Assistive System for Navigating Complex Realistic Simulated World Using Reinforcement Learning

Faruk Ahmed
Department of Electrical and Computer Engineering
The University of Memphis
Memphis, Tennessee 38152
Email: mfahmed@memphis.edu

Md Sultan Mahmud
Department of Electrical and Computer Engineering
The University of Memphis
Memphis, Tennessee 38152
Email: mmahmud@memphis.edu

Mohammed Yeasin
Department of Electrical and Computer Engineering
The University of Memphis
Memphis, Tennessee 38152
Email: myeasin@memphis.edu

***Abstract*—Finding a free path without obstacles or situation that pose minimal risk is critical for safe navigation. People who are sighted and people who are blind or visually impaired require navigation safety while walking on a sidewalk. In this paper we develop assistive navigation on a sidewalk by integrating sensory inputs using reinforcement learning. We train the reinforcement model in a simulated robotic environment which is used to avoid sidewalk obstacles. A conversational agent is built by training with real conversation data. The reinforcement learning model along with a conversational agent improved the obstacle avoidance experience about 2.5% from the base case which is 78.75%.**

I. Introduction

The main goal of this research is building navigational assistance on sidewalks for people who are blind or visually impaired. Avoiding obstacles or finding a free path with minimal risk of collision is an essential part of a safe navigation. Most of the reported literature assume static and intransient natural obstacles to build simplified navigational assistance. Such an approach is not suitable for a complex dynamic world full of uncertainties in the form of the transient obstacles, (e.g., puddle, scooter, and pothole) and activities (i.e., people riding bike, walking with pet, meeting and greeting etc.).

Image-based obstacle avoidance has been very popular in the past [1], [2]. This solution depends the visible light. However, for the visually impaired, it is difficult to capture a machine interpretable image. Besides, the diurnal cycle poses an additional layer of difficulties in classifying or annotating the image [3]. Even perfect "image annotation or classification" based systems are not suitable in rendering meaningful feedback to navigate a complex and dynamic world. More than often such a system induces higher cognitive load or ambiguities in the mind of users. To avoid such a problem, we present a novel approach to solve the navigation in a complex and dynamic world full of uncertainties.

We approach the problem from a point of view that is based on the concept of "free path", which poses minimal or no threat (i.e., risk of collision). Instead of modeling only the obstacles, we integrate sensory inputs in a reinforcement learning (RL) to develop an assistive solution to safely navigate on a sidewalk. We use a simulated environment along with a conversational agent to demonstrate the utility of a safer navigational system.

To implement our idea we use the concept of point cloud (PC). PCs are a set of data points in space [4] and usually constructed using range sensors (e.g., Intel RealSense, Microsoft Kinect). PCs contain both RGB and depth information (RGB-D). In addition, PCs are robust against variation in diurnal cycle and lighting conditions. There are a number of range sensors (i.e., https://rosindustrial.org/3d-camera-survey) available for building PC effectively. Some of those are bulky, less energy efficient and some are smaller as well as energy efficient.

II. Related works

Assistive technology solutions for the visually impaired drew the attention of researchers as a prominent research area in the mid-90s. Researchers have conducted studies and developed applications to improve the mobility of the visually impaired. Generally, two types of applications are available for visually impaired, *a) sensor based* and *b) computer vision based*.

A. Sensor based techniques

Most of the sensor based applications directly or indirectly use sensor data.

1) Directly using sensor data: Drishti [5] and GuideCane [6] used GIS information hosted on a central server. They continuously queried the server for GPS information to facilitate navigation. GuideCane used an ultrasonic sensor and embedded computer to detect obstacles, but the field of view of the sensor was very narrow. To circumvent the problem, Shoval, Ulrich, and Borenstein (2003) proposed an array of ultrasonic sensors mounted on a belt [7]. However, the belt became too bulky, along with being power and resource hungry. GuideCane therefore, along with other smart cane projects, focused on obstacles that are of head-height to make it lighter [8]–[10]. There is a talking navigation cane that allows voice command and provides navigation information via audible messages and haptic feedback [11]. They used the GPS to accomplish the localization of the user. A beacon-based navigation system is more accurate and provides far

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

better navigation help. But the deployment of several beacons is an expensive task and needs an expert for the implementation [12]. Recently WiFi-based positioning has drawn attention [13], [14]. This type of positioning and navigation system determines the approximate position of cellular devices by using radio frequency (RF) signals and a triangulation mechanism. It depends on the signal strength and phase, signal transmission time and angle of arrival along with channel state information. Indoor environments are complicated because of multiple access point transmission. Signals are affected by the adjacent and co-channel interference [15]. That is why this method is less reliable both indoors and outdoors. One system using this approach is ppNav, a mobile app which helps navigate based on a previous navigator's trace. It constructs trace from ubiquitous WiFi signal along with visual features [16]. Chen reported a mobile robot navigation algorithm which fuses the odometry and compass data. They used an extended Kalman filter algorithm for the fusion [17].

