Kinect Analysis of Obstacles & Feedback for the Visually-Impaired

Samarth H. Shridhar, John B. Prewitt, Utkarsh Borikar

Engineering and Technology

GSMST

970 McElvaney Ln NW, Lawrenceville, GA 30044

Table of Contents

Introduction

The purpose of creating the Konnect Wearable is to provide visually-impaired people an inexpensive, easy-to-manufacture navigation system that facilitates existing visual therapy strategies and bridges the gaps between the visually-impaired and the tangible world through lightweight, applied machine learning and affordable mechatronics. Although endeavors to create electronic-assistive systems for the blind are numerous, cost, feasibility, and insignificant results hamper the possibility of opening visual therapy to methods beyond the white cane. A comparative study on the efficacy of ultrasonic white canes revealed no significant impact of their usage, despite the canes accurately detecting nearly 70% of the surrounding objects (Santos et al., 2020). Beyond the electronic canes' futility, the combination of machine learning and visual rehabilitation have cultivated expensive systems. The heavy utilization of LiDAR cameras for data collection inflates such devices' monetization, thus promulgating a financially ineffectual device in a pool of 20.6 million visually-impaired Americans ("Occupational Therapy Services," n.d.). With accommodative visual therapy already costing the average American between 2,000 and 6,000 dollars, affordability is integral in supplying visually-impaired individuals with inexpensive assistive technologies.

Machine learning is predictive modeling based on datasets fed into algorithms such as random forests, in which the algorithms identify patterns within the data and produce categorical predictions. Hence, the methodology is useful in developing assistive technology because, unlike the relatively linear output of sensors, machine learning yields a probabilistic classification of inputted data into a specified category. Objects surrounding a visually-impaired person can then be classified under a label, such as "stairs" or "wall." However, current researchers seeking to infuse deep learning strategies into rehabilitative technology are obstructed by their image data usage. Image data prompts the usage of convolutional neural networks (CNNs), computationally expensive algorithms that produce generally mixed results without an exorbitant amount of training data. CNNs examine the intrinsic details of image vectors, thus requiring large datasets for accurate predictions. For instance, a study utilizing the Kinect sensor for object detection received inconclusive results because their utilization of CNNs required exhaustive data (Shimakawa et al., 2017). Nevertheless, algorithmic loopholes such as employing the cuboid method (subdividing datasets into smaller sections) increase the amount of data, but the CNN is fundamentally inefficient without appropriate computational power that would raise the price of the device (Kowalczuk & Szymański, 2019). Thus, a central issue with combining machine learning and assistive technology is designing an efficient system that can produce significant results without spiraling cost-efficacy. With holistic surveys concluding nearly 42% of visually-impaired people suffer encounters with physical obstacles, scalable, accurate, and efficient devices must be produced to seal the disparity of independence between able and visually-disabled people. Therefore, while LiDAR technology in conjunction with CNNs produces accurate results, a much cheaper proxy could be constructed from exalting the relatively poor-performing and affordable Kinect sensor through algorithmic strategies, with similar if not the same scope of function. Additionally, the Kinect can be integrated into a holster for use around a user's waist, serving as a wearable communicating obstacle detection alarms in real-time.

Hypotheses

Null: If a visually impaired person utilizes the Konnect Wearable, there is no statistically significant difference in the navigational time through the course than a case in which the device was not utilized.

Alternative: If a visually impaired person utilizes the Konnect Wearable, the visually impaired persons will experience a decrease in navigational time through their environment compared to usage without the system and with traditional navigation methods (i.e. white cane).

