V-Eye: A Vision-Based Navigation System for the Visually Impaired

Ping-Jung Duh, Yu-Cheng Sung, Liang-Yu Fan Chiang, Yung-Ju Chang, and Kuan-Wen Chen

***Abstract*—Numerous systems for helping visually impaired people navigate in unfamiliar places have been proposed. However, few can detect and warn about moving obstacles, provide correct orientation in real time, or support navigation between indoor and outdoor spaces. Accordingly, this paper proposes V-Eye, which fulfills these needs by utilizing a novel global localization method (VB-GPS) and image-segmentation techniques to achieve better scene understanding with a single camera. Our experiments establish that the proposed system can reliably provide precise locations and orientation information (with a median error of approximately 0.27 m and 0.95°); detect unpredictable obstacles; and support navigating both within and between indoor and outdoor environments. The results of a user-experience study of V-eye further indicate that it helped the participants not only with navigation, but also improved their awareness of obstacles, enhanced their spatial awareness more generally, and led them to feel more secure and independent while walking.**

***Index Terms*—Visually impaired, navigation system, user study, global localization, scene understanding.**

I. Introduction

WALKING on an unfamiliar street can be a challenge for people with visual impairments. They can be exposed to danger when encountering unpredictable obstacles, or easily lose their way. To address these issues, most of the better designs for navigation systems for the visually impaired have adopted sensor-based approaches, for example, involving the global positioning system (GPS) or beacons [8], [20]. However, GPS is not always precise enough to be effective for this purpose, especially indoors. Beacons, on the other hand, require pre-deployment, and thus are better suited to supporting indoor rather than outdoor navigation. In other words, neither of these general approaches by itself can effectively and economically support visually impaired individuals' navigation between indoor and outdoor environments. Moreover, both share two additional major limitations: difficulty in identifying unexpected obstacles, and problems with obtaining precise orientation information in real time. Being unaware of unexpected dynamic obstacles is self-evidently dangerous, while a lack of up-to-date orientation information means that users will take longer than necessary to plan and adjust their routes. To address both these problems, we propose a vision-based solution. With the rapid development of multimedia and computer vision technologies, vision-based positioning [28], [31], [33], [43] and semantic segmentation [24], [34] have been widely explored and they are also the key components of a navigation system for direction guiding and obstacle avoidance, respectively. In addition, several multimedia research [42], [49] have shown audio or haptic feedback are useful to convey visual information to visually impaired people.

Amid a recent wave of research on computer vision, various vision-based solutions have been developed to guide the visually impaired [6], [25], [29]. For example, Kanwal *et al.* [27] used depth cameras to estimate the distance between users and objects in their surroundings. However, the weight of such cameras makes equipment relying on them difficult to wear; nor are they well suited to outdoor environments, at least during periods of bright sunlight. To overcome these drawbacks, we developed V-Eye, a navigation system that uses a lightweight and inexpensive monocular camera as its major sensor for both global localization and scene understanding (Fig. 1). As this paper will demonstrate, V-Eye can determine the exact position and orientation of visually impaired people in real time, and recognize the key elements of the surrounding scene accurately, including the foreground and moving objects. In addition, V-Eye enables visually impaired people to navigate between indoors and outdoors. Its novel localization approach, VB-GPS (Vision-based Global Position System), combines two vision-based localization approaches – visual SLAM [31] and model-based localization (MBL) [37] – to achieve real-time, drift-free, and highly accurate positioning with a median error of approximately 0.27m and 0.95°. Additionally, to understand the scene surrounding the user, V-Eye acquires both constant and dynamic information using semantic segmentation (Fig. 2). Our user study of V-Eye with eight visually impaired people walking on a university campus found that most of them could successfully complete tasks at their typical speed; and all participants said

Manuscript received October 16, 2019; revised March 10, 2020; accepted June 2, 2020. Date of publication June 10, 2020; date of current version May 26, 2021. This work was supported in part by the Ministry of Science and Technology of Taiwan (MOST 107-2221-E-009-148-MY2, in part by MOST 108-2221-E-009-067-MY3, and in part by MOST 108-2218-E- 369-001-). The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Han Hu. *(Corresponding author: Kuan-Wen Chen.)*

Ping-Jung Duh and Liang-Yu Fan Chiang are with the Institute of Multimedia Engineering, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: jonntw211@gmail.com; aidr10030625@gmail.com).

Yu-Cheng Sung is with the Institute of Computer Science and Engineering, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: abc87941@gmail.com).

Yung-Ju Chang and Kuan-Wen Chen are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: armuro@cs.nctu.edu.tw; kuanwen@cs.nctu.edu.tw).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2020.3001500

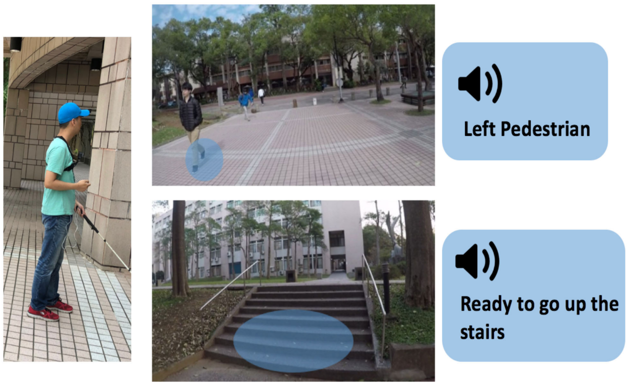


Fig. 1. The proposed system's monocular camera enables it to provide warnings of obstacles and navigation information, including locations and orientation data, based on both high-precision global positions and scene-understanding results. Audio feedback is then given to the user.

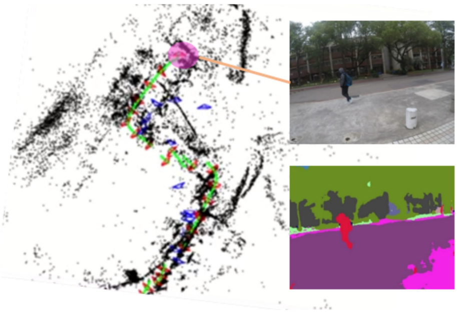


Fig. 2. The localization and semantic-segmentation techniques used in V-Eye to help its users identify their global positions and recognize the key elements of their surroundings, respectively.

that V-Eye was helpful in enhancing their awareness of obstacles, and made them feel more secure and independent while navigating.

The contributions of this paper are as follows. First, it proposes a novel vision-based localization method. Second, its vision-based navigation system is capable of calculating precise positions and orientations, thus helping visually impaired people stay on track during navigation in real-time. Third, it identifies unexpected dynamic obstacles, to reduce the danger to visually impaired users. And fourth, it supports navigation in and between indoor and outdoor environments with a single, easy-to-carry monocular camera.

II. Related Work

Much research on providing navigation assistance to visually impaired pedestrians has been performed with sensor-based or vision-based approaches, and will be reviewed below. And in addition to comparing the proposed global positioning system, this section will discuss vision-based localization and its likely future development.

A. *Navigation Systems for the Visually Impaired*

The most common sensor-based method for outdoor navigation relies on GPS [20], but it is not precise enough for the purposes of the present research, especially in indoor environments. For example, a TP3 system for supporting wayfinding by the visual impaired was found to be critically limited by a lack of adequate data for both indoor and outdoor environments [50]. Another recent solution [8], [38] uses beacons in combination with a smartphone app. However, beacon localization also faces some constraints. First, complete route navigation necessitates the deployment of many beacons, which carries considerable installation and maintenance costs. Second, signal collision may occur if the distances between beacons are too short, and result in incorrect location information. Lastly, this approach cannot easily obtain orientation information, or detect dynamic obstacles at all.

For these reasons, subsequent vision-based research has focused on the use of IR-based depth cameras [27], [29] to support indoor and outdoor navigation by pedestrians with visual impairments. While such technology is useful up to a point, experimental results reveal periodic failures of their infrared sensors, especially outdoors, and that the systems are so heavy that they are difficult for users to carry.

A more accessible approach is a context-aware wayfinding system that guides the visually impaired along a route via QR-code detection using a smartphone camera [25]. However, as with beacons, the deployment of large numbers QR codes at specific outdoor locations is costly. Moreover, none of the systems discussed above can detect unexpected obstacles, or provide up-to-the-minute orientation, or work equally well indoors and outdoors.

Little attention has been paid to the potential of a single monocular camera, combined with computer vision, to improve the effectiveness of such systems. In recent years, semantic image segmentation [3], [24], [34], which deciphers images of surrounding scenery by assigning labels to image pixels, has become an essential component of computer vision. Recent studies [3], [24], [34] that have employed deep-learning network architectures point toward the superiority of such vision-based approaches over sensor-based ones. After careful consideration of such prior work, we chose a vision-based method with a single camera to establish a navigation system that can function effectively both within and between indoor and outdoor environments, and which incorporates a semantic-segmentation system for scene understanding into its localization system for position and orientation estimation.

