OmniSense

Dhanush, Kautilya, Shangmin, Adeland

Premise

- Visually impaired individuals are exposed to dangerous situations and should have more support and accommodations to ensure their safety and well-being. The WHO indicates that 1 billion people have some level of disability, and it's increasing now more than ever!

- Monostatic radio frequency devices, such as software-defined radios (SDRs), have the potential to map environments and provide improved safety for visually impaired users. They're extensively used in radar technologies (esp. 80s).

Application

- The utilization of monostatic radio frequency devices allows for the early detection of potential obstacles or dangers approaching the user.

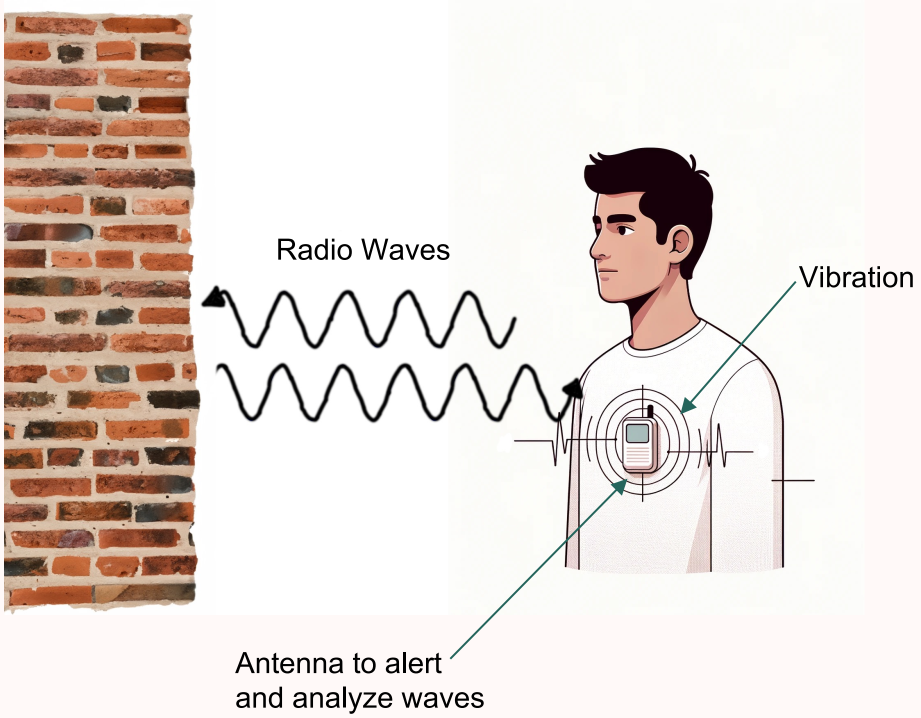
- This usage of radio waves allows for the constant mapping of an area omnidirectional around the user, keeping them updated their surroundings.

Proof of Concept

- Utilizes portable wearable device that emits waves at a constant frequency to notify user of oncoming obstacles.

- The device leverages monostatic radar system to quantify the object's distance, speed and direction!!

- Object provides feedback through vibration. So, for sure we know that there is something and where it is. Question is, how do we dodge it.

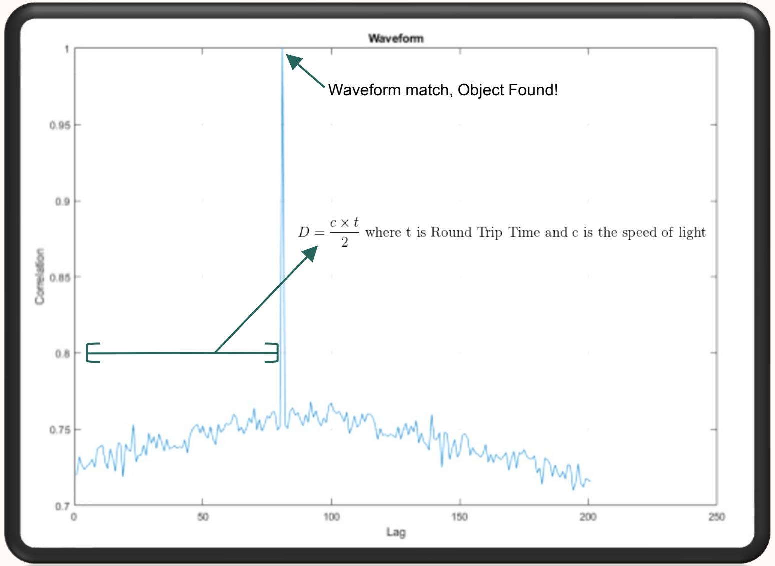


Simulations and Graphs

Why Graphs?

