# A Wearable Robotic Device for Assistive Navigation and Object Manipulation

Lingqiu Jin, He Zhang, and Cang Ye

*Abstract*— **This paper presents a hand-worn assistive device to assist a visually impaired person with object manipulation. The device uses a Google Pixel 3 as the computational platform, a Structure Core (SC) sensor for perception, a speech interface, and a haptic interface for human-device interaction. W-ROMA is intended to assist a visually impaired person to locate a target object (nearby or afar) and guide the user to move towards and eventually take a hold of the object. To achieve this objective, three functions, including object detection, wayfinding, and motion guidance, are developed. Object detection locates the target object's position if it falls within the camera's field of view. Wayfinding enables the user to approach the object. The haptic/speech interface guides the user to move close to the object and then guides the hand to reach the object. A new visual-inertial odometery (VIO), called RGBD-VIO, is devised to accurately estimate the device's pose (position and orientation), which is then used to generate the motion command to guide the user and his/her hand to reach the object. Experimental results demonstrate that RGBD-VIO outperforms the state-of-the-art VIO methods in 6-DOF device pose estimation and the device is effective in assistive object manipulation.**

## I. Introduction

There are approximately 253 million people with visual impairment worldwide [1]. Among them, about 36 million are blind. Vision loss limits their ability to live independently and deteriorates their quality of life. Since age-related diseases are the leading causes of vision loss and the population continues to age, more people will go blind. Therefore, there is a dire need in developing new assistive technology to help the blind live independently and improve their quality of life.

Object manipulation, such as grabbing a cup of coffee, taking a hold of a door handle, etc., is a common task that a blind or visually impaired (BVI) person needs to perform in day-to-day life. Often, the target object is far. In this case, a BVI person must first locate the object, walk towards it, and then grasp it. Taking the task of manipulating a kettle on a countertop as an example, a VBI person needs to get to the kitchen first, locate the kettle, move closer to the countertop and the kettle, and then grasp the kettle. Vision loss makes this a challenging task for the BVI, particularly in an unfamiliar environment. Because it involves independent mobility, object recognition, and guiding hand-movement to reach the target object. To mitigate the difficulties, researchers have developed several robotic navigation aids (RNAs), to assist the BVI in wayfinding. Monocular camera [2], [3], stereo cameras [4], [5], RGB-D cameras [6], [7], and 3D time-of-flight cameras [8], [9] have been used by these RNAs for pose estimation. However, these systems were designed only for assistive wayfinding. Some vision-based devices have been introduced for object detection [10], [11] or object manipulation [12], [13]. These devices are helpful only if the target object is inside the camera's field of view. To address the shortcomings, an assistive device for object manipulation must provide three functions in a package: object detection, wayfinding, and movement guidance (i.e., guide the user to move towards the object and then guide the hand to reach it).

In this paper, we present a wearable robotic object manipulation aid (W-ROMA) to assist the BVI to locate a target object, move to its vicinity, and take a hold of it. It provides both assistive wayfinding and assistive object grasping functions to a BVI user. In the paper's context, we assume that the BVI is a white cane user and W-ROMA is to enhance the white cane's function. Also, W-ROMA does not deal with the rest of the object manipulation issue after the hand reaches the target object. The main contributions of this paper include: 1) We developed a hand-worn assistive device that provides object detection, wayfinding, and motion guidance to assist the BVI with object manipulation tasks in their daily lives; 2) We proposed a new VIO method, called RGBD-VIO, which tightly couple the data from an RGB-D camera and an IMU to estimate device pose for assistive wayfinding and motion command generation; 3) We designed an effective human-device interface (a haptic interface and a speech interface) to guide the user's hand movement.

## II. Related Work

### A. Related Wearable Robotic Assistive Device

In the literature, wearable assistive devices have been developed to provide navigational guidance to a blind person for wayfinding [2], [4], [6], or object manipulation [12], [13]. Treuillet *et al.* [2] proposed a body-mounted system that uses a monocular camera to estimate its location and heading in a GPS denied environment and guides the VI to walk along a safe route by using audio feedback. Pradeep *et al.* [4] introduced a wearable stereo-vision system for the BVI. It uses a stereo camera to estimate the user's egomotion and generate a 3D point cloud map, based on which it guides the BVI to steer away from the obstacle and walk in a traversable area towards the destination. Later, this system was improved in [6] by replacing the stereo camera with an RGB-D camera as the RGB-D camera can provide denser depth data of the scene. These RNAs, merely based on computer vision, are prone to error when the operating environment is not feature-rich. Therefore, they cannot provide accurate and reliable navigational guidance in a feature-sparse environment. In contrast, our W-ROMA employs a new VIO method that can

This work was supported by the National Eye Institute of the NIH under Award R01EY026275. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agency.