Variables

The research is subdivided into two stages, classification and experimentation. The classification stage gauges the accuracy of classified objects from the algorithm's predictions. The experimentation stage tests the efficiency of utilizing the Konnect Wearable by examining the walking speeds of various trials. The independent variable of the classification stage will be the depth data captured and manipulated by the user. Since the Kinect moves with the user, the streams of depth values continuously change within a bound of zero to 4095 millimeters. The independent variable for the experimentation stage project is the type of device used to aid or hinder the user's navigational sense as they traverse a controlled environment. Varying conditions encompass no aid (normal vision), blindfold a white cane with a blindfold, the Konnect with a blindfold, and the Konnect with a white cane and blindfold. Normal vision and blindfolding serve as controls. This independent variable has no units because the type of aid is being manipulated. The dependent variable of the classification stage was the theoretical and empirical accuracies outputted by the model. The model classifies an object under these labels: "No Obstacle", "Known Obstacle", "Upstairs", "Downstairs", "Wall without Floor", and "Wall with Floor". Subsequently, the accuracy is the percentage of correctly identified classifications. The dependent variable of the experimentation phase is the time required to traverse throughout the environment when paired with a specific condition. The values will be the amount of time it takes to fully complete a controlled course, measured in seconds with a stopwatch. The controlled variables involved in the classification stage are within the Kinect model. The same Kinect sensor, code, and laptop will be used for computational consistency. Additionally, the machine learning algorithm (FastRandomForest) will remain constant throughout the classification process. As for the experimentation process, the following components will remain constant: the individual in the experiment, the white cane, the blindfold, the holster, the environment tested in, and the Kinect device utilized.

Materials & Software

Materials	Quantity : Price
Microsoft 1414 Kinect V1 Sensor	1 : $10
Mobile Phone / Camera	1 : $0
3D Printer w/ Filament	1 : $25
Laptop	1 : $0
White Cane	1 : $10
Backpack	1 : $0
Cardboard / Styrofoam	N/A : $0
Microsoft ML.NET (Employment of Machine Learning Algorithms)	N/A : $0
Kinect V1.8 SDK (Utilization of Kinect Sensor)	N/A : $0
Visual Studio 2019 (Integrated Development Environment)	N/A : $0

Procedure

Part A: Designing The Code (Full Codebase)

- **Initializing the Kinect:** Using inspiration from the Kinect SDK sample code, create a series of event handlers that will connect the Kinect to the codebase and activate the sensor when a depth and frame are taken.
- **Getting Depth and Color Frames:** Create algorithms as event listeners that trigger when the respective frames are ready. The depth frame will be where the majority of the code is found and called whereas the color frame is used to align the line profiles to be used in data collection. In these event listeners, create a writable bitmap that will map the frame objects to a continuous image stream displayed on the GUI.
- **Intuitive Graphical User Interface (GUI):** Inside the Visual Studio (VS) integrated development environment, create a new XAML application. Inside this XAML application there should be a button that fires an event for data collection (Export to CSV) as well as display the continuous image streams from the frame events. On the XAML application there also are drop down boxes to set the proper target for each line profile in data collection as well as labels that can be set to display the machine learning prediction for each line profile.
- **Export to CSV:** This event handler will fire when the Export to CSV button on the GUI is clicked. Once the button is clicked, a series of for loops will read down the seven-line profiles (Far Left, Left, Close Left, Center, Close Right, Right, Far Right) spaced 80 pixels apart on the Kinect frame and collect their depths. These for-loops then write the depths of these pixels into a comma-separated text file alongside the designated target given by the dropdown boxes on the GUI. However, a major issue comes with the Kinect's inability to determine depths of reflective or transparent objects. In order to combat this, two smoothing

algorithms were implemented with the data collection process to predict these unknown depths: one smoothing in real time and another post data collection, named FirstSmooth and DoubleSmooth respectively.

- **First Smooth (Null Depth Evaluator):** Unknown depth finder scans around the given unknown pixel in "bands" searching for close pixels. There is a 3-band threshold for the smoothing to ensure that the program maintains its minimal computational intensity. If it gets at least three known depth pixel values in these 3-bands the average of them is taken and extrapolated to the unknown depth of the pixel which is then stored into the csv file by the export to csv algorithm that calls this function on all unknown pixels.
- **Double Smooth:** Following data collection, the double smooth algorithm is run on the .csv file and analyzes depths along the respective line profile and attempts to find a small portion of 0s (<6) are found in between two known depth values then sets them to be the average of the two surrounding known values. Using this, the smoothing is able to minimize computational intensity as it no longer needs to smooth in real time while displaying the depth frames and all computer power can be used in smoothing.
- **Min-Max Algorithm:** (See Appendix A for graphs) The Min-Max algorithm is integral in normalizing the distribution of the data arrays prior feeding them into the machine learning algorithm. Because the range of depth is bound between 0 and 4095 millimeters, there is room for disparities and skew between data points. The result is variables with significant disparities that may not contribute equally to model fitting, resulting in the possibility of bias while training. The Min-Max algorithm scans the data points within the bitstreams and inputs them into the formula $\frac{x-x_{min}}{x_{max}-x_{min}}$, where *x* signifies a data point. Thus, every data point within an individual bitstream falls into the range of [0, 1], standardizing the *n*-dim.

