A Co-Robotic Cane for Blind Navigation

Cang Ye, Soonhac Hong and Xiangfei Qian
Department of Systems Engineering
University of Arkansas at Little Rock, AR, USA
Email: cxye@ualr.edu

Abstract—**This paper presents a new robotic navigation aid, called Co-Robotic Cane (CRC). The CRC uses a 3D camera for both Pose Estimation (PE) and Object Recognition (OR) in an unknown indoor environment. The 6-DOF PE method determines the CRC's pose change by an egomotion estimation method, called Visual Range Odometry (VRO), and the Iterative Closest Point (ICP) algorithm and reduces the pose integration error by a pose graph optimization algorithm. The PE method does not require any prior knowledge of the environment. The OR method detects indoor structures (stairways, doorways, etc.) and objects (tables, computer monitors, etc.) by the Gaussian Mixture Models. Some of structures/objects (e.g., stairways) may be used as navigational waypoints and the others for obstacle avoidance. The CRC is a co-robot. It may detect human intent and collaborate with its user in performing a navigation task. The proposed CRC is the first in its kind.**

Keywords—robotic navigation aid, robot pose estimation, egomotion estimation, pose graph optimization, object recognition.

I. INTRODUCTION

According to World Health Organization, there are 285 million visually impaired worldwide, of which 39 million are blind and 246 million are with low vision. Visual impairment causes loss of independent mobility and results in severe deterioration in the quality of life. Although substantial advances have been made in computer vision and robotics, independent mobility still remain challenging to a visually impaired individual using a mobility tool. Up to date, a white cane remains the most efficient mobility tool to the blind due to its powerful haptic feedback and low cost. However, a white cane is a point-contact device that cannot provide a "full picture" of an object or the surroundings. Also, a white cane has a limited range and is thus too nearsighted for path planning. In addition, a white cane is unable to provide location information for wayfinding.

To address these limitations, Robotic Navigation Aid (RNA) has been introduced to replace or enhance a white cane. A major challenge in developing an RNA is the portability requirement that limits its size, computing power and power supply. These constraints exclude the use of multiple sensors and high-end computers as seen in autonomous robots [1]. As a result, an RNA may only provide very limited assistance to the visually impaired. The existing RNAs can be classified into three categories: Robotic Wheelchair (RW) [2], Robotic Guide-Dog (RGD) [3, 4] and Electronic White Cane (EWC) [5, 6, 7]. A RW is well suited for a blind person who loses the ability to move. However, a RW gives its user an unpleasant sense of being controlled. Safety concerns will keep the visually impaired away from using RWs for their mobility needs. A RGD indicates a walkable direction towards the destination to the blind user. In this case, the user moves by his/her own. A RGD can be passive or active. A passive RGD [3] provides the travel direction to the user by steering its wheels and the user pushes the RGD forward. A passive RG gives its users a sense of safety that they are controlling the device, but it brings extra workload and may cause fatigue to the user. An active RGD [4], however, generates an additional forward movement that guides the user to the destination. The RGD may take on a certain payload without requiring the user to push and therefore it does not cause fatigue to the user. However, the robot-centric motion may cause uncomfortable feeling of being controlled. An EWC is a handheld device that detects obstacle(s) in its vicinity. The Nottingham obstacle detector [5] uses sonar for obstacle detection. The C-5 laser cane [6] triangulates range using three pairs of laser-/photo-diodes. The "virtual white cane" [7] measures range by a baseline triangulation system comprising a laser pointer and a miniaturized camera. The user receives multiple range measurements by swinging the device. In spite of the portability, these devices only provide limited obstacle information to the user due to the restricted sensing capability. They do not provide location information either. Also, an EWC may limit or deny the use of a conventional white cane.

Recently, a portable indoor localization aid is introduced in [8]. A sensor package, comprising an Inertial Measurement Unit (IMU) and a 2D laser scanner, is mounted on a white cane and used for 6-DOF device Pose Estimation (PE). The PE method predicts the device's pose using the data of the IMU and a user-worn pedometer and updates the prediction by an extended Kalman filter using the laser scans (observation). The method requires a known map of the environment (with vertical structure) to compute the intersection between the laser scans and the structural planes of the environment.

In this paper, a new RNA, called Co-Robotic Cane (CRC) is presented. The CRC uses a 3D camera for both Pose Estimation (PE) and Object Recognition (OR) in unknown indoor environments. The PE method does not require any prior knowledge of the environments. The OR method detects indoor structures and objects. Some of them (e.g., stairways) will be used as navigational waypoints and the others for obstacle avoidance. The CRC is a co-robot. It may detect human intent and collaborate with its user in navigation.