The authors are with the Computer Science Department, Virginia Commonwealth University, Richmond, VA 23284 USA (e-mail: cye@vcu.edu).

perform well in a feature-sparse area due to the use of inertial data for pose estimation. Furthermore, these systems focus on assistive wayfinding whereas our W-ROMA can also detect the target object and guide the user to take a hold of it. Concerning object manipulation, the handheld-camera-based system [12] detects the target object by using the SIFT features and indicates the object location to the user by sonification so that s/he can move the hand up/down and left/right to align with the object and while approaching it. The wearable visual aid [13] employs a glass-mounted camera to detect a target object and provides auditory feedback to help the user track it inside the camera's field of view. These systems, however, lack depth information to effectively guide the hand's forward/backward movement to reach the target object. Our system uses an RGB-D camera to obtain the depth information about the target object, based on which it generates navigational commands (forward/backward, left/right movement, etc.) and uses an array of vibrating motors to convey to the user the desired hand movement to reach the target object.

## B. *Related Visual-Inertial Odometry (VIO)*

The proposed W-ROMA employs a visual-inertial navigation system (VINS), consisting of an RGB-D camera and an inertial measurement unit (IMU), to estimate the device's pose by RGBD-VIO. Our survey on the related work is thus focused on RGB-D-camera-based VIO methods. In this regard, Laidlow *et al.* propose a dense RGB-D-inertial SLAM method in [14]. The method combines the residuals of the photometric per-pixel measurements, geometric point-to-plane distances, and IMU preintegration to form a cost function. The system's optimal motion state is found, via a Gauss-Newton iterative process, to be the one that minimizes the cost function. However, this method requires the use of GPU for processing the dense camera data in real-time and is thus unsuitable for a resource-limited mobile platform [15], [16]. Ling *et al.* propose a sparse-feature-based VIO method [17] that does not require GPU speedup. The method extracts computationally effective ORB features [18] and uses a 3D-3D-perspective n-point (PnP) method [19] to compute the visual structure (including the camera poses and the features' positions). It then aligns the visual-odometry-estimated camera poses with the IMU-estimated poses (IMU integrations) to compute the VINS' initial state, including the IMU's poses, velocities, bias, and gravity direction, and estimate the system's pose afterward by minimizing the cost function that takes into account the residuals of the visual and inertial measurements. Shan *et al.* extend this work later by [20] that uses corner points [21] as the visual features and employs a 3D-2D PnP [22] method to build the visual structure. Their method also uses the RGB-D camera's depth data to estimate the VINS' motion state via a nonlinear optimization process after the system's initialization: A visual feature upon its first observation is assigned an inverse depth using the depth data from the RGB-D camera. The inverse depth value remains constant since then. Differently, our RGBD-VIO allows the state estimator to update the inverse depth for a distant feature (z>3 m) throughout the optimization process to improve the pose estimation accuracy. Our method also adds extra edges (i.e., constraints) among the pose variables by using the related depth measurements. Furthermore, our method uses a hybrid-perspective-n-point (HPnP) method to build the visual structure in the state initialization process. HPnP requires fewer 3D points than 3D-2D PnP, resulting in a higher success rate of state initialization.

## III. THE W-ROMA

As depicted in Fig. 1, the W-ROMA prototype, is composed of two main units: a sensing unit and a guiding unit. The sensing unit employs an Occipital Structure Core [23] (SC) sensor, which is connected to and powered by a Google Pixel 3 smartphone via a USB-C cable. The SC has a built-in IMU, a color camera, and a stereo IR camera (for depth measurement). These sensors form an RGB-D-camera-based VINS for device pose estimation, 3D mapping, and target-object detection. The computational tasks will be performed by the smartphone. The guiding unit determines the desired hand movement and generates a proper vibration pattern to guide the user to move his/her hand to reach the target object. Six vibrating motors are installed on the surface of W-ROMA. They are controlled by the Bluno Nano board (powered by Pixel 3 via a USB-C splitter cable), which communicates with Pixel 3 through Bluetooth. A speech interface (using a Bluetooth headset) is used as an additional user interface for human-device interaction.

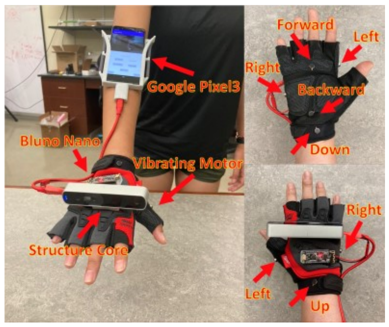

Fig 1. W-ROMA hardware system

Fig. 2 depicts the pipeline of W-ROMA's software system. The software runs entirely on Pixel 3. It processes the imaging and inertial data from SC for device pose estimation, 3D mapping, and object detection. The TensorFlow Lite Object Detection API [24] is used to detect the target object from the SC's color image(s). The API employs the quantized MobileNet SSD model [25] that has been trained with the COCO dataset [26] for object detection. RGBD-VIO processes the IMU and RGB-D data from SC to determine the 6-DOF device pose, which is then used to generate navigational commands to guide the hand movement.

