Smart Walker for the Visually Impaired

Jonathan Guilbe, Christopher Feltner, Sharare Zethabian, Siavash Khodadadeh, Lotzi Bölöni, and Damla Turgut
{jguilbe, chris.feltner, siavash.khodadadeh, sharare.zethabian}@Knights.ucf.edu, {lboloni, turgut}@cs.ucf.edu
Department of Computer Science
University of Central Florida

Abstract

Navigation of environments is a complex challenge for individuals who are visually impaired, and these individuals are typically aided by tools such as canes or guide dogs. However, individuals who are both visually and mobility impaired encounter greater difficulty, since conventional aids do not integrate well with walkers or rollators. Current depth sensing cameras, when combined with affordable micro-computing devices, have the potential to aid such individuals. We propose a smart walker which can recognize obstacles which may endanger the user, as well as obtain their distance from the user. We then convey this information through a haptic interface on the walker.

Goals

1. Create a solution which is low-cost.
2. Create a solution which is portable, and comfortable for the user.
3. Provide feedback on distance from obstacles to the user.
4. Provide feedback to the user without use of audio since visually impaired individuals have a greater reliance on their sense of hearing for navigation.

Background

Microsoft Kinect is an RGB-D camera, making it capable of producing both color images and depth images. In a depth image, each pixel value is representative of the distance from the camera.

We use Point Cloud Library (PCL) to create point clouds from distance information. A point cloud is an array of points with X, Y, and Z values in space. We use PCL to perform manipulation, filtering, and 3D perception operations on the cloud.

A scene captured in a color image, depth image, and point cloud (from left to right)

Raspberry Pi is a small-form computing device. It is extremely energy-efficient, but its processing capabilities are limited.

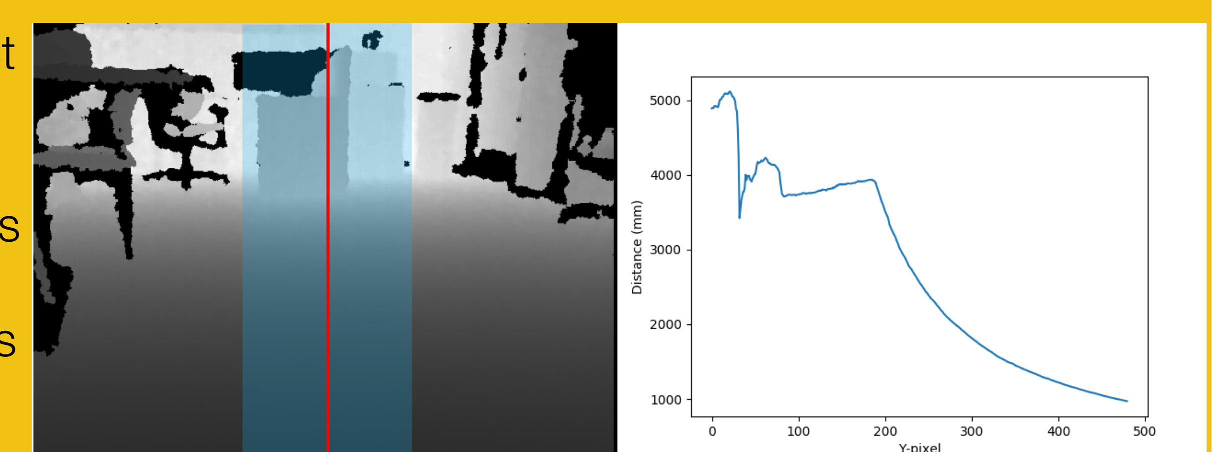
Methods

Approach 1 – Using Depth Images

- Create a Distance over Y-Pixel curve.
- Look for sudden changes in slope of the curve [2].
- Obstacles represented by decreases, stairs represented by large increases.

1) Remove Error Values and Generate Curve

- Remove Kinect error state values
- Average across a column
- Store values as an array



2) Find changes in slope

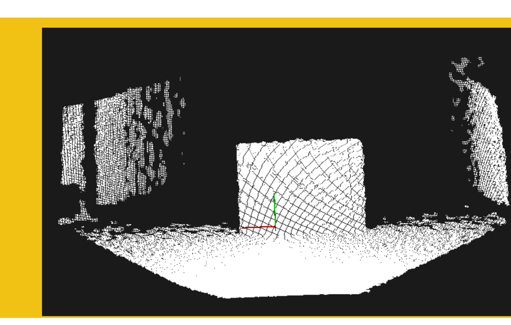
- Iterate through the array, checking the slopes of pairs of points.
- Break from the iteration if the slope is larger or smaller than thresholds.

