic change. The open legs occupy more pixels than overlapping legs. In Fig. 5, we can see that in frames 80-180, the gait feature curve is periodic because the pedestrian is walking. Comparatively speaking, the pedestrian stands still in frames 0-80 so that the curve shows no periodic feature.

b) Smartphone gait feature is the synthetic acceleration magnitude of the smartphone that is measured by the built-in inertial sensors in the smartphone. The accelerometer delivers three-dimensional acceleration $A(a_x,\ a_y,\ a_z)$. Note that the raw acceleration $A$ has a gravity component $G(g_x,\ g_y,\ g_z)$, which can be measured by the built-in gravity sensor in the smartphone. The relative acceleration $A^*$ is the acceleration without gravity component, in other words, $A^*$ is the synthetic motion acceleration of the smartphone whose magnitude $|A^*|$ is calculated by

$$|A^*| = \sqrt{(a_x - g_x)^2 + (a_y - g_y)^2 + (a_z - g_z)^2} \quad (4)$$

In our work, the pattern of synthetic acceleration magnitude $|A^*|$ is defined to be the gait feature of the smartphone. Similarly, $|A^*|$ is periodic only when the pedestrian is walking.

- **Heading azimuth:** Another feature is the heading azimuth of the pedestrian. In the video, the heading azimuth can be easily extracted by the history path in the last several frames. Differently, in the smartphone, the heading azimuth $\theta$ is defined as the azimuth of the peak acceleration, and can be calculated as follows

$$A_e = A^*_{peak} R_o \quad (5)$$

$$\theta = tan^{-1} \frac{A_e[2]}{A_e[3]} \quad (6)$$

Equation (5) and (6) present a usual practice of Android, where $A^*_{peak}$ denotes the acceleration vector with peak L2-norm value (see marks with red triangles in Fig. 5) in smartphone-based coordinates. Symbol $A_e$ is the acceleration vector in earth coordinates, while $R_o$ is a $3 \times 3$ rotatin matrix of the angle of the smartphone which can be dynamically measured by the built-in magnetometer in the smartphone.

Since we have the features from both video sequence and smartphone, we can identify the target pedestrian who is holding the smartphone in the crowd in the video. The matching rate $M$ between the smartphone feature and video feature can be achieved from Equation (7).

$$M = \alpha M_{ws} + \beta M_{sf} + \gamma M_{hd} \quad (7)$$

In the above equation,

- ✧ $M_{ws}$ denotes the matching rate of walking state in a video sequence of $k$ frames, where walking state indicates whether a pedestrian is walking in a particular frame,
- ✧ $M_{sf}$ is calculated as the difference of the two step frequencies, aiming to measure the similarity of step frequencies, and
- ✧ $M_{hd}$ is the matching rate of heading azimuth in the last k frames, the system has a threshold to judge whether two heading azimuths match in one frame.

The coefficients $\alpha$, $\beta$ and $\gamma$ can be assigned by the user. Generally, $\alpha$ and $\beta$ are bigger while $\gamma$ is much smaller.

Smartphone ceaselessly sends inertial sensor data to the server, then the smartphone features are extracted from sensor data and are used to match the video features of all the pedestrians in the video. The pedestrian in the video with largest matching rate with the features from a smartphone is considered to be the target pedestrian who is holding the smartphone. The system then sends the corresponding position data back to the smartphone.

## IV. Experimental results

In this section, we show our experimental results to evaluate the accuracy of the proposed indoor positioning system.

The experiment site is set in a room with a size of 12m × 10m, the floorplan is show in Fig. 6. A surveillance camera with 480p resolution is installed on the wall at $(6, 0)$ with 3-meter height above the floor. In the process of experiment, the target pedestrian walks along a marked route holding a smartphone with built-in inertial sensors.

The smartphone used experiment is Huawei Honor 8 with Android 7.0 operate system. The framerate of video sequence is 20fps. The main positioning algorithm is implemented in MATLAB on Ubuntu 16.10 using MatConvNet deep learning framework [16]. The computer hardware configuration is: Intel i7-4790k CPU with 16G memory + NVIDIA GTX1080 GPU.