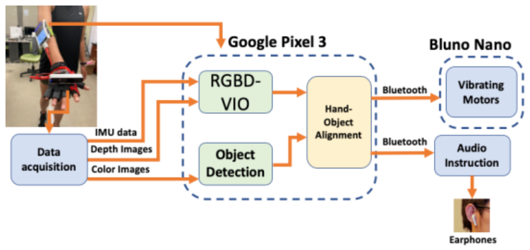

Fig. 2. Software pipeline of the W-ROMA

The Google Pixel 3 smartphone [27] is equipped with a Snapdragon 845 processor and 4 GB RAM. It possesses adequate computing power for our application. It is noted that our system can be migrated onto any ARCore-supported devices [28] that are compatible with SC. The smartphone also powers the SC sensor and the Bluno Nano board.

The SC sensor consists of an RGB-D camera and a Bosch BMI055 IMU. The camera measures depth by using a pair of global shutter IR cameras, which incur less image distortion and motion-induced image blur. As a result, SC provides more accurate depth measurements than other off-the-shelf RGB-D cameras. It has a 59°×46°×70° field of view and captures color images with a 640×480 resolution at 30 Hz. The measurable depth range is from 0.3 m to 5 m. The camera's accurate depth measurement ensures accurate pose estimation. The depth camera works both indoor (using structured light) and outdoor (using stereovision). The IMU provides 6-axis motion measurements (3-axis rotational rate and 3-axis acceleration) at a rate up to 500 Hz. The IMU and RGB-D data streams are synchronized. The whole unit only weighs 52.5 grams. The integration of the high-quality structured light camera and low-noise IMU, as well as the compact size and light weight make SC an ideal sensor for W-ROMA.

Six vibrating motors and a Bluno Nano board [29] forms the guiding unit powered by the smartphone. Bluno Nano is used to control the vibrating motors due to its low power consumption and compact size. The guiding unit communicates with the phone via Bluetooth. As shown in Fig. 1, each of the six vibrating motors indicates one of the six directions for the desired hand movement: up, down, left, right, backward, and forward. A combination of two of the four vibrating motors (up, down, left, right) indicates one of the four diagonal directions of movement: up-right, up-left, down-right, and down-left. Once Bluno Nano receives a command from Pixel 3, it generates the corresponding vibrating pattern and retains it for one second.

## IV. Motion Guidance

The coordinate systems of W-ROMA are depicted in Fig. 3. The IMU (body) and camera coordinate systems are denoted by $\{B\}/(X_bY_bZ_b)$ and $\{C\}/(X_cY_cZ_c)$, respectively. The initial $\{B\}$ is taken as the world coordinate system $\{W\}$. In this paper, the superscripts $b$ and $c$ indicate a variable in $\{B\}$ and $\{C\}$, respectively. The transformation matrix from $\{B\}$ and $\{C\}$ is precalibrated and denoted $T_c^b = [R_c^b;\ \mathbf{t}_c^b]$, where $R_c^b$ stands for the rotation and $\mathbf{t}_c^b$ the translation. Using SC's depth data, we can construct the 3D point cloud, which is denoted by $\boldsymbol{P}^{c_k}$ for the *k^th^* camera frame. It can be transformed and described in $\{C_n\}$ (i.e., the camera's coordinate system when the *n^th^* camera frame was captured) by $\boldsymbol{P}^{c_n} = T_{c_k}^{c_n}\boldsymbol{P}^{c_k}$.

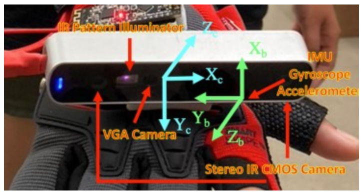

Fig 3. IMU and camera coordinate systems for the W-ROMA

The hand coordinate system is denoted by $\{H\}/(X_hY_hZ_h)$ and its origin $O_H$ is located at the image center as shown in Fig 4d. The center point of the target object expressed in $\{H\}$ is denoted by $\boldsymbol{P} = [X_h, Y_h, Z_h]^T$. If $Z_h \leq 0$, the "backward" command will be generated, indicating that the user should move the hand backward. Otherwise, the system reports the distance $Z_h$ to the user and the projection of $\boldsymbol{P}$ on the virtual image plane $(X_hO_HY_h)$, denoted $\tilde{\boldsymbol{p}}$, is used to determine the required lateral hand movement for alignment with the target object and thus the proper navigational command. For example, if $\tilde{\boldsymbol{p}}$ is in the right green region, the command "right" will be generated. If $\tilde{\boldsymbol{p}}$ is in the down-left blue region ([-157.5°, -112.5°]), the command will be "down left".

In the real world, object manipulation scenarios can be classified into three categories: i) case I: the target object is close and inside the views of both the color and depth cameras; ii) case II: the target object is beyond the SC's depth range but inside the color camera's view; iii) case III: the target object is outside of both color cameras' views, but it was observed and detected earlier. The guidance scheme is designed and explained as follows.