Approach 2 – Using Point Clouds

We use an approach similar to Pham et. al. [3], described in the figure below. As in Li et. al. [4], after processing we consider remaining points to be obstacles. If there are more points than our threshold value remaining, we identify the point with the minimum z-value as the obstacle distance.

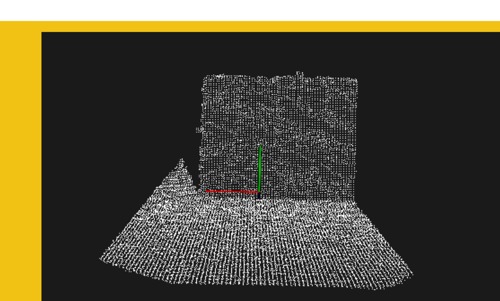
1) Acquire

PCL and OpenNI are used to generate a point cloud.



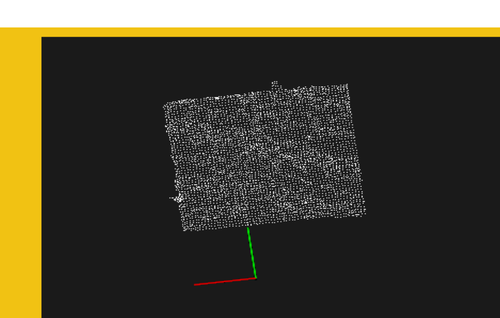
2) Downsample

A Pass-Through filter is applied to remove points beyond our bounds. A Voxel Grid Filter is used to reduce the number of points, improving runtime.



3) Segment

RANSAC is used to segment the largest plane parallel to the z-axis. This is assumed to be the floor. The minimum z-value among remaining points is considered the closest obstacle.



Results

In order to evaluate our methods, both approaches were used to detect obstacles at a known distance from the device.

In our evaluation, both methods were able to detect the presence of an obstacle, and were able to determine distance to within 10 cm of the actual distance.

An image of the walker

Environment	Approach 1	Approach 2	Actual Distance
Empty Hallway	*	*	*
Wall	130 cm	120 cm	128 cm
Downstairs	+	+	+
Upstairs	129 cm	119 cm	125 cm
Box in Path	187 cm	172 cm	182 cm

- "*" Represents no obstacle found
- "+" Represents obstacle found at close range

Future Work

Future goals include:
- Improving downwards drop and stair detection
- Creating a point cloud approach which can be run on an embedded device
- Implementing haptic feedback
- Find an alternative method for sunlit environments

Acknowledgements

The support for this work was provided by the National Science Foundation REU program under Award No. 1560302. Any opinions, findings, and conclusions and recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[1] S. Zehtabian, S. Khodadadeh, R. Pearlman, B. Willenberg, B. Kim, D. Turgut, L. Bölöni, and E. A. Ross. Supporting rehabilitation prescription compliance with an IoT-augmented four-legged walker. In Accepted to appear in 2nd Workshop on AI for Aging, Rehabilitation and Independent Assisted Living (ARIAL'18), July 2018.
[2] Nuria Ortigosa, Samuel Morillas, Guillermo Peris-Fajarnés, "Obstacle-Free Pathway Detection by Means of Depth Maps," Journal of Intelligent Robotic Systems, vol. 2011, pp. 63:115-129
[3] Huy-Hieu Pham, Thi-Lan Le, and Nicolas Vuillerme, "Real-Time Obstacle Detection System in Indoor Environment for the Visually Impaired Using Microsoft Kinect Sensor," Journal of Sensors, vol. 2016, Article ID 3754918, 13 pages, 2016. https://doi.org/10.1155/2016/3754918.
[4] B. Li, X. Zhang, J. P. Munoz, J. Xiao, X. Rong and Y. Tian, "Assisting blind people to avoid obstacles: An wearable obstacle stereo feedback system based on 3D detection," 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, 2015, pp. 2307-2311.