- **LightGBM Algorithm:** Using the ML.NET Model Builder’s text classification option, the normalized dataset of depth streams was fed into the model for a training time of 600 seconds. ML.NET trains the model on ten varying algorithms to determine the model with the highest accuracy. Incidentally, the LightGBM algorithm, a variation of a Random Forest algorithm, presented the highest accuracy (micro-accuracy of 81.20%).
- **Estimating Depths**: There are a series of small algorithms that determine the estimated depth of a classified target. This classification is then run through a switch case that determines which algorithm will be run to estimate the distance of the target from the Kinect. For example, the Wall without Floor classification algorithm uses an implementation of RANSAC, an algorithm which detects planes relative to a perpendicular surface and estimates the depths of the wall disregarding the plane (in this case, the floor).
- **Hierarchy:** The Hierarchy algorithm is used to determine which callout to make to the user using the speech to text. The algorithm is called once every three seconds and makes the text-to-speech call to the user. This algorithm takes three factors into account: Depth, Classification, and Value Proximity to Center. The Hierarchy uses these three factors to rank each bitstream consumed by the Konnect and then sends auditory feedback to the user with the System.Speech.Synthesis and .SpeakAsync() methods of the Speech Synthesizer library.

Part B: Data Collection

1. Ensure Kinect is properly connected to the computer and powered on.
2. Begin in a darkened room to prevent stray light from skewing results.
3. Ensure the environment has the necessary objects: stairs, obstacles, and several walls.
4. Close any concurrent programs to ensure computational efficiency. Connect the computer to

a power source to prevent loss of data.

5. Run the “Main.Window.XAML” application via Visual Studio. The GUI containing the RGB camera and the Grayscale camera will appear.
6. Position the Kinect Camera on your waist to simulate data collection from an appropriate height. Avoid rapid movements to prevent sensor recalibration.
7. In the GUI, match the depth bars (thin black columns) to the desired target: No Obstacle, Known Obstacle, Upstairs, Downstairs, Wall with Floor, Wall without Floor.
    a. Several unique targets may appear in one frame. In the frame, seven columns (far left, close left, left, center, right, close right, and far right) should be assigned the appropriate label (No Obstacle, Known Obstacle, Upstairs, Downstairs, Wall with Floor, Wall without Floor). Each black bar and label represent a bitstream of depth values with a corresponding object. Hence, seven unique data points are collected per frame, ensuring ample data is collected while expediting the collection process. See Appendix D for the completed GUI.
8. Once appropriate labels are assigned within the frame, click the “Export to CSV” button at the top-center of the GUI.
9. Repeat steps 2-7 until ample data (8500+ data points) has been collected.
    a. To check the number of data points collected, examine the number of filled columns within the .csv file.
10. Open the “DoubleSmooth.py” file in an appropriate IDE/Text Editor. Run the script, and enter the .csv file’s path into the terminal.
    a. The DoubleSmooth algorithm removes any outstanding outliers or data labeled as “trash” in the collection process. Ensure Python (version 3.x) is installed for this

script to perform accordingly.

11. Open the “MinMax.py” file in an appropriate IDE/Text Editor. Run the script, and enter the CSV file’s path into the terminal.
    a. The MinMax algorithm normalizes the distribution of the collected data. Ensure Python (version 3.x) is installed for this script to perform accordingly.