*Case I:* At the *k^th^* camera frame, the object detection module [24] detects the target object on the color image and determines the center of the object's bounding box denoted $\mathbf{P}^{C_k}$. $\mathbf{P}^{C_k}$ can be transformed into $\{H\}$ by $\mathbf{P}^{H_k} = T_C^H\ \mathbf{P}^{C_k}$. Figure 4a shows an example of this scenario. First, the object detection module recognizes the bowl and returns the coordinates of a bounding box (Fig. 4a). Then, it is checked on the depth image (Fig. 4b), if there is a sufficient number of data points around the center of the bounding box (i.e., within the square image patch in green as shown in Fig. 4b). If yes, the center point of the target object is obtained by using the data points within the patch. The center point is then transformed into $\{H\}$ to determine the hand-object misalignment and thus the desired hand movement and the navigation command.

*Case II:* W-ROMA detects the target object on the *k^th^* camera image frame. The desired hand movement is computed based on the location of the object bounding box's center point. Since the object is beyond the depth camera's range, the device reports to the user that the target object is found but is in a distant place. In this case, the device will signal the user to walk towards the object. The needed command to retain hand-object alignment will be computed and conveyed to the user while s/he is walking towards the target object.

*Case III:* As the target object was observed and detected earlier, its point cloud is transformed into the current hand coordinate system to determine the required hand movement. For example, the user walks away from the target object (bowl) after it was observed and detected (in Fig 4a), making it afar and become not visible on the camera's current image as shown in Fig. 4c. The center point (i.e., the center of the green square in Fig. 4b) that was computed earlier is transformed to the current camera coordinate system (see the green dot in Fig. 4d). By reprojecting the transformed center point onto the virtual image plane (see Fig. 4e), it can be found that the bowl lies in the "right" sector. Therefore, the navigation command is "move right" and the right vibrating motor will vibrate for one second, signaling the hand to move right.

In summary, as long as the target has been observed and its point cloud has been generated, the system can accurately estimate the relative 3D distance as well as the misalignment between the target and the hand. The information will then be used to guide the hand movement to reach the object. To achieve this function, W-ROMA's pose must be accurately and reliably tracked. This will be achieved by the proposed RGB-VIO method.

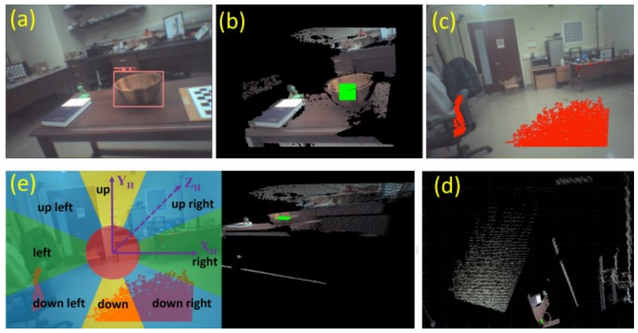

Fig. 4. (a) Target object (a bowl) detected on an image; (b) Depth image (texture-mapped); (c) The bowl is not visible on the image (pixels with depth data are shown in red); (d) Top (XZ) view of the point cloud data corresponding to Fig. 4c; (e) The target object's center projected onto the virtual image plane of current (the entire point cloud is transformed for visualization).

## V. RGBD-VIO

RGBD-VIO consists of two components, frontend feature tracking and backend state estimation. The feature tracking component extracts corner features [21] from an image and tracks them across images by using KLT [30]. A fundamental-matrix-based RANSAC process is implemented to remove the outliers. keyframes are selected based on the average parallax difference and managed by using a sliding window. The tracked features in all keyframes within the sliding window are passed to the backend process to estimate the VINS' motion state. The backend state estimator starts with a sophisticated initialization process and then proceeds with a nonlinear optimization process for state estimation.

### A. *Initialization*

The initialization procedure first implements a vision-only structure from motion to build a visual structure (including the camera poses and features' positions). To keep the computational cost low, only keyframes within the sliding window are used. Sparse features are extracted and tracked over these frames and the feature correspondences are used to build the visual structure. Since depth data are available from the RGB-D camera, a visual structure with a known scale (compared to an arbitrary scale in [33]) can be obtained. First, the pose change between two frames with sufficient parallax is computed by using a Perspective-n-Point (PnP) method. Second, the depths for those visual features (on the two frames) that have no depth data from the RGB-D camera are computed by triangulation. Third, the device pose is estimated by using all visual features from the keyframes within the sliding window. Finally, a global Bundle Adjustment is used to compute the poses for all keyframes and the features' positions by minimizing the total feature reprojection error. After the visual structure is constructed, the initialization process uses a visual-initial alignment pipeline [20] to estimate the VINS's initial state, including the IMU's poses, velocities and biases.

The accuracy of the pose change estimation (PCE) in the first step determines the accuracy of the triangulated feature points which further affects the visual structure construction and the visual-inertial alignment. Therefore, the reliability of the initialization is dependent on the PnP method for computing the PCE. VINS-RGBD [20] employs the 3D-2D-PnP method [22] to compute the PCE since depth data are available. However, we find that it often results in an inaccurate PCE when the number of visual features with depth data is low. To mitigate this issue, we propose a hybrid PnP (HPnP) by decoupling the rotation and translation computation. The rationale is that visual feature correspondences with unknown depth contain the camera's rotation information. Therefore, they can be used to compute the rotation. HPnP first employs the 2D-2D PnP method [31] to estimate the rotation by using all visual features (w/ and w/o depth data). It then determines the inliers and uses the inliers with a depth measurement to estimate the camera's translation by minimizing the reprojection error of the de-rotated inliers.

