2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

# DLWV2: a Deep Learning-based Wearable Vision-system with Vibrotactile-feedback for Visually Impaired People to Reach Objects

Meng-Li Shih[1], Yi-Chun Chen[1], Chia-Yu Tung[1], Cheng Sun[1], Ching-Ju Cheng[1], Liwei Chan[2], Srenivas Varadarajan[3], and Min Sun[1]

***Abstract*— We develop a Deep Learning-based Wearable Vision-system with Vibrotactile-feedback (DLWV2) to guide Blind and Visually Impaired (BVI) people to reach objects. The system achieves high accuracy in object detection and tracking in 3-D using an extended deep learning-based 2.5-D detector and a 3-D object tracker with the ability to track 3-D object locations even outside the camera field-of-view. We train our detector with a large number of images with 2.5-D object ground-truth (i.e., 2-D object bounding boxes and distance from the camera to objects). A novel combination of HTC Vive Tracker with our system enables us to automatically obtain the ground-truth labels for training while requiring very little human effort to set up the system. Moreover, our system processes frames in real-time through a client-server computing platform such that BVI people can receive real-time vibrotactile guidance. We conduct a thorough user study on 12 BVI people in new environments with object instances which are unseen during training. Our system outperforms the non-assistive guiding strategy with statistic significance in both time and the number of contacting irrelevant objects. Finally, the interview with BVI users confirms that our system with distance-based vibrotactile feedback is mostly preferred, especially for objects requiring gentle manipulation such as a bottle with water inside.**

## I. INTRODUCTION

In Oct. 2017, an estimate of 253 million people lives with vision impairment [1]. For these Blind and Visually Impaired (BVI) people, tasks that seem trivial for those with normal sight–such as reaching a cup on a table (see Fig. 1 (a))–are challenging if they do not memorize the location of all objects. As a consequence, BVI people typically have superior memory skills and keep everything in a specific order [2]. Unfortunately, memorizing everything might hinder them to multitask in daily life and such strategy will fail when living in a shared space with others.

Several systems [3], [4], [5] have been proposed to assist BVI people in reaching objects. All of them are based on computer vision techniques. Some of them propose to mount camera close to head [4], wrist [3] or both [5]. For interfacing the systems with BVI users, some of them propose to use tactile [3], audio [4] or both [5]. Despite various system designs, all of them rely on handcraft visual features (e.g., SIFT and HOG) and classical object recognition methods which are typically less robust. As a result, most systems are not reliable enough to conduct a thorough user study without relying on simplified assumptions or even the wizard of oz prototype (i.e., assuming perfect perception).

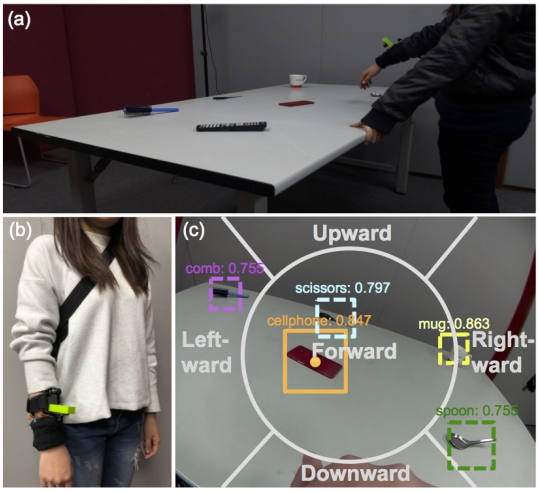


Fig. 1: The wearable system can guide users to reach target objects in new environments. (a) A blind user searching the desired object using vibration guidance. (b) Our wearable system. (c) Hand-camera view. The boxes are detected bounding boxes, and the solid-line one is the target object. White lines divide the guidance direction into upward, left-ward, downward, and rightward regions.

We propose DLWV2–a Deep Learning-based Wearable Vision-system with Vibrotactile-feedback–aiming to robustly guide BVI people to reach objects in real-time (see Fig. 1 (b)). Our system consists of "Perception" and "Guidance" modules. For "Perception", we leverage modern deep learning models to become significantly more robust than existing systems. In particular, our system achieves high accuracy in object detection (see Fig. 1 (c)) and tracking in 3-D using an extended deep learning-based 2.5-D detector and a 3-D object tracker with the ability to update 3-D object locations even outside the camera field-of-view. We need to train our deep learning-based detector with a large number of images with 2.5-D object ground-truth (i.e., 2-D object bounding boxes and distances from the camera to objects). One of our main contribution is to propose a novel combination of HTC Vive Tracker[1] with our system which enables us to automatically obtain the ground-truth labels for training while requiring very little human effort to set up the system. Moreover, our system processes frames in real-time through a client-server computing platform. Our "Guidance" module generates distance-based vibrotactile feedback to convey the

*We thank MOST 107-2633-E-002-001, National Taiwan University, Intel Corporation and Delta Electronics.

[1]National Tsing Hua University, Taiwan, shihsml@gmail.com
[2]National Chiao Tung University, Taiwan
[3]Intel Cooperation, California, USA

[1]https://www.vive.com/us/

direction and distance information to BVI users. In this way, the BVI users can expect when the objects are getting closer.