B. *Vision-Based Localization Systems*

Vision-based positioning technologies can be divided into two broad categories: visual simultaneous localization and mapping (SLAM) and image-based localization. Visual SLAM [17], [26], [28], [31], [33], [40], [44], [47], which runs in real time and is widely used in robotics, concurrently constructs a 3D map and estimates camera angles relative to its starting point. However, it is hindered by accumulated errors; its inability to recover from tracking failures especially when the camera rotates or moves rapidly; and its camera pose estimations being relative rather than absolute. Image-based localization [7], [9], [22], [23], [30], [37], [43], [51], on the other hand, arrives at camera pose by matching a query image with image datasets or pre-trained

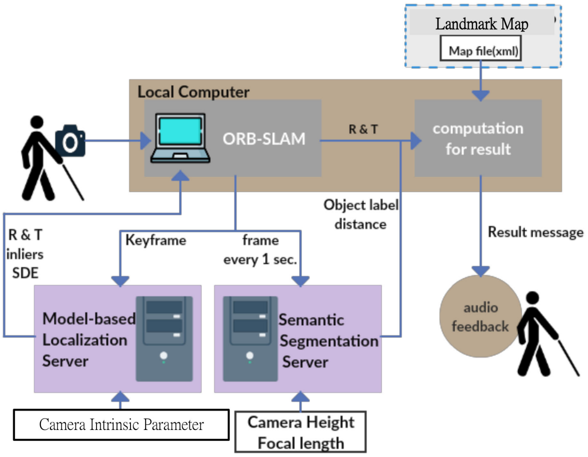


Fig. 3. Architecture of the proposed system.

models. Its main advantage is that it provides a global position, and thus is drift-free. However, it is time-consuming, and does not work well on scenes that differ from those used in its training period, even in relatively minor ways such as illumination. This is a critical, and as-yet unsolved problem for real-world applications.

To fulfill V-Eye's design requirements for real-time, robust, high-precision, and global localization, a novel vision-based positioning method called VB-GPS is proposed in this paper. It integrates both the categories of vision-based positioning technology mentioned above, in an effort to circumvent their limitations while retaining their benefits. Middelberg *et al.* [32] made a similar attempt to achieve real-time, global positioning, but their results were unreliable, because their integration strategy incorrectly assumed that neither visual SLAM nor MBL would experience any localization failures.

III. V-EYE NAVIGATION SYSTEM

The proposed system utilizes five hardware components: two servers, one for MBL and the other for semantic segmentation, a local computer, a wearable camera, and a smartphone for audio feedback. Fig. 3 shows the system architecture. The camera captures real-time images and is connected to the local computer, which conducts the process of visual SLAM – in our implementation, ORB-SLAM [31] – for relative-pose estimation. An integrator merges all the information received from visual SLAM and MBL to estimate final position results, and transmits the keyframes to external servers. A keyframe is an image with a large enough disparity from the previous frames; usually, one keyframe is acquired every 1 to 2 seconds. Based on a chosen keyframe, the system employs a model-based approach to estimate global camera pose in the MBL server [9] while simultaneously performing scene understanding once per second in the semantic-segmentation server. Apart from camera frames, the intrinsic parameters of the camera and its height above the floor/ground are indispensable inputs, which our system assumes will be calibrated in advance. Each time the servers finish the estimation of a frame, the local computer is sent the location information and segmentation results, including objects with distances, by the segmentation server. Local visual SLAM then modifies the position from the data received, and then utilizes the finalized position when searching for landmarks information in the system's landmark map (see III.C, below). After acquiring all these essential elements, the system is able to determine what output message to send, following the rules discussed in Section III.D, below, and send it to the user via a text-to-speech conversion tool.

A. *Localization System: VB-GPS*

To achieve real-time, drift-free, high-precision localization, we devised VB-GPS, and configured our system around it as shown in Fig. 4. A laptop serves as the local computer. It runs visual SLAM to estimate relative poses in real-time, and integrates results from both visual SLAM and MBL with an integrator. The MBL server is a powerful PC that runs MBL [9] to determine the 6-DoF camera pose of the keyframe image in the coordinate system of a pre-constructed 3D point cloud model. Because the model is pre-constructed and fixed, it is possible to align the pre-constructed model to the real world in advance, and obtain global positions.

Our localization system runs on a laptop as local computer with an Intel Core i7-6700HQ (8 cores @ 2.40 GHz) and 16 GB RAM, and a MBL server with an Intel Core i7-6700 (8 cores @ 3.40 GHz) and 8 GB RAM without GPU acceleration. The video sequence was captured by a GoPro5. In the implementation, we resize the images to 640×360 pixels for acceleration. It has shown the proposed system can run in real-time, i.e. 30 fps, to provide global positions.

The three main components in our localization system are visual SLAM, MBL, and the integrator. Visual SLAM runs ORB-SLAM [31] in real time, estimates the relative pose of every frame, and sends the results to the integrator. It also extracts the keyframe, which is the image with a large enough disparity compared with previous frames, and sends it to the MBL server (approximately every 1 to 2 s). After receiving a keyframe, MBL will match the keyframe with a pre-constructed 3D model to estimate global pose, and then send the pose, the number of inliers, and the degree of feature distribution to the integrator at the local computer. It should also be noted here that MBL does not need to run in real time, because the timestamp of the transmitted keyframe is recorded and used in the integrator. Finally, the integrator merges all the information received from the visual SLAM and MBL components, and handles the failures of both to provide the final positioning results. More details concerning these three components will be explained as follows.

1) Visual SLAM: To the best of our knowledge, ORB-SLAM [31] is the most reliable and real-time visual SLAM for monocular camera. In our implementation, we use ORB-SLAM as the visual SLAM method. Here, we use only the relative pose every two consecutive frames provided by ORB-SLAM in our system. In addition, we send a keyframe selected by ORB-SLAM to the server for global pose estimation to eliminate accumulated errors.

Furthermore, as mentioned in the previous section, ORB-SLAM or visual SLAM methods are unreliable under rapid rotation and motion. If these conditions happen, ORB-SLAM will

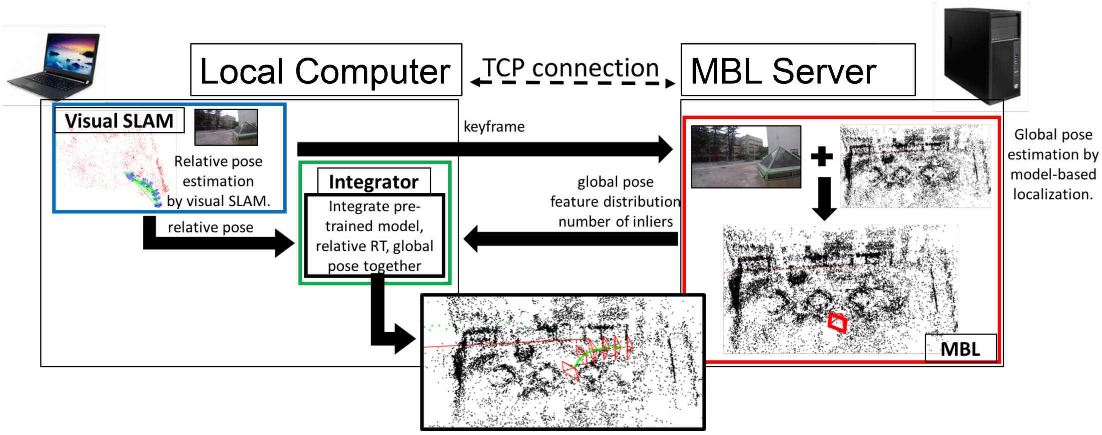

Local Computer
TCP connection
MBL Server
Visual SLAM
Relative pose estimation by visual SLAM.
relative pose
keyframe
Integrator
Integrate pre-trained model, relative RT, global pose together
global pose
feature distribution
number of inliers
Global pose estimation by model-based localization.
MBL

Fig. 4. System overview of VB-GPS, showing its three main components: visual SLAM, MBL, and an integrator.

lose tracking, and pose estimation will be interrupted until the camera moves back to the previous path and loop closing recovers the system. In our implementation, we solve this problem by storing the located camera pose on the global server, and restarting ORB-SLAM in the client when the ORB-SLAM system is interrupted because of tracking loss. Here, ORB-SLAM checks the lost tracking event for every frame. If there are insufficient number of matched features (i.e. fewer than 15 in the implementation) between current frame and current reference keyframe, the state of system will be set as lost tracking. Note that in the VB-GPS, visual SLAM only provides relative poses, and thus it is easy to replace ORB-SLAM with another method if a new and better-performing visual SLAM is proposed in the future.

2) Model-Based Localization: To estimate the global position of each keyframe in the MBL server, we build it with our previous work [9], which is one of the state-of-the-art MBL method. During the training period, we first use an SfM [48] algorithm to construct a 3D point cloud model. After receiving a keyframe, we extract features from the image and then estimate the 2D-to-3D correspondences to calculate the global camera pose of the keyframe in the 3D model with RANSAC. Unfortunately, MBL is mainly relied on 2D-to-3D feature matching and thus may sometimes output inaccurate locations due to scene changes (such as illumination changes) or view angle differences between current testing and previous training period.