2) Modeling obstacle from sensor data: Probabilistic inertial-visual odometry (PIVO) was developed for an occlusion-robust navigation system [18]. In this work, the Inertial Measurement Unit (IMU) sensors and the monocular camera information are fused to construct odometry. The application is robust even if the camera is covered for an extended amount of time. However, this is not usable by the visually impaired people because the camera and the IMU sensors have to be at a specific orientation. [19] constructed the model of left turn, right turn, and stairs from IMU data. This model they have used to perform wayfinding.

B. Computer vision techniques

There are applications to utilize the computer vision techniques along with sonar, IMU, and LiDAR sensor data. Some applications capture video and use traditional computer vision techniques to identify obstacles. Others uses deep neural network to model the obstacles.

1) Direct use of computer vision: In [20] Michels et al., they used ground truth laser distance data to train depth estimator. The output of the estimator becomes input for a controller trained by reinforcement learning.

2) Modeling obstacle from computer vision: Ahmed et al. constructed a sidewalk image dataset and trained different deep neural networks to identify the obstacles [1]. Once an obstacle is identified, the audible label of that obstacle is played to the visually impaired as feedback. This approach suffers from inefficient feedback to avoid the obstacle. Moreover, it works only with daylight with a drawback of the diurnal affect.

To provide efficient feedback for obstacle avoidance, Ahmed et al. applied image captioning [2]. The generated caption of the image is played as feedback. In this approach, the caption does not always contain sufficient information so that it can be avoided. In addition, the diurnal cycle affects the image as well as the caption.

III. Reinforcement learning for modeling free-path

Researchers are spurred to improve the mobility of the visually impaired by devising an obstacle avoidance mechanism. There are vision-based solutions to model the obstacles [21], [22]. The obstacles are modeled using a traditional computer vision algorithm or a modern deep neural network (DNN) [23]. Both traditional and DNN algorithms have a limited capacity of modeling dynamic nature and a huge number of obstacles. Dynamic nature refers to stationary and moving obstacles along with their sizes, shapes, motion speed, and colors. The dynamic number refers to the unknown number of obstacles. Any object that blocks the mobility of the people is an obstacle. The convolutional neural network (CNN) is one of the popular object recognition models and is heavily used; [1], [3] it is inspired by ImageNet [24] Large Scale Visual Recognition Challenge (ILSVRC).

There is research to combine camera and Inertial Measurement Unit (IMU) sensors [1] with improving the model of obstacles. In this approach, the system becomes too complex. Simple sensor-based algorithms are prevalent nowadays to reduce complexity. Yang, Wang, Lin, Bai, Bergasa, and Arroyo (2018) proposed pairs of sensors for this purpose [25]. RealSense R200 and IMU are mounted on smart glass at eye-level and RealSense RS410 at waist level. This system is efficient to detect low-lying obstacles. Wang, Yang, Hu, and Wang described stixel representations of a 3D world combined with pixel-wise semantic segmentation for navigation aid [26], [27].

The technologies mentioned in related work in the above para graphs are limited to a certain class of obstacles. For example, the CNN based models are capable of recognizing only the classes of obstacles that belong to the training set. Moreover, the model has to see the obstacle beforehand. To overcome the shortcomings and to simplify the navigation on a sidewalk, the novel idea in this research is "free-path." The idea of free-path is to find a safe area on a sidewalk instead of trying to model the dynamic environment of obstacles. We utilized a RealSense D435 depth camera as well as the custom LiDAR to collect PC of the sidewalk. The PC is then used to model the free-path using reinforcement learning.

Another aspect is that the position of the dynamic obstacles has to be communicated to the visually impaired. This group would take necessary action based on that. In a situation with a dynamic obstacle, the outcome of actions performed by visually impaired people is delayed. For example, to avoid a bike rider, the visually impaired may stop and stand to the side or keep walking in a certain direction. They do not know if the bike rider is avoided until they pass. In this case, her beginning actions (e.g., stopping, standing aside, or walking) are delay rewarded. To model this behavior, the RL is a perfect fit for both static and dynamic obstacles. The reason is explained

[1]This is an electronic device that measures and reports orientation, velocity, and gravitational forces through the use of accelerometers and gyroscopes and often magnetometers

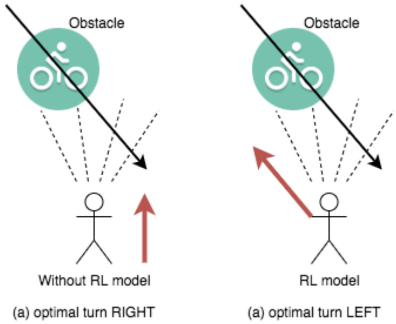


Fig. 1. Optimal turn decided by RL model.

with an example. Let us say a biker is approaching a user in a crossing pattern from left to right as in figure 1 (a). A model without RL will see an immediate empty space in front and will decide that as a free path, whereas the biker will reach that space after some time. On the other hand, the model with RL takes the movement of the biker into account and decides to move left instead of going forward as seen in figure 1 (b). In this research, we choose RL to teach the robot the dynamic and static nature of the obstacles.