We conduct a thorough user study on 12 BVI people in new environments with object instances which are unseen during training. Our system outperforms the non-assistive guidance strategy with statistic significance in both time and the number of contacting irrelevant objects. Finally, the interview with BVI users confirms that our system with distance-based vibrotactile feedback is mostly preferred, especially for objects requiring gentle manipulation such as a bottle with water inside.

## II. Related Work

Several systems have been proposed to help Blind and Visually Impaired (BVI) people live better. Magalhes and Kohn [6] propose a posture stabilization system to help avoid falling. Systems assisting object reaching [3], [4], [5] and navigation [7], [8] have also been proposed. These systems typically consist of a perception module and a guidance module. In the following, we discuss their perception and guidance modules.

*1) Perception:* The perception module is typically based on classical computer vision technique which is not robust enough to conduct a thorough user study. The PalmSight system [3] relies on handcraft features such as SIFT, HoG to detect or track objects with a binocular camera on the palm. Thakoor et al. [4] propose a system with the head-mounted camera. However, the BVI people need to turn their head to find the object as the camera is mounted on the glasses. Besides, their method also relies on handcraft features. Similarly, the Third Eye system [5] contains a glasses and glove with cameras and relies on handcrafted features for perception. Recent advances in deep learning inspire us to build a system using efficient and robust CNN-based object detection [9].

*2) Guidance:* Most of the guidance modules use either audio channel [4], vibrotactile [3], [6] or both [7], [5] to convey guidance to BVI users. However, as mentioned in [7], audio may interfere with environment and/or overload the BVI users sensory capabilities. Hence, we focus on vibrotactile feedback in our guidance module.

User studies have been conducted on some of the previous systems [3], [4], [5]. However, their systems are typically not reliable enough to conduct a thorough user study without relying on simplified assumptions (e.g., recognizing the same object instance as in training) or even the wizard of oz prototype (i.e., assuming perfect perception). In contrast, we conduct a thorough user study on 12 BVI people in new environments with object instances which are unseen during training in order to test our system.

## III. Our System

Our system aims to guide BVI people to reach objects more accurately and efficiently with the following requirements:

1) A compact and low-cost solution.
2) A low latency and highly reliable system.
3) Natural user interfaces.

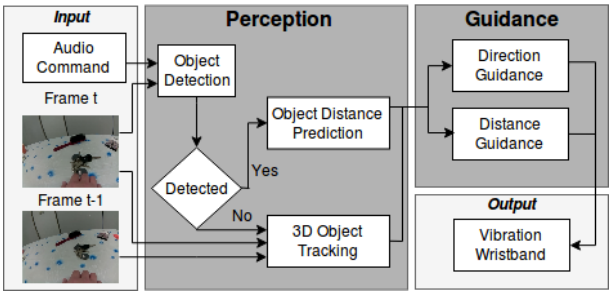


Fig. 2: System overview. The inputs are processed by the "Perception" module, then followed by the "Guidance" module. Finally, the guidance will be mapped to a vibration wristband as the system output. Notably, in the "Perception" module, if the target object is detected, its distance will be predicted and the process continues. If not, the 3-D object tracker will be activated to track the object location from the previous frame.

Hence, we design a wearable system using commodity hardware components (Sec. III-D). We further leverage a state-of-the-art deep learning algorithm (Sec. III-B) to attain high reliability and utilize a client-server computing platform (see implementation details in Sec. III-D) to achieve low latency. Finally, we use speech recognition to command the system and vibration feedback to guide users (Sec. III-C) without overwhelming the users sensory and cognitive capacities.

To achieve this task, we incorporate two essential components: vision perception and vibration guidance. In this chapter, we will give an overview of our system and then present each component and their implementation in detail.

### A. System Overview

Our system has two types of inputs: (1) a sequence of frames captured from a fish-eye camera and (2) an object category recognized by an on-device speech recognizer. The inputs are first processed by the "Perception" module; then, by the "Guidance" module. In the "Perception" module, our object detector aims to estimate the 3-D location of the target object. When the detector fails to detect the object, we propose a object tracker to keep tracking the 3-D location. The object tracker will track the previous estimated 3-D object location with respect to the new camera coordinate system. Next, in the "Guidance" module, our system calculates the action in order to steer the user's hand closer to the target object. The action turns into a vibration pattern on the wrist as the output of our system to interact with the user. The architecture and flow diagram of the system is shown in Fig. 2. In the following, we introduce our "Perception" and "Guidance" modules.

### B. Perception

To estimate the 3-D object location, the "Perception" module consists of a real-time 2.5-D object detector and real-time 3-D object tracker. Specifically, the 2.5-D object detector provides the direction and distance information of the target object. Given this information, we compute a 3-D point as a proxy of the 3-D object location. During the guidance, if the detector misses the target object, the 3-D

object tracker will track the 3-D object location in the current camera coordinate system due to the camera motion on the wrist. The combination of detector and tracker ensures us to continuously estimate the 3-D object location even under challenging detection condition. Next, we will detail the 2.5-D object detector and 3-D object tracker.

*1) **2.5-D Object Detector**:* The 2.5-D object detector is designed to complete two real-time tasks: (1) localizing the objects in the 2-D frames, (2) predicting the distance between the hand camera and the objects. In this way, our system can guide BVI people to reach the objects with both direction and distance of the objects in mind. In order to achieve real-time performance, we choose YOLO9000 to be our object detection architecture. However, it is only able to predict the 2-D locations (i.e., 2-D bounding boxes) of objects, but not the distance. Therefore, we design a function to predict the distance. Furthermore, we employ the function to automatically collect 2-D bounding boxes at scale.