Here, we deal with the illumination differences between training and testing, by constructing a model pool instead of using only a single model. In [9], we have shown a small number of models is sufficient for the daytime scenarios (eight models, including of sunny, cloudy, rainy, etc.) and proposed a model update algorithm. In our implementation of VB-GPS, we build three models for each outdoor task, where they are built in the sunny morning (about 10 am.), sunny afternoon (about 2 pm.) and cloudy day. Moreover, the proposed VB-GPS does not assume all the results from MBL should be correct and will compensate the error with visual SLAM, and thus it is more robustness to scene changes. Therefore, only a small number of models is required in our system. An experiment to evaluate its robustness to scene changes is shown in Section IV.B.

However, even with such model pool to deal with illumination changes, MBL may sometimes output inaccurate locations due to scene changes or view angle differences between current testing and previous training period. To eliminate the influence of such failures in our system, we propose a method to verify the correctness of the MBL results. Our system records the number of inliers and the degree of feature distribution, and then sends them to the integrator together with the global positions. Inliers represent good matching features that satisfy the geometrical constraints between the 2D image and the 3D model. In general, low accuracy occurs often with few inliers. Furthermore, Sattler [37] points out the importance of feature distribution in the image, and that a greater number of inliers cannot represent the absolutely right. That is, the accuracy will decrease when the matched features are centralized in a particular region. On the contrary, it often leads to more reliable localization results when features are evenly distributed. Therefore, we quantify the feature distribution as a number and send it to the integrator as another condition to verify the correctness of the global pose. The degree of feature distribution is calculated by fitting an ellipse, which is estimated by the standard deviations of all 2D feature points along two main axes, to representing the feature points first, and we then get a value proportional to the area of the ellipse, which represents the degree of feature distribution. Smaller values mean a higher degree of feature centralization.

Table I shows an evaluation of how the number of inliers and the degree of feature distribution influence the positioning accuracy and passing ratio of MBL. Here, we test all the keyframes (424 frames) in the test sequence of scene *Square* in Section IV. As we can see, omitting the MBL results when there are insufficient number of inliers or their distribution area is small will improve the positioning accuracy of MBL but also decrease the number of passing frames. In our implementation, we selected 40 for the threshold of number of inliers and 1/8 of the image area for the threshold of feature distribution, which provide both high precision and high passing ratio in our experiments.

3) Integrator: This component integrates both results from visual SLAM and MBL, and outputs the final positioning results. The flowchart is shown in Fig. 5. The integrator receives relative pose from visual SLAM at every frame, and outputs the final results directly if no results are received from MBL. Or, if MBL sends the global pose, number of inliers, and degree of feature

TABLE I

POSITIONING ACCURACY AND PASSING RATIO OF MBL FOR SCENES *SQUARE* WITH DIFFERENT THRESHOLDS OF THE NUMBER OF INLIERS (20, 40, AND 60) AND THE DEGREE OF FEATURE DISTRIBUTION (1/4, 1/8, AND 1/16), WHICH ARE THE RATIO OF AREA OF FITTED ELLIPSE TO THE AREA OF IMAGE

	Threshold for degree of feature distribution								
	1/4			**1/8**			**1/16**		
Threshold for number of inliers	**Passing Ratio (#Frames)**	**Positioning Accuracy**		**Passing Ratio (#Frames)**	**Positioning Accuracy**		**Passing Ratio (#Frames)**	**Positioning Accuracy**	
		Mean	**Median**		**Mean**	**Median**		**Mean**	**Median**
20	257/424	0.60m	0.20m	386/424	0.60m	0.23m	400/424	0.61m	0.26m
40	220/424	0.38m	0.16m	295/424	0.41m	0.20m	297/424	0.43m	0.19m
60	30/424	0.32m	0.10m	51/424	0.37m	0.28m	60/424	0.39m	0.26m

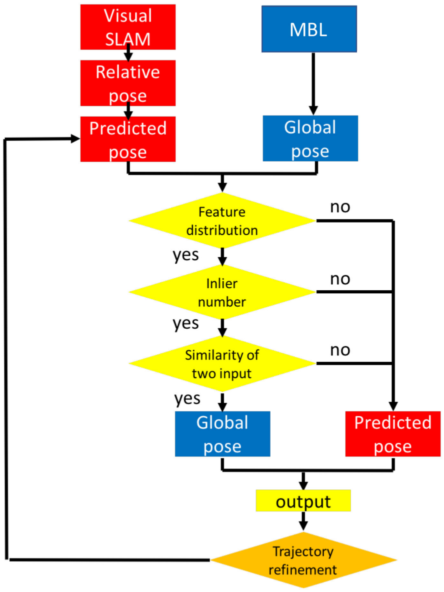


Fig. 5. The flowchart of Integrator, which integrates the results from visual SLAM and MBL.

distribution of a keyframe, the integrator integrates both results and updates the current trajectory.

Before integrating both results, where one is the predicted pose estimated from the relative poses of visual SLAM and the most recent keyframe, and the other is a global pose calculated from MBL, we have to align both poses in the same coordinate system in advance. To achieve this aim, we first compute the global pose of the first keyframe of visual SLAM and consider it as the origin of our trajectory. Then, because monocular visual SLAM is without scale information, we use the poses of the first two keyframes in visual SLAM and the corresponding poses of the same frames in MBL server to calculate the scale relationship between server and client. However, the scale of the trajectory estimated by visual SLAM may not be always fixed as a result of the tracking failures, and may affect the integration results. We solve this scale drift problem by using an adaptive scale revision method to adjust the scale over time. In short, it allows the relative scale between visual SLAM and MBL being changeable. Where we update the scale at every integration process and use the global distance between two keyframes from MBL to refine the scale of visual SLAM. After that, we will have a local pose predicted by visual SLAM and a global pose estimated by MBL in the same coordinate system.

We then determine whether current global pose estimated by MBL is reliable or not. If the global pose is considered reliable, we will output the global pose as the current position and apply trajectory refinement to optimize the trajectory, or the local predicted pose will be regard as current pose. To decide whether the global pose from MBL of current keyframe is reliable or not, we check it by the following steps.

Firstly, we check the degree of feature distribution, i.e., ellipse area values derived by standard deviation ellipse (SDE) [19], and ignore the MBL results when its value is smaller than a threshold, which is set to one eighth of the image area in our implementation. Secondly, the number of inliers is checked. A smaller value is considered to indicate fewer good matches between the keyframe and the 3D point cloud model. Therefore, we discard the results with insufficient inliers (40 in our implementation). Thirdly, to ensure the accuracy of the global pose, we compare the global pose from MBL with the pose predicted by the relative pose, and compute their similarities in rotation and distance. According to our observations, ORB-SLAM usually provides a good rotation estimation, and thus if the rotation difference between the two estimations is larger than a threshold angle t_a, we consider it incorrect. In addition, the distance between the predicted and global pose should not be too large owing to our adaptive scale revision and relative pose estimation, so we discard the global pose when it is far from the predicted one by checking its distance against a threshold t_d. In our implementation, we use 10° for t_a, 1.2 m as t_d for outdoor conditions, and 0.6 m as t_d indoors. The details of the integration algorithm are shown in Algorithm 1.

Once the global pose of the keyframe is selected, trajectory refinement will be required to optimize the trajectory by updating the pose of the target keyframe from the predicted local pose by visual SLAM to the estimated global pose by MBL. Here, we apply the trajectory refinement to the section between the latest keyframe and the previous one to make the final trajectory smooth. Fig. 6(a) shows an example to represent the method, where A, C, B represent the position of the latest keyframe estimated by MBL, the position of previous keyframe estimated by MBL, and the position of latest keyframe estimated from the relative poses of previous frames with visual SLAM, respectively. A and B will usually different because of drift error of visual SLAM. After the integration process, B will be corrected to A,

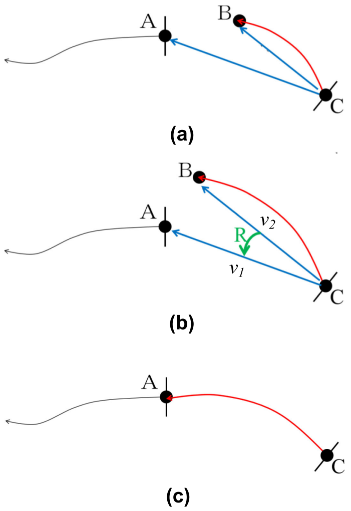


Fig. 6. An example of trajectory refinement: (a) represents the original trajectory without any refinement, (b) trajectory with the scale-refinement, and (c) the final trajectory after trajectory refinement.