IV. Definition of free-path

Generally, the sidewalk consists of static and dynamic obstacles. The dynamic obstacles have motion. The visually impaired person walking on the sidewalk has motion as well. The mean comfortable walking speed of adults between 20 and 70 years of age ranges approximately from 100 cm/s to 150 cm/s [28].

Suppose $\chi = (M, d)$ is a discrete metric space from euclidean space $\mathbb{R}^n$, where $M \subset \mathbb{R}^n$ is the set of points and d is the distance metric. The density of M in the ambient euclidean space may not be uniform due to perspective distortion. There exists a set of functions f that takes χ as input and produces clusters satisfying a set of constraints (e.g., points at a given neighborhood distance or color) [29]. In this research $n = 3$, meaning the spaced is three dimensional.

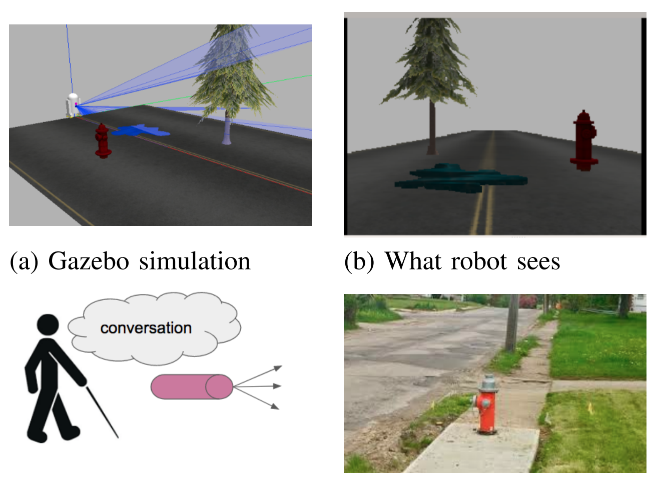
In the given χ the *free path* is defined as $f(\chi) = \phi$ which indicates there is no obstacle along the direction of interest. Let us assume that $f(\chi) = C$, where C is a set of clusters in χ. The *threat level* t is inversely proportional to the distance of the cluster c_i ($C \in \{c_1, c_2, ..., c_i\}$), that is $t \propto \frac{1}{d_i}$ [30], [31].

V. Building RL model

In order to build an RL model, there has to be an agent and environment. The agent placed in this environment can learn from the interaction with the environment. Building a real environment, especially a sidewalk, to train an agent is expensive. Moreover, there must exist a practical way of implementing the punishment mechanism every time the agent makes mistakes. To understand the complexity of a real sidewalk and to study the feasibility of the system, the simulated environment is extremely suitable. It is easy to program and modify, and various types of agents can be placed in the environment. In addition to that, implementing different algorithms as well as training and testing models is much easier than the real environment. Based on this reasoning, we selected the simulation to train the RL model and the real environment to test it.

The RL model is trained in the Gazebo [32] simulation environment. We placed a robot on a virtual sidewalk, where there are obstacles (e.g., curb and grass beside the sidewalk, pothole, cone, fire hydrant, electric scooter, electric pole, dumpster, and tree). The RL algorithm stays in Robot OS (ROS). In this setup, we let the robot walk on the sidewalk with $10,000$ episodes and $1,000$ steps in each episode. The physics engine of the Gazebo environment makes it easy to detect collisions, falls, displacements, and other physical measurements. It also provides a way to set the base speed of the robot. We set the base speed equal to the mean walking speed of men. Whenever the robot collides with an obstacle or falls down by going off of the sidewalk, it gets penalized by one, and the simulation resets and the robot starts from the initial position. There are rewards of $+1$ for actions which do not cause collisions or falls. We present a depiction of a metaphor between the simulated sidewalk and the real sidewalk in figure 2. Once the RL model is built, then it was transferred to the device for the testing and evaluation.

We named the combination of free-path finder and conversational agent as "Augmented Guide (AG)" because it helps people to find free-path through the RL model and it keeps people informed about the ambiance through the conversational agent.



(a) Gazebo simulation (b) What robot sees

(c) Real sidewalk (d) What AG sees

Fig. 2. Analogy with AG and Gazebo simulation.

The following aspects solve the free-path problem using RL

- Different actions yield different rewards. For example, when trying to avoid an obstacle in a sidewalk, going left may lead to an avoidance, whereas going right may cause collision.
- Reward for an action is conditional on the state of the

environment. In figure 1, going left may be ideal at a certain position in the path, but not at others.
- Rewards are delayed over time. This just means that even if going left (Fig 1) is the right thing to do, we may not know it until the obstacle is completely out of sight.

We have defined the environment, state, action, and reward in terms of sidewalk in the following manner.

