Computer Science Behind Our Prototype

Autonomous Walking Group

Rick van Oosterhout

Domain

The target group of our project is the **visually impaired**. Our goal is to create an assistive device that allows them to move freely in indoor environments. The most important thing to enable this is **obstacle detection**. The importance of this was indicated by our stakeholders, discovered during 'deprivational' testing and through user interviews. Another clear example of this is the white cane, which is the most used assistive device among the visually impaired. **So, the focus of our project was to enable visually impaired users to navigate an indoor environment independently, without bumping into any obstacles or objects.**

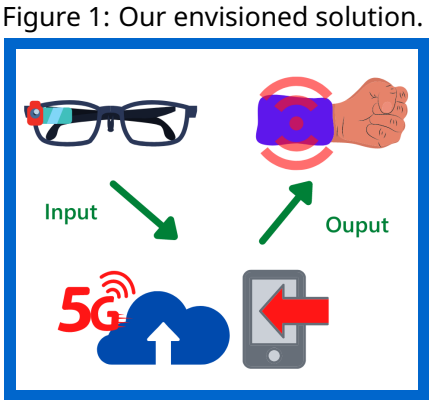
Relevant Computer Science

Multiple aspects of Computer Science could be relevant for this project, depending on the approach and the type of prototype. For our approach, it involved three main areas, **Computer Vision**, **Artificial Intelligence** and **Networking**. Where, my main concern was creating a computer vision model that would generate the feedback for the user. Later, **Reverse Engineering** was required, because the software of the sleeve was not made for Windows.

Envisioned Solution

Our initial vision of the solution consists of a set of **glasses**, with an integrated camera and controller, and a **vibrating wristband** as an output device. The processing of the camera images would be offloaded using **5G networks**. This solution is shown as a diagram in Figure 1.

Figure 1: Our envisioned solution.



Rationale

We decided on glasses because they are **inconspicuous** and glasses with cameras already exist. Offloading would help in keeping the device compact, since it would require less processing power (thus a smaller cpu and battery), which is required to make the device **inconspicuous**. The offload using 5G should maintain a **low latency**, which is crucial for an assistive device, because the feedback must be near instantaneous. Lastly, the output device should use small vibration devices (e.g. a wristband). The reason for this is that the user should not suffer a **sensory overload** when using the device, which would be the case if audio feedback was used. Again, this device should be small to ensure that the complete system is **inconspicuous**.

Challenges

Obviously, the envisioned solution is a good focus point. However, with the resources and time given, we could not realize this vision. So, the goal of the project was to create a prototype based on our envisioned solution. While working toward this prototype, we encountered **multiple challenges**, some of them were:

- Working with **large existing software and documentation**; The camera documentation was complex, had many different pages and topics, but was not quite clear
- Creating a reliable **AI-based model**; More about this in the next section
- Obtaining **resources**; It took much longer to obtain the sleeve than we initially agreed on; We were able to use the sleeve for the last 2 weeks
- Working with the **sleeve**; the software that came with it was not open source or made for Windows. This required some reverse engineering to integrate the sleeve in the system

Innovation Space

Initial Model

Our initial prototype was an **AI** model that mainly focused on object recognition. We expected that this would allow the system to obtain a **lot of information** about the environment, which would be useful for the user. This model was created using the **Oak-D Lite** stereo depth camera, this device was **easy to obtain** from our stakeholders and therefore allowed us to start working on the model as soon as possible. The model tried to **recognize objects** on the screen and estimate the **relative position in 3D**. However, it was difficult to get reliable detection for objects that were only partially visible. So, the **critical flaw** of this model was that many objects close to the user would not be classified as an obstacle.

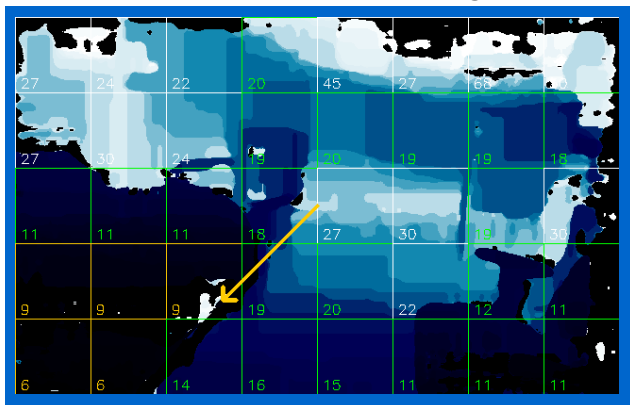
Prototype

Our final prototype does not use **Artificial Intelligence**, since we needed a reliable model and the critical flaw (mentioned in the previous section) was detected with limited time remaining. This would not allow a new AI to be trained and tested properly. Instead, the model uses **statistics** to determine the obstacles and dangers around the user. When an obstacle is close, the sleeve will vibrate. The location of the vibration is a **mapping** of the position in the camera's view. So, the feedback provides information about the horizontal and vertical location. The **intensity** is provided by the frequency and intensity of the vibration. In the end, this new model is merely a **classification algorithm** that uses depth information as **input features** and will provide a combination of direction and intensity as output.

Detailed Workings

The visualization in Figure 2 shows a visualization of the processing that takes place. The system uses two separate camera sensors to mimic **human binocular vision**. The **VPU** (Vision Processing Unit) of the camera does the **disparity matching** to estimate the depth of all pixels in the frame. Next, the individual pixels are combined into groups to create a **grid**. For each cell in the grid, we drop the **invalid measurements**, these are either produced by noise or objects that are only visible to one of the two cameras. The depth estimation is given in millimeters for each pixel, as a result you have a lot of data for your model. With all these individual pixels, it is difficult to create a classification. Therefore, we apply **feature reduction** by **aggregating** the pixels for each cell into a set of bins. These bins are groups of depths, so for instance all depth values between 0mm and 124mm are in bin 0, between 125mm and 249mm in bin 1, etc. For each cell, we take the **largest bin** and return the value of this bin. For each cell, we compare the value of the largest bin with predefined **threshold values**, and based on these thresholds we assign an intensity (*OFF*, *SOFT*, *MEDIUM* or *INTENSE*) to each cell. Lastly, we look for the region in the camera view with the highest intensity. This is done by computing the **mean** for each horizontal region (left, center, right) and vertical region (top, center, bottom) separately. Next, we define the classification of the direction, by selecting the **largest mean** of both the horizontal and vertical regions. The direction is combined with the **maximum intensity** value to form the final classification and output signal.
All the settings of the model (bin size, thresholds, number of rows, number of columns, etc.) can easily be adjusted in the model. This was very useful during the limited testing that we were able to do. Besides, this also allows for **fine-tuning** in the future to improve the model even further. For our prototype, we tuned these values such that they worked well during testing. In the end, these values work for the model, since we did not have any issues that were caused by bad model settings during our demonstrations.

Figure 2: Visualization of the prototype depth model. Darker pixels are closer than lighter pixels and the numeric values indicate the largest bin of each cell.



Alternatives

Instead of a depth based model, we believe that a model that recreates the environment as a **3D model** would be more suited. Mainly because the prototype does not provide reliable feedback for obstacles that are below knee height. This 3D model should check whether stepping forward is possible and if not in what direction it would be possible. This information can then be provided using a mapping similar to that of our prototype. The **downside** to this alternative is that it would be much more intensive and would require a different camera setup. This is also the power of our prototype, it is very **undemanding** and could easily be optimized to be even less demanding.