Part C: Construction/Testing:

1. Create trained ML.NET LightGBM text classifier model from collected data
2. Write the code designating text-to-speech alerts so that audio feedback of obstacles is possible.
3. Download/compile all the code, including other parts of the procedure.
4. Design and 3D print out a model holster in Inventor; it will be used to secure the Konnect around the user’s waist and enable live feedback. See Appendix B for the Konnect Holster.
5. Assemble the jig by attaching a Kinect sensor and a belt for securing the holster in place. Equip the human participant with the Kinect jig and a backpack and wire the jig with a laptop that is concealed within the backpack. See Appendix C for a subject fitted with the device.
6. Prepare Obstacle Courses. Include two obstacles courses with miscellaneous obstacles (walls, objects, empty space, etc.). The other two obstacle courses should focus on performance with stairs; one of these should be upstairs only, while the other should be downstairs only. See Appendix E for a rudimentary course illustration.
7. Create a test case randomization system to reduce the possibility of the participant memorizing segments of a course and altering results. First, make a table with the four blindfold scenarios as the column headers and the four courses. Number each of the elements of this table with a number from one to sixteen. Randomly generate numbers between one

and sixteen that represent each element, and the order each element appears in will be the order testing occurs. Make sure to not repeat the same element. Please find below the aids used in this table:

   a. 1st Scenario (Negative Control / Blindfold): Test the time to complete each course with full blindness but no visual aids. This is a control that tests how difficult it is for a blind person to travel through each course.
   b. 2nd Scenario (Blindfolded with White Cane): Test the time to complete each course blindly but with a white cane to map out the environment.
   c. 3rd Scenario (Blindfolded with Konnect Device): Test the time to complete each course blindly but with the Konnect Device to detect obstacles in the environment.
   d. 4th Scenario (Blindfolded with White Cane and Konnect Device): Test the time to complete each course blindly but with both the white cane and Konnect Device as aids. Use the distinct functions of each aid together to raise efficiency.

8. Create a new table using the same test case randomization system. However, this time, the only scenario will be normal vision without a blindfold. Instead of numbering the elements one to sixteen, number them one to four.
   a. 1st Scenario (Positive Control / Normal Vision without Blindfold): Test the time to complete each course with clear, corrected vision without any visual aids. This control tests how quickly one can casually travel throughout each course.
9. Record the time to complete trials in the order they were randomly generated, as well as the participant's accuracy (how many obstacles were hit) and Konnect classification accuracy when needed. Complete the normal vision trials iff all blindfolded trials have been completed.

10. Determine statistical significance between assistive aid categories

Data

Figure 1: Sensor Data ([Full Data Table](#))

Target	0	1	2	3
Side Wall	0.82	0.86	0.86	0.9
Side Wall	1	1	0.953125	0.890625
Side Wall	0.868056	0.913194	0.868056	0.868056
No Obstacle	0.809267	0.809267	0.828125	0.828125
No Obstacle	0.890984	0.890984	0.890984	0.890984
Wall with Floor	0.917591	0.917591	0.917591	0.917591

Figure 2: Traversal Times

	Normal Vision	Blindfolded	White Cane	Konnect	Konnect With White Cane
Course #1 Times (s)	17.24	118.21	98.45	86.34	71.08
Course #2 (Downstairs) Times (s)	11.03	48.72	58.27	50.84	39.30
Course #3 (Upstairs) Times (s)	16.76	67.31	63.10	48.72	36.79
Course #4 Times (s)	14.64	53.97	48.69	44.89	39.30

Figure 3: Kruskal Wallis Test

Kruskal-Wallis H Test

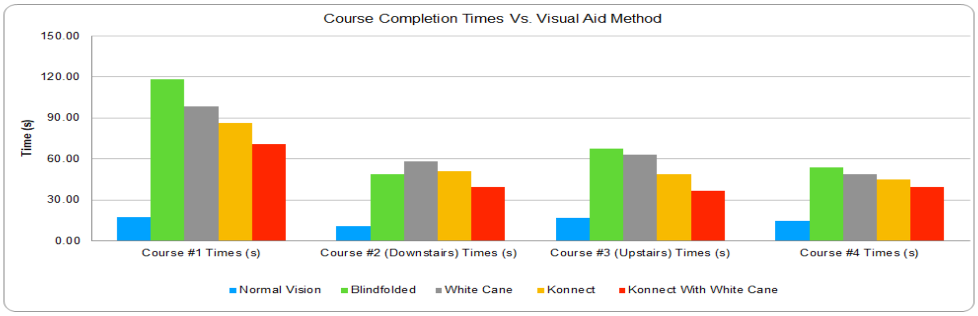
	Normal Vision (Control)	Blindfold (Control)	White Cane	Konnect	Konnect With Cane	
			12	11	10	
			8	7	2.5	
			9	7.5	1	
			6	4	2.5	
R			35	29.5	16	
n			4	4	4	12
R^2 / n			306.25	217.5625	64	587.8125
H						6.216346154
df						2
p						0.04468251
α						0.05
sig						yes