**Distance from detection.** Rather than introducing another distance prediction model, we want to directly decode the distance $d$ from the size of the object 2-D bounding box. This is possible, since, intuitively, the closer the hand is to an object, the larger the size of the object is to be captured in a 2-D image. More precisely, the object size in a 2-D image is roughly proportional to the inverse distance $1/d$. Therefore, we use the diagonal length $l$ of the object bounding box to represent the object's size in the 2-D image. Then, we define the relationship between the diagonal length and the object's distance using as below,

$$d = \frac{\alpha_c}{l}, \tag{1}$$

where $\alpha_c$ is the hyperparameter of the object category $c$. As we assume objects in a given category are of equal size, the value of $\alpha_c$ is a constant within a certain object category. To find the scalar $\alpha_c$, we manually annotated 100 bounding boxes in different distance for each object category and fit $\alpha_c$ with the least square error. To obtain the distance information from the bounding boxes, we simply use Eq. 1 to calculate the distances according to the diagonal lengths of bounding boxes. In this way, our detector can predict both the locations and distances of the objects.

**Scaling-up 2-D box annotation.** An important insight from Eq. 1 is that ground-truth 2-D bounding boxes for training our detector can be directly computed from 3-D object locations provided by HTC Vive Tracker. We first project a 3-D location onto the image to obtain a 2-D object center $o$ and distance $d$. Then, we can compute the diagonal length $l$ of a box using Eq. 1. Given the object center $o$ and box diagonal length $l$, we can obtain the 2-D object box for training without human annotation. In Sec. IV, we describe how to obtain 3-D object locations using HTC VIVE Tracker at scale.

**Correcting 2-D box annotation** To adapt the distortion introduced by a fish-eye camera, 2-D bounding boxes need to be corrected. The correction consists of two steps. First, we shift the location of the box center from $(x, y)$ to $(x_c, y_c)$ by using Eq. 2, Eq. 3, and Eq. 4 from OpenCV [10].

$$r = \sqrt{x^2 + y^2} \tag{2}$$

$$x_c = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + (2p_1 xy + p_2(r^2 + 2x^2)) \tag{3}$$

$$y_c = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + (p_1(r^2 + 2x^2) + 2p_2 xy)\ , \tag{4}$$

where $(x, y)$ represent the original point $o$ on the image plane, $(x_c, y_c)$ is the new point after correction, $(k_1, k_2, k_3)$ are barrel distortion parameters, and $(p_1, p_2)$ are tangential distortion parameters. Second, we resize the 2-D bounding box by multiplying its height and width by $scale_x$ and $scale_y$, where $scale_x = \frac{\partial x_c}{\partial x}$ and $scale_y = \frac{\partial y_c}{\partial y}$. Finally, we can acquire modified bounding boxes for YOLO9000 object detection.

**Implementation** We train YOLO9000 network on our dataset for 35000 iterations with a starting learning rate of 0.001 and divide it by 10 at 10000 and 20000 iterations, using pre-trained Darknet neural network framework [11]. We use a weight decay of 0.0005 and momentum of 0.9.

*2) **3-D Object Tracker**:* The 3-D object tracker is designed to track the 3-D object location when the target object is outside the camera field-of-view. Since we assume the target object is stationary, we decide to estimate the camera motion rather than directly track the object. In this way, our system can keep tracking the object when the target object is outside the camera view or the detection of the target object is missing.

**Camera motion estimation** Given two consecutive frames, we propose to apply a deep learning based visual odometry method (DeepVO [12]) to estimate the 6-DoF camera motion with absolute scale. Note that DeepVO has two advantages over geometry-based methods: (1) geometry-based methods cannot estimate the absolute scale of the camera motion with a monocular camera, and (2) DeepVO can be trained in an end-to-end fashion without explicitly handle the distortion caused by the fish-eye camera. In order to train DeepVO, we propose to collect a large number of image sequences with ground truth 6-DoF camera motion in absolute scale via HTC Vive Tracker.

**Tracking 3-D object location.** We represent the 6-DoF camera motion using a rotation matrix $R$ and a translation vector $T$. Given the predicted $R$ and $T$ from DeepVO, we transform the previous 3-D object location $P_{t-1}$ into the updated 3-D object location $P_t$ in the current camera coordinate at time $t$ as follows,

$$P_t = R \cdot P_{t-1} + T \tag{5}$$

**Implementation** Similar to DeepVO [12], we fine tune the pre-trained FlowNet [13] on our dataset to predict the camera motion. We train DeepVO network on our dataset for 10000 iterations with a starting learning rate of 0.0001 and divide it by 2 at 3000 and 6000 iterations. We use a weight decay of 0.00004 and momentum of 0.9. For our dataset, we predict rotation represented by Euler angles and translation in Euclidean space.

The 3-D object tracker works with the 2.5-D object detector in order to compensate detection failures. In particular,

the 3-D object tracker will stop if (1) the target object has been re-detected, or (2) the target object hasn't been detected for 3 seconds.