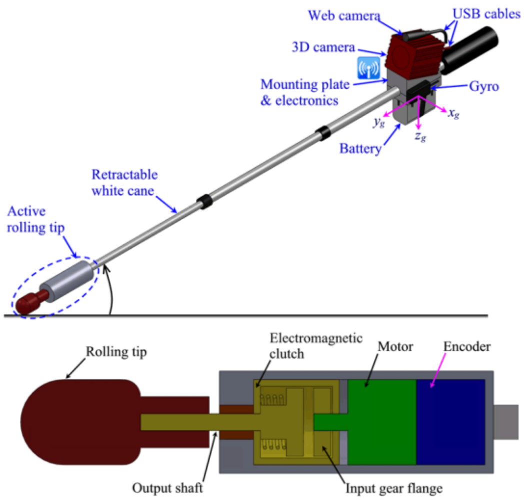
II. OVERVIEW OF CO-ROBOTIC CANE

The proposed CRC is depicted in Fig. 1. The CRC is a computer-vision-enhanced white cane where the computer vision enhancement allows the blind traveler to "see" better

This work was supported in part by the NSF under grant IIS-1017672, the NIH under award R01EB018117, and NASA under award NNX13AD32A. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

and farther. The CRC provides its user the desired travel direction in an intuitive way and offers a friendly human-device interface. The computer vision system, comprising a 3D camera, a 3-axis gyro and a Gumstix Overo® AirSTORM COM computer (inside the mounting plate), provides two functions: PE and OR. The CRC's pose is used to provide location information to the traveler and to register 3D data points in a 3D map. The Gumstix reads data from the camera and gyro and relay the data to a backend computer through WiFi. The off-board computer processes the data for PE and OR and send the results back to the Gumstix for navigation. This arrangement is due to the Gumstix's limited computing power. The CRC has an active rolling tip that may steer the cane to the target direction. The CRC also has a speech interface for human-device interaction. A headset is connected to the Gumstix via Bluetooth and used to communicate with the CRC through speech. A keypad will be created on the cane's grip and serve as an alternative human-device interface.

The CRC has two navigation modes—active mode and passive mode—that are selectable to the user by controlling the so-called Active Rolling Tip (ART) device. The ART comprises a rolling tip, a servo motor assembly with an encoder, and an electromagnetic clutch. The rolling tip is connected to the motor through the clutch. When the CRC is in passive mode, the clutch is disengaged. The rolling tip and the output shaft rotate freely when the user swings the CRC. In this case, the speech and tactile interface provide the desired travel direction and object information to the user. When the CRC is in active mode, the clutch is engaged. The motor drives the rolling tip on the ground and steers the CRC to the target direction. Assuming that the CRC swings from A to B with no slip (Fig. 2), the cane's yaw angle change is given by $\Delta\psi = \Delta\alpha R/CL\cos\theta$, where $\Delta\alpha$ is the motor's rotation angle obtained by the encoder, C is the gearhead reduction ratio, R is the rolling tip's radius, and L is the cane's length. This means that the cane's turn angle can be accurately controlled by the motor. In the active mode, the CRC provides the desired travel direction to a destination, waypoint (e.g., a stairway) or point of interest (e.g., a table), as well as for obstacle avoidance.



Section View of the Active Rolling Tip

Fig. 1. The Co-Robotic Cane

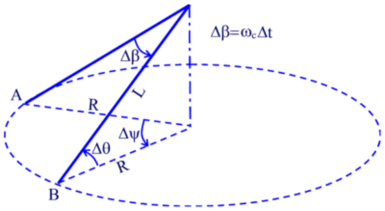


Fig. 2. The CRC Swings from A to B

Mode selection is made through the keypad or a more intuitive Human Intent Detection Interface (HIDI). The HIDI measures the user's compliance with the ART's motion by analyzing the data from the encoder and 3-axis gyro (Fig. 1). When the user is compliant with the ART, the CRC's angular velocity (ω_g) measured by the gyro should agree with that obtained from the encoder data ($\omega_c = R\Delta\alpha/CL\Delta t$). If the user swings the cane when it is in the active mode, substantial slip at the rolling tip is produced. The slip is given by $= \omega_c - \omega_g$. If S is above a threshold, a motion-incompliance, indicating the user's intent to use the CRC in the passive mode, is then detected and the CRC is switched to the passive mode. The CRC is switched back to the active mode if a motion-compliance is detected.