Algorithm 1: Integration of Predicted and Global Poses

```
Input: Predicted pose from client, $P_c$; Global pose from
  MBL server, $P_s$; Number of inliers, $N_i$; Area of feature
  distribution, $A_f$; inlier threshold, $t_i$; distance threshold,
  $t_d$; angle threshold, $t_a$.
if $A_f < \frac{1}{8}$ image Area then
  return $P_c$;
else if $N_i < t_i$ then
  return $P_c$;
else if $distance\ between\ P_c\ and\ P_s > t_d$ then
  if $angle\ between\ P_c\ and\ P_s > t_a$ then
    return $P_c$;
  end if
else
  return $P_s$;
end if
```

but there is still a gap between previous trajectory from C to B and the new keyframe position A. To fill the gap and make the trajectory from C to A smooth, we apply the trajectory refinement here. First, we apply the revised scale estimated by adaptive scale revision, which makes the scale between C to B and the scale between C to A being the same, as shown in Fig. 6(b). After that, denote two vectors v_1 and v_2 the vector from C to A and the vector from C to B, respectively (Fig. 6(b)). We then calculate a rotation matrix ($\mathbf{R}$ in Fig. 6(b)) from v_2 to v_1. Finally, we apply the rotation matrix $\mathbf{R}$ to all images of the trajectory between C and B. It will refine the position B to A and make the trajectory smooth as shown in Fig. 6(c). Here, the trajectory refinement method will ensure the positions of keyframes estimated from MBL fixed, the relative rotation angles between two consecutive frames estimated from visual SLAM fixed, and also make the final estimated trajectory smooth.

It should be borne in mind that the proposed integration method is flexible, and allows the methods used in both the client server (for relative-pose estimation) and the MBL server (for global positions) to be replaced in the future if better methods arise. Moreover, our system's built-in assumption that the results from visual SLAM and/or MBL will sometimes be incorrect will render it more reliable in the real world than previous systems incorporating those methods.

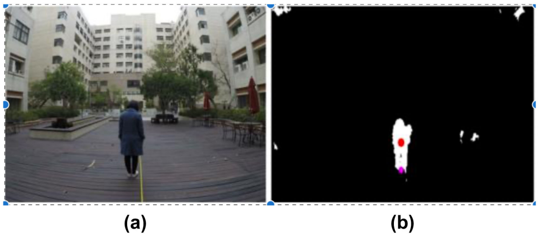


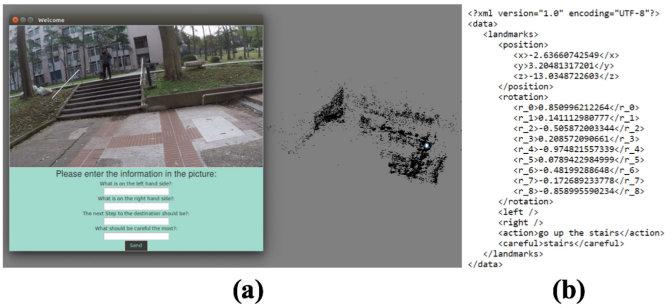
Fig. 7. An example of (a) original image and (b) its segmentation results of human label by FRRN. Red point represents the cluster center estimated by Mean Sift and pink point is the interaction point between the object and the ground plane.

B. Semantic-Segmentation System

Our system incorporated a semantic-segmentation system based on FRRN [34]. Pre-existing trained models used with outdoor/street datasets, such as Cityscapes [10], can perform test procedures in real time. Notice that we do not use object-level detection method here, it is because object-level detection method can detect only specific and pre-trained object, such as human, car, etc. On the contrary, semantic-segmentation will try to segment all pixels in the image and label even unknown object. We think it will be a more flexible and safer choice for visually impaired people navigation. For street scenes, FRRN was found to be adequate to our system's requirements, since we were primarily interested in dynamic objects or other unexpected obstacles that could not be obtained from the 3D model maps already in the localization server. An example of semantic segmentation is shown in Fig. 2. After recognizing objects, the system calculates their respective distances from the user. For each class of object, considering every labeled pixel a feature, a density estimation is arrived at based on feature space-based analysis. For clustering, we chose the robust approach Mean Shift [11], which assigns points to move toward the closest cluster iteratively until convergence occurs. An example of Mean Shift clustering is shown in Fig. 7. After mean-shift clustering, we are able to compute cluster centers (the red point in Fig. 7(b)); and along a vertical line from the center, a point of interaction (the pink point in Fig. 7(b)) between the object and the ground plane can be determined as the foot position of the object. To estimate the distance between the user and an obstacle using only the monocular camera, we adopted an imaging-geometry approach [36]. In our experiments, this yielded an estimate of 4.9 miles for an actual distance of 5 miles: an accuracy sufficient for our system's purposes.

C. Landmark Map

In the same way as typical navigation software provides navigation instructions when its users are close to a decision point,



Please enter the information in the picture:

What is on the left hand side?:

What is on the right hand side?:

The next Step to the destination should be?:

What should be careful the most?:

