Smart Walker: Smart Device to Assist Walkers

Dipankar Chakravort, Dongxin Liang, Hassan Yassine, Toma Suciu, Hodgson Tetteh, Farouq Adepetu

1. Abstract

In this proposal, we state the purpose and objective of our device that will assist people who need walkers/canes. We call this device Smart Walker. We describe our motives for wanting to make the device, the different phases we will go through and all the technologies that can be used to help us create it.

2. Project Context

The intended purpose of our project is to design and implement a smart walker that serves to enhance quality of life for those with poor balance and mobility, particularly patients who are visually impaired. Traditional medical-grade walkers provide support and balance, but still rely on the individual using the device to make a judgment call as to whether there is a clear, unobstructed path forward. In cases where the patient in question is visually impaired, this burden is offloaded to a second individual, who is assumed to accompany the patient. We propose the creation of a smart, depth-aware walker that will notify the operator when there are objects in their path, thereby minimizing, if not entirely preventing, the risk of injury in situations in which accompaniment is not possible.

3. Project Description

Navigating complex, unknown environments is an integral part of human life, but for individuals with disabilities this task is more challenging. According to a report published by the WHO in 2004, almost 82% of blind people worldwide are 50 years or older, 16% of which reported the use of a mobility device. Unfortunately, conventional navigation aids for the blind are not designed to be used in conjunction with traditional mobility aids, such as walkers. A common technique for walker users who are blind is to stop periodically and observe their surroundings using a white cane, which is slow, tedious, and reduces their mobility. Some visually impaired individuals need to be guided by a sighted individual to get around, which decreases the autonomy of the visually impaired individual. Our project is a smart walker for the blind that provides feedback about obstacles in the user's path of travel. This solution attempts to make the user safer, more mobile, less dependent on caretakers, and most importantly a solution that is low-cost.

In doing so, we aim to create a solution that is lightweight, portable, and comfortable for the user and it should be completed on the device itself, since a mobile network connection is not always assured. In addition to that, we would design the walker with the intention of providing feedback to the user about the distance to the obstacles, rather than simply alerting her to the presence of an obstacle and our preference is to do it without the use of audio, to avoid impeding the user's sense of hearing.

Our walker will consist of a standard off-the-shelf four-wheeled rollator with a Microsoft Kinect camera mounted on the basket and angled towards the floor. The Kinect power supply might be modified to be connected to a 12V DC rechargeable battery. Data processing would be performed by a Raspberry Pi microcomputer. There are multiple approaches for recognizing obstacles. Our most likely approach would be one that involves an analysis of depth images produced by the Kinect RGB-D camera. The algorithm searches for sudden changes in depth along the user's assumed direction of travel. We will show that the low computational requirement of this algorithm allows it to run on low-cost, low-power computing devices, hence the use of Raspberry Pi.

Vibration motors are attached to the handles of the rollator, allowing the walker to provide feedback to the user. The use of the Microsoft Kinect sensor contributes towards a low-cost design. Whereas in what is currently available on the market, the cost of sensors such as laser ranging devices is on the order of thousands of dollars, while in our project the cost is on the order of tens of dollars. In addition to that, the decision to use tactile (haptic) over audio feedback is because for visually impaired people, the sense of hearing is used to supplement the lack of eyesight while navigating. Requiring the user to wear earphones could impede navigation since it reduces their ability to hear ambient sounds.

If we had to optimize the feedback system, we would have to set it up in a way that the feedback would consist of three so-called "modes": close, mid-range, and far. Each mode is assigned a different vibration intensity, with far being the least intense and close being the most intense. But in our case, we will be using a binary system with an "ON" and "OFF" approach. It simplifies the process but would be a starting point to any future optimization.

4. Project Organization

The organization of this project is divided into 5 phases.

A. Research

In this phase we determine what the smart walker should do. We identify the scope of the project and get a basic understanding of the walker's functionalities. We also investigate what technologies can be used to build the smart walker.

B. Analysis

Analyze all the technologies we believe can be used to build the walker. This is the phase we test and determine which technologies we are going to use from the technologies we researched.

C. Design

Each individual component will get its own individual design. Then we bring it together to make the smart walker. We will show how the components give information to each other and how each component processes the information they received. Design a way to attach the device to a walker/cane

D. Implementation

During the implementation phase we will be using the agile methodology. We will setup milestones for each individual component and use sprints consisting of two weeks each. We will use Github and its features to let people work on different parts of the project and keep all relevant code/files in one place. We will discuss our progress once a week on Discord and any challenges/hurdles that we need to overcome, if any. The result after this phase should be a working product.

