A Wearable Assistive Technology for the Visually Impaired with Door Knob Detection and Real-Time Feedback for Hand-to-Handle Manipulation

Liang Niu[1,2*] Cheng Qian[1,2,*] John-Ross Rizzo[2,3,*] Todd Hudson[3] Zichen Li[1,2]
Shane Enright[3] Eliot Sperling[3] Kyle Conti[3] Edward Wong[1,2]
Yi Fang[1,2,4,†]
[1] NYU Multimedia and Visual Computing Lab, USA
[2] NYU Tandon School of Engineering, USA
[3] NYU Langone Medical Center, USA
[4] NYU Abu Dhabi, UAE

Abstract

The visually impaired are consistently faced with mobility restrictions due to the lack of truly accessible environments. Even in structured settings, people with low vision may still have trouble navigating efficiently and safely due to hallway and threshold ambiguity. Assistive technologies that are currently available do not provide door and door-handle object detections nor do they concretely help the visually impaired reaching towards the object. In this paper, we propose an AI-driven wearable assistive technology that integrates door handle detection, user's real-time hand position in relation to this targeted object, and audio feedback for "joy stick-like command" for acquisition of the target and subsequent hand-to-handle manipulation. When fully envisioned, this platform will help end users locate doors and door handles and reach them with feedback, enabling them to travel safely and efficiently when navigating through environments with thresholds. Compared to the usual computer vision models, the one proposed in this paper requires significantly fewer computational resources, which allows it to pair with a stereoscopic camera running on a small graphics processing unit (GPU). This permits us to take advantage of its convenient portability. We also introduce a dataset containing different types of door handles and door knobs with bounding-box annotations, which can be used for training and testing in future research.

1. Introduction

1.1. Background

According the World Health Organization circa 2014, there were 39 million people suffering from blindness worldwide with 82% of them at or above the age of 50. Additionally, there were 246 million people with low vision. In just the United States, billions of dollars are spent per annum towards direct and indirect medical cost for vision-related illness. In fact, the total economic impact of blindness and visual impairment is estimated to be approximately 3 trillion dollars globally.

Low vision, an impairment of visual information acquisition and/or processing, is well-known to hinder both spatial perception and object detection. This creates a myriad of functional mobility difficulties, including but not limited to trips, falls, and head injuries. Even in situations where an object may be partially visualized or roughly localized, difficulties abound. On average, human eyes are horizontally separated by about 65 mm; each eye has a slightly different view of the surrounding world. By comparing these two views, our brain can infer not only depth, but also 3D motion in space. Thus, people with visual impairment lose this crucial environmental information not only when both eyes are affected by pathology but even when one eye is affected, resulting in an incomplete sense of depth and distance.

Current mobility aids within the assistive technology space, such as canes and adaptive mobility devices, focus on improving gross environmental perception to augment safety and improve efficiency. These macroscopic views do not afford granular details about environmental constraints or potential environmental hazards that are "out of reach" of the device itself and often provide only partial situational awareness. This is not only limited to outdoor navigation

*Equally contributed
†Corresponding author: Yi Fang (yfang@nyu.edu)

but also indoor navigation. One such difficulty at a categorical level is hallway and threshold ambiguity, where individuals have problems locating a threshold, the related door handle or knob, and manipulating this targeted object for seamless travel. Despite automatic doors and audio-enabled, spoken-word prompts representing an ideal, there always seems to be a lack of truly accessible environments for those with low vision. While the present norm of outfitting environments with accessible hardware is certainly feasible and presently obtainable, the scale of the endeavor to achieve universal application is near impossible.

Given recent progress in neural networks and deep learning, many achievements have been made in image detection and computer vision. Recently, much scientific research has expanded into 3D scenarios, with the additional step of extracting depth information. 3D cameras, with two lenses aligned horizontally to capture images simultaneously, work biomimetically to human eyes. By calculating the differences (or disparities) between every pixel in the left eye image with its corresponding pixel in the right eye image, "disparity images" or "disparity maps" can be reconstructed. Given the distance between the center of the two lenses and the focal length, depth information can be obtained based on the two camera images. Thus, depth sensing and motion tracking become not only possible, but also computationally affordable and statistically accurate. This paper is developed based on these two areas: computer vision / deep learning neural network for object detection, and depth sensing for expanding such spatial localization from 2D to 3D environment.