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
   <landmarks>
      <position>
         <x>-2.63660742549</x>
         <y>3.20481317201</y>
         <z>-13.0348722603</z>
      </position>
      <rotation>
         <r_0>0.850996212264</r_0>
         <r_1>0.141112980777</r_1>
         <r_2>-0.505872003344</r_2>
         <r_3>0.208572090661</r_3>
         <r_4>-0.974821557339</r_4>
         <r_5>0.0789422984999</r_5>
         <r_6>-0.48199288648</r_6>
         <r_7>-0.172689233778</r_7>
         <r_8>-0.858995590234</r_8>
      </rotation>
      <left />
      <right />
      <action>go up the stairs</action>
      <careful>stairs</careful>
   </landmarks>
</data>
```

Fig. 8. An example of our landmark-map approach. (a) The 3D point cloud of the scene. Clicking a point on the model will show the corresponding image taken at that point, following which, four questions are asked. (b) After sending the answers, the information will be saved in an xml file.

V-Eye provides visually impaired people with information about nearby obstacles and navigation instructions when they are close to those obstacles and decision points [45]. Static obstacles such as stairs, fences, etc., and decision points are collectively referred to as *"landmarks"* for purposes of the present research. To continuously provide obstacle and navigational information, we created a landmark map within our system, firstly, by designing an interface that allowed people to manually add landmarks for visually impaired people. Fig. 8(a) shows an example of landmark information being entered using the interface. When a point in the 3D model on the right is clicked, the system will show some candidate images whose localizations are close to the clicked point. Then, humans can select one image for annotation or click another point on the model. After the selection, the corresponding image taken at this point appears on the left, followed by four questions for collecting the relevant information about it. These are *"What is on the left-hand side?" "What is on the right-hand side?" "What is the next step to the destination?"* and *"What should you be most careful about?"* The answer data is automatically saved in an XML file (Fig. 8(b)) which is then read by our system. In that file, a landmark is described in terms of six factors: position, rotation, objects on the left, objects on the right, obstacles that the user should be careful about, and the actions he/she should follow.

D. Obstacle Warnings and Navigation Messages

V-Eye provides visually impaired people with two types of messages: obstacle warnings and navigation instructions. Of the two, we consider the former more important; thus, V-Eye checks for obstacles first, and if no obstacles are detected, provides navigation instructions second. More specifically, the system estimates the global position and orientation of the user. Meanwhile, the semantic-segmentation procedure is performed to detect nearby moving obstacles, and a warning message is generated if one or more are detected. Obstacles are classified into two types: *moving obstacles*, such as other pedestrians and cars, and *static obstacles*, such as lamp posts and railings. Some prior research [21] used meters or feet as the distance units in notifications. However, people who were born blind tend to have less understanding of such units of length than their sighted counterparts [46]. Thus, our system notifies users at pre-set distances, i.e., 2m from a static obstacle (following [41], [52]) and 4m from a moving one, without specifying in the messages what those distances are. The reason the threshold was doubled for moving obstacles was to add a further margin of safety in cases where an obstacle is moving very quickly. Warning messages for both types of obstacles are structured in the format *"direction + subject"* (e.g., *"left, pedestrian"*) as recommended in prior research [13], [39].

When no nearby obstacles are detected, the V-Eye system checks whether, based on the user's current orientation and position, he/she is staying on the right track. It then delivers a navigation message in the format *"ready to + action"* when the user is 2m away from a turning point; and a second message, in the format *"action"*, when the distance is less than 1m. When the user has arrived at the destination, the system generates an arrival message that includes some information about the destination. A demonstration video of V-Eye can be seen at "https://youtu.be/ifwMCcuQSc8", and Fig. 9 is a flowchart of how the system generates its warning and navigation messages.

IV. Evaluation of VB-GPS

In this section, we show the evaluation of the proposed localization method, VB-GPS, first before evaluating the V-Eye system with a user study in Section V. We compare it with three stat-of-the-art methods in six scenes and then demonstrate how the proposed method compensates the failure of MBL when illumination of the scene changes. It shows the robustness of VB-GPS even when MBL becomes worse.

A. Comparison With State-of-the-Arts

To evaluate VB-GPS, we compared it with three state-of-the-art positioning methods – ORB-SLAM [31], MBL [9], and PoseNet [22] – across six scenes. Fig. 10 shows the scenes for comparison, including indoor and outdoor. Among them, the scenes *Square*, *Stairs*, and *Corner* contain challenging routes with multiple rotations in a short period of time. Scene *Garden* contains a simple route, but with pure rotations. Scenes *Plaza* and *Indoor* are easier scenes, with slow motion and smooth rotations. Like our own system, MBL and PoseNet require training data, so we used the same training images for all three of them. We sampled the test sequences at 6 fps for offline SfM [48] to obtain benchmarks for accuracy evaluation, which is a standard way of evaluating vision-based positioning algorithms in outdoor environments [22], [23], [32].

A qualitative comparison results for scene *Square* are shown in Fig. 11. Table II shows quantitative comparison results for all scenes. VB-GPS outperformed the other three methods, with a median error of approximately 0.27m and 0.95° on average. As we can see, MBL performs well except in scene *Indoor*, because the camera view may change a lot in indoor scenes when there is even a slight difference between the view angles of the training and test trajectories, making MBL difficult to match features. A scene with multiple rotations easily causes worse relative pose estimation with ORB-SLAM, but it outperform MBL in indoors because visual SLAM has a smaller drift in small area. PoseNet leads to the worst performance in our experiments, because it

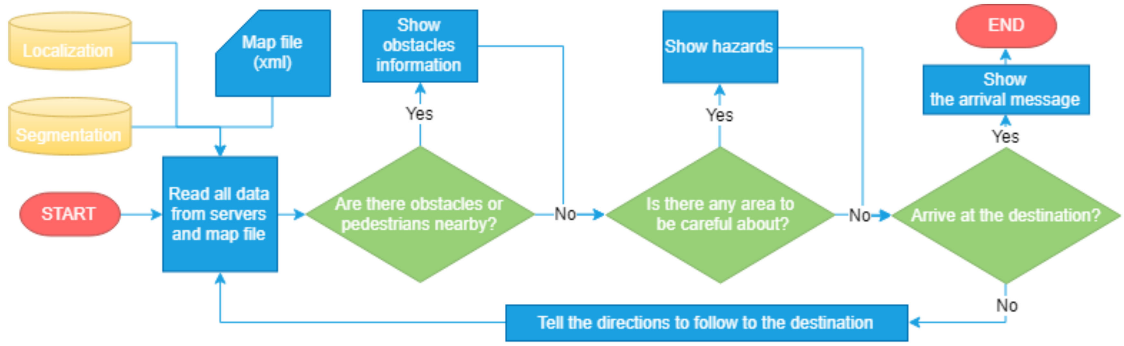

Localization
Segmentation
Map file (xml)
START
Read all data from servers and map file
Show obstacles information
Yes
Are there obstacles or pedestrians nearby?
No
Show hazards
Yes
Is there any area to be careful about?
No
END
Show the arrival message
Yes
Arrive at the destination?
No
Tell the directions to follow to the destination

Fig. 9. System flow for the generation of warning and navigation messages.

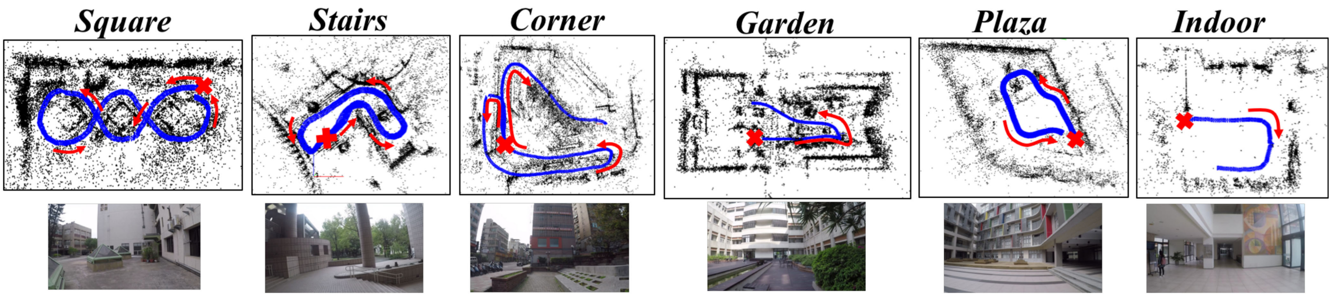


Fig. 10. Six test scenes. The top row shows the 3D point cloud model and test trajectory of each scene, and the second row is one sample image of test video sequences in that scene.

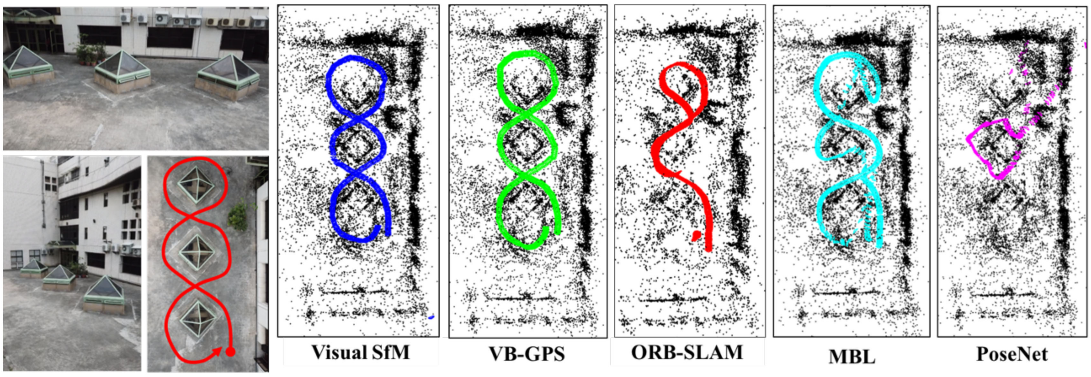


Fig. 11. Comparison of the proposed method (VB-GPS) with the state-of-the-art methods: ORB-SLAM [31], MBL [9], PoseNet [22], and our benchmark generated by Visual SfM [48]. The left-side images show the scene and test trajectory (red curve).

TABLE II
POSITIONING ERRORS OF VB-GPS, ORB-SLAM [31], MBL [9], AND POSENET [22], WITH BOLD FONT SHOWING THE MINIMA

	#Frames		VB-GPS (with refinement)		VB-GPS (without refinement)		ORB-SLAM		MBL		PoseNet	
scene	Train	Test	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Square	240	2164	**0.39m** 1.94°	**0.23m** 0.85°	0.44m **1.53°**	0.26m **0.80°**	2.09m 3.45°	1.39m 1.77°	1.08m 4.98°	0.66m 1.21°	6.83m 132°	6.36m 146°
Stairs	378	2284	**0.13** **1.54°**	**0.09m** **0.93°**	0.25m 1.57°	0.19m 1.04°	3.97m 2.21°	1.65m 1.34°	1.09m 6.66°	0.44m 1.12°	7.35m 81°	7.56m 78.3°
Corner	412	3415	**0.40m** **1.24°**	**0.31m** **0.72°**	0.68m 1.76°	0.59m 0.92°	6.30m 10.97°	5.67m 3.16°	0.50m 3.93°	0.46m 0.88°	10.08m 114°	10.08m 121°
Garden	399	2379	**0.81m** **1.45°**	**0.57m** **1.15°**	0.88m 1.63°	0.69m 1.18°	11.72m 1.27°	2.44m 0.89°	2.05m 8.92°	0.98m 1.41°	16.9m 163°	17.2m 174°
Plaza	355	1958	**0.33m** 1.91°	**0.23m** 0.68°	0.44m 1.06°	0.32m 0.78°	3.02m **0.86°**	0.74m **0.59°**	0.77m 6.82°	0.53m 1.88°	13.3m 165°	11.4m 175°
Indoor	194	681	**0.21m** **1.70°**	**0.21m** **1.36°**	0.25m 2.48°	0.21m 1.90°	0.32m 2.81°	0.13m 1.68°	X X	X X	3.79m 100°	4.44m 120°

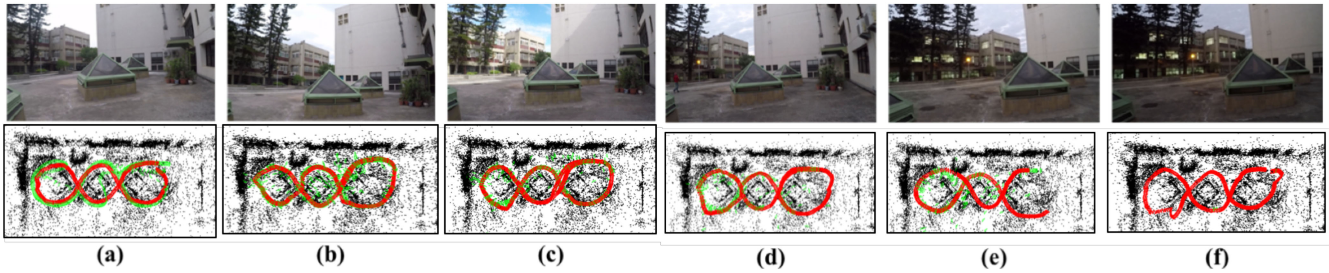


Fig. 12. Results of comparison between VB-GPS (in red) and MBL (in green) under different illumination conditions, where training at 16:00 on one day, and test at (a) the same time slot as training and at (b) 9:00, (c) 12:00, (d) 17:00, (e) 17:30, and (f) 17:50 on a different day.