- They are helpful for understanding the relationship between variables through simulations.
- We have exhausted all other options in conducting scientific experiments to prove this concept.
- Generate meaningful data, which can we can use to product inferences and conclusions, thereby arriving towards the proof of the concept.
- The complexity of the SDRs in radar technology is immense:

Iteration 1: Detect Object

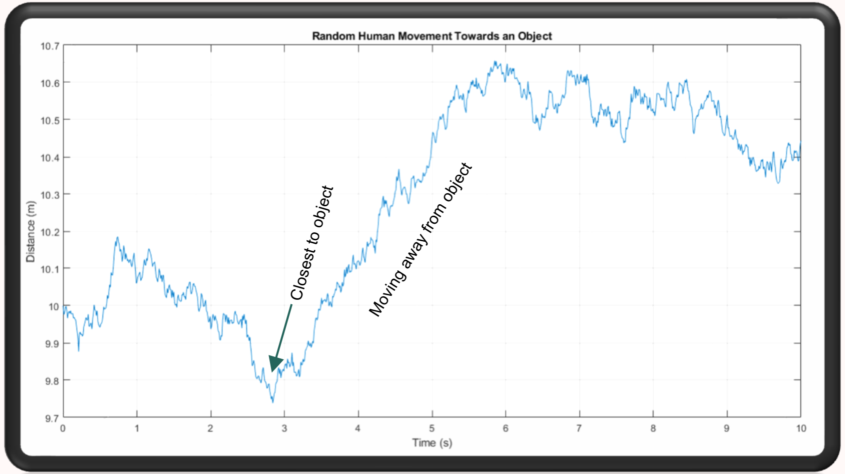


Waveform Correlation v Lag Graph

PSEUDOCODE

- Generates two waveforms: ref_waveform and recv_waveform.
- Calculates the cross-correlation between the two waveforms using the xcorr() function.
- Normalizes and plots the cross-correlation data.
- Saves the graph values to a CSV file.