1.2. Related Works

With recent improvements in powerful GPUs and algorithms, the processing time for object detection and recognition has been significantly reduced without jeopardizing accuracy. Deep neural networks (DNN), especially convolutional neural networks (CNN), are so far the most efficient and productive model for such computer vision tasks. One of the first networks to exploit the DNN construct for object detection was the R-CNN, a CNN-based classifier [5]. While surpassing previous models by a large margin, its computational burden limits its speed. The Spatial pyramid pooling (SPP)-based network [6] alleviated the issue by computing CNN features once per image. The Fast-RCNN [4] improved further on the SPPNet with ROI pooling layers, and its successor, the Faster-RCNN [17], significantly increased detection speed by making the object-proposal network differentiable. This body of work has laid a solid foundation for computer vision tasks, followed by many productive applications with fast detection [1, 15]. Preliminary experiments and explorations, mostly based on mobile platforms, have been conducted in real-time. Projects such as real-time object detection on a mobile device [2], door handle detection based on shape detection and color segmentation [9, 10], laser perception [19], and combination of multiple features [24, 7] have all achieved initial success.

Several researchers have taken the research one step further and integrated 3D-image detection models with larger systems that require more human interaction or provide similar assistance. Notable projects include Camera Input-Output system [20], sensor fusion with infrared for obstacle identification [18], RGB-D image based detection [23], indoor staircase detection [3], and wearable systems for enhanced monitoring and mobility [21]. However, there are a few that focuses on the visually impaired and goals or tasks that involve character detection [13] or travel aid [12], even in 3D glasses [11].

1.3. Our solution: Door Knob Detection with Joy Stick Control

To address the mobility issue mentioned previously, we have developed a wearable assistive technology that detects door handles and user's real-time hand position in relation to this targeted object. The system also provides audio feedback for "joy stick-like command prompts" for hand-to-handle manipulation, allowing the user to conveniently reach out for the door handle with initial spatial guidance and improved threshold navigation efficiency.

Figure 1 illustrates the pipeline of this proposed project, which has three major components. The first component is the deep learning neural network that can perform two-class object detection – door/door handles and human hands, and it will return the corner coordinates of the bounding box enclosing the object. The second component is the image flow and depth information extractor (stereo camera). The third component is a hard-coded program that integrates the information flow: combining the position of the detected objects along with the depth information extracted from the camera to obtain its 3D spatial location. It then transforms such information into descriptive sentences and outputs it with synthesized voice to the users, providing joy stick-like commands to the user, assisting in-hand control with feedback assistance.

Figure 2 demonstrates the system of devices in a nutshell. The image is captured by our stereo camera (1) mounted in front of the user's chest; then the data are processed in the portable GPU (2) carried in the bag, finally the audio output containing localization information will be sent to the user via a Bluetooth headset (3), helping them navigate and locate the door knob.

2. Method

To complete the proposed project, the system must integrate image fetching, real-time object detection, and real-time depth computation into a wearable device to provide