in particular requires a large amount of training data to learn the neural network. On the contrary, the proposed VB-GPS can work well even with fewer training data. Briefly, the two main reasons for this were (1) that the test sequences' many rotations sometimes led to tracking failures of visual SLAM, and (2) that the training and test trajectories were not exactly the same, which caused the results of MBL and PoseNet to suffer.

B. Robustness Against Scene Changes

To evaluate how the proposed method compensates the error of MBL with visual SLAM when the performance of MBL decreasing. Here we design a scenario of illumination changes to evaluate the robustness of our method, and compare it with MBL [9], which is the only method besides VB-GPS that worked well in the previous experiment. In scene *Square*, we collect multiple test sequences over different time slots (Fig. 12). Here, we train the 3D model for VG-GPS and MBL with the same video sequence taken at 4:00 pm on one day. The results are shown in Fig. 12. It shows even when MBL is not reliable for most of frames due to illumination changes, VB-GPS still can provide a sufficiently good trajectory estimation compared to that of ORB-SLAM or MBL in Fig. 11. It is because the proposed intergrator will not assume both ORB-SLAM or MBL being always correct and try to complement each other.

V. User Study

A. Task Design

To evaluate our system design in various situations while ensuring the participants' safety, we chose a university campus as our test site. We designed three travel tasks, one in an indoor environment and two in outdoor ones. The indoor area comprised straight passageways; the first outdoor environment included a number of buildings; and the second was a more open and wider square. All three tasks were linked into a single task scenario, as explained below.

Task 1: You want to discuss a project with a professor in another department. Your friend drops you off at the front door of the building where the meeting is to take place. However, your friend has a class now, so you have to find the office by yourself.

Task 2: At the end of the meeting, the professor recommends that you borrow a book related to your project topic. The professor cannot help you with this because he has another meeting. However, he has asked a librarian to wait for you with the book at the front entrance to the library. Therefore, you need to go to the library to pick up the book.

Task 3: You have successfully obtained the book and it is time to take a bus back home. Your friend is joining you for this journey and meeting you at a particular bus stop, which you now need to walk to.

Fig. 13 shows the routes of the above task scenario, along with some examples of landmarks at which the participants received navigation messages. For example, in Task 1 (Fig. 13(a)), the system asked participants to be ready to turn right at point #1; told them that they were walking down a straight passageway at point #2; asked them to be ready to turn left at point #3; and told them they had arrived at the meeting place. Similarly, during Task 2 (Fig. 13(b)), the participants received a turn-left message at point #1 and were told to be careful while going up the stairs at point #2. The system provided information about the grass and stones at point #3, and notified the users when they had almost arrived at the library. And in Task 3 (Fig. 13(c)), the participants were told they should be cautious while walking down the stairs at first, and informed of the tactile guide path on the right-hand side. Then, they received a turn-left instruction at point #3, and finally were told that they had arrived at the correct bus stop.

B. Participants

We recruited four male and four female participants, all of whom were visually impaired and aged 22 to 35. All were completely unfamiliar with the test environment. One (P2) had been totally blind since birth; five had poor vision from birth and became totally blind during elementary school (P3, P5, P6, P7, P8). P4 was blind in the right eye and had poor vision in the left; and P1 had poor vision due to cerebral palsy. All habitually used canes when traveling, and were familiar with smartphones. Three participants (P2, P3, and P7) had prior experience of participating in navigation research. Table III shows the demographic information and quantitative results of user study. To further demonstrate how well the proposed solution supported the participants, we also recorded the time of a normal person (referred to as P0) walking along the same trajectory of each task with normal speed as the baseline.

C. Procedure

During the experiment, each participant was asked to perform all three of the tasks described above while wearing a GoPro camera, which transmitted image frames to a laptop via Wi-Fi.

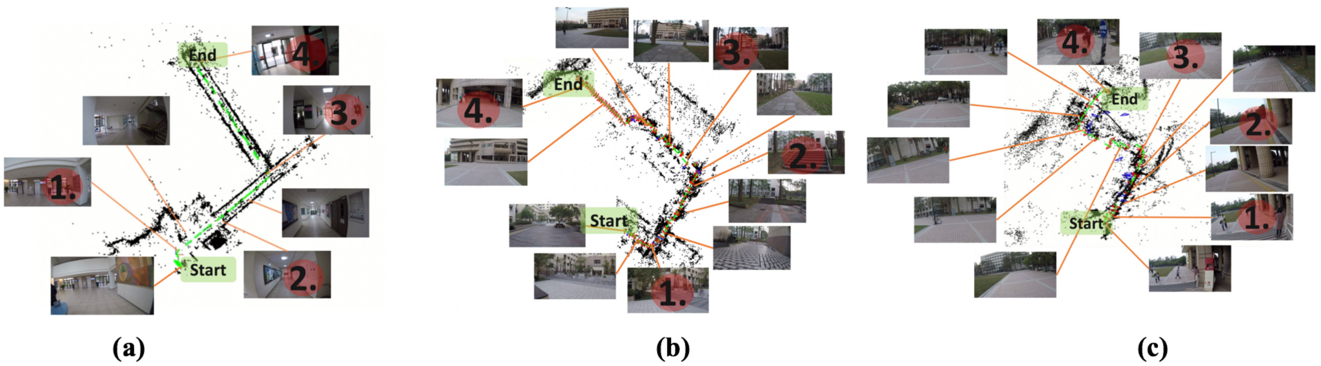


Fig. 13. The routes for (a) Task 1, (b) Task 2, and (c) Task 3, with some examples of features from the system's landmark map.

TABLE III
THE DEMOGRAPHIC INFORMATION AND QUANTITATIVE RESULTS OF THE TEST INCLUDING HOW MUCH TIME (T) IT TAKES TO FINISH EACH TEST, HOW MANY TIMES THE PARTICIPANTS HAVE ENCOUNTERED DIFFICULTY (D) AND THE NUMBER OF MESSAGES (N) THEY RECEIVE IN EACH TASK. WHERE P0 IS A NORMAL PERSON WALKING ALONG THE SAME TRAJECTORY OF EACH TASK WITH NORMAL SPEED

	Gender	Age	Visually Impairment	Mobility Aid	Congenital blindness /Adventitious blindness	Time for three tasks (t1/ t2 / t3)	Difficulty times for three tasks (d1/ d2 / d3)	Number of messages during tasks (n1/ n2/ n3)
P0	Male	38	-	-	-	50s/2'30s/1'44s	-	-
P1	Male	35	Amblyopia due to CP	White cane	Congenital	48s/2'43s/1'48s	0 / 0 / 0	7/ 16/ 14
P2	Male	31	Blindness	White cane	Congenital	1'15s/4'49s/1'48s	2/ 4/ 2	9/ 31/ 22
P3	Male	23	Blindness	White cane	Amblyopia and become blindness	1'18s/4'16s/2'53s	0/ 1/ 1	7/ 22/ 16
P4	Female	32	Blind with right eye, amblyopia with left eye	White cane	Congenital	45s/2'14s/1'27s	0/ 0/ 0	8/ 17/ 11
P5	Female	27	Blindness	White cane	Amblyopia and become blindness	2'18s/7'24s/4'01s	1/ 2/ 2	15/ 24/ 14
P6	Female	22	Blindness	White cane	Amblyopia and become blindness	2'30s/8'59s/5'45s	1/ 4/ 2	20/ 34/ 16
P7	Female	24	Blindness	White cane	Amblyopia and become blindness	46s/4'25s/2'29s	0/ 2/ 1	7/ 25/ 19
P8	Male	23	Blindness	White cane	Amblyopia and become blindness	1'12s/4'31s/3'21s	1/ 1/ 2	9/ 25/ 21

At the start of the study, we provided a brief description of the system and asked participants to answer a background questionnaire. They were then given a system tutorial, which included explanations of the audio feedback and specific messages, followed by a training session on a campus route about 10m in length that was not otherwise included in the experiment. After this training session, the participants were asked to try their best to perform the three tasks, with no time limit, and to tell the study moderator when they thought they had arrived at the final destination. The moderator followed each participant with a laptop running our system through a 4G portable Wi-Fi router, while two other members of the research team recorded video and took notes. After they had completed all tasks, the participants were asked to fill out a questionnaire about the system and participate in a semi-structured interview about their experiences. All interview sessions were audio- and video-recorded for subsequent data analysis, with the participants' permission.

VI. RESULT

All participants completed all three tasks successfully, but the lengths of time this took them varied widely, as shown in Table III. Those participants who rarely took part in orientation and mobility (O&M) training (P5, P6) spent twice as long as the others finishing the tasks. Except for the two with amblyopia, i.e., P1 and P4, who could stay on track easily, the participants had difficulty correcting their direction, and as a result, received an increasingly large number of messages as the experiment went on. Unpredictable factors also played roles in users' outcomes. For example, P2 had more difficulty completing Task 2 than the others because he encountered more pedestrians during it; and P5 received more messages on Task 1 than others, presumably because of her relatively low self-reported spatial awareness. For P6, who usually walked with her parents, the experiment represented her first-ever solo walk, and she unsurprisingly received the most messages. Despite their varying levels of eyesight and experience of walking unassisted, however, none of the participants went the wrong way or failed to reach their three destinations. Our observations and the feedback from the study are discussed in the sections that follow.

A. *Positive Feedback*

Our participants reported that our system, considered as an information provider, was useful to them when traveling in unfamiliar environments, because it helped them to know more

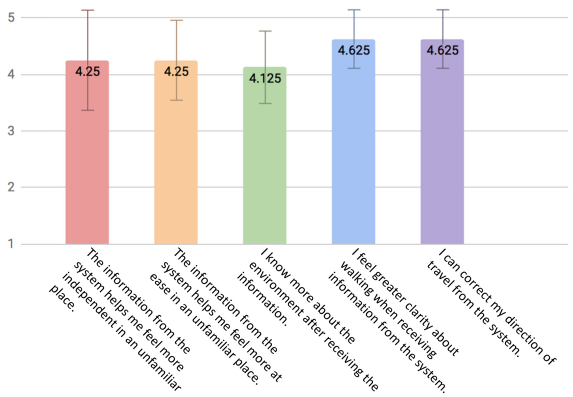


Fig. 14. Questionnaire feedback from the participants, with error bars indicating ± standard error from the mean.

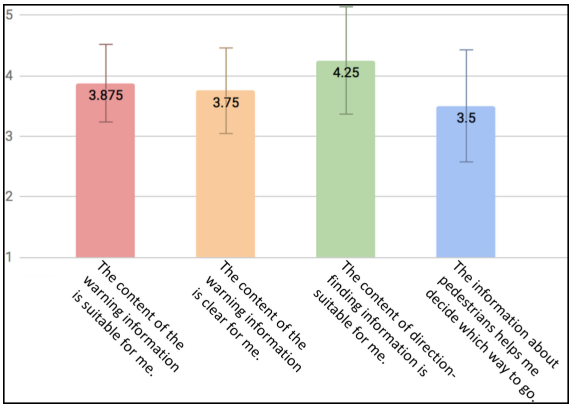


Fig. 15. Questionnaire feedback from the participants regarding the V-Eye system's information, with error bars indicating ± standard error from the mean.