E. Deployment

In this phase we will make sure all our hardware and software are working as intended.

5. Technology Landscape

All the technologies needed to build the device.

A. RGB-D Camera

With the gradual development of disruptive technologies such as autonomous driving, 3D cameras are used for object recognition, behavior recognition, and scene modeling related applications. It can be said that RGB-D cameras are the eyes of terminals and robots. RGB-D cameras are also called 3D cameras. As the name implies, the camera can detect the depth of field distance of the shooting space.

The pictures taken by 2D color cameras can see all the objects in the camera's angle of view and record them, but the recorded data does not include the distance of these objects from the camera. We can only judge which objects are far away from us and which are closer through the semantic analysis of the image, but there is no exact data. The RGB-D camera solves this problem. Through the data obtained by the RGB-D camera, we can accurately know the distance of each point in the image from the camera, so that by adding the (x, y) coordinates of the point in the 2D image, we can get the three-dimensional space coordinates of each point in the image. The real scene can be restored through three-dimensional coordinates, and applications such as scene modeling can be realized.

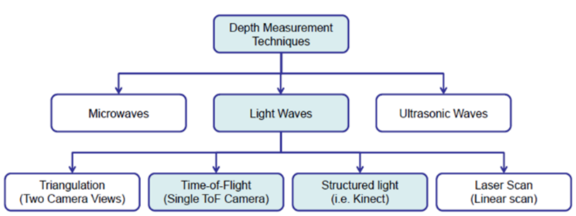

Figure 1: Depth Measurement Techniques

We will be discussing the technologies that use Light Waves.

1) Structured-light (Structured-light)

2) Time-of-flight method (TOF)

3) Stereo Vision (Triangulation)

1. Structured Light Depth Camera - The basic principle of structured light is that light with certain structural characteristics is projected onto the object to be photographed through a near-infrared laser, and then collected by a special infrared camera. This kind of light with a certain structure will collect different image phase information due to the different depth areas of the subject, and then use the arithmetic unit to convert this structure change into depth information to obtain a three-dimensional structure. To put it simply, the three-dimensional structure of the object being photographed is obtained through optical means, and then the obtained information is further applied. Usually, an invisible infrared laser with a specific wavelength is used as the light source. The light emitted by it is projected on the object through a certain code, and the distortion of the returned code pattern is calculated through a certain algorithm to obtain the position and depth information of the object.

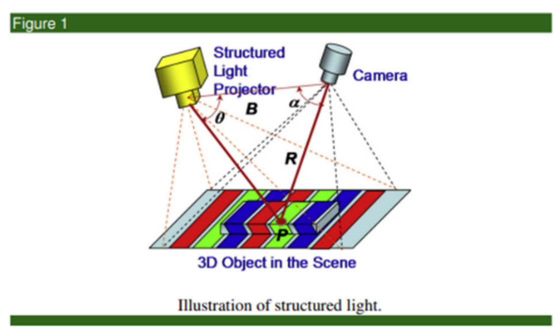

Figure 2: Illustration of Structured Light

2. Time of Flight - Time of Flight, specifically is to obtain the exact distance of the target by continuously emitting laser pulses to the target, and then using the sensor to receive the reflected light, and by detecting the flight round-trip time of the light pulses. Because of the speed of light laser, it is not feasible to directly measure the flight time, and it is generally achieved by detecting the phase shift of the light wave modulated by a certain method. Time of Flight methods can be divided into two types according to different modulation methods: Pulsed Modulation and Continuous Wave Modulation. Pulse modulation requires a very high-precision clock for measurement and needs to emit high-frequency and high-intensity lasers. At present, most methods of detecting phase shift are used to realize the Time of Flight function. To put it simply, it emits a processed light that will reflect when it hits an object to capture the time of the round trip. Because the speed of light and the wavelength of the modulated light are known, the distance to the object can be calculated quickly and accurately.

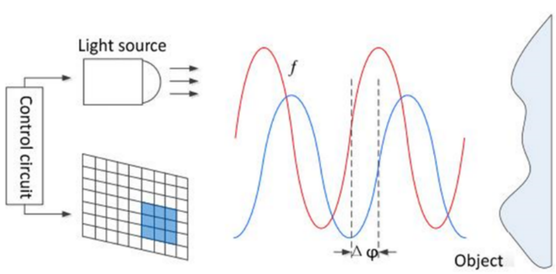

Figure 3: Time of Flight

3. Stereo Vision - Stereo Vision is another type of computer vision. The various objects that humans can see in the world are due to our visual system. If we want robots to distinguish objects like humans, we also need to give robots eyes. When we see an object, we can estimate whether the object is closer or further away from us. Then we need to give the robot more than a system that can recognize the object. And this kind of system also needs to be able to judge how far away the obstacle is, and the function of the stereo vision system can well meet the requirements.