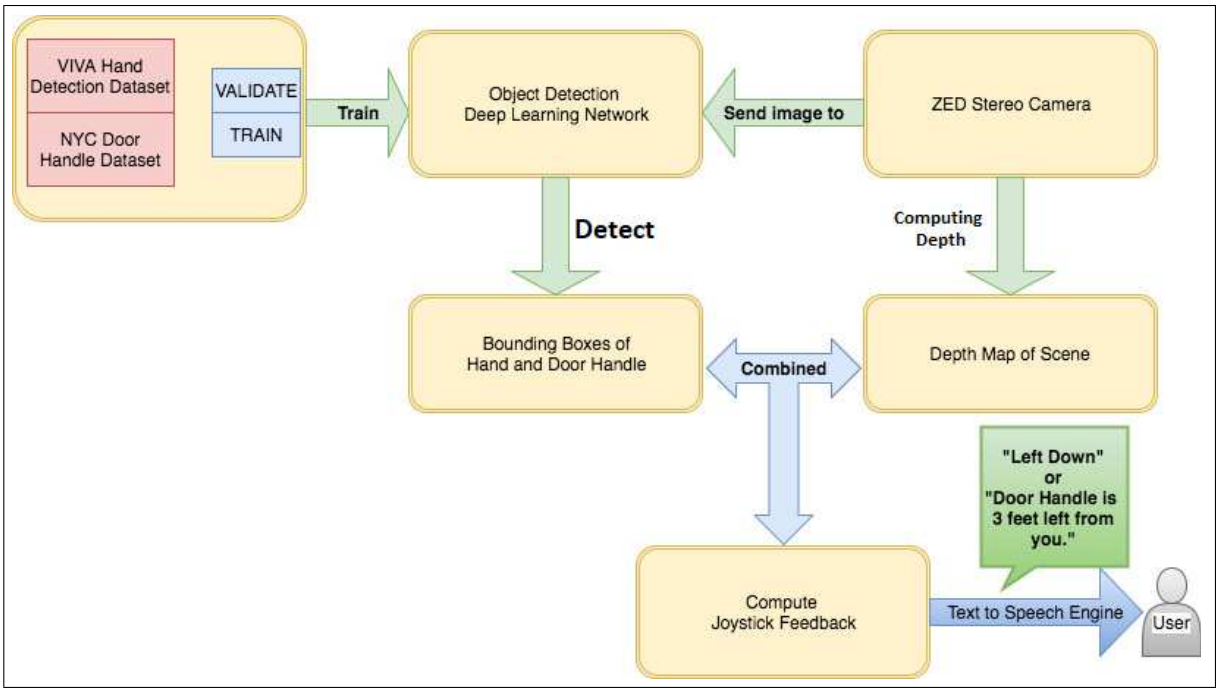


Figure 1: Pipeline of our proposed project.

real-time feedback. It can inform the user about the relative position of the door handle, help him or her locate it, guide the initial reach, and alert them in real-time if they are deviating from the desired direction with joy stick-like prompts. When the door handle is within reach, the output information would indicate the best route to reach towards the spatial target relative to a common starting point. As the individual moves their hands in an attempt to reach out, grasp, and manipulate the door handle, the system will provide synthetic sensory feedback relative to an idealized trajectory from start to target with the joy stick-like commands if the user is deviating from the ideal path.

2.1. Hand and Door Knob Detection

This section presents the deep neural networks to detect door/door handle and hand. The model is based on YOLOv2 [16] because of its relatively high detection speed and accuracy. Real-time object detection is one of the major components for the "joy stick-like command" system for assisting the visually impaired. Deep learning methods are state-of-the-art for such a task given that the architecture has proven to be effective in detection among different object domains. The architecture we choose can detect objects with high speed because it divides input images into grids and raises bounding-box proposals, which saves the GPU time by significantly decreasing the proposal volume. For every grid, the network will output B bounding boxes as proposals, and those bounding boxes that have a higher confidence than the threshold will be sent to NMS (Non Maximum Suppression) to generate the final output.

2.1.1 Architecture

The architecture of the model is a Convolutional Neural Network containing 22 convolutional layers and no fully connected layer. The network contains a layer of 32 convolutional filters with height and width of 3 and stride of 1. This layer is followed by Leaky ReLU, max pooling of pixels in a window of height and width of 2 and strides of 2, one layer of 64 convolutional filters with the same height, width, and stride as before, same Leaky ReLU, and the same max-pooling. The max pooling layer is followed by 3 layers of 128, 64, and 128 convolutional filters with stride of 1 and height and width of 3, 1 and 3, respectively. This layer is then followed by Leaky ReLU and another max pooling layer with size of 2 and stride of 2. The max pooling layer is followed by 3 layers of 256, 128, and 256 convolutional filters with the same height, width, and stride as before. They are all followed by Leaky ReLU and the same max pooling layer. Then, the max pooling layer is again followed by 5 convolutional layers of 512, 256, 512, 256, and 512 filters with size of 3, 1, 3, 1, and 3 and stride of 1. They are all followed by Leaky ReLU activation and a max-pooling layer like before. After that, there are 7 more convolutional layers of 1024, 512, 1024, 512, 1024, 1024,

Figure 2: Major components of the navigation system: 1. ZED™ camera for visual data acquisition, 2. Portable GPU carried in the back bag for data processing, and 3. Bluetooth headset for audio output.