III. 3D TOF Camera

A SwissRanger SR4000 [9] is used for 3D perception. The SR4000 is a small-sized (65×65×68 mm^3) CMOS 3D camera. It illuminates the environment with modulated infrared light and measures ranges up to 5 meters (accuracy: ±1 cm) for each imaging pixels (176×144 pixels) based on phase shift. The camera produces range and intensity data simultaneously at a rate up to 54 frame/s. The SR4000 has much better range measurement accuracy for distant objects and data completeness than a stereovision system and a RGB-D camera (e.g., Microsoft Kinect). This may result in better performance in PE and OR. In addition, the SR4000 has a much smaller dimension that makes it suitable for the CRC**.**

IV. 6-DOF Pose Estimation

A. VRO and ICP for Pose Change Estimation

The VRO extracts and tracks the SIFT features [10] in two consecutive intensity image frames using the SIFT descriptors. As the features' 3D coordinates are known from the depth data, the feature tracking process results in two associated 3D point sets, $\{p_i\}$ and $\{q_i\}$ for $i = 1, \cdots, N$. The rotation and translation matrices, R and T , between these two point sets may be determined by minimizing the error residual:

$$e^2 = \sum_{i=1}^{i=N} \left\| p_i - Rq_i - T \right\|^2 \qquad (1)$$

This least-squares data fitting problem is solved by the Singular Value Decomposition (SVD) method [11]. As feature descriptor based match may produce incorrect feature correspondences (outliers), a RANSAC process is used to reject the outliers. The resulted inliers are then used to estimate R and T, from which the camera's Pose Change (PC) is determined. Here, pose refers to the Euler angles (roll ϕ, pitch θ, yaw ψ) and X, Y, Z coordinates. To obtain accurate PC, a Gaussian filter is applied to both intensity and range data before VRO computation. The filtering reduces the data's noise ratio by ~60% [12]. In addition, data with low certainty values (provided by the camera) are discarded. By these data treatment, VRO's accuracy and repeatability are improved.

To overcome the VRO's performance degradation in a visual-feature-spare environment, an ICP-based shape tracker is used to refine the alignment of the two point sets and thus the CRC's PC. In this work, a convex hall is created using the 3D points of the visual feature correspondences. Data points within the convex hall are used for ICP calculation. This scheme substantially reduces the ICP process's computational time. The PC estimation method is termed VRO-FICP [13].

B. *Pose Error Minimization by Pose Graph Optimization*

A straight integration of the PCs over time, aka Dead Reckoning (DR), may accumulate a large pose error. Visual feature tracking such as Extended Kalman Filtering (EKF) and Pose Graph Optimization (PGO) have been proposed to reduce the error. Our recent study [14] shows that PGO has a better performance consistency than an EKF. Therefore, we use PGO to minimize the pose error in this paper.

Let $x = (x_1, \dots, x_N)^T$ be a vector consisting of nodes $x_1, \dots, x_N$, where x_i for *i*=1, …, *N* is the camera's pose at *i*. Let z_{ij} and Ω_{ij} be the mean and the information matrix of a virtual measurement between nodes *i* and *j*. Let $\hat{z}_{ij}$ be the expected value of z_{ij}. The measurement error $e_{ij}(x_i, x_j) = z_{ij} - \hat{z}_{ij}(x_i, x_j)$ and Ω_{ij} are used to describe the edge connecting nodes *i* and *j*. Fig. 3 shows a pose graph with 5 nodes.

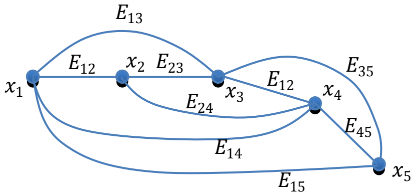


Fig. 3. A pose graph with five nodes: $E_{ij} =< e_{ij}, \Omega_{ij} >$ represents the edge between nodes *i* and *j*.

A number of approaches are taken to ensure a quality graph for PGO. When adding node x_i to the graph, $z_{(i-1)i}$ (i.e., PC between *i* and *j*) may be unavailable if VRO-FICP fails. In this case, we assume a constant movement and use $z_{(i-2)(i-1)}$ to create node x_i, i.e., $e_{(i-2)(i-1)}$ and $\Omega_{(i-2)(i-1)}$ are used for edge $E_{(i-1)i}$. For non-consecutive nodes, no edge is created if VRO-FICP fails or the PC uncertainties are large. We use the following bootstrap method [15] to compute PC uncertainties: (1) Compute a PC using $K = \min(0.75N, 40)$ samples randomly drawn from the *N* correspondences given by VRO-FICP; (2) Compute 50 PCs by repeating step 1 and calculate the standard deviation. Finally, Ω_{ij} is computed as the inverse of the diagonal matrix of the PC uncertainties. Compared with the existing works [16] that assume a constant uncertainty in graph construction, our method is advantageous in that the use of the predicted PC uncertainties may result in a more accurate graph and thus improve the PE result.