Iteration 2: Find Distance

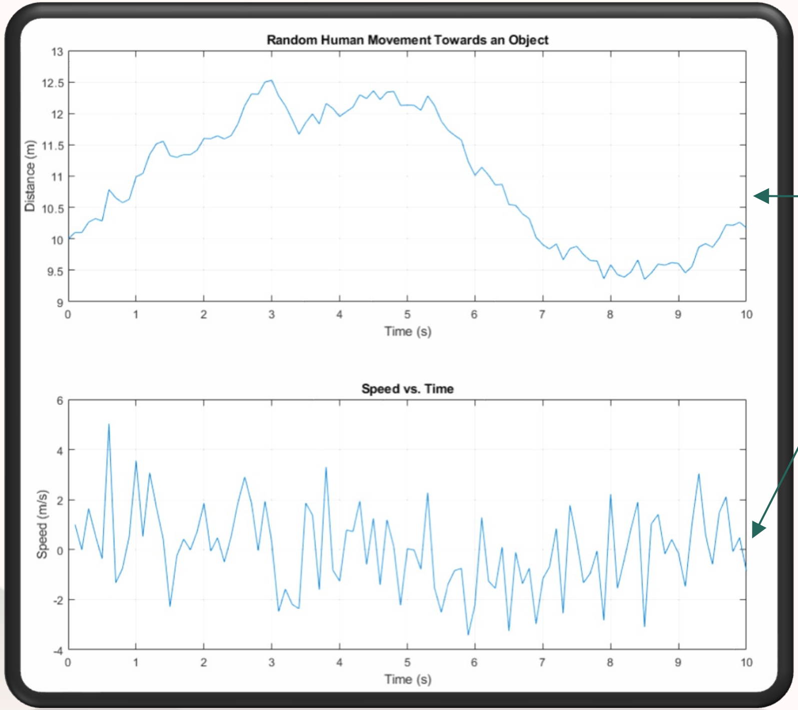


Distance v Time Graph

PSEUDOCODE

- Defines constants such as speedOfSound or frequency f0 (24e9).
- Calculates the position of the human position using the random walk values deltaDistance.
- Calculates the Doppler shift dopplerShift using the transmit frequency f0, the speed of sound speedOfSound, and the velocity of the human velocityHuman.
- Plots and saves the distance data to a CSV file.

Iteration 3: Calculate Speed



- Take lower sample sets for less noise resulting is less input but more accurate result.

Distance v Time Graph

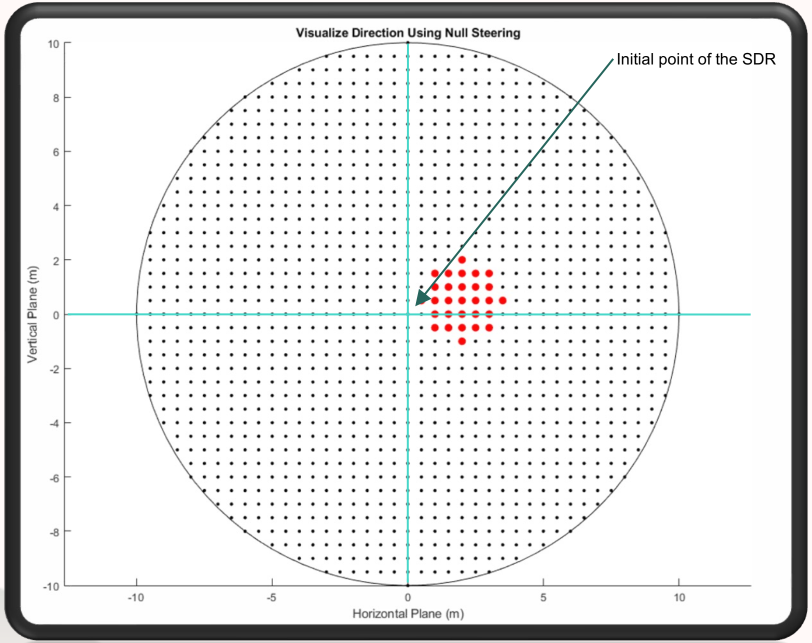
Speed v Time Graph

$\text{Speed} = \int \vec{s}\, dt$ where s is distance

PSEUDOCODE

- `Set constants like speed of sound and initial distance.`
- `Find speed by taking the derivative of the position over time.`
- `Calculate Doppler shift based on speed and constants.`
- `Plot position and speed graphs and store to CSV.`

Iteration 4: Pinpoint Direction



→ Detected object

→ No data (No object)

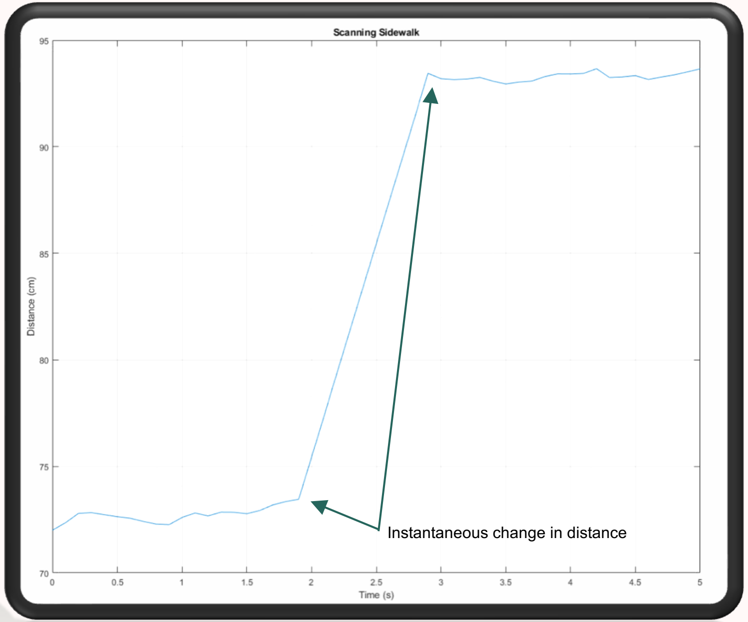
→ The field of view for the SDR device

- All values are treated as Boolean (T/F)