Fig. 6 illustrates a sample of experimental result data. The experimenter walks along the black dashed line, while the blue solid line indicates position coordinates reported by the proposed system. The positioning error is defined as the L1-norm difference between reported position and real position. The average positioning error is 8.81cm when the experimenter encounters no occlusion circumstances. When the experimenter is partly blocked throughout the whole route, as shown by the red solid line, the average error increases to 20.26cm. In general cases, the error changes between the values of two limit cases. Fig. 7 show the cumulative distribution function (CDF) curves of the error in two cases. Without occlusion, the error of almost 95% samples is smaller than 20cm, while with occlusion, the error of about 90% of the samples is under 50cm. The accuracy is enough for most indoor pedestrian positioning applications.

Table 1 compares our methodology with related works. RF-based methods including fingerprinting (FP) [17] and received signal strength indicator (RSSI) [18] are easy to use, but RF based systems have big error and cannot give orientation

TABLE I. COMPARISON OF THE POSITIONING PERFORMANCE OF SMARTPHONE-BASED INDOOR POSITIONING SYSTEMS

| Reference | [17] | [18] | [19] | [20] | [21] | Proposed |
|---|---|---|---|---|---|---|
| Method | RF FP | RF RSSI | PDR | VLC | Fusion method | Camera |
| Easy to use | Yes | Yes | Yes | No | No | Yes |
| Report orientation | No | No | Yes | Yes | Yes | Yes |
| Robustness | Medium | Medium | Low | Medium | High | High |
| Overhead | Low | Low | Low | High | High | Low |
| Average error (cm) | 230~460 | 161 | 305 | 5 | 14 | 8.81~20.26 |

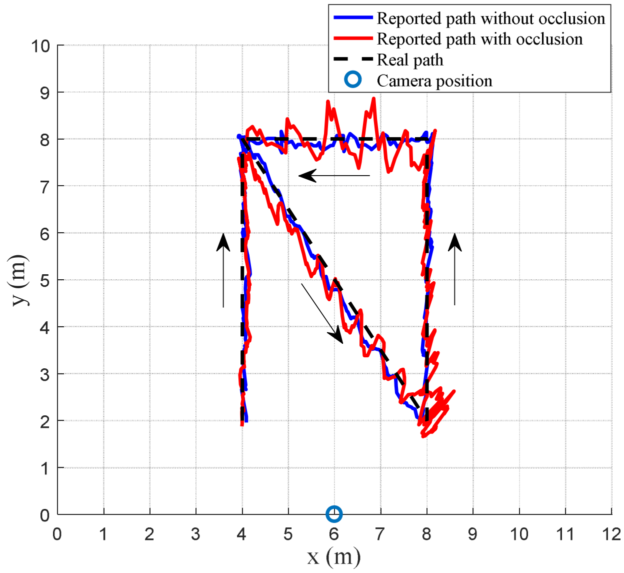


Fig. 6. The reported and real positional trajectory of a pedestrian.

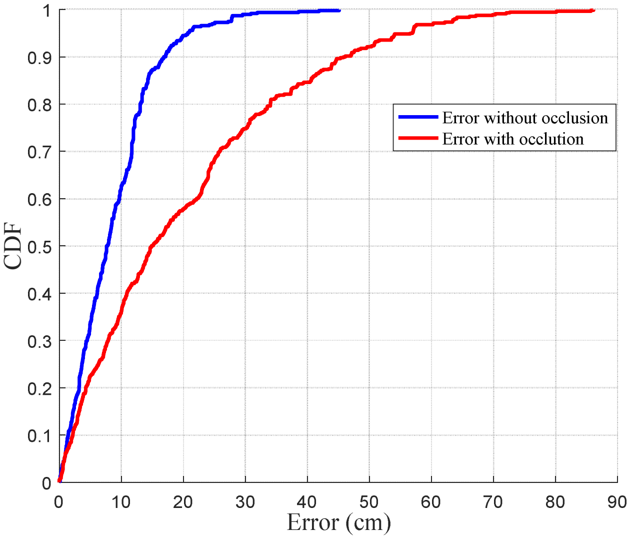


Fig. 7. CDF of positioning error.

information. PDR system [19] show even bigger error. The VLC-based system [20] and the fusion method [21] integrating VLC and PDR are difficult to use because the users have to aim the smartphone camera to the light source, meanwhile modulating light source is costly. Relatively speaking, our proposed indoor positioning system is very easy to use, and performs relatively high accuracy with low overhead.

## V. CONCLUSION

This paper has presented a novel indoor positioning system integrating low-resolution surveillance camera and smartphone inertial sensors. The system takes advantage of the surveillance cameras that are already set in the building and locate the pedestrians in the video. A CNN-based tracking algorithm is applied to handle occlusion situations, meanwhile the target pedestrian is identified via gait features and orientation information. The average positioning error is 8.81~20.26 centimeters, which is fairly low. Our work is a successful attempt to apply the idea of computer vision in indoor positioning area.