Environment: The sidewalk environment consists of static and dynamic obstacles. The static obstacle does not move whereas the dynamic obstacle moves. The sidewalk has a curb and it has brick pavement. There is grass beside the sidewalk which is different in color than the sidewalk itself.

State: The state space is a set of all possible relative positions of agents and the obstacles on the sidewalk. That is why the number of states is infinite. The agent finds useful information from the states to make the right action.

Action: There are five actions namely stop, left, forward, right, and backward. The agent encounters infinite number of states and takes one of these actions in the action space set.

Reward: If an action performed by the agent causes collision, then the reward is -1. The agent keeps getting $+1$ as a reward until there is no collision.

With the above environment, the Gazebo simulation is created. We implemented three algorithms, Q-learning, SARSA, and the Deep Q-learning network (DQN). The optimal parameters found for those algorithm are listed in table I.

TABLE I
PARAMETERS FOR THE LEARNING ALGORITHMS

Parameters	Q-learning	SARSA	DQN
learning rate	0.5	0.5	0.001
discount factor	0.9	0.9	0.95
exploration probability	0.1	0.1	0.1
exploration decay	0.99	0.99	0.99

VI. ACTIVE INTERFACE: CONVERSATIONAL AGENT

In daily life, any matter not apparent to the user becomes more transparent through conversation. That is why the teachers request that students ask questions, and the managers ask the employee to ask questions. Through conversation, the real scenario becomes evident.

In this research, we are adopting this concept. The user communicates with the agent, and the agent talks about what it sees ahead. Through the conversation, the ambiance becomes more apparent to the user. The agent mentions any obstacle on the walkway to the user. How to avoid that obstacle depends on the user. The AG device will not command to do a particular action. Instead, the user decides the next action based on the conversation. This conversational agent is an active interface.

For the basic understanding of the conversational agent, we introduce few keywords from the literature.

Intent: The intent is the end meaning of what the user is trying to say. For example, if the user says, “Find the fire hydrant” the intent can be classified as to find the obstacle.

Entity: An entity is to extract useful information from the user input. From the example above, “Find the fire hydrant” the entities extracted should be the *name* of the obstacle. The name, for example, is a fire hydrant.

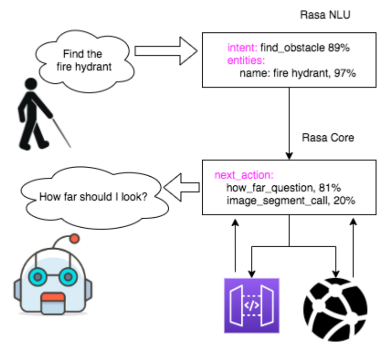


Fig. 3. Block diagram of Rasa NLU and Rasa Core.

Stories: Stories define the sample interaction between the user and the conversational agent connecting intent and action performed by the agent. In the example above, the agent got the intent of finding the obstacle and entities like the name of the obstacle, but still, there is an entity missing, how far should it look? That would create the next action from the agent.

Actions: Actions are the operations performed by the agent. It could be either asking for some more details to get all the entities, integrating with some APIs, or querying the RL model to get any information.

Templates: The templates are the sample replies from the agent which can be used as actions.

The conversational agent, a software system, enables a user to talk with it in natural language. RASA, an open-source machine learning framework, serves as the engine of the conversational agent. It is easy to customize. We can build, deploy, or host RASA internally in our server or environment with complete control. Confidential conversation data cannot be shared with a third party. The majority of the conversational agent’s tools available are cloud-based and provide software as a service. We cannot run them internally in our environment, and we need to send data to the third party. With RASA, there is no such issue.

The RASA is comprised of two main components *Rasa NLU* and *Rasa Core*. Rasa NLU is a library for natural language understanding (NLU), which does the classification of intent, extracts the entity from the user input and helps the agent to understand what the user is saying. Rasa Core, on the other hand, is a conversational agent framework with machine learning-based dialogue management capabilities. It takes the structured input from the NLU and predicts the next possible best action using a probabilistic model like long short-term memory (LSTM) recurrent neural network. Rasa NLU and Rasa Core are independent, and we can use NLU without Core, and vice versa. But using both NLU and Core enhances performance. A block diagram of RASA is shown in figure 3.

Three types of files are necessary to train Rasa NLU, NLU training file, Stories file, and Domain file. The training file contains some training data with user inputs along with the mapping of intents and entities present in each of them. The more varying examples we provide, the better the agent's NLU capabilities become. The Stories file contains sample future interactions between the user and the agent. Rasa Core creates a probable model of interaction from each story. The Domain file lists all the intents, entities, actions, templates, and more information. The conversational data obtained from the WoZ experiment is converted to text and processed to create the above-mentioned training files. The training files are stored in the markdown format. The samples form an NLU file which is presented in listing 1.

```
## intent:greet
- hey
- hello
- are you there?
- are you ready?
- ready?

## intent:greet_ask
- Yes ready, are you ready?
- Ready, want to start?.
- I am here, start walking?

## intent:greet_normal
- yes
- yap
- let's go

## intent:find_obstacle
- Find [obstacle](obstacle)?
- What is [there](obstacle)?
- What is [that] (obstacle)?
- Do you see [anything](obstacle)?
- [There] (obstacle)?
- [Here] (obstacle)?
- This [way](obstacle)?
- That [way](obstacle)?

## intent:find_distance
- [Where](distance)?
- How [far](distance)?
- How long to [reach](distance)?
- Is it [close](distance)?
- Is it very [close](distance)?

## intent:bye
- bye, let me know
- bye now
- i am here, bye
```

Listing 1. Samples from an NLU file.