PSEUDOCODE

- `Generate spatial data points within the SDR's FOV.`
- `Emphasize the detected objects its immediate vicinity.`
- `Construct a graphical representation.`
- `Persist the highlighted data as a CSV file.`

Simulating Edge Cases

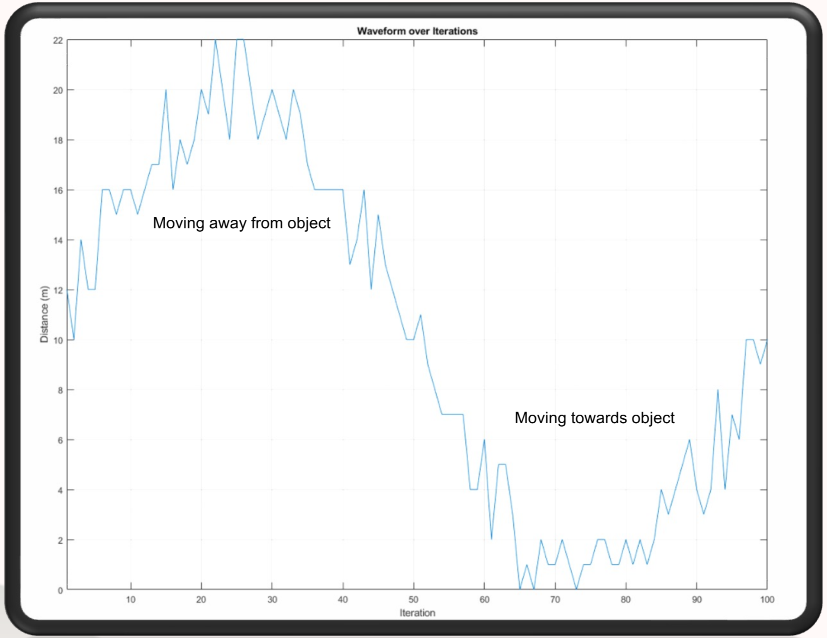


- Handle edge case, such as a drop at the edge of a sidewalk.

PSEUDOCODE

- `Initializes a random walk for distance deltaDistance with added noise.`
- `Introduces a single outlier for a limited duration to the delta distance deltaDistance.`
- `Calculates the position of the human using the random walk values deltaDistance.`
- `Plots the distance between the human and the object as a function of time.`
- `Saves the distance data to a CSV file.`

Alternate Simulation with Waveform

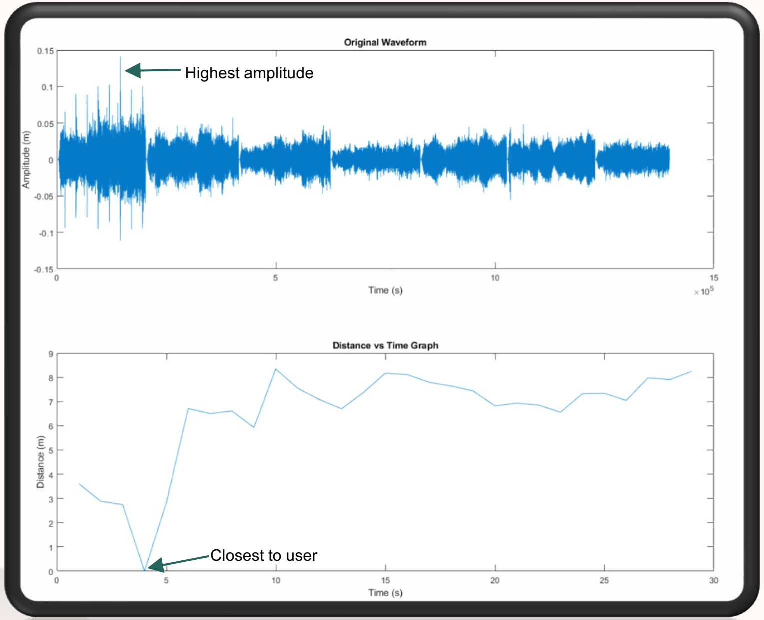


- One iteration of waveform detection every second.
- Calculate distance of object based on response.