### C. Guidance

Most electronic aids for BVI people transmit signals to the users via either haptic, audio format or both. Wang et al. 2017 [7] referred that audio feedback can not only be indistinguishable in noisy environments, but interfere with the perception of auditory environmental cues on which BVI people depend for situational awareness. Consequently, we choose to use a vibrotactile feedback wristband to deliver vibration guidance to BVI people for less possible interference from the surroundings. Before illustrating the vibration guidance, we first show how user wears our device in Fig. 4 (a). Since we mount the camera very close to the right-hand wrist, we set hand position as the origin in the camera coordinate. We also set the positive x-axis, positive y-axis, and positive z-axis of the camera to be aligned with the rightward direction, upward direction, and forward direction of right-arm.

To guide the user, we design a vibration system using five vibration motors (see Fig. 4 (c)). Each motor corresponds to a guidance direction (i.e., upward, downward, leftward, rightward, or forward). Moreover, each vibration period maps to guidance distance - specific ranges of distances between the camera and the target object. In the following, we describe how 3-D object location is mapped to different direction and distance guidance, and we give more details of the design of the vibration wristband.

*1) Direction Mapping:* We map a 3-D object location into five directions: "forward", "rightward", "upward", "leftward", and "downward". The procedure consists of three steps. First, we connect the camera center and the 2-D object location to form a ray in 3-D referred to as the object ray (see Fig. 3). Then, we calculate the angle $\theta$ between the object ray and the forward axis (z-axis) of camera coordinate. When this angle is less than 30 °, the guidance direction is mapped to "forward". Otherwise, we project the object ray onto the 2-D plane defined by the x-axis and y-axis. We refer the projection as projected object ray. The angle $\phi$ between the projected object ray and the positive x-axis is used to map a 3-D object location to "rightward", "upward", "leftward", or "downward" directions when $\phi$ falls in (-45°(315°), 45°], (45°, 135°], (135°, 225°], or (225°, 315°], respectively (see Fig. 3).

*2) Distance Mapping:* The goal of distance mapping is to notify users how far they are to the target objects through varied vibration period. For instance, when a user's hand is getting closer to the target object, the vibration will be triggered more frequently. We describe the mapping from the distance to vibration period $v$ as Eq. 6. In our pilot user study, subjects find it sufficient to start distance guidance when the distance from the target object is within 0.5 meters. Hence, we set the upper limit distance as 0.5 meters in Eq. 6.

$$v = \frac{\min(d, 0.5)}{0.5} \times 0.6 \ , \tag{6}$$

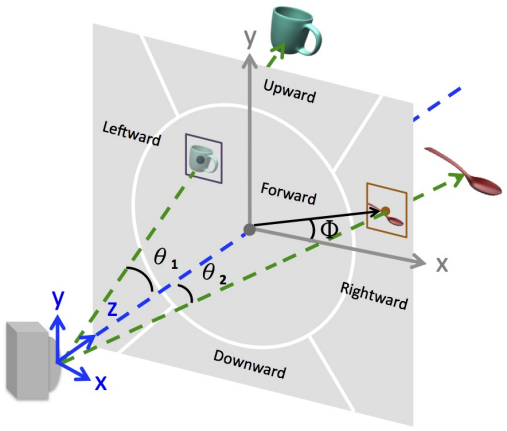


Fig. 3: Illustration of direction mapping. First, the camera center and the object are connected to form an object ray (green dashed line). Angle $\theta$ is between the object ray and the z-axis (blue dashed line). $\theta_1$ is less than 30°, so the guidance direction for the mug is mapped to "forward". $\theta_2$ is more than 30°, so the spoon's ray is projected onto a 2-D plane (black solid line). Then we calculate the angle $\phi$ between the projected object ray and the x-axis. Here $\phi$ falls in (-45°(315°), 45°], so the guidance direction for spoon is mapped to "rightward".

where $v$ is vibration period in seconds. When the distance between target objects and users are more than 0.5 meters, the vibration period is set to a constant. On the other hand, when the distance between target objects and users are less than 0.5 meters, the vibration period will become linearly proportional to the distance. Through Eq. 6, the distance information can be conveyed to users clearly.

*3) Vibration wristband:* Our vibration wristband comprises a sports wristband and five micro vibration motors (see Fig. 4 (c)). One of the five micro vibration motors is on the "front-ring" of the sports wristband, and it will be activated when the guidance direction is "forward". The other four micro vibration motors are on the "rear-ring" of the wristband, located on the right side, left side, up side, and down side of the wrist. These four vibration motors will be activated when the guidance direction is mapped to "rightward", "leftward", "upward", and "downward", respectively. Moreover, the shorter vibration period indicates the object is closer to the hand, and vice versa. We tune the strength of the vibrotactile feedback to ensure that users can feel and interpret the vibrotactile feedback correctly.

### D. Hardware Component

To command the system and vibrotactile feedback, we employ a speech recognition technique [14] in our system. Users can utilize audio command to specify the object they intend to reach. A microphone and a button are attached to the vibration wristband (see Fig. 4 (c)). After pressing down the button, the user's command will be recorded and classified into one of the ten object categories (e.g., "Wallet"). Then, the target object will be specified.