## VI. ACKNOWLEDGEMENT

The authors would like to thank Prof. Xin Li at Duke University for helpful discussions.

## REFERENCES

[1] H. Liu, H. Darabi, P. Banerjee and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067-1080, Nov. 2007.

[2] R. Harle, "A Survey of Indoor Inertial Positioning Systems for Pedestrians," in *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1281-1293, Third Quarter 2013.

[3] J. Armstrong, Y. A. Sekercioglu and A. Neild, "Visible light positioning: a roadmap for international standardization," in *IEEE Communications Magazine*, vol. 51, no. 12, pp. 68-73, December 2013.

[4] H. S. Kim, D. R. Kim, S. H. Yang, Y. H. Son and S. K. Han, "An Indoor Visible Light Communication Positioning System Using a RF Carrier Allocation Technique," in *Journal of Lightwave Technology*, vol. 31, no. 1, pp. 134-144, Jan.1, 2013.

[5] Kuo. Ye Sheng, P. Pannuto, and P. Dutta. "Demo: Luxapose: indoor positioning with mobile phones and visible light." in *ACM International Conference on Mobile Computing and Networking* , 2014:299-302.

[6] D. Li, L. Xu and E. D. Goodman, "Illumination-Robust Foreground Detection in a Video Surveillance System," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 10, pp. 1637-1650, Oct. 2013.

[7] G. Ding, W. Chen, S. Zhao, J. Han and Q. Liu, "Real-Time Scalable Visual Tracking via Quadrangle Kernelized Correlation Filters," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 140-150, Jan. 2018.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 1725-1732.

[9] T. h. Chen, T. y. Chen and Z. x. Chen, "An Intelligent People-Flow Counting Method for Passing Through a Gate," *2006 IEEE Conference on Robotics, Automation and Mechatronics*, Bangkok, 2006, pp. 1-6.

[10] S. Li and Y. Hai, "Easy Calibration of a Blind-Spot-Free Fisheye Camera System Using a Scene of a Parking Space," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 232-242, March 2011.

[11] R. Cucchiara, C. Grana, M. Piccardi, A. Prati and S. Sirotti, "Improving shadow suppression in moving object detection with HSV color information," in *2001 IEEE Intelligent Transportation Systems. Proceedings*, Oakland, CA, 2001, pp. 334-339.

[12] K. S. Arun, T. S. Huang and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698-700, Sept. 1987.

[13] H. Nam and B. Han, "Learning Multi-domain Convolutional Neural Networks for Visual Tracking," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 4293-4302.

[14] C. Huang, S. He, Z. Jiang, C. Li, Y. Wang and X. Wang, "Indoor positioning system based on improved PDR and magnetic calibration using smartphone," *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, Washington DC, 2014, pp. 2099-2103.

[15] B. Shin *et al*., "Implementation and performance analysis of smartphone-based 3D PDR system with hybrid motion and heading classifier," *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, Monterey, CA, 2014, pp. 201-204.

[16] http://www.vlfeat.org/matconvnet/

[17] K. Chen, C. Wang, Z. Yin, H. Jiang and G. Tan, "Slide: Towards Fast and Accurate Mobile Fingerprinting for Wi-Fi Indoor Positioning Systems," in *IEEE Sensors Journal*, vol. 18, no. 3, pp. 1213-1223, Feb.1, 1 2018.

[18] Z. Li, T. Braun, X. Zhao, Z. Zhao, F. Hu and H. Liang, "A Narrow-Band Indoor Positioning System by Fusing Time and Received Signal Strength via Ensemble Learning," in *IEEE Access*, 2018, no. 99, pp. 1-1.

[19] K. Nguyen-Huu, K. Lee and S. W. Lee, "An indoor positioning system using pedestrian dead reckoning with WiFi and map-matching aided," *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo, 2017, pp. 1-8.

[20] Y. Han, Q. Cheng and P. Liu, "Indoor positioning based on LED-camera communication," *2016 IEEE International Conference on Consumer Electronics-China (ICCE-China)*, Guangzhou, 2016, pp. 1-4.

[21] Z. Li, A. Yang, H. Lv, L. Feng and W. Song, "Fusion of Visible Light Indoor Positioning and Inertial Navigation Based on Particle Filter," in *IEEE Photonics Journal*, vol. 9, no. 5, pp. 1-13, Oct. 2017.