Specifically, given the rotation $R_{ji}$ between the camera coordinate system $i$ and $j$, we can estimate the translation $\mathbf{t}_{ji}$ by the following scheme: Assuming $M$ pairs of visual features, $\{\mathbf{p}_k^i, \mathbf{p}_k^j\}$ for $k = 1 \ldots M$, are observed on frames $\mathcal{F}_i$ and $\mathcal{F}_j$, where $\boldsymbol{p}_k = [u_k, v_k, 1]^T$ represents a visual feature in the normalized coordinate system. The corresponding 3D point is $\boldsymbol{P}_k = [X_k, Y_k, Z_k]^T$. $\boldsymbol{P}_k^j$ can be expressed by

$$Z_k^j \boldsymbol{p}_k^j = \boldsymbol{R}_{ji} \boldsymbol{p}_k^i + \mathbf{t}_{ji}. \tag{1}$$

By eliminating $Z_k^j$ from the equation group in (1), we obtain

$$\begin{aligned} (\boldsymbol{R}_1 - u_k^j \boldsymbol{R}_3) \boldsymbol{p}_k^i + t_1 - u_k^j t_3 = 0 \\ (\boldsymbol{R}_2 - v_k^j \boldsymbol{R}_3) \boldsymbol{p}_k^i + t_2 - v_k^j t_3 = 0 \end{aligned} \tag{2}$$

where $\boldsymbol{R}_i$ and $t_i$ $(i = 1, 2, 3)$ are the $i^{th}$ row of $\boldsymbol{R}_{ji}$ and $\mathbf{t}_{ji}$, respectively. Since each feature can provide a 2-D constraint, at least two feature points are needed to estimate $\mathbf{t}_{ji}$. For the $k^{th}$ feature point, the residual vector of $\mathbf{t}_{ji}$ is:

$$\boldsymbol{r}_k = (\boldsymbol{A}_k \boldsymbol{t}_{ji} - \boldsymbol{b}_k), \tag{3}$$

where $\boldsymbol{A}_k = \begin{bmatrix} 1 & 0 & -u_k^j \\ 0 & 1 & -v_k^j \end{bmatrix}$, $\boldsymbol{b}_k = \begin{bmatrix} (u_k^j R_3 - R_1) \boldsymbol{p}_k^i \\ (v_k^j R_3 - R_2) \boldsymbol{p}_k^i \end{bmatrix}$. The optimal translation $\mathbf{t}_{ji}^*$ that minimize $\sum_{k=1}^{M} ||\boldsymbol{r}_k||^2$ is obtained by using the Levenberg-Marquardt algorithm.

### B. *State Estimator*

An iterative optimization process is employed to solve the nonlinear state estimation problem by using the keyframes and the associated IMU data within a sliding window. The state vector of W-ROMA is defined as $\boldsymbol{\chi} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n, \lambda_1, \lambda_2, \ldots, \lambda_m\}$, where $\mathbf{x}_i = \{\mathbf{t}_{b_i}^w, \mathbf{v}_{b_i}^w, \mathbf{q}_{b_i}^w, \mathbf{b}_a, \mathbf{b}_g\}$ for $i = 1, \cdots, n$ is the IMU's motion state, including the translation, velocity, rotation, accelerometer bias, and gyroscope bias, at the time when the $i^{th}$ keyframe is captured. We use $\boldsymbol{\psi}_i = \{\mathbf{t}_{b_i}^w, \mathbf{q}_{b_i}^w\}$ to denote the device pose and $\mathbf{R}_{b_i}^w$ the rotation matrix corresponding to quaternion $\mathbf{q}_{b_i}^w$. $n$ is the size of the sliding window ($n = 10$ in this work) and $m$ is the total number of features inside the sliding window. $\lambda_k$ $(k = 1 \ldots m)$ is the inverse-depth of the $k^{th}$ feature. Differing from the method in [20] that initializes the inverse-depth of all visual features by using the depth data from the RGB-D camera and keep the values constant, RGBD-VIO allows the state estimator to update λ for a distant feature (z>3 m) throughout the optimization process. The optimal state vector $\boldsymbol{\chi}^*$ is the one that minimizes the following cost function:

$$F(\boldsymbol{\chi}) = \| {}^{o}\boldsymbol{r}\|^2 + \textstyle\sum_j \| {}^{u}\boldsymbol{r}_{\boldsymbol{j}}\|^2 + \sum_{k,j} \rho\left(\| {}^{p}\boldsymbol{r}_{\boldsymbol{kj}}\|^2\right) + \sum_k \| {}^{d}\boldsymbol{r}_{\boldsymbol{k}}\|^2 + \sum_{k,j} \rho\left(\| {}^{d}\boldsymbol{r}_{\boldsymbol{kj}}\|^2\right) \tag{4}$$

where $\rho()$ is the Cauchy loss function [32]. ${}^{o}\mathbf{r}$, ${}^{u}\mathbf{r}_j$, and ${}^{p}\mathbf{r}_{kj}$, represent the residuals for the prior information from marginalization, IMU preintegration (between keyframes $j$-1 and $j$), and feature reprojection (for those features without depth measurements), respectively. The definitions of ${}^{o}\mathbf{r}$, ${}^{u}\mathbf{r}_j$ and ${}^{p}\mathbf{r}_{kj}$ are referred to [33]. The other terms, ${}^{d}\mathbf{r}_k$ and ${}^{d}\mathbf{r}_{kj}$, are defined by using the inverse-depth measurement and they are explained as follows.