The PGO process is to find the node-configuration x^* that minimizes the following nonlinear cost function:

$$F(x) = \sum_{ij} e_{ij}\left(x_i, x_j\right)^T \Omega_{ij}\, e_{ij}\left(x_i, x_j\right). \qquad (2)$$

A numerical approach based on the Levenberg-Marquardt (LM) algorithm is used to solve the optimization problem. The LM approximates $e_{ij}\left(x_i, x_j\right)$ by its first order Taylor expansion around an initial guess $\tilde{x}$:

$$e_{ij}\left(\tilde{x}_i + \Delta x_i, \tilde{x}_j + \Delta x_j\right) = e_{ij}(\tilde{x} + \Delta x) \approx e_{ij}(\tilde{x}) + J_{ij}\Delta x \quad (3)$$

Here, J_{ij} is the Jacobian of $e_{ij}(x)$ computed at $\tilde{x}$. Substituting (3) into (2), we obtain

$$\begin{aligned} F(\tilde{x} + \Delta x) &\approx \sum_{ij}(a_{ij} + 2b_{ij}\Delta x + \Delta x^T H_{ij} \Delta_x) \\ &= a + 2b\Delta x + \Delta x^T H \Delta x, \end{aligned} \qquad (4)$$

where $a = \sum_{ij} a_{ij} = e_{ij}^T \Omega_k e_{ij}$, $b = \sum_{ij} b_{ij} = e_{ij}^T \Omega_{ij} J_{ij}$, and $H = \sum_{ij} H_{ij} = J_{ij}^T \Omega_k J_{ij}$. $F(\tilde{x} + \Delta x)$ can be minimized in term of Δx by solving the linear system

$$(H + \lambda diag(H))\Delta x^* = -b, \qquad (5)$$

where λ is a damping factor whose value is adjusted at each iteration by the LM algorithm. The linearized solution is then obtained by

$$x^* = \tilde{x} + \Delta x^* \qquad (6)$$

The PGO process iterates the linearization in Eq. (4), solution in Eq. (5) and update step in Eq. (6) until an optimal x is found. Details on the PGO are referred to [17]. In this paper, we use the GTSAM C++ library [18] for PGO.

V. 3D Object Recognition

The state-of-the-art graph-based OR method [19] is of NP complexity. For computational efficiency, we propose a Gaussian-Mixture-Model (GMM) based OR method as depicted in Fig. 4. It consists of five main procedures: range data acquisition, plane extraction, feature extraction, GMM Plane Classifier design and training, and plane clustering. Each of them is described in this section.

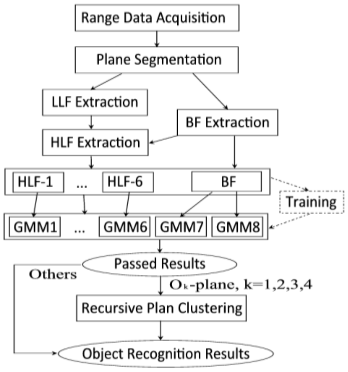


Fig. 4. Diagram of the proposed object recognition method

A. *Range Data Acquisition and Plane Segmentation*

Using the estimated poses, the 3D range data of the SR4000 (captured at different location) are registered to form a large 3D point cloud map. The point cloud data is then segmented into *N* planar patches, P_1, P_2, …, P_N, by the NCC-RANSAC method [20] proposed by the authors.

B. Features and Feature Vectors

Features are the properties of a planar patch that describe its intrinsic attributes and geometric context, i.e., Inter-Plane Relationships (IPRs) representing the geometric arrangement with reference to another planar patch. The features of a planar patch are described by a feature vector that will be used to classify the patch into one belonging to a specific type of objects. For each of the N patches, P_1, P_2, …, P_N, a feature vector construction procedure is performed to assign each of them a feature vector. In this paper, we define three classes of features—Basic Feature (BF), Low Level Feature (LLF), and High Level Feature (HLF).

Basic Features: BFs are local features that describe a patch's intrinsic attributes. They serve as the identity of the patch and play important roles in OR. Similar to [19], three BFs—Orientation, Area and Height (OAH)—are defined for a planar patch. They are computed as the angle between the patch's normal and Z-axis, size of the patch, and the maximum Z coordinate of the points in the patch, respectively. The computational details are omitted for simplicity.