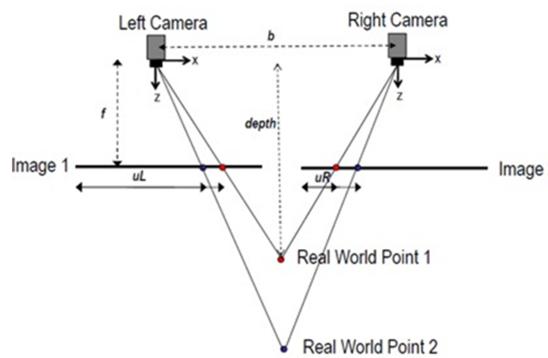

Figure 4: Stereo Vision

B. *Object Detection with Stereo-Vision*

Just like the two cameras in Figure 4 we can take the two images generated by the left and right cameras. Both images are identical and differ by only the Baseline represented in Figure 4 as b. Assuming we get one point in the image generated by the left camera **(Ul, Vl)** we then shoot out a ray from the left camera to the reference point. We also get the corresponding point from the right camera whose image is identical to the left camera but shifted to the right. Assuming we can get the corresponding point in the right image, then we can simply find the intersection of the two rays corresponding to the same point and use that to find the position of the object in space using the four equations given by Figure 6.

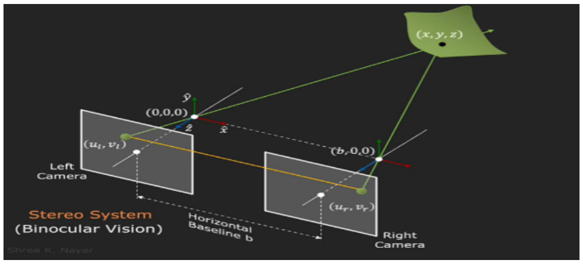

Figure 5: Stereo System

Simple Stereo: Depth and Disparity

From perspective projection:

$$(u_l, v_l) = \left(f_x \frac{x}{z} + o_x, f_y \frac{y}{z} + o_y\right) \qquad (u_r, v_r) = \left(f_x \frac{x-b}{z} + o_x, f_y \frac{y}{z} + o_y\right)$$

Solving for (x, y, z):

$$x = \frac{b(u_l - o_x)}{(u_l - u_r)} \qquad y = \frac{b f_x (v_l - o_y)}{f_y (u_l - u_r)} \qquad \boxed{z = \frac{b f_x}{(u_l - u_r)}}$$

Figure 6: Depth and Disparity

The Z refers to the depth or distance from our walker in our case , however if we pay attention to the equations referenced in the Figure 6, we notice that all equations depend on the Ul-Ur. This is referred to as the disparity, the depth is inversely proportional to the disparity, and the disparity increases with increasing separation of the two cameras also called the Baseline.

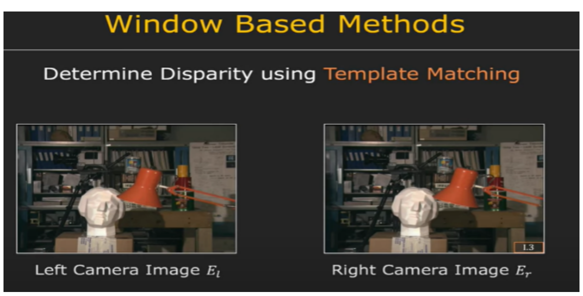

Figure 7: Window Based Methods

Figure 8: Disparity Map (Ground Truth)

Figure 7 and Figure 8 demonstrate how a simple disparity map can show how close or far an object is. Using a disparity map or a depth map we can determine which path around the user is safe to travel.