For the $k^{th}$ visual feature on the $j^{th}$ keyframe, the residual is given by $\mathrm{e}_k = 1/Z_k - \hat{\lambda}_k$, where $Z_k$ is the feature's actual depth measurement provided by the RGB-D camera and $\hat{\lambda}_k$ is the estimate of the inverse-depth. Assuming a Gaussian distribution $\gamma \sim N(0, \sigma_\gamma)$ for the error of the SC's image disparity, the standard deviation of the inverse-depth error is given by $\sigma_\lambda = \sigma_\gamma/(f * l)$, where $f$ and $l$ are the focal length and the baseline of the structured light camera, respectively. Then, ${}^{d}\mathbf{r}_k$ can be computed by ${}^{d}\mathbf{r}_k = \mathrm{e}_k/\sigma_\lambda$.

For the $k^{th}$ feature that was first observed on the $i^{th}$ keyframe as $\mathbf{p}_{\boldsymbol{k}}^{\boldsymbol{i}} = [u_k^i, v_k^i, 1]^T$ and then tracked onto the $j^{th}$ keyframe as $\mathbf{p}_{\boldsymbol{k}}^{\boldsymbol{j}} = [u_k^j, v_k^j, 1]^T$ with depth measurement $Z_k^j$, the estimated 3D coordinate can be computed by $\widehat{\boldsymbol{P}}_{\boldsymbol{k}}^{\boldsymbol{j}} = [\hat{X}_k^j, \hat{Y}_k^j, \hat{Z}_k^j]^T = \mathbf{R}_{c_i}^{c_j} \boldsymbol{p}_{\boldsymbol{k}}^{\boldsymbol{i}}/\hat{\lambda}_k^i + \mathbf{t}_{c_i}^{c_j}$, where $\mathbf{R}_{c_i}^{c_j}$ and $\mathbf{t}_{c_i}^{c_j}$ are the rotation matrix and translation from $\{C_j\}$ to $\{C_i\}$, respectively. We define the residual vector ${}^{d}\mathbf{r}_{kj} = \Sigma_{c_j}^{-1/2}\left[X_k^j - \hat{X}_k^j, Y_k^j - \hat{Y}_k^j, 1/Z_k^j - \hat{\lambda}_k^j\right]^T$, where $\Sigma_{c_j}$ is the measurement covariance and it is given by $\Sigma_{c_j} = diag\left(\sigma_x^2, \sigma_y^2, \sigma_\lambda^2\right)$ with $\sigma_x = \sigma_y = \mathrm{Z}_k^j \sigma_\tau/f$, where $f$ is the focal length, $\sigma_\tau$=1.5 pixels is the image noise, and $\sigma_\lambda$ is the inverse-depth noise.

It is noted that RGB-VIO automatically degrades itself into VIO if depth data is unavailable or an IMU-based pose estimation system if both depth and visual data are unavailable.

Table I: Comparison of methods: RMSE of the estimated trajectory of each method. In each row, the best result is bolded. TL - Trajectory Length.

| Dataset | TL (m) | RGBD-VIO | VINS-Fusion | VINS-RGBD |
|---|---|---|---|---|
| 1 | 15.3 | **0.134** | 0.178 | 0.137 |
| 2 | 23.53 | **0.12** | 0.193 | 0.153 |
| 3 | 22.73 | 0.174 | **0.166** | 0.206 |
| 4 | 65.6 | **0.43** | 0.583 | 0.581 |
| 5 | 84 | **0.486** | 0.677 | 0.725 |
| Mean | 42.232 | **0.269** | 0.359 | 0.360 |

## VI. EXPERIMENTS

### A. RGBD-VIO Performance Evaluation

We compare RGBD-VIO with the other two state-of-the-art RGB-D-camera-based VIO methods, VINS-Fusion (RGB-D version) [34] and VINS-RGBD [20]. We collected the SC's visual and inertial data by handholding it and walking in our laboratory where the ground truth trajectories of the SC were obtained by using an OptiTrack motion capture system. We acquired five datasets (three short and two long trajectories) for comparison. For each dataset, we compared the root mean square error (RMSE) of the estimated trajectories by different VIO methods. The results are tabulated in Table I. In each row, the smallest RMSE is bolded. The results in the table show that RGBD-VIO has the smallest RMSE in four of the five experiments and its mean RMSE (0.269) is smaller than VINS-Fusion (0.359) and VINS-RGBD (0.360). On average, RGBD-VIO reduces the RMSE by 25.2% and 25.4% when compared to VINS-Fusion and VINS-RGBD, respectively. This demonstrates that RGBD-VIO has a more accurate pose estimation accuracy than the other methods. In addition, a qualitative result is shown in Fig. 5, where the point cloud map built for dataset 3 by using the RGBD-VIO-estimated poses is rendered and the trajectories estimated by the three methods are plotted. The quality of the map reflects the RGBD-VIO's good performance in pose estimation.