Low Level Features: To classify a patch into a constituent element of a model object, the patch's IPR must be considered in addition to its BFs. The following 9 IPRs are defined for patch P_i with reference to patch P_j: (1) *plane-distance* is a value representing the minimum distance between the points of P_i and the points of P_j; (2) *plane-angle* represents the angle between P_i and P_j; (3) *parallel-distance*, denoted l_i^j, represents the distance between two parallel planes, P_i and P_j, and is computed as the mean of the distance from the centroid of P_i to patch P_j and the distance from the centroid of P_j to patch P_i; (4) *projection-overlap-rate* is a value representing to what extent P_i overlaps P_j and it is calculated as the area ratio of the projected P_i (onto P_j) to P_j; (5) *projection-distance* is the minimum distance from the points of P_i to plane P_j; (6) *is-parallel* describes if P_i is parallel to P_j (1: parallel, 0: not parallel); (7) *is-perpendicular* describes if P_i is perpendicular to P_j (1: perpendicular, 0: not perpendicular); (8) *is-coplanar* describes if P_i is co-planar with P_j (1: coplanar, 0: not coplanar); (9) *is-adjacent* describe if P_i is adjacent to P_j. (1: adjacent, 0: not adjacent). Due to the SR4000's noise, threshold values are used in computing the above LLFs. The details are omitted. Since there are N patches, an N×N matrix is formed to record each of the 9 IPRs among the N patches. Each matrix is called an LLF. In an LLF matrix, an element at (i, j) describes the IPR of P_i with reference to P_j. For example, *is-parallel*(i, j)=1 indicates that P_i is parallel to P_j.

High Level Features and Feature Vectors: The goal of OR is to identify one of the eight objects, namely, doorway, hallway, stairway, parallelepiped, monitor, table, ground and wall, from a scene. In this work, each planar patch is classified to a plane belonging to one of the eight objects (models). To this end, we define 6 mutually exclusive High Level Features (HLFs) as shown in Fig. 5. Each HLF represents a set of particular IPRs that exists in a model. The HLF extraction is a process to identify the HLF for each of the N planar patches and assign each patch a HLF vector. The BFs extracted earlier are used in constructing the HFL vector. We first construct a BF vector [O, A, H] for a planar patch. Each BF is then extended based on the patch's HLF. For a patch with HLF-1, HLF-2, HLF-3 or HLF-6, we add parameter d into its BF vector to form a HLF vector [O, A, H, d]. For a patch with HLF-5, we add parameter d_1 and d_2 to form a HLF vector [O, A, H, d_1, d_2]. For a patch with HLF-4, we simply use the BF vector as the HLF vector. A plane is treated as an object if it is a wall/ground. Such a plane does not have an IPR. We therefore simply use its BF vector as the HLF vector. A plane with an HLF as depicted in Fig. 5 is called a complex plane, while a wall/ground plane an elementary plane.

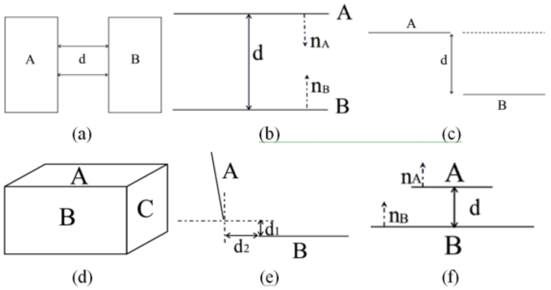


Fig. 5. Definition of HLFs: (a) HLF-1: co-planar with distance *d;* (b) HLF-2: parallel and face-to-face with distance *d;* (c) HLF-3: step-shape with distance *d;* (d) HLF-4: parallelepiped-shape; (e) HLF-5: side-above with distance d_1 and d_2: d_1 is the *A*'s *projection-distance* while d_2 is the minimum distance between *A*'s projection-points on *B* and *B*'s points*;* (f) HLF-6: above-with-distance *d*. *A*, *B* and *C* represent any three patches out of the *N* planar patches.

For each of the N planar patches, the LLF matrix is analyzed to detect the HLFs and the corresponding feature vector is assigned to the patch for each detected HLF. The HLF vector assignment process also generates 6 matrices, $\boldsymbol{Q_1}$ **,…,** $\boldsymbol{Q_6}$, each of which records the N planar patches' IPRs for HLF-1, …, HLF-6, respectively. After HLF vector assignment, the HLF vectors are sent to the GMM Plane Classifier (GMM-PC) (in Fig. 4) for plane classification.

C. GMM Plane Classifier

The GMM-PC consists of 8 GMMs each of which has been trained using data captured from a particular type of objects and is thus able to identify a plane related to that type of object when the relevant HLF vector is present. For simplicity, we call a GMM for detecting object type O_k, for $k = 1, \cdots, 8$, an O_k-GMM. Here, O_1, O_2,…, O_8 represent doorway, hallway, stairway, parallelepiped, monitor, table, ground and wall, respectively. We call a plane belonging to object O_k an O_k-plane and a scene with object O_k an O_k-scene. In the GMM-PC, the O_i-GMM receives all HLF-i vectors and identify if each vector's associated patch is an O_i-plane (here, $i = 1, \cdots, 6$). The ground-GMM and the wall-GMM receive the remaining BFs and classify each of the associated planar patches into a ground-plane or a wall-plane.