As shown in Fig. 4 (b), a CCD Raspberry Pi camera is mounted on the right elbow to capture image sequence. The


Fig. 4: (a) Blue coordinate system is the camera coordinate, and the green one offsets from the blue one, representing hand-moving directions. (b) CCD camera. (c) Vibration wristband and its perspective. The green spots are brushless micro vibration motor for different direction guidance.

image sequence will be transmitted to the remote server, which serves as a work station, via WLAN by MJPG package for further analysis. In the remote server, the image sequences will be converted to vibration guidance by our perception module and transmitted back to the vibration wristband. The total latency is about 0.3 seconds when the frame rate is 5 FPS and the resolution is $480 \times 640$. Besides, we mount a microcomputer in a shoulder bag to control the camera, microphone, button and vibration motors.

## IV. Dataset

To assist BVI people in reaching daily-life objects more efficiently and accurately, our system primarily focuses on detecting ten daily life object categories, which include mugs, keys, combs, wallets, spoons, remotes, scissors, cellphones, banknote, and plastic bottles. For each object category, we collected eight instances for training and two others for evaluation. To collect the dataset, the collector needs to wear the camera along with an HTC Vive Tracker on the elbow and follow a data collection procedure which consists of the following three steps. First, put multiple objects on a table randomly and annotate their 3-D locations by another HTC Vive Tracker. Next, start recording the image sequence and swing the elbow to let the camera capture the objects at different viewpoints. Finally, move the elbow close to the object gradually and stop recording the video once the object is reached. Through such procedure, the camera pose and all 3-D location of objects corresponding to every image are automatically annotated.

To train our object detector, we need annotated ground-truth bounding boxes which cannot be acquired directly from the procedure above. However, using the method mentioned in Sec. III-B, we can automatically generate a large number of bounding boxes for the objects, while only 1000 bounding boxes need to be annotated manually to estimate $\alpha$ in Eq. 1. We also present the collection procedure in the supplementary video.

In addition, our camera is allowed to capture the objects from any viewpoint, which increases the variety of our dataset. As a result, our detection model can learn more diverse features through this dataset. Our dataset includes 648 image sequences, 47855 images for training, and 57 image sequences, 13118 images for testing.

TABLE I: Result of 2.5-D Object Detection. Our 2.5-D object detector achieves reliable quantitative results for conducting user study (above 90% on Precision, above 72% on Recall, and less than 0.13m on distance error).

| Category | Precision (%) | Recall (%) | Distance Error (cm) |
|---|---|---|---|
| **key** | 99.3 | 80.0 | 9.58 |
| **mug** | 99.9 | 74.5 | 13.95 |
| **plastic bottle** | 99.9 | 72.3 | 9.51 |
| **comb** | 94.9 | 96.1 | 10.53 |
| **wallet** | 97.7 | 85.3 | 13.37 |
| **spoon** | 95.4 | 93.7 | 13.65 |
| **remote** | 99.7 | 93.2 | 13.70 |
| **scissors** | 90.0 | 99.6 | 13.83 |
| **banknote** | 99.2 | 85.9 | 8.42 |
| **cellphone** | 99.9 | 80.5 | 11.76 |

TABLE II: Result of 3-D Object Tracker. Translation error (cm) and rotation error (degree) of baseline and tracker system.

| | Rotation (deg) | Translation (cm) |
|---|---|---|
| **Baseline** | 4.32 | 3.92 |
| **3-D Object Tracker** | 2.00 | 2.77 |

## V. Perception Module Validation

We validate the reliability of detection and tracking algorithm on our testing dataset before the user study.

**2.5-D Object Detector** We evaluate the 2.5-D object detector on the testing set of our dataset. Among all object categories, the precision rates are above 90%, and the recall rates of object classes are above 72% when we set the detection threshold equal to 0.4 (see Tab. I). Moreover, the distance errors of object classes are all lower than 0.13 meter, which is shorter than the length of a human palm. Such validation results prove our detector can be used to estimate 3-D object location in an acceptable range at first sight. Further user experiment supports our inference (see Sec. VI).

**3-D Object Tracker** As described in Sec. III-B.2, we estimate the camera motion (i.e., rotation and translation) to update the 3-D object location. Because reaching object is a sequential action, the camera pose and 3-D object location according to the camera are within small deviation between adjacent frames. Therefore, a baseline is to ignore the camera motion (i.e., do not update the 3-D object location). For camera motion estimation, we evaluate the performance of our tracker and the baseline using translation error in centimeter and rotation error in degree. Tab. II shows that our tracker achieves lower errors than the baseline method in camera motion. Moreover, we are interested in the tracking effect when the object locates outside the camera view or the detection of the object is missing. We compute the Euclidean error in centimeter between the predicted 3-D position of the target object and the ground-truth 3-D position annotated by

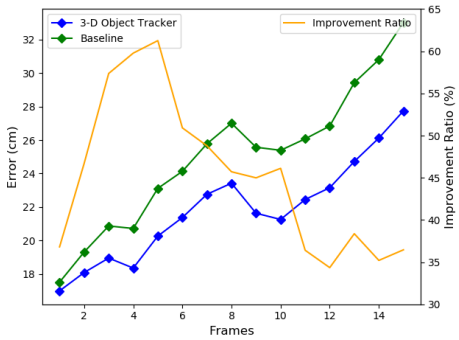


Fig. 5: Error of 3-D object tracker at frames after object detector fails to detect the target object.

HTC Vive Tracker. As shown in Fig. 5, the error of the predicted 3-D object location grows as the error of estimated camera motion accumulated over time. Nevertheless, it can be alleviated by over 34% with our tracking method. Therefore, the object tracker is more effective to track the target object outside the camera view than the baseline.