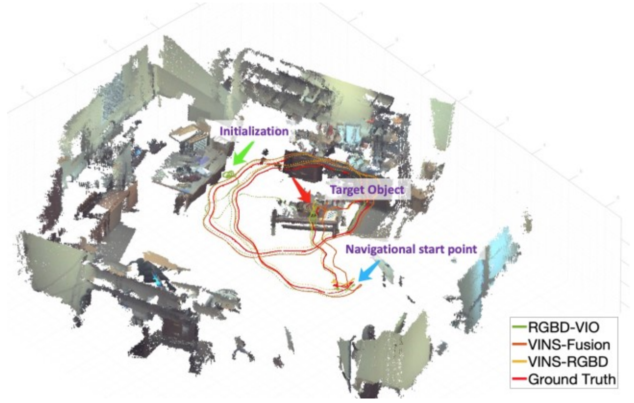

Fig. 5. The point cloud (texture-mapped) and estimated trajectories for dataset 3. The place where the system was initialized, the target object, and the navigation start point are labeled by the green, red and blue arrows, respectively.

### B. Experiment with W-ROMA prototype

We recruited five sighted volunteers to test the W-ROMA prototype for object manipulation. We asked each of them to perform an object manipulation task five times w/ and w/o the device. The time taken to complete the task was recorded for each experiment. If a volunteer could not grasp the target object in 2 minutes or s/he picked up a wrong object, the task was terminated, and this test was considered as unsuccessful. Otherwise, it is a successful one. For each subject, we also recorded the number of successful trials (NST).

The volunteers were blindfolded during the five trials. The target object was a wooden bowl (Fig. 4a). At the beginning of each trial, the volunteer was standing in front of the bowl, and they were asked to move W-ROMA around slowly. This way, the system can successfully detect and locate the bowl and meanwhile initialize RGBD-VIO for pose tracking. Then the volunteer was accompanied by a sighted person and walked to another place (labeled as the navigation start point in Fig. 5). By making the target object out of the camera's view, we intended to test W-ROMA's three functions: wayfinding – guiding the user to walk to the vicinity of the target object; object detection – detecting the target object once it appears in the camera's view, and motion guidance – generating effective motion commands to guide the user's hand to grasp the target object. If any of these functions fails, the user may fail to grab the bowl or needs more time to search for it. When a subject started to search the bowl by following the instructions from W-ROMA, we started timing. W-ROMA sends the volunteers a voice command and vibration pattern every two seconds. At the beginning of a W-ROMA-aided test, the volunteer was guided by a sighted person to touch the bowl and then escorted to the navigation starting point. Then, s/he started searching for the bowl and we started timing.

Table II summarizes the experimental results. It can be seen that with the assistance of W-ROMA, the total success rate of trials was improved three times, from 32% to 96%. And the average time for task completion was halved, from 29.1s to 15.6 s. The results demonstrate that our W-ROMA can effectively help the person in wayfinding and object manipulation.

Table II Experiments with humans: In each row, the best result is bolded. NST – Number of Successful Trials, X – Failed in all five tests.

| Subject | w/ W-ROMA | | w/o W-ROMA | |
|---|---|---|---|---|
| | NST | Mean Time(s) | NST | Mean Time(s) |
| 1 | **5** | **15.2** | 3 | 39 |
| 2 | **5** | 13 | 2 | **12** |
| 3 | **4** | **14** | 2 | 31 |
| 4 | **5** | **20** | 1 | 30 |
| 5 | **5** | **12** | 0 | X |
| Mean | **4.8** | **15.6** | 1.6 | 29.1 |

## VII. Conclusion

This paper presents a new hand-worn assistive device that can assist a visually impaired individual for wayfinding and object manipulation. The device uses a Google Pixel 3 as the computing platform and a Structure Core sensor as the perception sensor. The device can track the user's position, detect the target object in the surroundings, and guide the user to grasp the object. Based on the estimated device pose, the system computes the location of the target object in the current camera coordinate system and uses this information to generate the proper navigational command and convey the command to the user by using a haptic interface (an array of vibrating motors) and a speech interface. The proposed W-ROMA is demonstrated by experiments to be capable of effectively guiding the user to grasp a target object.

To achieve reliable and accurate pose estimation, a new method called RGBD-VIO is proposed. It exploits the camera's depth data in the initialization and the optimization processes to improve pose estimation accuracy. In the initialization phase, we use a hybrid PnP pipeline to reliably initialize the VINS system. Furthermore, we incorporate the depth measurements into the nonlinear optimization process to improve pose estimation accuracy. Experimental results demonstrate that the proposed method outperforms the state-of-the-art VIO methods in pose estimation accuracy.