A GMM is a probability density function represented as a weighted sum of M Gaussian component densities given by:

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} \omega_i g(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \qquad (7)$$

where $\boldsymbol{x}$ is a D-dimensional (in our case, D=3, 4 or 5) vector with continuous values, ω_i, $i = 1, \ldots, M$, are the mixture weights, and $g(x|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), i = 1, \ldots, M$, are the component Gaussian densities with mean vector $\boldsymbol{\mu_i}$ and covariance matrix

$\boldsymbol{\Sigma_i}$. The mixture weights satisfy $\sum_{i=1}^{M} \omega_i = 1$. The complete GMM is parameterized by $\boldsymbol{\mu_i}$, $\boldsymbol{\Sigma_i}$ and ω_i for $i = 1, \ldots, M$. We denote these parameters collectively by λ from now on for conciseness. In this paper, the configuration (M and λ) of an O_k-GMM is estimated by training the GMM using a set of HLF vectors obtained from a number of O_k-scenes. The maximum likelihood estimate of λ is iteratively obtained by the Expectation Maximization (EM) method. The value of M is determined by repeating the training process with an increasing i and observing the trained GMM's output $p(\boldsymbol{x}|\lambda)$. If the mean of the output difference between an I-component GMM and an (I+1)-component GMM is below a threshold, we let M=I because more Gaussian component densities will not change the GMM's probability density.

We acquired 500 datasets from different O_k-scenes. After plane segmentation and feature extraction, we obtained a set of HLF vectors from each dataset. We then trained the O_k-GMM using these HLF vectors. The training process determined the O_k-GMM's configuration (M and λ). The smallest probability density p_{min}^{k} of the trained O_k-GMM for the training data is recorded as the threshold for plane classification in a later stage.

After the training, each GMM of GMM-PC is able to classify a patch into one of the eight plane types using the patch's HLF vector. To be specific, a planar patch's HLF (or BF if an HLF is unavailable) is presented to the relevant GMM. The GMM's output, $p_k(\boldsymbol{X}|\lambda)$ for $k = 1, \cdots, 8$, is compared with p_{min}^{k}. If $p_k(\boldsymbol{X}|\lambda) > p_{min}^{k}$, the planar patch is classified as an O_k-plane. In case that a patch is classified as both an elementary plane and a complex plane, the complex plane classification overrides the elementary one. All extracted HLF vectors are presented to the GMM-PC and the corresponding planar patches are classified into the eight types of planes. The GMM-PC results in 8 arrays, G_k for $k = 1, \cdots, 8$, each of which stores the indexes of the planar patches that have been classified as O_k-planes.

D. *Recursive Plane Clustering*

In this stage, the classified planes, O_k-planes for $k = 1, \cdots, 4$, are recursively clustered into a number of objects, O_k for $k = 1, \cdots, 4$. This means that the doorway-/hallway-/stairway/parallelepiped-planes are grouped into doorway(s) /hallway(s)/stairway(s)/parallelepiped(s) by a Recursive Plane Clustering (RPC) process. A monitor-/table-/ground-/wall-plane is treated as a standalone object and thus no further process is needed. Four recursive procedures, RPC-1, …, and RPC-4, process G_1, …, G_4, respectively and cluster the neighboring object planes into the four types of objects by analyzing $\boldsymbol{Q_1}$,…, $\boldsymbol{Q_4}$.

VI. Experimental Results

A. *Pose Estimation*

The PE method has been validated in both visual-feature-rich and visual-feature-sparse environments using our preliminary CRC prototype [21]. The prototype has PE and OR functions but it is not equipped with the ART (which is currently being fabricated). We carried out 9 experiments (Group I) in feature-rich environments and 5 (Group II) in feature-sparse environments. In each experiment, the CRC user walked with the cane in a looped trajectory (path-length: 20-40 meters). The CRC's Final Position Error (FPE) (in percentage of path-length) is used as the overall accuracy of PE. The PGO method's FPE is compared with that of the baseline (DR method) and the percentage Error Reduction (ER) indicates the performance of PGO. The results are tabulated in Table I. The PGO method reduces the FPEs in all cases in feature-sparse environments. For feature-rich environments, the PGO slightly increases the FPEs in 4 cases but substantially decreases the FPEs in the other 5 case. Overall, the method improves the PE results in feature-rich environments. It can be seen that the PGO achieves a much larger and more consistent PE improvement (over DR) in a feature-sparse environment. This property will benefit indoor navigation where feature-spare scenes occur from time to time.