and 1024 filters with filter size of 3, 1, 3, 1, 3, 3, and 3 and stride of 1. These layers are followed by the same Leaky ReLU but not the max-pooling layer. Finally, there are two convolutional layers for proposal outputs which can either have 1024, 35 (door knob and hand), or 30 (only door knob or only hand) filters with size of 3 and 1, and stride of 1. These two layers are followed by Leaky ReLU and a linear activation function. The number of filters in the last layer is based on the equation: $(\#classes + \#coords + 1) * (B)$. In this model, we let $B = 5$, and the coordinates are x, y, w, h, which represents the center coordinates of the object and the height and width of the bounding box. So if there is only one class, the filter number is 30; if there are both hand and door handle to detect, the filter number is 35. After the neural network outputs the tensors, proposals with high enough confidence will be sent to NMS to generate the final results. Also, the network uses anchoring to help with the accuracy. Our anchor boxes are generated from clustering VOC detection dataset by k-means of $k = 5$ [15].

2.1.2 Implementation

The idea behind this high processing speed is that the model applies a single neural network to the whole image and the whole network consists of only convolutional layers and no fully connected layer. It divides the image into $S \times S$ grid cells and chooses the box with argmax(IOU)(Intersection Over Union). Every grid gives out B bounding boxes, each containing 5 pieces of information: (x, y, w, h, prb) where x, y are coordinates of the box center, w and h represents the width and height of the box, and prb represents the probability that an object is inside this box $P(\text{Object})$. Prior detection systems usually repurpose classifiers and localizers by applying them at multiple locations and scales. Each region of the application would yield a score corresponding to the probability of target object and regions with high scores are considered to be detections. Every grid outputs $P(\text{Class}|\text{Object})$ for C classes. And since $P(\text{Class}) = P(\text{Class}|\text{Object}) \times P(\text{Object})$, threshold can be applied to filter recognition results. Then, classes and small bounding boxes from the grids will be integrated into real bounding boxes and labels. The network is trained on a GPU-based deep learning framework called Darknet [14]. Since it is implemented in C and CUDA, the speed of training or inference is relatively high, which meets the need of the real-time detection task.

2.2. Depth extraction and "joy stick-like command" control

Following image acquisition, two slightly different pictures are captured by the two lenses positioned horizontally relative to each other. By performing a matching process for every pixel in the left eye image and its counterpart in the right eye image, we would end up with an map or image where every pixel contains the disparity (distance) value. Then, the depth information can be extracted from this reconstructed image [8].

Assuming the target object appears in the camera images, "Joy stick-like command" utilizes a combination of the aforementioned detection results to give the user feedback of the spatial location of their hand relative to the targeted door handle by guiding them to reach out for and manipulate the handle. The general framework is presented in Algorithm 1 and elaborated below. Figure 3 illustrates how such "joy stick-like command" is performed. In essence, it contains three steps (given that both hand and door handle are detected):

- Step 1. While performing object detection, the model

will obtain the coordinates for both objects. Then, it will dynamically compare the depth information to determine whether the door handle is within reach (whether the door handle's "z coordinate output" is close to the hand's "z coordinate output", i.e. less than a preset threshold, say approximately 24-30 inches)

- Step 2. If the comparison yields "False", it will prompt the user to move towards the door handle by output the position information without the vertical difference (e.g. "door knob is 3 feet in front of you and 1.2 feet to your left"). Many door handles are relatively standardized in verticality as well.

- Step 3. If the comparison yields "True", it will help the user to reach towards the knob by only focusing on the x and y differences and ignore the depth information (i.e. only report horizontal and vertical distance between the hand and the door knob).

Data: Images fetched from camera
Result: Give navigation information to user
initialization of stereo camera, detection model, and TTS(Text to Speech Synthesizer) engine;
while *getting image from the camera* **do**
  send image to model, detect hand and door handle in the image;
  if *nothing is detected or only hand is detected* **then**
    continue;
  else if *only door handle is detected* **then**
    compute door handle coordinates using bounding box location;
    tell user where the door handle is in camera coordinate system;
  else both hand and door handle are detected
    compare the location/depth of hand and door handle;
    tell user how to move his or her hand towards the door handle;
  end
end

Algorithm 1: "Joy stick-like command" Feedback System

3. Experiments and Testing

3.1. Data collection and annotation

3.1.1 Door handles

Our dataset contains two major components. The first and most important one is the door handle dataset. Since no dataset is perfectly tailored to our goal, we decided to build our own dataset by collecting images organically instead of searching and downloading them from Amazon or Home Depot. We had team members taking photos of all kinds of door handles in different locations and scenes and at varying vantage points. The camera was always held in a position that approximates the position the stereo camera when mounted and integrated into the wearable system. Starting from the buildings and properties of New York University, the team gradually expanded their reach to the city streets for more images. The image library includes door handles, knobs, levers, latches, and many other types. In this paper we will be focusing on the knob category given its characteristic shape features and distinctive properties; it is also very standard and common around the world.