Figure 4: Mann Whitney U Tests

Pairwise Mann-Whitney tests

group 1	group 2	p-value	U-stat	mean
Normal Vision	Blindfolded	0.030382822	0	57.5
Normal Vision	White Cane	0.030382822	0	52.25
Normal Vision	Konnect	0.030382822	0	43.25
Normal Vision	Konnect With White Cane	0.029401048	0	32
Blindfolded	White Cane	0.885233914	7	5.25
Blindfolded	Konnect	0.383630327	4.5	14.25
Blindfolded	Konnect With White Cane	0.191266987	3	25.5
White Cane	Konnect	0.470486422	5	9
White Cane	Konnect With White Cane	0.191266987	3	20.25
Konnect	Konnect With White Cane	0.191266987	3	11.25

Figure 5: Per-Course Traversal Times



Discussion

The overarching trend observed was that utilizing the Konnect with a white cane was consistently faster than the other tested visual aids methods, as portrayed in Figure 5. To confirm statistical significance, a Kruskal-Wallis Test was run to evaluate variance over all visual aid methods, and a Mann Whitney U test was run to evaluate pairwise correlations. The independent variable is the visual aid method with the dependent variable being course traversal time.

Because of the relatively small amount of data collected and the resultant non-uniform distribution, we selected non-parametric alternatives to ANOVA and Student's T-tests, the Kruskal-Wallis Test (Figure 3) and Pairwise Mann Whitney U Tests (Figure 4), respectively. The α-value for both tests was set to 0.05 so that we could achieve a 95% confidence interval. The Kruskal-Wallis test reported a p-calculated value of ~0.045. The Mann Whitney U tests reported a p-calculated value of ~0.470 between white cane and the Konnect, and ~0.191 between white cane and Konnect with the white cane and Konnect with Konnect with the white cane. The control case, normal vision, was omitted from the tests because of the inherent statistical significance it bears over the other cases. Every p-calculated value pertaining to pairs with the normal vision case was <0.05. Therefore, the test case would have skewed the Kruskal-Wallis test results because of its inherent significant difference from the other test cases. The blindfold case was omitted because it also serves as a control. Though the blindfold case did not render statistical significance from the Konnect and white cane trials, the cause is attributed to qualitative analysis. Courses such as course #2 (downstairs) and course #3 (upstairs) had railings that successfully assisted the individual, with or without the white cane. Thus, although the blindfold case is insignificant, the Kruskal-Wallis test's objective is not to compare already insignificant data but to identify underlying disparities between the cases that were not controls. Generally, the blindfold cases reported the slowest times across all four trials, followed by white cane and Konnect individually (with differing orders). The quickest (disregarding normal vision, the inherent fastest time) time was the Konnect and White Cane used in conjunction.

The Kruskal Wallis test's p-calculated value, ~0.045, is less than the chosen α-value of 0.05. Therefore, the Kruskal-Wallis test confirms statistical significance between using the white cane, Konnect, and Konnect with white cane, rejecting the null hypothesis. However, the Kruskal-Wallis test fails to identify where the disparity lies within the dataset. Thus, the Pairwise Mann Whitney U test identifies intergroup correlations to pinpoint areas of possible significance. Although the permutations between white cane, Konnect, and Konnect with white cane resulted in p-calculated values greater than 0.05 and subsequently statistically insignificant, there is a consistent difference between pairs that test with Konnect with cane and those that do not. The permutation white cane vs. Konnect returns a relatively high p-calculated value,~0.470, which is substantially larger than the α-value of 0.05. However, the p-calculated values for the permutations containing Konnect with cane (white cane vs. Konnect with cane, Konnect vs. Konnect with cane) are ~0.191, insignificant but closer to 0.05, indicating the usage of Konnect with cane is notably nearer statistical significance than cases without its presence. Hence, this disparity bolsters the conclusion of the Kruskal-Wallis test and therefore supports the claim that using the Konnect device with a white cane is significantly different than the individual Konnect and white cane usage when traversing an indoor environment. According to the Kruskal-Wallis table, the Konnect with cane case contains the majority of the highest-ranked values from the dataset. The values are ranked from greatest (low rank) to least (high rank). The Konnect with cane category thus contains the fastest times for traversing the courses. Hence, as confirmed by the Kruskal-Wallis and Mann Whitney U tests, using the Konnect device with a white cane is statistically faster (in seconds) than other assistive methods.