Table I PE performance of the PGO method in various environments

Group I	FPE [%]		ER [%]	Group II	FPE [%]		ER [%]
	DR	PGO			DR	PGO	
1	4.45	**2.65**	1.80	1	3.01	**1.15**	1.86
2	2.97	**2.37**	0.60	2	3.58	**2.50**	1.08
3	**2.71**	3.07	-0.36	3	2.42	**2.23**	0.19
4	2.66	**2.50**	0.16	4	4.17	**2.75**	1.43
5	**1.2**	1.33	-0.13	5	2.21	**1.64**	0.57
6	**2.03**	2.28	-0.25				
7	2.05	**0.46**	1.59				
8	5.7	**4.71**	0.99				
9	**3.3**	3.79	-0.49				
Mean			0.43	Mean			1.02
Standard Deviation			0.86	Standard Deviation			0.67

Fig. 6 depicts the result of experiment 1 in Group I. It can be observed that the FPE of the PGO algorithm is much smaller than that of the DR method.

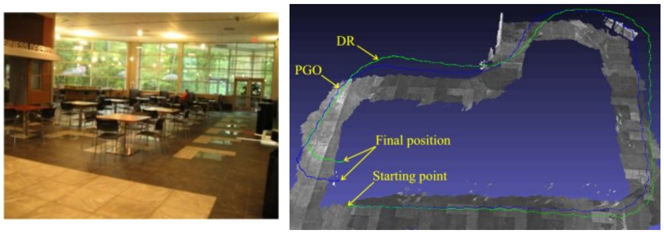


Fig. 6. Pose estimation in a feature-rich environment (experiment 1 in Group I). Left: snapshot of scene; Right: CRC's trajectories estimated by the DR and PGO methods plotted over the 3D point cloud map.

Table II Success rates in object recognition

Object Type	Type I Data	Type II Data	Average
Stairway	0.967	0.900	0.933
parallelepipe	1.000	0.933	0.967
Hallway	1.000	1.000	1.000
Doorway	0.933	0.867	0.900
Table	1.000	0.967	0.983
Monitor	0.900	0.800	0.850
Wall	1.000	1.000	1.000
Ground	1.000	1.000	1.000

B. *Object Recognition*

We collected 60 data sets from each type of scenes, half (Type I) from the same scenes that were used for GMM training and the other half (Type II) from similar scenes with the same type of objects. We ran the OR method on the 480 (8×60) datasets and evaluated its performance in term of the success rate of OR. The results are tabulated in Table II. For each object type, the success rates on Type I data are slightly higher than that on Type II data because the GMM was trained using the similar type of data. The fact that the success rate on Type II data is over 86.7% (except for monitor scenes) indicates that the trained GMMs generalize well in OR.

Fig. 7 depicts OR results of a doorway, a hallway, a parallelepiped (box), and grounds. In all cases, the proposed method detects and labels the objects correctly. Fig. 8 shows OR results in two real-world wayfinding scenarios. In each case, the CRC users walked with the cane from points 1 to 2. The CRC registered the 3D range data (captured at different locations along the path) into a 3D point cloud map using the estimated poses. The OR method successfully detected the objects and labeled them in both cases. The results demonstrate that the PE method is sufficiently accurate for the navigation of the CRC and the OR method is robust to the errors of the 3D maps due to the PE errors.

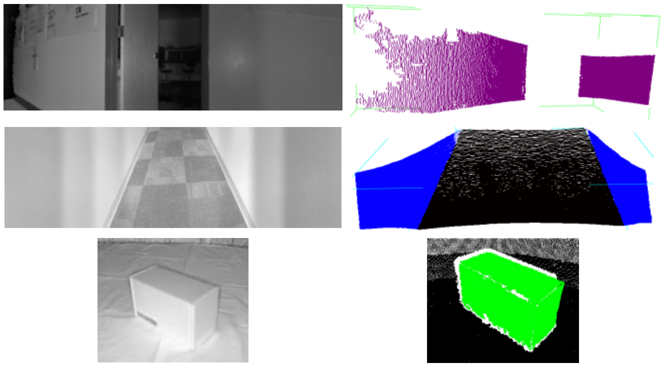


Fig. 7. Experimental results with a doorway, hallway and box. Left: intensity image; Right: 3D view of the detected objects (doorway in purple, hallway in blue, box in green, ground in black).

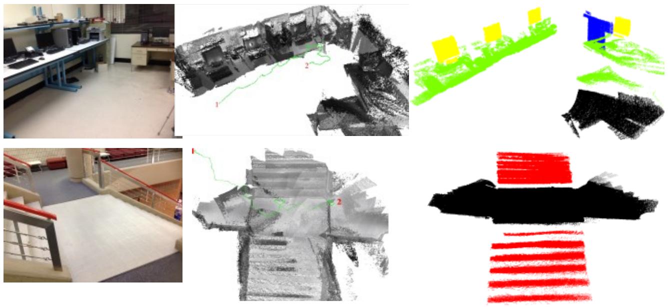


Fig. 8. Experimental results with realistic wayfinding scenarios—a laboratory/stairway scene. Left: snapshot of scene; Middle: registered 3D point cloud map; Right: 3D view of the detected objects (tables in green, monitors in yellow, grounds in black, stairway in red).