Once the RL model is trained, AG integrates that model. Through this RL model the device sees obstacles and recommends an action. The AG does not dictate the turn or move; it gives the ambient information about the obstacle, and the person decides which direction to move.

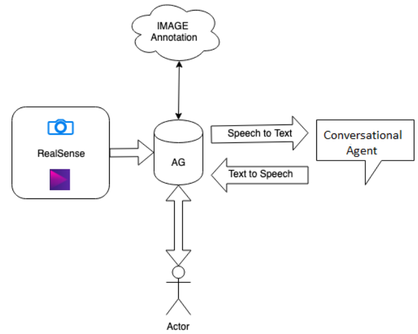


Fig. 4. Block diagram of AG.

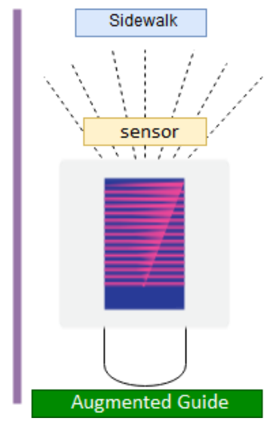


Fig. 5. Working principal of augmented guide for visually impaired.

VII. Augmented Guide (AG)

AG contains two essential modules free-path finder and conversational agent. The free-path finder module uses RL to find the obstacle-free path, and the CA helps the visually impaired be informed about the ambiance through active conversation.

There are camera and LiDAR sensors connected to the computing engine (e.g., Raspberry Pi, Jetson Nano) in the AG. The upgraded version of AG uses only the RealSense depth camera, which provides both RGB and LiDAR data, to make it lighter and smaller in size. The block diagram of AG is shown in figure 4 and the working principal is shown in figure 5.

VIII. Evaluation

In the Gazebo simulated training environment, the robot is equipped with the depth camera. From this depth camera, the robot can sense depth, color, and texture from the RGB sensor. Figure 6 shows sample pictures from the depth camera. The blobs in the pictures are laser beams which form PC. Picture (a) is a depth image taken during the daytime, (b) is an infrared image also taken during the daytime, and (c) is an infrared image taken at night. The PC is the input to RL both on the real sidewalk as well as in simulation. The base moving speed of the robot is set to the mean walking speed of men to make the simulation close to the real sidewalk. The lighting condition is set to ambient light, which gives an illusion of daylight. We were also able to set the wind speed of the ambient environment. There were ways to create a sidewalk with a slippery surface, with ice, snow, and slope. However, to avoid the extreme complexity of the implementation, we skipped these aspects within this research scope.

(a) (b) (c)

Fig. 6. Sample pictures obtained from depth camera.

In the training environment, we examined the learning of the three algorithms. The same sidewalk was used for training all of these. Q-learning, State-Action-Reward-State-Action (SARSA), and Deep Q-Network (DQN) belong to the list of training algorithms. The Q-learning algorithm is off-policy meaning it learns based on the action obtained from another policy e.g., greedy approach. Whereas the SARSA algorithm is on-policy that learn based on the action performed by the current policy instead of the greedy approach. Both Q-learning and SARSA are not generalized because these algorithms have to experience a state before learning. That is why these are not generalized and performs poorly in a huge number of states.

Within 200 episodes, the DQN learned best among the three, and SARSA learned better than the Q-learning. Figure 7 shows these findings. The derivative of the learning curve of the reward increased over the number of episodes. In other words, we can say that the more interaction the robot makes with the obstacles, the more it learns to avoid them. That is why the reward increases after a couple hundred episodes.

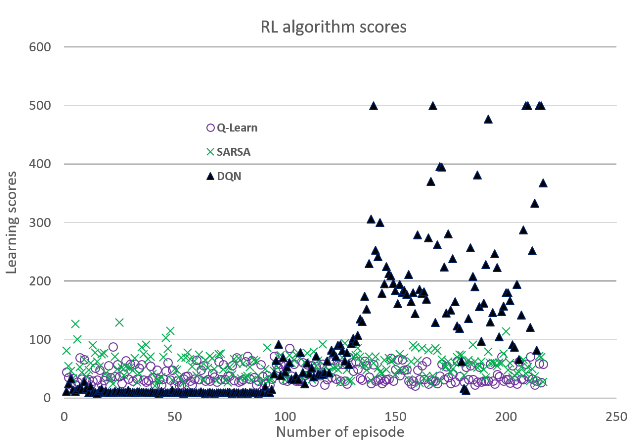


Fig. 7. Learning score comparison of Q-learning, SARSA, and DQN.