## VI. User Studies

### A. Experimental Setup

To measure the efficacy of the proposed system, we design a task of finding daily necessities. In the task, subjects need to find the target object on a table in limited time. Before the formal user study, we have conducted a pilot experiment in order to define the following experiment setup.

*1) Environment:* The user study is conducted in new environments and on the object instances that are unseen during training. The starting point of the experiment is 50 centimeters in front of the table. For the table-top setup, we define three levels of distance from the starting point (see Fig. 6). Before each trial, three to four objects are placed in each distance level randomly. In this way, subjects would not memorize the target object location as we shuffle the desktop layout. Furthermore, to exclude outliers, each trial has a time limit of 60 seconds, which is two times longer than the average time of all trials in the pilot test.

*2) System Settings:*

- Baseline: Subjects will search the target objects with their right hands without any assisted system.
- System 1: Our system will detect the target objects on the table and use vibration guidance to guide subjects to reach the targets.
- System 2: A variant of System 1. We remove distance information from vibration guidance. That is, the vibration period is uniform at all distances.
- Oracle: We use HTC Vive Tracker to obtain ground-truth 3-D object location. It is an oracle system combining perfect perception and our distance-based guidance method (i.e., the upper-bound performance of our system). However, this system is limited to environments equipped with HTC Lighthouse and Vive Tracker. The distance between two base stations of HTC lighthouse has to be less than 5 meters.

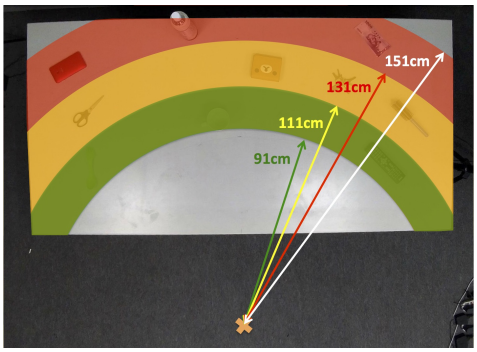


Fig. 6: Environment: The starting point (orange cross) is 50 cm in front of the desktop. Green: Distance level "Near", 91cm to 111cm from starting point. Yellow: Distance level "Medium", 111cm to 131cm. Red: Distance level "Far", 131cm to 151cm.

TABLE III: Measurements of time and number of superfluous contacts. Means and standard deviations for the four settings along with the p-values ($*p < 0.05$, $**p < 0.01$) which tested by one-way ANOVA. If the p-value is lower than 0.05, there exist certain significantly different groups.

| | Time Consumption (sec.) | Number of contacts | Success Rate |
|---|---|---|---|
| **B** | $23.60 \pm 13.08$ | $5.28 \pm 3.32$ | 93.06% |
| **S1** | $16.86 \pm 8.76$ | $0.41 \pm 0.82$ | 98.61% |
| **S2** | $18.27 \pm 9.45$ | $0.55 \pm 1.12$ | 95.83% |
| **Oracle** | $12.89 \pm 6.52$ | $0.49 \pm 0.69$ | 100% |
| **p-value** | ** **p<0.01** | ** **p<0.01** | $p = 0.08$ |

In each setting, subjects are asked to reach and find target objects six times, two times for each distance level. To prevent the learning effect in repeated experiments, we set the order of settings based on a Latin Square design.

*3) User:* We conducted a user study with 12 totally blind users. It took 80 minutes for each participant to finish the experiment. In the experiment, the participant would wear the wearable assistive system we proposed (see Fig. 4 (a)). Before the experiment of each setting, we gave them instruction on utilizing our system to let them be familiar with the settings. At the end of the experiment, we interviewed the subjects experience.

Through this experimental setup, we can appropriately measure the proposed system's efficacy.

### B. Time and Superfluous contacts

To highlight the efficiency and accuracy when using our systems, we did a comparison of the four systems on time consumption and the number of superfluous contacts with irrelevant objects. Tab. III shows that S1, S2 outperform the baseline in average time consumption conspicuously. Aside from time consumption, the assistive systems can also reduce 5.28 times superfluous contacts in baseline to less than one time. Among all assistive systems, Oracle consumes less time than S1 and S2 do. The difference is due to the camera's limited angle of view in S1 and S2. In S1 and S2, the target object needs to be captured by our camera to activate the vibrotactile guidance. Hence, subjects have to spend time on

TABLE IV: Measurements of time consumption and number of superfluous contacts within each distance level. S1 and S2 guide subjects to reach target objects with less time consumption and number of contacts in every distance level than baseline.

| | Time Consumption (sec) | | | Number of Contacts | | |
|---|---|---|---|---|---|---|
| | **Near** | **Medium** | **Far** | **Near** | **Medium** | **Far** |
| **B** | 23.07 | 22.25 | 25.50 | 5.09 | 4.82 | 5.95 |
| **S1** | 14.76 | 17.63 | 18.23 | 0.25 | 0.48 | 0.50 |
| **S2** | 17.23 | 17.11 | 20.63 | 0.33 | 0.65 | 0.68 |
| **Oracle** | 11.94 | 12.65 | 14.09 | 0.29 | 0.54 | 0.62 |