We speculate that utilizing both the white cane and Konnect results in the fastest and safest navigation because two different aids, haptic feedback and computer vision, facilitate movement in various methods. For example, the Konnect classifies obstacles and can provide feedback about those obstacles (stairs are difficult to identify with a white cane, and electronic white canes do not classify objects) The Konnect also has depth perception and provides a relative distance to the identified object. On the other hand, the white cane locates the exact position of obstacles and enables the user to map the environment. Because of this haptic and artificial intelligent duality, a blind user can utilize the Konnect and the white cane to identify obstacles and then classify them via the device. Once the Konnect detects an obstacle, the white cane can locate and avoid it. Hence, the addition of the white cane seems to manifest haptic familiarity and temporal control, increasing the success of traversing new environments.

There were several possible errors within our testing that would have been adjusted had the group more time and necessary resources. The first source of error resulted from the test subject learning the individual courses as they were repeated each four times. However, our empirical method sought to mitigate the chances of this occurrence. Grouping the Blindfold, White Cane, Konnect, and Konnect with Cane categories, we assigned a number to each of the sixteen cases. A number within the range of those numbers was assigned to the cases.

Subsequently, the test subject traversed the case that the random number generator selected. Therefore, the test subject’s course selection and type of assistive technology were randomized, preventing the test subject from gradually getting accustomed to a specific course. Although our subject was not a thoroughly blind individual accustomed to living life without vision, he did not invalidate the experiment but rather fortified it compared to a sole test subject with standard vision. The test subject has a -6.00 prescription, rendering him legally blind with uncorrected vision. This designation makes the test subject similar to an average blind individual. Additionally, according to a comparative study with blind and ordinary vision people, subjects with uncorrected visual impairments tended to yield more accurate, consistent results (Santos et al., 2020). However, since the subject was fifteen-years old, the efficacy of the Konnect can be generalized exclusively to visually-impaired populations around adolescence.

The first significant method our group would fundamentally change is training our model on more data to augment the accuracy of our classifications. Though the LightGBM model rendered a holistic accuracy of 81.2%, an instance of incorrect feedback could pose potential danger for the user. As a result of COVID-19 restrictions, only two data collection sessions spanning two hours were conducted, and more time would have permitted collection of more data. Despite the relatively small dataset, the data gathered was sufficient for a competitive empirical accuracy (96.2%). Nonetheless, additional data could have potentially raised the model's theoretical accuracy and subsequent empirical accuracy. As for the second part of data collection, the experimentation, having an increased number of trials for each course and aid combination would have increased our sample size and mitigated the statistical effect of outliers. Additionally, utilizing more courses and employing legally blind subjects would generalize our data over a broader demographic and reduce the probability that outliers sway our results.

Finally, if the group had more time, we would conduct a supplemental study that analyzes Konnect users' progression and compares the traversal efficiency of a new user to that of an individual who is accustomed to the Konnect.