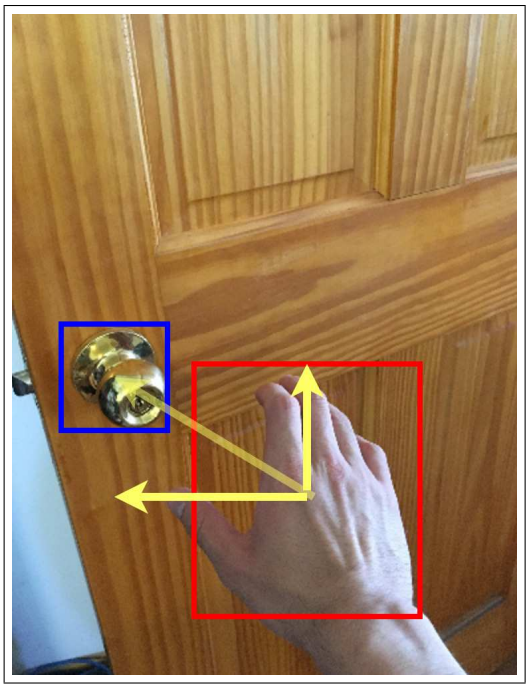


Figure 3: Demonstration of joy stick control.

Next, all photos were manually annotated with bounding boxes as the ground truth label. For better processing speed and consistency, all photos are also downsized to a standard resolution of 1200 by 900 and stored in JPEG format. This preprocessing standardizes the coordinates of the bounding boxes, where x and y now range from 0 to 1200 and 0 to 900, respectively.

After several weeks of work, our team had collected approximately 4000 photos for training and testing; among them, 1000 images contain door knobs. Some examples of the door knob pictures with bounding boxes are shown in Figure 4.

(a) (b)

Figure 4: Examples of door knob photo.

3.1.2 Hand images data

In order to perform "joy stick-like command" control and inform hand orientation, the user's hand also has to be detected in real-time. For this purpose, the popular benchmark dataset for the Vision for Intelligent Vehicles Applications (VIVA) Hand Detection Challenge (http://cvrr.ucsd.edu/vivachallenge/index.php/hands/hand-detection/) was used for training the hand classifier. This dataset consists of large scale hands data (from 54 videos collected in naturalistic driving settings) with 2D bounding boxes around, covering illumination variation, hand movements, and common occlusion. These photos were taken from 7 possible viewpoints, including first person view, which is the most common viewpoint in our model (similar to Figure 5) since the camera is mounted in front of our user's chest.

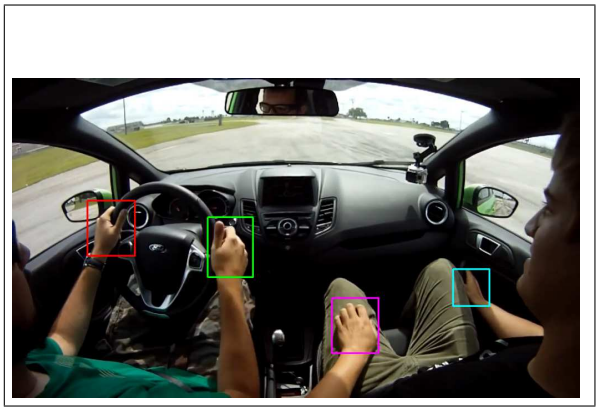


Figure 5: Example of the hand pictures with annotation. Figures collected from the VIVA challenge website (http://cvrr.ucsd.edu/vivachallenge/index.php/hands/hand-detection/)

3.2. Training classifiers

As mentioned before, there are 175 door knob pictures in total, 160 of them were used for training and the rest for testing. A neural network (called model 1 in this case) was created to only focus on the door knob. It was first pre-trained on ImageNet, then fine-tuned with the door knob dataset.