TABLE V: Post-hoc Tukey HSD Test results on the measurements. The critical value of the Tukey HSD $Qstatistic$ based on the $k = 4$ groups and $v = 275$ degrees of freedom for the error term, for significance level $\alpha = 0.01$ and 0.05 (p-values) in the Standardized Range distribution. We present the significance (p-value) of the observed $Qstatistic$ of $Q_{i,j}$.

| | Time Consumption (sec.) | | Number of contacts | |
|---|---|---|---|---|
| **Pair** | **Q statistic** | **p-value** | **Q statistic** | **p-value** |
| B vs S1 | 5.78 | ** **p<0.01** | 21.67 | ** **p<0.01** |
| B vs S2 | 4.54 | ** **p<0.01** | 21.67 | ** **p<0.01** |
| B vs Oracle | 9.22 | ** **p<0.01** | 22.20 | ** **p<0.01** |
| S1 vs S2 | 1.2 | 0.80 | 0.66 | 0.90 |
| S1 vs Oracle | 3.47 | 0.07 | 0.37 | 0.90 |
| S2 vs Oracle | 4.67 | ** **p<0.01** | 0.30 | 0.90 |

scanning the table with the hand camera first. In contrast, Oracle can locate 3-D positions of target objects and users' hands directly, so it is able to guide the users immediately after the experiment starts.

Also, we analyze the results at 3 distance levels (see Tab. IV). We find users spent less time and made less number of superfluous contacts using S1 and S2, and that these systems are more efficient than baseline at all distance levels.

On the other hand, we did a Tukey HSD test to identify the significance of the difference in each pair group (see Tab. V). The results support our previous summary that S1 and S2 are significantly different from baseline, and can guide subjects reach target objects more efficiently and accurately.

The comparison between S1 and S2 highlights the importance of distance-based vibrotactile feedback. Although the difference is less significant than all the other pairs, in a post-study interview, 9 out of 12 subjects considered the distance information provided by the vibration guidance helpful. Hence, we further design an experiment to highlight the importance of distance information in Sec. VI-D.

Overall, less time consumption and the less number of superfluous contacts prove that our system is more efficient and accurate in guiding BVI people to reach target objects.

### C. Hand Search Space and Hand Moving Trajectory

Our system helps users do efficient, local search around the target object instead of exhaustive search over the whole table. To support this argument, we monitor hand moving trajectory using the birds' eye view camera (see Fig. 7). In the baseline setting, the subject's hand tended to carry out an exhaustive search. In S1 and S2 settings, the trajectories show that subjects follow instructions to reaching the object: (1) reaching the table, (2) scanning the objects on the table with

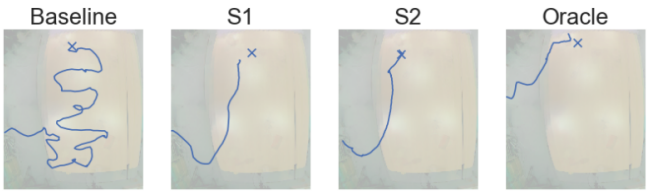


Fig. 7: Typical trajectory patterns of different system settings.

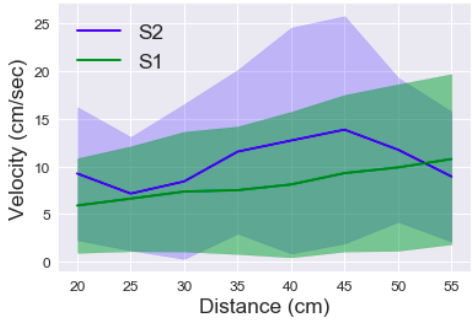


Fig. 8: Blue and green solid lines are users' mean hand-moving velocities over distance under S1 and S2 systems, respectively, while blue and green shaded areas are the ±standard deviation.

the camera until the target object is detected, and (3) heading toward the target object with vibration guidance (see the supplementary video). In Oracle setting, the trajectory shows that subjects headed toward the object by the vibration.

In addition, we use a violin plot in Fig. 9 to show distance distribution compared across each system. In the baseline setting, distance distribution is uniform over all the distance. In contrast, S1, S2, and Oracle all have high density distribution over short distances, which indicates that users tend to do a local search when they are close to the targets with assistive systems.

### D. Object Distance Effect

There are scenarios that the desired objects need to be reached carefully. In such case, the distance information becomes essential. Hence, to identify the benefit of distance information, subjects are asked to reach an easily knocked-down plastic bottle using S1 and S2. Then, we analyze the relation between hand-moving speed and the distance between the hand and the object. We find that subjects would establish a rough object distance in their mind after several trials in experiments. Hence, they tended to lower the hand-moving speeds roughly when they were approaching to an easily knocked-down plastic bottle. However, with the distance information, their hand-moving speeds can decrease more steadily than in S2 (see Fig. 8). Also, among all 12 subjects, 9 of them expressed that the distance information helped them establish precise object positions in mind with more confidence. Furthermore, in Fig. 9, since S1 shows higher distribution density at short distances than S2 does, we infer that users will do local searches with distance information. Finally, we conclude that (1) the distance information is proved necessary through varied vibration period, (2) the local search property of S1 is stronger than that of S2.