about such environments; to correct their direction of travel when needed; and to feel more confident and independent (specifically, due to the detection of obstacles). Fig. 14 shows the rating from the participants. Most thought that they could obtain sufficient information from our system. Two areas where the system received relatively high amounts of positive feedback are examined in turn in the following two subsections.

Spatial awareness and sense of security. V-Eye allowed the participants to be aware of unexpected dynamic as well as static obstacles near them. As a result, they reported that the system helped them to build a sense of security and to feel more relaxed. As P2 said, *"It definitely makes me feel more at ease and more secure that I can know what things are around and notice the direction of the obstacles."* P8 noted, *"I feel secure because I can understand which way to go and walk at my normal speed, which is usually slower in an unfamiliar place."* Moreover, although P8 was carrying the cane that he normally used when walking, he did not actually make use of it during the experiment, and ascribed this to his confidence in the V-Eye system.

Responsive adjustment facilitates independence. Our system enabled the participants to know where to go and to quickly revise their direction of travel when they made wrong turns, via a simple relative-direction prompts. It was probably due to this feature that none of the participants had to backtrack during their journeys. P5 mentioned that this feature *"helped me a lot since I turn to a wrong direction very often."* Likewise, P7 reported, *"I always have to remember which way to turn by using my cane to hit something. I don't have to memorize so many things if I use the system."*

All participants reported feeling more independent while using the system and most ascribed such feelings to the system's directional information. P6, who had never walked alone before, said: *"I am very happy to walk by myself and don't have to ask others for help."*

P1 commented that, thanks to the system, *"I do not have to ask other people in advance to accompany me when I need to walk in an unfamiliar place."* Similarly, P3 said, *"I didn't know how to go to the destination, but I was really able get there by myself using the system. This was unlike my previous experiences in unfamiliar places."*

B. Individual Differences

Although the participants' feedback was generally positive, we found individual differences in their responses to the information provided, which seemed to be related to their different backgrounds and habits. These findings are divided into four categories in the four subsections below.

Wanting more distance information about obstacles. As noted above, when a participant was within 2m of a static obstacle, our system issued a warning message. Most of the participants quickly became accustomed to this and adjusted their routes in accordance with the information provided. Nonetheless, not every participant thought that such notifications were suitable (see Fig. 15). One, P8, who had a clearer concept of distance than others in the sample, wanted to know how long the stairway was in Task 3; and P2 mentioned that, although he did not have a clear concept of distance, he was used to listening to navigation systems' distance information.

Differing responses to *ready* messages. All participants thought it was useful to receive a *ready* message first, and subsequently a *reminder* message. However, we observed that they had varied reactions to *ready* messages. Four participants (P1, P2, P3, P4) maintained their existing courses until the *reminder* messages were received; but the others (P5, P6, P7, P8) tended to perform the action immediately, i.e., too soon.

Relative direction. Our participants mostly considered that the relative-direction information that our system provided was clear, but again, this reaction was not unanimous. P2, for example, commented that *"Although relative direction is okay, I have had a complete O&M training and I think clock directions will be more suitable for me."*

Information about pedestrians. As shown in Fig. 15, the participants' ratings of the information about pedestrians was relatively low, presumably due to their wide range of opinions about how such information ought to be provided. At one extreme, two participants (P4, P8) thought it was not really necessary to know such information at all, whereas P1 thought it was so helpful that he revised his route immediately if other pedestrians were nearby. P2 and P6 stopped for a moment when they heard the information, and P2 said he actually looked forward to hearing the next instruction regarding what direction he should head in. Lastly, P3, P5 and P7 maintained their existing speeds and directions when hearing this type of information, yet said it might be useful if they wanted to ask people for help.

C. Discussion

Based on our analysis of the above findings, we recommend that users be given a tutorial that includes how to customize the system's settings according to their individual habits and needs. The four settings that we propose to allow users to customize in a future version are discussed in turn below.

Distance information. Users should be able to choose whether or not they want to have distance information included in the instructions they receive, because not every visually impaired person has a strong concept of distance. Those who want simple messages can keep our default settings, while others can ask for additional distance information, e.g., in the formats *"direction + distance + obstacle"* or *"ready to + action + distance"*.

Distance threshold for ready messages. Because the participants' reactions upon hearing ready messages varied sharply, users who tend to maintain their course and speed on hearing them should use our default settings (i.e., 2m). On the other hand, those who tend to take action instantly on hearing them should be able to reduce their distance thresholds to points nearer to upcoming action points.

Clock vs. relative directions. Depending on their backgrounds, and in particular, the extent of their O&M training (if any), clock directions might be more practical for some visually impaired people than our system's existing method of providing directional information.

Pedestrian information. Because perceptions of the usefulness of our system's pedestrian information varied so widely among our participants, future users should be allowed to determine whether they need such information or not.

D. Limitations

This study revealed several limitations of V-Eye that must be acknowledged here. One is that the system encounters delays when communication between the camera and the laptop, and/or between the laptop and the server, is poor; thus, the stability of the entire system should be improved. Although it might be more stable if all the system can run on embedded systems or wearable devices, such as a smartphone. However, we think the computation overhead will charge too much energy of battery of the device. In addition, with the increasing development of communication technologies, such as 5G, we think using internet connection and cloud computing, as the proposed system, will become more feasible in the near future.

Additionally, the system design necessitated that we build a model and prepare information about static objects in advance via a landmark map. However, we believe that it would be feasible to obtain such information via a crowdsourcing approach [4], [35]. And lastly, the localization and scene-understanding components of our system did not run well on rainy days or at night. Though this is a common problem in computer vision, which many researchers have tried to find solutions to, it must be solved before our system's potential to help visually impaired people can be fully realized.

VII. Conclusion

Unlike most recent work in a similar vein, the current study has proposed a novel guidance system, V-Eye, that integrates localization with scene understanding and a vision-based approach using a monocular camera. The proposed system can detect and warn visually impaired individuals about both static and moving obstacles; correct their orientation; and allow them to navigate within and between indoor and outdoor spaces. The participants' responses to using our system indicate that it is a viable and effective approach that enabled them to explore an unfamiliar environment without human assistance. Although it is still a prototype, our system represents a promising avenue for future research aimed at enhancing the spatial awareness of visually impaired people traveling in unfamiliar environments.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski, "Building rome in a day," in *Proc. Int. Conf. Comput. Vision*, 2009, pp. 1–8.

[2] G. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[4] C. Cardonha *et al.*, "A crowdsourcing platform for the construction of accessibility maps," in *Proc. Int. Cross-Disciplinary Conf. Web Accessibility Article*, 2013, pp. 1–4.

[5] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[6] F. Catherine, S. Azenkot, and M. Cakmak, "Designing a robot guide for blind people in indoor environments," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interact. Extended Abstr.*, 2015, pp. 107–108.