## References

[1] R. Boourne, *et al.*, "Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis," The Lancet Global Health, vol. 5, no. 9, pp. E888-E897, 2017.

[2] S. Treuillet, *et al.*, "Body Mounted Vision System for Visually Impaired Outdoor and Indoor Wayfinding Assistance," *CVHI*, 2007

[3] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot.* Autom., vol. 12, no. 5, pp. 651-670, 1996

[4] V. Pradeep, G. Medioni, and J. Weiland, "Robot vision for the visually impaired." *CVPR Workshops*, 2010, pp. 15-22.

[5] J. M. Saez, F. Escolano, and A. Penalver, "First Steps towards Stereo-based 6DOF SLAM for the Visually Impaired," *CVPR Workshops*, 2005, pp. 23–23.

[6] Y. H. Lee and G. Medioni, "RGB-D camera based navigation for the visually impaired," *Computer Vision and Image Understanding*, vol. 149, pp. 3-20, 2016.

[7] N. Kanwal, E. Bostanci, K. Currie, and AF. Clark, "A navigation system for the visually impaired: a fusion of vision and depth sensor," *Applied Bionics and Biomechanics*, 2015.

[8] H. Zhang and C. Ye, "An indoor navigation aid for the visually impaired," *Proc. IEEE Int. Conf. Robotics and Biomimetics*, 2016, pp. 467-472.

[9] H. Zhang and C. Ye, "Plane-Aided Visual-Inertial Odometry for 6-DOF Pose Estimation of a Robotic Navigation Aid," *IEEE Access*, vol. 8, pp. 90042-90051, 2020.

[10] S. Nanayakkara, *et al.* "EyeRing: a finger-worn input device for seamless interactions with our surroundings", *Proc. Augmented Human International Conference* , 2013, pp. 13-20.

[11] M. Waisbourd, *et al.*, "The impact of a novel artificial vision device (OrCam) on the quality of life of patients with end-stage glaucoma," *Investigative Ophthalmology & Visual Science*, vol. 56, no. 7, 2017.

[12] B. Schauerte, M. Martinez, A. Constantinescu, "An Assistive Vision System for the Blind that Helps Find Lost Things," in *Proc. Int. Conf. on Computers for Handicapped Persons*, vol. 2, pp. 566-572, 2012.

[13] K. Thakoor, *et al.*, "A system for assisting the visually impaired in localization and grasp of desired objects," *Proc European Conference on Computer Vision*, 2014, pp. 643-657.

[14] T. Laidlow, *et al.*, "Dense RGB-D-inertial SLAM with map deformations," *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2017, pp. 6741-6748.

[15] M. Li and A. Mourikis, "Vision-aided inertial navigation for resource-constrained systems," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1057-1063.

[16] H. Zhang and C. Ye, "An indoor navigation aid for the visually impaired," *IEEE International Conference on Robotics and Biomimetics,* 2016*,* pp. 467–472.

[17] Y. Ling, *et al.*, "RGBD inertial odometry for indoor robot via keyframe-based nonlinear optimization," *Proc. IEEE International Conference on Mechatronics and Automation*, 2018, pp. 973–979.

[18] R. Ethan, *et al.*, "ORB: An efficient alternative to SIFT or SURF." *IEEE International conference on computer vision*, 2011, pp. 2564-2571.

[19] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. CVPR*, vol. 1, pp. I-I, 2014.

[20] Z. Shan, R. Li, and S. Schwertfeger, "RGB-D-inertial trajectory estimation and mapping for ground robots," *Sensors*, vol. 19, no. 10, pp. 2251, 2019.

[21] J. Shi, *et al.*, "Good features to track," *Proc. CVPR*, 1994, pp. 593–600.

[22] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155, 2009.

[23] Structure core - specs data. https://structure.io/structure-core/specs.

[24] Tensorflow lite examples, https://www.tensorflow.org/lite/examples,

[25] J. Huang, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," *Proc. CVPR*, 2017, pp. 7310–7311.

[26] TY. Lin, *et al.*, "Microsoft coco: Common objects in context," *Proc. European Conference on Computer Vision*, 2014, pp. 740–755.

[27] G. Davidson, "Google Pixel: The Complete Beginner's Guide," *Van Helostein*, 2017.

[28] ARCore supported devices. https://developers.google.com/ar/ discover/ supported-devices.

[29] https://wiki.dfrobot.com/Bluno_Nano_SKU_DFR0296.

[30] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," Proc. Int. joint conf. Artificial intelligence, 1981, pp. 674–679.

[31] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision," *Cambridge university press*, 2003.

[32] PJ. Huber, "Robust estimation of a location parameter," *Annuals of Math Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[33] T. Qin, P. Li, and S. Shen, "VINS-MONO: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004-1020, 2018.

[34] https://github.com/ManiiXu/VINS-Fusion-RGBD.