The object detection network will return bounding boxes with corresponding "confidence", i.e. the probability that it contains our target object, door knob. For example, if the model returns a box with a confidence value of 0.8, it is "80% sure" that there is a door knob inside this box. Hence, we can set up a proper threshold such as 0.6 to filter out all the uncertain cases and reduce false positives. Some success and failed/inaccurate cases of the Model 1 predictions are shown below. The model successfully detected almost all door knobs appearing in the pictures (14 out of 15 testing samples) with any reasonable confidence threshold between 0 and 0.6, although lower thresholds tend to let our model return more false alarms. Moreover, among those successful cases, the model accurately and decisively highlights the localization of door knobs with high confidence. When a confidence threshold of 0.6 is set, the model prediction acts as if it is the ground truth data, where the bounding boxes look almost like human annotations – no false positive or false negative among the 14 selected testing samples (Figure 6).

For experimentation purposes, different thresholds have been tested in order to observe its impact as well as to search for the optimal value. When the threshold drops to 0.2, only one picture starts with a false positive; when it drops even lower to 0.005, about one third (5 out of 15) of the tests resulted in failed/inaccurate predictions. Aside from some clean detections, there are some samples that give multiple bounding boxes with low threshold settings (0.005) and they are listed in Figure 7.

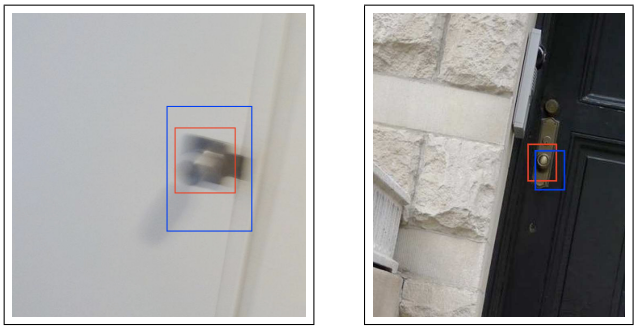
In Figure 7a, despite the fact that both the shape and features of such a blurry door knob are barely recognizable by human eye, our model still detected it with confidence. It was also accompanied by another larger blue box, presumably because the blurriness obscured the boundaries. Nonetheless, the general position (the centroid) is not significantly affected.

Secondly, in Figure 7b, the model gives two boxes, but this time the centroid coordinates are distinctively different. Although it correctly covers all parts of the knob, the blue box is severely skewed and it includes too much marginal space. The cause behind this skewness is somewhat unclear, but the dark color can be a possible factor.

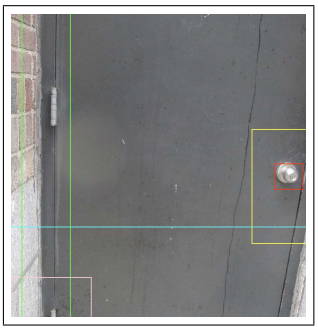
Next, in the last example (Figure 7c), the model shows a lack of confidence in its own prediction by labeling multiple objects/areas as the target object; the majority of them do not even make sense to human judgment. Such unrea-

(a) (b)

Figure 6: Successful cases.



(a) (b)



(c)

Figure 7: Inaccurate cases.

sonable prediction is expected to emerge only under the circumstances of extremely low confidence threshold such as 0.005; they are mostly likely to be suppressed in practice with a regular threshold level.

The second classifier (called Model 2), which targets human hands, was trained using the VIVA hand detection dataset mentioned above. 5000 hand pictures were randomly chosen from the dataset for training and the remaining 500, along with 50 hand photos that we manually collected, were used for testing. The training session used almost the same parameter settings in terms of learning rate and batch operation. Since the VIVA dataset extracts hand images from videos collected from a driving environment, our initial concern was generalizability. Our test results confirmed our suspicion: when we test Model 2 with the photos from the VIVA dataset, it performs well; however, if we apply it to the indoor photos that we have taken, the network often fails – it either can not detect the hand or gives multiple candidates, many of which are just false alarms.

3.3. Wearable Assistive System

As shown before in Figure 2, an end user wears a prototype of our wearable assistive technology for the visually impaired with door knob detection and real-time joy stick-like command for hand-to-handle manipulation, the detailed information about the major components are listed below.