The testing environment of the RL model is the real sidewalk. A visually impaired person volunteered to test the prototype. The IRB approval of the blind-ambition umbrella project is used for this testing as it involves human subjects. Five hundred feet of the u-shape sidewalk was selected for the evaluation of the prototype. There were trees, an electric pole, a pothole, a dumpster, an iron fence, a visible curb, a bollard, and a fire hydrant on this sidewalk. We manually placed a couple of electric scooters, yellow construction cones, and water to form a puddle. The user was mostly happy about being aware of upcoming objects ahead of time. He could quickly point to any direction and ask “What is there?” and receive names of the segmented objects. The obstacles which stand above the ground were found easily, but the ground level obstacles such as the pothole and puddle was hardly found. On a narrow sidewalk, the RL got confused with the sidewalk fence (not the construction fence) as an obstacle. Though there are limitations, according to the volunteer, the overall performance of the assistive device was found to be satisfactory.

RASA [33] framework is the base engine for building a conversational agent. To train it, it requires conversational data, which we have obtained from the WOZ experiment on the sidewalk. We carefully annotated the spoken sentences of the visually impaired into proper intent, and we identified the entities and actions from those. Executing actions requires developing a service engine. An entity is passed as a parameter to the action. The RASA stack provides a light-weight SDK for this purpose. We used this SDK to develop the action end-point.

The input and output of the conversational agent is text. From an audio input device, the speech is converted to text and fed into the agent. The reply from the agent was again converted back to speech and sent to the audio output device. We have used the speech-to-text engine for the speech to text conversion, and it generated words with correct spelling. Because the RASA stack always receives words with correct spelling, we did not have to train it with incorrectly spelled words. For example, we avoided training the conversational agent with the variation of “hi”, “hey”, or “hai”.

The Bluetooth headset acts as an audio input and output interface. This device connects to the prototype of the assistive device and provides a partial scope of the private conversation. That is, people may hear what the visually impaired person is asking for, but they cannot hear what the device is replying.

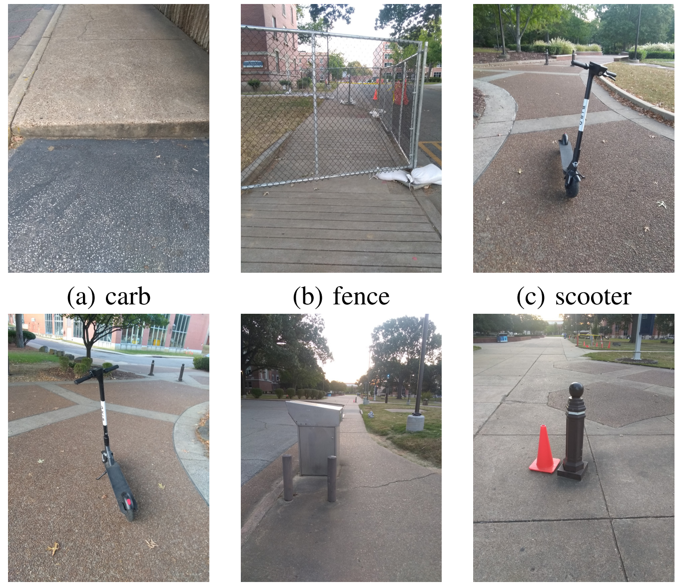
We show the basic block diagram (see Figure 4) of the prototype. The user has the option to ask the AG to take a picture and segment it. Amazon Rekognition does the segmentation of images in AG. The text-to-speech and speech-to-text service is used from Google. Of course, to use the Google and AWS services, there is a need for internet connectivity.

Table II contains the results obtained from a test simulation. In the testing phase, we let the robot walk from one side to the other side of the sidewalk $10,000$ times which is the number of episodes. The robot found the construction cone most of the time but failed to see the pothole. It is reasonable, because the pothole is on the ground whereas the construction cone, fire hydrant, stopper, and electric scooter stand above the ground. Among the above ground level obstacles, the AG is less able to detect the electric scooter than other obstacles. This less detection is due to the size and shape of the scooter.

For comparison, we selected the base case as 78.75% “in field obstacle detection accuracy” from [2]. We used the AG on the real sidewalk and found that the average accuracy measured is about 81.29%. The obstacle avoidance experience improved about 2.5%. It detected and talked about various obstacles. Few important obstacles are shown in Figure 8.