Conclusion

The hypothesis that utilizing the Konnect would facilitate the user's traversal to the extent their travel time is significantly lower was confirmed by the statistical analyses of the data. Because the Kruskal-Wallis Test value had a p-calculated value less than 0.05, the group concluded that using a Konnect with a white cane is the statistically fastest visual aid method. Additionally, the Konnect is comparably accurate compared to existing technologies, with a theoretical accuracy of 81.2% and an empirical accuracy of 96.2%, determined out of 104 test classifications. Therefore, the Konnect is an affordable alternative to numerous assistive technologies and functions similar to if not superior to rival devices. The Konnect pursuit accomplished elevating an inexpensive, mass-produced sensor to pioneer the actively researched field of combining machine learning and assistive technology. The device has practical applications for affordable visual therapy that can extend into parts of the world without sustainable access to expensive equipment. The Konnect, costing nearly $15, provides a low-priced technology that can process, classify, and communicate the proximity of many obstacles with ample computational efficiency and time complexity. Compared to the praised assistive technology, The Sound of Vision, the Konnect identifies a substantial range of potential obstacles while also classifying them—a function that The Sound of Vision lacks. Additionally, the Konnect is scalable. With necessary driver code changes, the software system can be used with LiDAR, Infrared, or virtually any camera that captures point cloud sequences. Other popular technologies such as TapTapSee and Drishti Navigation utilize GPS for obstacle identification and are subsequently restricted to GPS-distinguishable zones. However, the Konnect functions in a significantly broader range of environments. The electronic cane, a comparably versatile device, does not detect obstacles to the accuracy of the Konnect (72%) and is not statistically faster than the white cane, the primary device it is attempting to exalt.

There are areas where the Konnect can experience future improvements. The most notable change could be giving Konnect a public platform via open-source hosting. With publicly accessible databases, people worldwide can gather data themselves with the software's user-friendly graphical interface and contribute to increasing Konnect's accuracy. To make the Konnect less intrusive, a microcontroller such as the Meadow F7 can downsize the hardware's form factor. Lastly, appending haptic interfaces to our existing audio interface can provide feedback to blind and deaf people, broadening the Konnect's applicative demographic. With a score of applications and increased affordability compared to traditional assistive devices, the Konnect can be used to facilitate the visually-impaired further and ignite more research into fusing machine learning and rehabilitative technology.

Literature Cited

Filipe, V., Fernandes, F., Fernandes, H., Sousa, A., Paredes, H., & Barroso, J. (2012). Blind navigation support system based on Microsoft Kinect. Procedia Computer Science, 14, 94-101. doi:10.1016/j.procs.2012.10.011

Hoffmann, R., Spagnol, S., Kristjánsson, Á., & Unnthorsson, R. (2018). Evaluation of an audio-haptic sensory substitution device for enhancing spatial awareness for the visually impaired. Optometry and vision science: official publication of the American Academy of Optometry, 95(9), 757–765. https://doi.org/10.1097/OPX.0000000000001284

Kowalczuk, Z., & Szymański, K. (2019, September 10). Classification of objects in the LIDAR point clouds using Deep Neural Networks based on the PointNet model. Retrieved September 27, 2020, from https://www.sciencedirect.com/science/article/pii/S2405896319304331

Sharma, K. Kinect sensor based object feature estimation in depth images. International Journal of Signal Processing, Image Processing and Pattern Recognition, 8(12), 237-246. https://dx.doi.org/10.14257/ijsip.2015.8.12.23

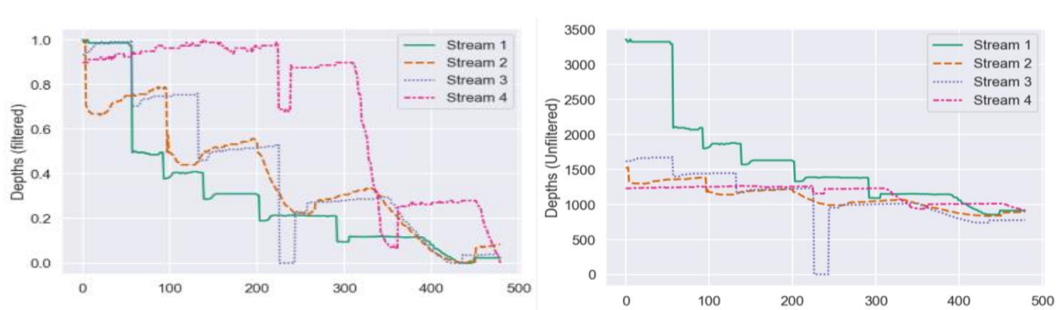
Shimakawa, M., Akutagawa, R., Kiyota, K., & Nakano, M. (2017). A study on staircase detection for visually impaired person by machine learning using RGB-D images. Proceedings of The 5th IIAE International Conference on Industrial Application Engineering 2017. doi:10.12792/icisip2017.080