- ZED™ Stereo Camera from Stereo Labs can capture high resolution images (up to 2200) with a detectable depth range of 0.3 to 20 meters and a field of view of 110-degrees. The detection accuracy can be as precise as 1 millimeter, or 0.1 degree in terms of visual angle. It provides a detailed API for image rendering and it will serve as the "eyes" of our assistive technology. In practice, the camera will be slightly lower than the user's eye height, but is close to the height of human hands and particularly good for wide and central view of the core body, including mid-body, in addition to high-body and low-body; this can be ideal for a wide range of navigation needs.

- Next, the system needs a "brain" to process all the visual signals for final output. We use the Jetson TX1 / TX2 from NVIDIA™ for the convenient portability and power/computation profile the hardware affords. To run this with a stereoscopic system such as the ZED™ camera, Darknet was first compiled with CUDA and OpenCV, then the model will display the predicted classes (in our case, either hand or door knob) as well as the bounding boxes drawn on top of it. In addition, we integrated the depth information into audio output for user feedback, so he/she can accomplish the door knob manipulation required for door opening in a more efficient manner.

- As a control system, this wearable device can provide constant voice feedback to the user by detecting relative position of the hand and door knob when "in view" on the stereo camera. By comparing the depth/distance, the system will tell users whether they need to keep moving, stretch their arm farther (along

the z-axis), or just maintain depth and modify 2D placement of hand on the door in a handle-centric direction (along the pertinent x- and y- axes). This is a task ideal for the Bluetooth bone conduction-based headset given its ability to maintain intact air conduction for hearing, low power needs [22], wireless connection, and ease of wearing. All detected/computed information will be extracted and structured into a meaningful and high-impact English phrase or sentence and then synthesized as a person's voice for audio output. A Text To Speech (TTS) Engine is deployed for this task. Presently, we have implemented a robotic voice as more natural voice generation requires more complex neural networks and more computation resources, which will inevitably affect energy cost and overall performance.

The whole system works in the following manner: First, when the user is approaching the threshold, the door knob will most likely be the only initial object detected by the Neural Network. One of the user's hands is likely to be on the handle of a white cane, ideally positioned at mid-line and perhaps just out of view of the camera and the other will be on the side of the body instead of the front (mid-line). The system will then tell the user the position of the door knob with depth by outputting a sentence like "Door knob is 5 feet ahead of you", letting the user know how far away he/she is. At this point, no information regarding the hand will be presented at all. Once the user is near enough (e.g. depth of approximately 2.5-3 feet, a typical range for canes), the system will remind the user, permitting time to pause or switch hands for the cane to free up the dominant hand for reaching to the door knob. It is at this point the system will issue an initial spatial target command to be used as a guide for the reach in hand-to-handle manipulation, monitoring both door knob and human hand. Now it will return their respective positions and the system can automatically compute the spatial difference in all three dimensions, based on the centroid of the corresponding bounding boxes along an idealized trajectory from start to finish, updated over the course of the movement and ultimately to goal. See Figure 3, for an example, the coordinates of the hand (red box) is x_1, y_1, z_1, while the coordinates of the door knob is x_2, y_2, z_2, which can all be obtained with ZED™ camera. Then the program with TTS integrated will first compute three differences $x = |x_1 - x_2|, y = |y_1 - y_2|, z = |z_1 - z_2|$, where x, y, z denote horizontal, vertical, and depth distance, respectively. Next, TTS program will output a sentence like "Please move your hand x feet to your left/right, y feet up/down." via the Bluetooth headset to our users and help them with the orientation (of course x, y will be replaced with real values). The z value here usually serves as a threshold to control the structure of the sentence: if z is relatively large, the main objective is to grossly approach the door handle; if it becomes small, then the system will focus on the "joy stick-like command" part, meaning it will mainly output x and y. This final process is repeated every 2 seconds to provide real-time instruction, so even if the user missed the door knob on the first try, our system will help to adjust with feedback towards more refined strategies on subsequent attempts.

3.4. Future Work

Presently, the main issue is the stability of hand detection performance. Sometimes when it successfully detects both the door knob and human hand, the system works as intended; however, other times when hand is ignored by the camera, we will have an unfortunate domino effect: no hand position will be obtained, no spacial difference can be computed, no x, y, z values for audio feedbacks, and consequently, no navigation. We will further fine-tune the network parameters and improve its performance on the hand dataset to resolve such issues.