TABLE II
OBSTACLES AVOIDANCE RESULTS IN SIMULATION AND REAL WORLD

Obstacle	Image	% simulation	% sidewalk
Pothole		55	52
Construction Cone		91	92
Fire hydrant		92	93
Electric Scooter		74	71
Electric Pole		78	79
Dumpster		93	94
Tree		87	89
	average	81.428	81.285



(a) carb (b) fence (c) scooter

(d) scooter (e) dropbox (f) stopper

Fig. 8. Example obstacles detected by the assistive device

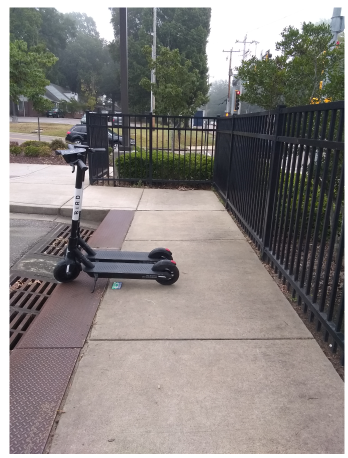


Fig. 9. AG got confused where scooter and sidewalk fence were together. It reported fence as obstacle instead of scooter.

The AG got confused with the obstacle in Figure 9. The real obstacle is the electric scooter but it was talking about the fence as an obstacle as well.

IX. Conclusion

In this research, we built an assistive technology prototype device. The purpose of this prototype is to augment the means of avoiding obstacles for the visually impaired. The obstacles include both static and dynamic nature, and the device is useful during a walk on the sidewalk.

We developed the free path approach instead of modeling the various obstacles. Reinforcement learning served as an essential tool for free path modeling. Also, to communicate the free path to the user, we incorporated the conversational agent trained on the RASA stack.

For modeling the free path, we created the simulated sidewalk and the 3D models of obstacles in Gazebo. We placed a robot in the environment, which learned to avoid obstacles through RL. When the RL was stable, we incorporated it into the AG.

The conversational agent is trained with the Wizard of OZ conversation data. This conversation is the starting point of the agent learning to talk. The user asks the agent about the ambient environment. The agent then talks back to the user with the necessary information. RASA collects that information from AWS, API, and the RL model. From this information about the ambient environment, the user decides to take necessary actions.

We observed some limitations of the assistive prototype system during the training and testing. One of those is that the Gazebo obstacles are a purely mathematical model. It means that the physics engine sees a tree as a box even though the tree has a particular shape. During testing, we found that this limitation did not matter much because the input to the RL was PC. The conversation tool sometimes takes a long time to respond. It could be dangerous in a situation where time is crucial, e.g., an oncoming car while crossing the road. The use of the WiFi network is another limitation. It could be solved by keeping the models and services all in the computing device, but that requires higher computing, storage, and battery capacity. As a trade-off, the WiFi is used. Also, the most critical obstacle, according to Ahmed et al., is the "slope" [1]. Our assistive device can not detect slope.

ACKNOWLEDGMENTS

This work was supported by Department of Electrical and Computer Engineering and Institute for Intelligent Systems (IIS) at the University of Memphis.

REFERENCES

[1] F. Ahmed and M. Yeasin, "Optimization and evaluation of deep architectures for ambient awareness on a sidewalk," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 2692–2697.

[2] F. Ahmed, M. S. Mahmud, R. Al-Fahad, S. Alam, and M. Yeasin, "Image captioning for ambient awareness on a sidewalk," in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*. IEEE, 2018, pp. 85–91.

[3] F. Ahmed, M. S. Mahmud, and M. Yeasin, "An interactive device for ambient awareness on sidewalk for visually impaired," in *2018 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2018, pp. 1–6.

[4] S. J. Lee, Y. S. Moon, N. Y. Ko, H.-T. Choi, and J.-M. Lee, "A method for object detection using point cloud measurement in the sea environment," in *2017 IEEE Underwater Technology (UT)*. IEEE, 2017, pp. 1–4.

[5] A. Helal, S. E. Moore, and B. Ramachandran, "Drishti: An integrated navigation system for visually impaired and disabled," in *Proceedings fifth international symposium on wearable computers*. IEEE, 2001, pp. 149–156.

[6] I. Ulrich and J. Borenstein, "The guidecane-applying mobile robot technologies to assist the visually impaired," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 31, no. 2, pp. 131–136, 2001.

[7] S. Shoval, I. Ulrich, and J. Borenstein, "Navbelt and the guide-cane [obstacle-avoidance systems for the blind and visually impaired]," *IEEE robotics & automation magazine*, vol. 10, no. 1, pp. 9–20, 2003.

[8] W. Wu, L. Au, B. Jordan, T. Stathopoulos, M. Batalin, W. Kaiser, A. Vahdatpour, M. Sarrafzadeh, M. Fang, and J. Chodosh, "The smartcane system: an assistive device for geriatrics," in *Proceedings of the ICST 3rd international conference on Body area networks*. ICST (Institute for Computer Sciences, Social-Informatics and …, 2008, p. 2.

[9] V. Singh, R. Paul, D. Mehra, A. Gupta, V. D. Sharma, S. Jain, C. Agarwal, A. Garg, S. S. Gujral, M. Balakrishnan *et al.*, "'smart'cane for the visually impaired: Design and controlled field testing of an affordable obstacle detection system," in *TRANSED 2010: 12th International Conference on Mobility and Transport for Elderly and Disabled PersonsHong Kong Society for RehabilitationS K Yee Medical FoundationTransportation Research Board*, 2010.