[7] D. M. Chen *et al.*, "City-scale landmark identification on mobile devices," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2011, pp. 737–744.

[8] H. E. Chen, Y. Y. Lin, C. H. Chen, and I. F. Wang, "Blindnavi: A mobile navigation app specially designed for the visually impaired people," in *Proc. ACM Conf. Extended Abstr. Human Factors Comput. Syst.*, 2015.

[9] K. W. Chen *et al.*, "Vision-based positioning for Internet-of-Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 364–376, Feb. 2017.

[10] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 3213–3223.

[11] Y. Cheng, "Mean shift mode seeking and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[12] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[13] J. Ducasse, M. Macé, M. Serrano, and C. Jouffrais, "Tangible reels: Construction and exploration of tangible maps by visually impaired users," in *Proc. ACM CHI Conf. Human Factors Comput. Syst.*, 2016, pp. 2186–2197.

[14] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[15] N. Dahlbäck, A. Jönsson, and L. Ahrenberg, "Wizard of OZ studies: Why and how," in *Proc. ACM Int. Conf. Intell. User Interfaces*, 1993, pp. 193–200.

[16] E. Eade and T. Drummond, "Scalable monocular SLAM," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2006, pp. 469–476.

[17] J. Engel, T. Schops, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 834–849.

[18] M. Fischler and R. Bolles, "Random sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[19] J. Gong, "Clarifying the standard deviational ellipse," *Geographical Anal.*, vol. 34, no. 2, pp. 155–167, 2002.

[20] S. Gilson, S. Gohil, F. Khan, and V. Nagaonkar, "A wireless navigation system for the visually impaired," *Capstone Project*, 2015.

[21] J. Guerreiro, D. Ahmetovic, K. M. Kitani, and C. Asakawa, "Virtual navigation for blind people: Building sequential representations of the real-world," in *Proc. Int. ACM SIGACCESS Conf. Comput. Accessibility*, 2017, pp. 280–289.

[22] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 2938–2946.

[23] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 6555–6564.

[24] B. Kang, Y. Lee, and T. Q. Nguyen, "Depth-adaptive deep neural network for semantic segmentation," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2478–2490, Sep. 2018.

[25] E. Ko and E. Y. Kim, "A vision-based wayfinding system for visually impaired people using situation awareness and activity-based instructions," *Sensors*, vol. 17, no. 8, 2017, Art. no. 1882.

[26] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proc. Int. Symp. Mixed Augmented Reality*, 2007, pp. 1–10.

[27] N. Kanwal, E. Bostanci, K. Currie, and A. F. Clark, "A navigation system for the visually impaired: A fusion of vision and depth sensor," *Appl. Bionics Biomechanics*, vol. 2015, Jul. 2015, Art. no. 479857.

[28] H. Luo *et al.*, "Real-time dense monocular slam with online adapted depth prediction network," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 470–483, Feb. 2019.

[29] Y. H. Lee and Gérard Medioni, "RGB-D camera based wearable navigation system for the visually impaired," *Comput. Vision Image Understanding*, vol. 149, pp. 3–20, 2016.

[30] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua. "Worldwide pose estimation using 3D point clouds", in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 15–29.

[31] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[32] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, "Scalable 6-DOF localization on mobile devices," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 268–283.

[33] J. C. Piao and S. D. Kim, "Real-time visual–inertial SLAM based on adaptive keyframe selection for mobile AR applications," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2827–2836, Nov. 2019.

[34] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full resolution residual networks for semantic segmentation in street scenes," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 3309–3318.

[35] M. Rice, R. Daniel, D. R. Caldwell, D. Mcdermott, and F. I. Paez, "Crowdsourcing techniques for augmenting traditional accessibility maps with transitory obstacle information," *J. Cartography Geographic Inf. Sci.*, vol. 40, no. 3: Selected Papers from ICC, pp. 210–219, 2013.

[36] G.P. Stein, O. Mano, and A. Shashua, "Vision-based ACC with a single camera: bounds on range and range rate accuracy," in *Proc. Intell. Veh. Symp.*, 2003, pp. 120–125.

[37] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2D-to-3D matching," in *Proc. Int. Conf. Comput. Vision*, 2011, pp. 667–674.

[38] D. Sato *et al.*, "NavCog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment," in *Proc. Int. ACM SIGACCESS Conf. Comput. Accessibility*, 2017, pp. 270–279.

[39] M. K. Scheuerman, W. Easley, A. Abdolrahmani, A. Hurst, and S. Branham, "Learning the language: The importance of studying written directions in designing navigational technologies for the blind," in *Proc. ACM CHI Conf. Human Factors Comput. Syst. Extend Abstract*, 2017, pp. 2922–2928.

[40] K. Schmid and H. Hirschmuller, "Stereo vision and IMU based real-time ego-motion and depth image computation on a handheld device," in *Proc. Int. Conf. Robot. Autom.*, 2013, pp. 4671–4678.

[41] M. A. Soto *et al.*, "DroneNavigator: Using leashed and free-floating quadcopters to navigate visually impaired travelers," in *Proc. Int. ACM SIGACCESS Conf. Comput. Accessibility*, 2017, pp. 300–304.

[42] P. M. Silva, T. N. Pappas, J. Atkins, and J. E. West, "Perceiving graphical and pictorial information via hearing and touch," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2432–2445, Dec. 2016.

[43] Y. Song, X. Chen, X. Wang, Y. Zhang, and J. Li, "6-DOF image localization from massive geo-tagged reference images," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1542–1554, Aug. 2016.

[44] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 6565–6574.

[45] R. Tscharn, T. Außenhofer, D. Reisler, and J. Hurtienne, "Turn left after the heater: Landmark navigation for visually impaired users," in *Proc. Int. ACM SIGACCESS Conf. Comput. Accessibility*, 2016, pp. 295–296.

[46] S. Ungar, M. Blades, and C. Spencer, "Teaching visually impaired children to make distance judgements from a tactile map," *J. Vis. Impairment Blindness*, vol. 91, pp. 163–174, 1997.

[47] V. Usenko, J. Engel, J. Stuckler, and D. Cremer, "Direct visual-inertial odometry with stereo cameras," in *Proc. Int. Conf. Robot. Autom.*, 2016, pp. 1885–1892.

[48] C. Wu, "VisualSFM: A visual structure from motion system," 2011. [Online]. Available: http://www.cs.washington.edu/homes/ccwu/vsfm/

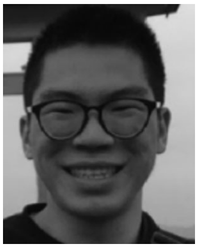
[49] Z. Wang and B. Li, "A bayesian approach to automated creation of tactile facial images," *IEEE Trans. Multimedia*, vol. 12, no. 4, pp. 233–246, Jun. 2010.

[50] R. Yang *et al.*, "Supporting spatial awareness and independent wayfinding for pedestrians with visual impairments," in *Proc. Int. ACM SIGACCESS Conf. Comput. Accessibility*, 2011, pp. 27–34.

[51] A. Zamir, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg, "Wide area localization on mobile phones," in *Proc. Int. Symp. Mixed Augmented Reality*, 2009, pp. 73–82.

[52] L. Zeng, M. Simros, and G. Weber, "Camera-based mobile electronic travel aids support for cognitive mapping of unknown spaces," in *Proc. Int. Conf. Human-Comput. Interact. Mobile Devices Serv.*, 2017, pp. 1–10.

Ping-Jung Duh received the B.S. degree from the Department of Computer Science from National Chiao Tung University, Hsinchu, Taiwan, in 2016 and the Master's degree from the Institute of Multimedia Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2019.



Yu-Cheng Sung received the B.S. degree from the Department of Computer Science from National Chiao Tung University, Hsinchu, Taiwan, in 2016 and the Master's degree from the Institute of Computer Science and Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2018.

Liang-Yu Fan Chiang received the B.S. degree in double specialty program of management and technology from National Tsing Hua University, Taiwan, in 2018 and the Master's degree from the Institute of Multimedia Engineering at National Chiao Tung University, Taiwan, in 2019. She is currently doing research with KU Leuven, Belgium.

Yung-Ju Chang received his B.S. degree from National Chiao Tung University, Taiwan, in 2005, and the M.S, and Ph.D. degrees from the University of Michigan, Ann Arbor, USA, in 2009 and 2016, respectively. He is currently an assistant professor with National Chiao Tung University, Taiwan. His research interests include in the area of human computer interaction and ubiquitous computing, with a focus on mobile attention, human intelligence interaction, mobile crowdsourcing.

Kuan-Wen Chen received the B.S. degree in computer and information science from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and the Ph.D. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2011. He is currently an assistant professor with the Department of Computer Science, National Chiao Tung University. From 2012 to 2014, he was a post-doctoral researcher with National Taiwan University, where he was an Assistant Research Fellow in the Intel-NTU Connected Context Computing Center from 2014 to 2015. His research interests include computer vision, pattern recognition and multimedia.