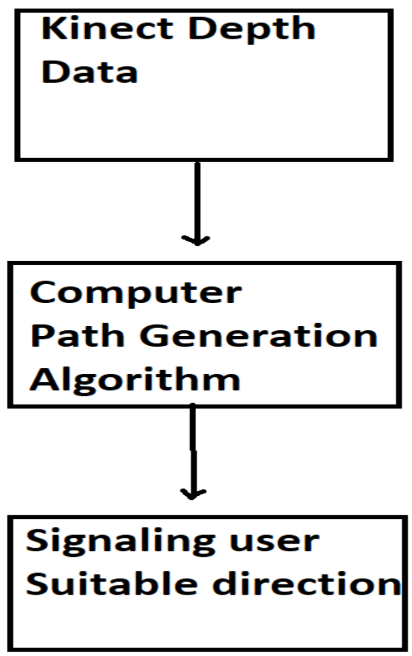

Figure 9: System Path Design

Figure 9 shows the simple high-level overview of the smart-walker system, the data generated from the Kinect or stereo cameras is sent to a computer to find the path suitable for the user to take, this path data can be sent to a specific section of the motors that indicate signal what direction the User might take.

C. *Machine/Deep Learning Algorithm*

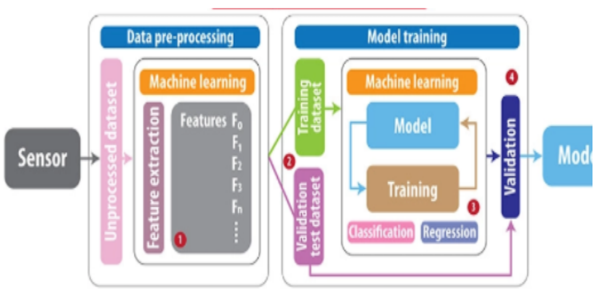

Figure 10: A general overview of the development and training scheme for a smart sensor.

A reliable smart model must be trained thoroughly with high-quality training datasets before being implemented into the smart sensor system. The model training development is split into two phases: data preprocessing and model training. Application-specific sensor data are compiled into unprocessed raw datasets and then processed by the data preprocessing step, in which the unprocessed datasets will involve preprocessing tasks such as handling null values, standardization, or encoding discrete class labels among others.

The following related discussion is separated into two distinct ML groups: Neural Network (NN) and non-Neural Network (non-NN) algorithms.

NN algorithms are highly efficient in terms of feature learning and extraction and require less manual input when compared with non-NN ML algorithms. A NN learns abstract features from a given dataset by the activation of neuron nodes that are within a NN, like a biological neural circuit.

Non-NN algorithms are more complex to configure and require more manual involvement to fine-tune ML parameters to achieve application outcomes. This can be attributed to the need for domain knowledge of the features in each sensor dataset to accurately develop an application specific non-NN smart model. Furthermore, the features can be either manually selected by a domain expert or "automatically" extracted by other ML algorithms.

D. *Feedback System*

When people interact with electronic devices, they use two out of the five senses, sight, and hearing. With haptic feedback we get to use another sense to interact with electronic devices, touch. This allows the computer to touch you back. When a different level of elevation is detected by the camera, a signal will be sent out to a haptic feedback system to alert the user. One type of haptic feedback is vibrotactile feedback systems. Vibrotactile feedback systems are tiny motors that create vibrations. For vibrotactile we have the option of using eccentric rotating mass (ERM) vibration motors or coin vibration motors.

E. *Feedback Technologies*

Eccentric Rotating Mass Vibration Motors (ERM) are vibration motors that spin a mass offset from the center of rotation. They have a cylindrical form factor, and the mass and rotating shaft are often exposed. One advantage of ERM is that they have strong vibrations. A disadvantage is that they are larger in size than Coin Vibration motors.

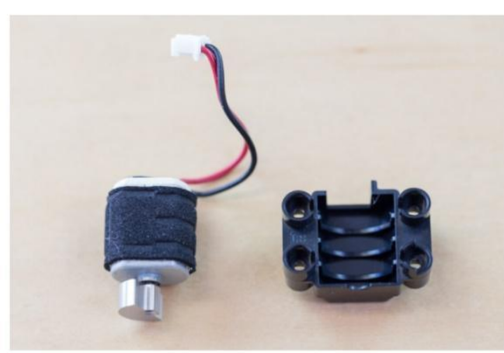

Figure 11: ERM vibration motor

Coin Vibration Motors “are flat, “pancake”. An advantage of Coin Vibration motors is that they have a more compact form than ERM’s and are smaller. A disadvantage is that the vibrations they produce are not as strong as ERM’s.