[10] M. H. A. Wahab, A. A. Talib, H. A. Kadir, A. Johari, A. Noraziah, R. M. Sidek, and A. A. Mutalib, "Smart cane: Assistive cane for visually-impaired people," *arXiv preprint arXiv:1110.5156*, 2011.

[11] R. S. Jesie, "Advanced talking navigation cane for visually impaired using capacitive touch keypad," in *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*. IEEE, 2015, pp. 1–5.

[12] D. Ahmetovic, M. Murata, C. Gleason, E. Brady, H. Takagi, K. Kitani, and C. Asakawa, "Achieving practical and accurate indoor navigation for people with visual impairments," in *Proceedings of the 14th Web for All Conference on The Future of Accessible Work*, ser. W4A '17. New York, NY, USA: ACM, 2017, pp. 31:1–31:10. [Online]. Available: http://doi.acm.org/10.1145/3058555.3058560

[13] C. Yang and H.-R. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.

[14] H.-H. Liu and Y.-N. Yang, "Wifi-based indoor positioning for multi-floor environment," in *TENCON 2011-2011 IEEE Region 10 Conference*. IEEE, 2011, pp. 597–601.

[15] M. S. Mahmud and M. F. Uddin, "Mitigating unfairness problem in wlans caused by asymmetric co-channel interference," *International Journal of Mobile Communications*, vol. 16, no. 3, pp. 307–327, 2018.

[16] Z. Yin, C. Wu, Z. Yang, and Y. Liu, "Peer-to-peer indoor navigation using smartphones," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1141–1153, 2017.

[17] W. Chen and T. Zhang, "An indoor mobile robot navigation technique using odometry and electronic compass," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, p. 1729881417711643, 2017. [Online]. Available: https://doi.org/10.1177/1729881417711643

[18] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, "Pivo: Probabilistic inertial-visual odometry for occlusion-robust navigation," *arXiv preprint arXiv:1708.00894*, 2017.

[19] F. Ahmed, M. S. Mahmud, and M. Yeasin, "Rnn and cnn for way-finding and obstacle avoidance for visually impaired," in *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*. IEEE, 2019, pp. 225–228.

[20] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 593–600.

[21] H. D. Escobar-Alvarez, N. Johnson, T. Hebble, K. Klingebiel, S. A. Quintero, J. Regenstein, and N. A. Browning, "R-advance: Rapid adaptive prediction for vision-based autonomous navigation, control, and evasion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 91–100, 2018.

[22] A. J. Barry, P. R. Florence, and R. Tedrake, "High-speed autonomous obstacle avoidance with pushbroom stereo," *Journal of Field Robotics*, vol. 35, no. 1, pp. 52–68, 2018.

[23] C. Zhou, F. Li, W. Cao, C. Wang, and Y. Wu, "Design and implementation of a novel obstacle avoidance scheme based on combination of cnn-based deep learning method and lidar-based image processing approach," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–11, 2018.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[25] K. Yang, K. Wang, S. Lin, J. Bai, L. M. Bergasa, and R. Arroyo, "Long-range traversability awareness and low-lying obstacle negotiation with realsense for the visually impaired," in *Proceedings of the 2018 International Conference on Information Science and System*. ACM, 2018, pp. 137–141.

[26] J. Wang, K. Yang, W. Hu, and K. Wang, "An environmental perception and navigational assistance system for visually impaired persons based on semantic stixels and sound interaction," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, pp. 1921–1926.

[27] M. Cordts, T. Rehfeld, L. Schneider, D. Pfeiffer, M. Enzweiler, S. Roth, M. Pollefeys, and U. Franke, "The stixel world," *Image Vision Comput.*, vol. 68, no. C, pp. 40–52, Dec. 2017. [Online]. Available: https://doi.org/10.1016/j.imavis.2017.01.009

[28] R. W. Bohannon, "Comfortable and maximum walking speed of adults aged 20—79 years: reference values and determinants," *Age and Ageing*, vol. 26, no. 1, pp. 15–19, 1997. [Online]. Available: http://ageing.oxfordjournals.org/content/26/1/15.abstract

[29] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 77–85.

[30] N. Morales, J. Toledo, L. Acosta, and J. Sánchez-Medina, "A Combined Voxel and Particle Filter-Based Approach for Fast Obstacle Detection and Tracking in Automotive Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1824–1834, Jul. 2017.

[31] S. J. Lee, Y. S. Moon, N. Y. Ko, H. T. Choi, and J. M. Lee, "A method for object detection using point cloud measurement in the sea environment," in *2017 IEEE Underwater Technology (UT)*, Feb. 2017, pp. 1–4.

[32] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[33] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open source language understanding and dialogue management," *arXiv preprint arXiv:1712.05181*, 2017.