PSEUDOCODE

- Simulates the movement of an object by generating a sinusoidal delay with added randomness.
- Cross-correlates the reference and received waveforms.
- Finds the lag value at the max correlation value, which is the estimated delay.
- Repeats steps 2-6 for a specified number of iterations.
- Plost and then saves the estimated delays to a CSV file.

Alternate Simulation with MP3



- Collected four second audio clips of the source getting farther from source.
- Merge all clips (PY), calculated amplitude.
- Use amplitude to calculate distance v time.

** This test case assumes all variables are optimal.*

PSEUDOCODE

- Reads the MP3 file and extracts the waveform.
- Denoises the waveform using a median filter.
- Calculates distance to the object by finding the maximum amplitude of the waveform and mapping it.
- Plots the waveform and distance over time.
- Saves the distance data to a CSV file.

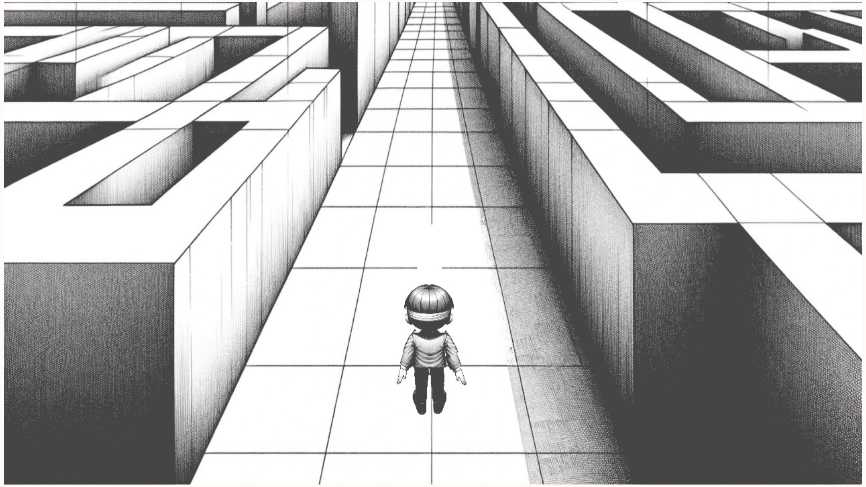
How we recorded data for MP3 graph

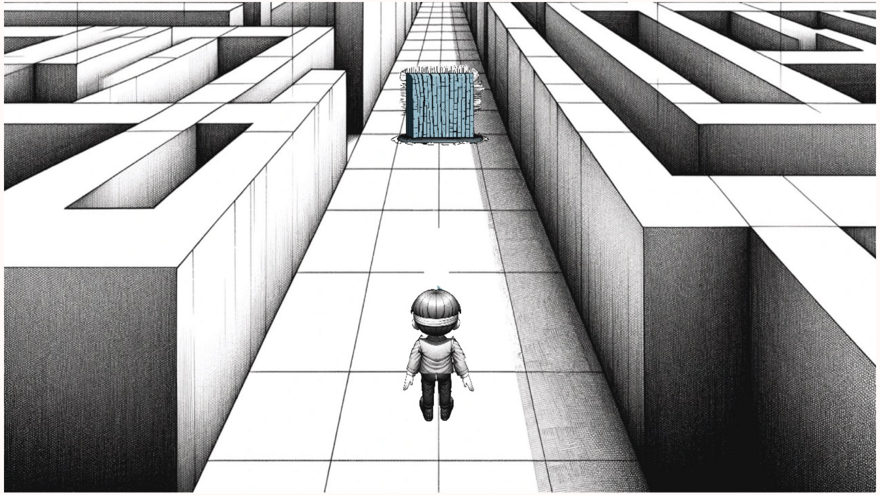
Trial 1

Trial 2

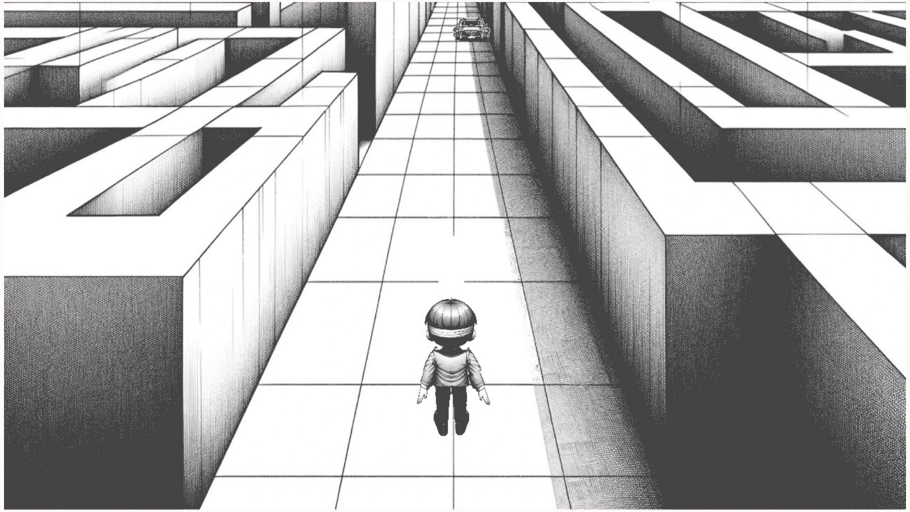
Visualization

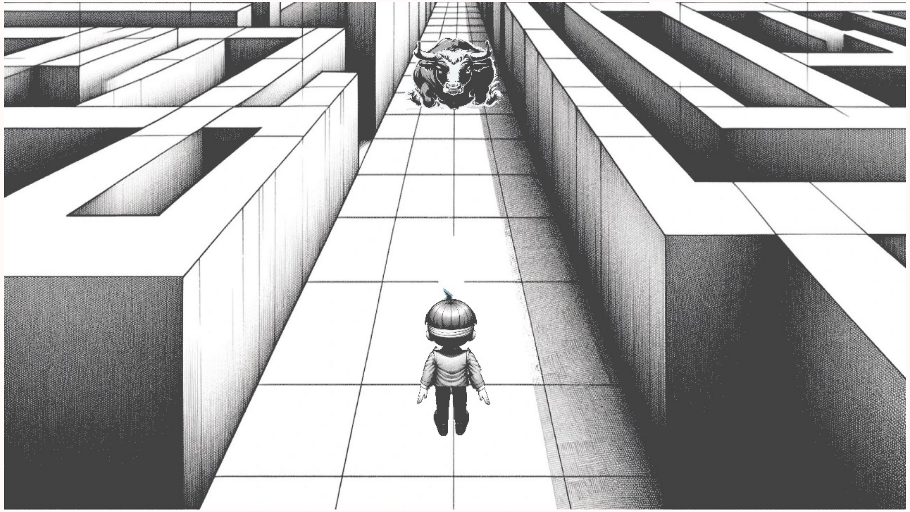
Static Visualization





Dynamic Visualization





Graph CSV Example Datasets

Waveform	
Lag (ms)	Correlation
0	0.723623032
80	1

Distance v Time	
Time (s)	Distance (m)
0.01	9.9759
9.96	10.389

Speed v Time	
Time (s)	Speed (m/s)
0.1	10.101
9.9	10.263

Null Steering	
X (m)	Y (m)
1	0
570	1

Edge Case	
Time (s)	Distance (cm)
2	75.449
3	93.201

Waveform Sim	
Iteration	Distance (m)
2	10
99	9

MP3 Sim		
Time	Amplitude(m)	Distance (m)
1	-5.5156e-06	3.6132
29	1.5764e-05	8.2497