### E. Failure Case

There are some failure cases worth discussion. When subjects fail to reach the target object within the time limit, we label the trials as the failure cases. In our user study, there are 5 out of 72 failure cases in the baseline setting, 1 failure cases in S1, 3 failure case in S2, while 0 failure case in Oracle. Next, we will list some factors that cause such failures. The failure cases in the baseline setting are due to human error as there is no assistance from any system. To illustrate, since some subjects search the table arbitrarily and do not exactly bear in mind which area they have searched, some objects are easy to be missed.

For S1 and S2, the failure reason consists of human error and functional error in the visual algorithm. For the human error, few subjects are not sensitive to the vibrotactile-feedback or may feel fatigue after receiving the vibrotactile-feedback for a while.

Next, there are two types of functional error. The first one is false positives detection. In most of the time, the false positive will vanish within few frames. However, if the false positive has existed for a period, subjects will reach the miss-classified object. The other type is occlusion. Our visual algorithm can detect the partially occluded target objects. However, if the target object is occluded severely, this will cause failure cases. In short, human error, false positives, and occlusion are in majority of all failure cases.

### F. Post-study Interview

In the post-study interviews, most of the users replied that this system can help them find desired objects in unfamiliar environments. Besides, 10 out of 12 subjects described our system a reliable assistive equipment after the experiments. Also, they considered our ability to detect unseen objects as a great feature. In this way, they do not need to collect their own database of personal objects. Some subjects also suggested that we can extend the search space from a 2-D plane (e.g., table) to a 3-D layout (e.g. refrigerator, cabinet) to further enlarge the capability of our system. We take this suggestion as great direction for future work.

## VII. Conclusion

We develop a Deep Learning-based Wearable Vision-system with Vibrotactile-feedback (DLWV2) to guide Blind and Visually Impaired (BVI) people for reaching objects. To train our system, a novel combination of HTC Vive Tracker with our system enables us to automatically obtain the ground-truth labels while requiring very little human efforts to set up the system. We conduct a thorough user study on 12 BVI people in new environments and object instances which are unseen during training. Our system outperforms the non-assistive guiding strategy with statistic significance. Finally, the interview with BVI users confirms that our system with distance-based vibrotactile feedback is mostly preferred.

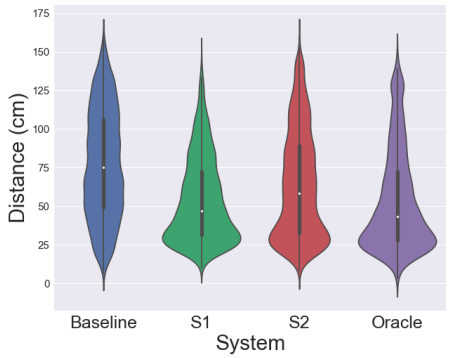


Fig. 9: Violin plot shows distance distribution compared across four system settings. Compared with the uniform distance distribution in baseline, S1 and Oracle show high density distributions over short distances. This indicates that users would spend more time to do local search when their hands are close to target objects. Within each violin, there is a box plot showing 25th, 50th (median), 75th percentile.

## References


[1] W. H. Organization, "Vision impairment and blindness," 2017. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs282/en/

[2] L. SCIENCE, "Blind people have superior memory skills," 2017. [Online]. Available: https://www.livescience.com/4503-blind-people-superior-memory-skills.html

[3] Z. Yu, S. J. Horvath, A. Delazio, J. Wang, R. Almasi, R. Klatzky, J. Galeotti, and G. D. Stetten, "Palmsight: an assistive technology helping the blind to locate and grasp objects," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-16-59, December 2016.

[4] K. Thakoor, N. Mante, C. Zhang, C. Siagian, J. Weiland, L. Itti, and G. Medioni, *A system for assisting the visually impaired in localization and grasp of desired objects*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Germany: Springer Verlag, 2015, vol. 8927, pp. 643–657.

[5] P. A. Zientara, S. Lee, G. H. Smith, R. Brenner, L. Itti, M. B. Rosson, J. M. Carroll, K. M. Irick, and V. Narayanan, "Third eye: A shopping assistant for the visually impaired," *Computer*, vol. 50, no. 2, pp. 16–24, Feb. 2017.

[6] F. H. Magalhes and A. F. Kohn, "Vibration-enhanced posture stabilization achieved by tactile supplementation: May blind individuals get extra benefits?" *Medical Hypotheses*, vol. 77, no. 2, pp. 301 – 304, 2011.

[7] H.-C. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarré, and D. Rus, "Enabling independent navigation for visually impaired people through a wearable vision-based feedback system," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6533–6540, 2017.

[8] S. Mattoccia and P. Macr, "3d glasses as mobility aid for visually impaired people," pp. 539–554, 09 2014.

[9] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[10] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[11] J. Redmon, "Darknet: Open source neural networks in c," *Pjreddie. com.[Online]. Available: https://pjreddie. com/darknet/.[Accessed: 21-Jun-2017]*, 2016.

[12] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2043–2050.

[13] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.

[14] B. McFee, M. McVicar, O. Nieto, S. Balke, C. Thome, D. Liang, E. Battenberg, J. Moore, R. Bittner, R. Yamamoto, D. Ellis, F.-R. Stoter, D. Repetto, S. Waloschek, C. Carr, S. Kranzler, K. Choi, P. Viktorin, J. F. Santos, A. Holovaty, W. Pimenta, and H. Lee, "librosa 0.5.0," Feb. 2017.