Occupational Therapy Services for Persons With Visual Impairment. (n.d.). Retrieved November 17, 2020, from https://www.aota.org/About-Occupational-Therapy/Professionals/PA/Facts/low-vision.aspx

Zhao, Yuhang & Kupferstein, Elizabeth & Tal, Doron & Azenkot, Shiri. (2018). "It looks beautiful but scary": How low vision people navigate stairs and other surface level changes. 307-320. doi:10.1145/3234695.3236359

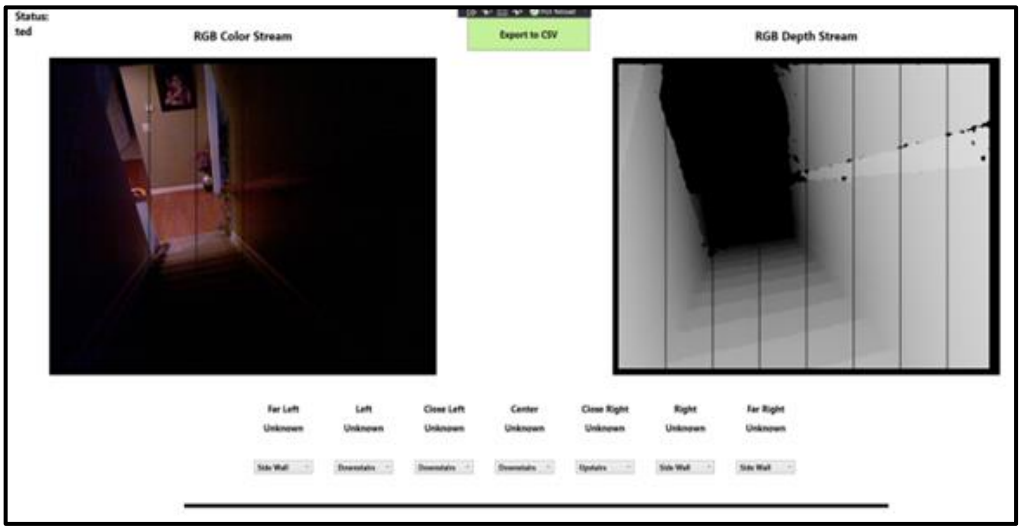
Appendix

A. MinMax (left) vs Raw (right)

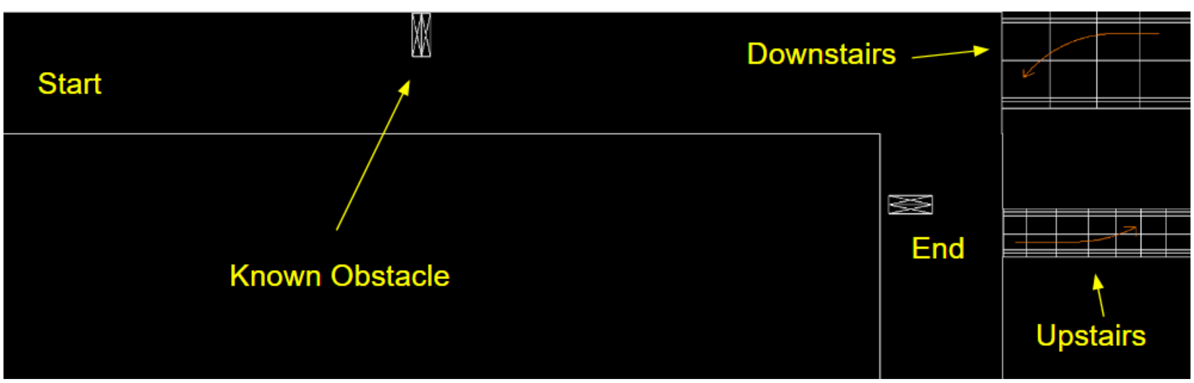


B. Konnect Holster 

C. 

D. Konnect GUI 

E. Course Design #1



Acknowledgements

The group would like to acknowledge Mr. Nguyen, who advised us to ensure our success, Clifton Prewitt, who equipped us with necessary materials, and Mr. Coddington, who generously allowed us to use his 3D printer.