VII. Conclusions

We have introduced a new RNA, called Co-Robotic Cane (CRC) for indoor navigation of a visually impaired individual. As a co-robot, the CRC may detect human-intent and use the intent to determine the device's use mode. In the active mode, the CRC may steer into the direction of desired travel and thus provide accurate and intuitive guidance to a blind traveler. As a computer-vision-enhanced white cane, the CRC processes the intensity and range data of a 3D camera for device Pose Estimation (PE) and Object Recognition (OR). The PE method integrates a visual tracker with a shape tracker for device Pose Change (PC) estimation and employs PGO to minimize the device pose error when integrating the PCs into the pose in the world coordinate. The OR employs GMMs to classify planar patches into ones belonging to the 8 object models. The classified planar patches are then clustered into objects, each of which is identified as one of the 8 types of objects. The GMM-based method produces probabilistic classification results that may be refined by probabilistic reasoning in the future. This is advantageous over those non-probabilistic methods.

Although the methods are proposed for the CRC, they may be used by other generic robotic applications or applied to some computer vision problems.

References

[1] C. Ye, "Polar Traversability Index: a Measure of Terrain Traversal Property for Mobile Robot Navigation in Urban Environments," in *Proc. IEEE Int. Conf. SMC*, 2007, pp. 2342-2347.

[2] C. Galindo, *et al.*, "Control Architecture for Human-Robot Integration: Application to a Robotic Wheelchair," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 5, pp. 1053-1067, 2006.

[3] I. Ulrich and J. Borenstein, "The GuideCane−Applying Mobile Robot Technologies to Assist the Visually Impaired," *IEEE Trans. Syst., Man, Cybern. A, Syst. and Humans*, vol. 31, no. 2, pp. 131-136, 2001.

[4] V. Kulyukin, *et al.*, "Robot-Assisted Wayfinding for the Visually Impaired in Structured Indoor Environments," *Autonomous Robots*, vol. 21, no. 1, pp. 29-41, 2006.

[5] D. Bissit and A. Heyes, "An Application of Biofeedback in the Rehabilitation of the Blind," *Applied Ergonomics*, vol. 11, no. 1, pp. 31-33, 1980.

[6] J. M. Benjamin, N. A. Ali, and A. F. Schepis, "A Laser Cane for the Blind," in *Proc. San Diego Medical Symposium*, 1973.

[7] D. Yuan and R. Manduchi, "A Tool for Range Sensing and Environment Discovery for the Blind," in *Proc. IEEE Computer Vision and Pattern Recognition Workshops*, 2004.

[8] J. A. Hesch and S. I. Roumeliotis, "Design and Analysis of a Portable Indoor Localization Aid for the Visually Impaired," *The International Journal of Robotics Research*, vol. 29, no. 11, pp. 1400–1415, 2010.

[9] http://www.mesa-imaging.ch/prodview4k.php

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Computer Vision*, vol. 2, no. 60, pp. 91-110, 2004.

[11] K. S. Arun, *et al.*, "Least square fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 698–700, 1987.

[12] S. Hong, C. Ye, M. Bruch and R. Halterman, "Performance Evaluation of a Pose Estimation Method Based on the SwissRanger SR4000," in *Proc. IEEE International Conference on Mechatronics and Automation*, 2012, pp. 499-504.

[13] S. Hong and C. Ye, "A Fast Egomotion Estimation Method based on Visual Feature Tracking and Iterative Closest Point," in *Proc. IEEE Int. Conference on Networking, Sensing and Control*, 2014, pp. 114-119.

[14] S. Hong and C. Ye, "A Pose Graph based Visual SLAM Algorithm for Robot Pose Estimation," in *Proc. World Automation Congress*, 2014.

[15] K. Cho, *et al.*, "Performance assessment through bootstrap," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 11, pp. 1185–1198, 1997.

[16] M. Kaess, *et al.*, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 3281-3288.

[17] G. Grisetti, *et al.*, "A Tutorial on Graph-Based SLAM," *IEEE Intelligent Transportation Systems Magazine,* vol. 2, no. 4, pp. 31-34, 2010.

[18] GTSAM 2.3.0 (https://collab.cc.gatech.edu/borg/gtsam/).

[19] X. Xiong and D. Huber, "Using context to create sematic 3D models of indoor environments," In *Proc. British Machine Vision Conf.*, 2010.

[20] X. Qian and C. Ye, "NCC-RANSAC: A Fast Plane Extraction Method for 3D Range Data Segmentation," *IEEE Transaction on Cybernetics*, 2014. (DOI: 10.1109/TCYB.2014.2316282)

[21] C. Ye, "Navigating a Portable Robotic Device by a 3D Imaging Sensor," in *Proc. IEEE Sensors Conference*, 2010, pp. 1005-1010.