In addition, we plan to add more photos to our database and extend the door knob detection function to more general door handle types, such as door latches, levers, pull-handles, etc. so that this system can be used in as many different environments as possible, enabling our users to walk more efficiently and safely through thresholds both indoors and outdoors.

4. Summary

In this paper, we have presented a wearable assistive technology for the visually impaired that can detect door knobs and the human hand in real-time, yielding pertinent spatial information with audio-based guidance and feedback with regards to "joy stick-like control" for hand-to-handle manipulation. We believe such system will enable users to navigate more safely and efficiently as threshold and hallway ambiguity is clarified and knobs and handles are more easily manipulated. We have also introduced a door handle dataset that can be used for model training in the future. While our results reveal that there are instances where the performance appears to be modest, our preliminary performance is promising and we will continue to improve the model and extend its functionalities for generalized door/door handle detection, universal hand detection, and extended wearable assistive technology functionality.

References

[1] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. *arXiv preprint arXiv:1607.07155*, 2016.

[2] T. Y.-H. Chen, L. S. Ravindranath, S. Deng, P. V. Bahl, and H. Balakrishnan. Glimpse: Continuous, real-time object recognition on mobile devices. In *13th ACM Conference*

on Embedded Networked Sensor Systems (SenSys), Seoul, South Korea, November 2015.

[3] A. Ciobanu, A. Morar, F. Moldoveanu, L. Petrescu, O. Ferche, and A. Moldoveanu. Real-time indoor staircase detection on mobile devices. In *Control Systems and Computer Science (CSCS), 2017 21st International Conference on*, pages 287–293. IEEE, 2017.

[4] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *arXiv preprint arXiv:1406.4729*, 2014.

[7] J. Hensler, M. Blaich, and O. Bittel. Real-time door detection based on adaboost learning algorithm. In *International Conference on Research and Education in Robotics*, pages 61–73. Springer, 2009.

[8] R. Jain, R. Kasturi, and B. G. Schunck. *Machine vision*, volume 5. McGraw-Hill New York, 1995.

[9] E. Jauregi, J. Martinez-Otzeta, B. Sierra, and E. Lazkano. Door handle identification: a three-stage approach. *IFAC Proceedings Volumes*, 40(15):517–522, 2007.

[10] E. Jauregi, B. Sierra, and E. Lazkano. *Approaches to door identification for robot navigation*. INTECH Open Access Publisher, 2010.

[11] S. Mattoccia et al. 3d glasses as mobility aid for visually impaired people. In *European Conference on Computer Vision*, pages 539–554. Springer, 2014.

[12] F. L. Milotta, D. Allegra, F. Stanco, and G. M. Farinella. An electronic travel aid to assist blind and visually impaired people to avoid obstacles. In *International Conference on Computer Analysis of Images and Patterns*, pages 604–615. Springer, 2015.

[13] A. A. Panchal, S. Varde, and M. Panse. Character detection and recognition system for visually impaired people. In *Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE International Conference on*, pages 1492–1496. IEEE, 2016.

[14] J. Redmon. Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`, 2013–2016.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[16] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[18] J.-R. Rizzo, Y. Pan, T. Hudson, E. K. Wong, and Y. Fang. Sensor fusion for ecologically valid obstacle identification: Building a comprehensive assistive technology platform for the visually impaired. In *Modeling, Simulation, and Applied Optimization (ICMSAO), 2017 7th International Conference on*, pages 1–5. IEEE, 2017.

[19] R. B. Rusu, W. Meeussen, S. Chitta, and M. Beetz. Laser-based perception for door and handle identification. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–8. IEEE, 2009.

[20] H. Shen, O. Edwards, J. Miele, and J. M. Coughlan. Camio: A 3d computer vision system enabling audio/haptic interaction with physical objects by blind users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, page 41. ACM, 2013.

[21] R. A. Shoureshi, J.-R. Rizzo, and T. E. Hudson. Smart wearable systems for enhanced monitoring and mobility. *Advances in Science & Technology*, 100, 2017.

[22] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen. How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pages 232–237. IEEE, 2012.

[23] S. Wang, H. Pan, C. Zhang, and Y. Tian. Rgb-d image-based detection of stairs, pedestrian crosswalks and traffic signs. *Journal of Visual Communication and Image Representation*, 25(2):263–272, 2014.

[24] X. Yang and Y. Tian. Robust door detection in unfamiliar environments by combining edge and corner features. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 57–64. IEEE, 2010.