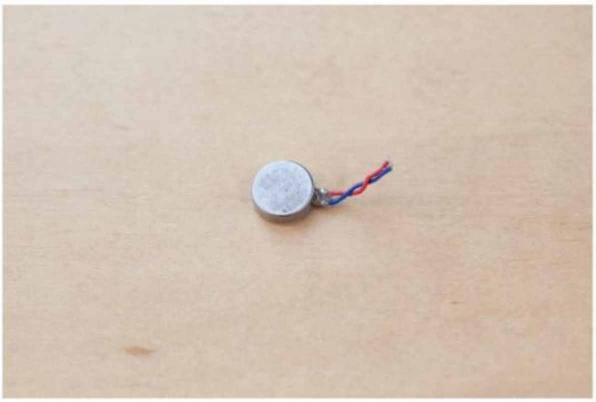

Figure 12: Coin Vibration Motors

Either an ERM or a coin vibration motor will be used with a raspberry pi to give haptic feedback to the user. The one used will depend on how much or how little vibration is needed to give proper feedback.

F. Raspberry Pi

A raspberry pi and a breadboard are needed to connect everything together. The raspberry pi will host the algorithm and the RGB-D camera will be connected to it through USB. The breadboard will host the vibration motor and will be connected to the raspberry pi pins. The Raspberry Pi 4 has everything needed to accomplish our objective. It has a processor to execute the algorithm, multiple USB ports to connect the RGB-D camera, and pins to connect the breadboard with the vibration motor.

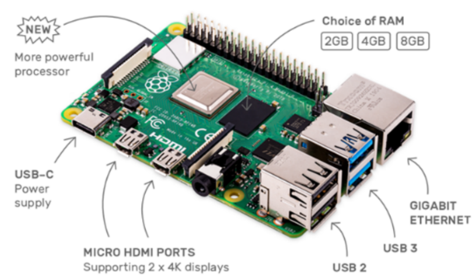

Figure 13: Raspberry Pi 4

6. Final Deliverable

The final deliverable is a device that is depth-aware and will notify the operator through haptic feedback when there is a change in elevation or objects in their path.

7. Architecture Design

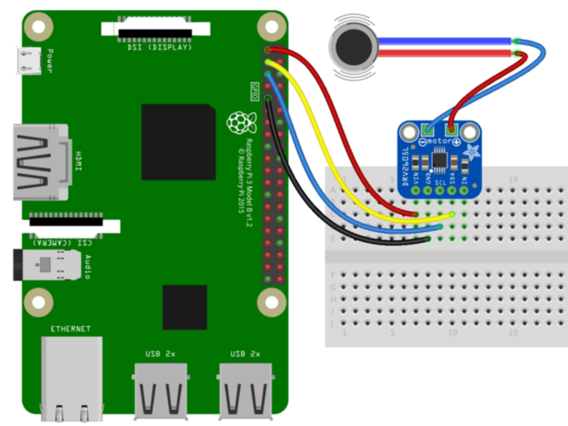

Figure 14: Raspberry Pi with Coin Vibration Motor

Figure 15: Xbox Kinect Camera

Will connect the Kinect to the USB ports of the Raspberry Pi.

References

L. Wang, N. Li, D. Ni, and J. Wu, “Navigation System for the visually impaired individuals with the Kinect and Vibrotactile Belt,” *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 2014.

L. Yang, B. Wang, R. Zhang, H. Zhou, and R. Wang, “Analysis on location accuracy for the Binocular Stereo Vision System,” *IEEE Photonics Journal*, vol. 10, no. 1, pp. 1–16, Feb. 2018.

N. Ha, K. Xu, G. Ren, A. Mitchell, and J. Z. Ou, “Machine learning‐enabled smart sensor systems,” *Advanced Intelligent Systems*, vol. 2, no. 9, p. 2000063, 2020.

R. Blenkinsopp, “What is haptic feedback?,” *Ultraleap*, 05-Apr-2019. [Online]. Available: https://www.ultraleap.com/company/news/blog/what-is-haptic-feedback/

R. Train, “Intro to haptic technology: Vibration Motors,” *Fictiv*, 07-Feb-2017. [Online]. Available: https://www.fictiv.com/articles/intro-to-haptic-technology-vibration-motors

“Raspberry pi documentation,” *Computers*. [Online]. Available: https://www.raspberrypi.com/documentation/computers/

Raspberry Pi, “Raspberry pi 4 model B specifications,” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

Y. He, B. Liang, Y. Zou, J. He, and J. Yang, “Depth errors analysis and correction for time-of-flight (TOF) cameras,” *Sensors*, vol. 17, no. 1, p. 92, 2017.