ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

ASSISTIVE TECHNOLOGIES CHALLENGE
P4 : LEFT SIDE WARNING

GOLFIER Ewan, LI Changling, SILVEIRA Joaquim & STEGMÜLLER Noah

Project supervision: IJSPEERT Auke, HUGHES Josie, MICERA Silvestro & SHAFIEE Milad

Date: 29th September 2023

Contents

1 Introduction

In the context of the Assistive Technologies Challenge 2023, this MAKE interdisciplinary project is organised by the Swiss association Hackahealth where the aim is to develop a personnalized solution to assist impairment of a real patient (hereinafter referred to as "the Challenger") by tackling one of his daily challenge. Its development is always carried out in ongoing collaboration with the Challenger who is the primary user of the final product. The solution development method follows the principles of the V-model, to try to meet the needs of the Challenger.
In this specific project, the aim is to help the person move around in a wheelchair by creating a device that allows to detect potential obstacles. This report presents this project by first providing the motivation, followed by the existing solutions and our final solution in a way so that one can reproduce it easily, from the manufacturing to the algorithm developed for the obstacle detection. Finally, further improvements are presented along with the conclusion.

2 Challenger presentation and motivation

2.1 Challenger presentation

Since the Challenger has an uncommon condition, naturally the first step is to understand it before tackling the project. Indeed, the Challenger suffers from a neuropsychological condition called spatial neglect. This is typically caused by a stroke, damaging mainly the right cerebral hemisphere [1]. This is a complex disorder, as it manifests itself quite differently in each patient [2]. But in most cases, a patient with spatial neglect will behave as if the left side of the sensory space does not exist. For example, they could ignore the left side of their body, despite the stimulations. An extreme case would be that a patient might not shave the left side of the beard, or does not eat the left half of their plate while complaining of being hungry. Most commonly, they will neglect the left side of the visual field. So they may frequently collide with objects or structures on their way[1]. This is the case and the main issue of the Challenger on which the project is centered.

2.2 Motivation

The Challenger aspires to take daily walks on their own while using their electrical wheelchair. Unfortunately, the chances of collision in an unpredictable outside world is fairly high when one tends to neglect what is happening on the visual left side. Plus, the frequent road works in the neighbourhood add some difficulty to the walks. This is why, as mentioned in the introduction, the aim of this project is to help the Challenger to move around in a wheelchair by creating a device that allows to detect potential obstacles. Due to this condition, the therapist also participated throughout the development the project, giving some advice or helping to define precisely the needs of the Challenger. For example, the device should warn the Challenger of the obstacles on the left in order to pay attention to them, instead of trying to control the wheelchair as one could imagine at the beginning of this project. The idea for the Challenger is to let as much as possible theirs independence and autonomy. Moreover, additional to the obstacle detection, the Challenger should also be reminded from time to time to pay attention to the left even when there is no obstacle, to try to mimic a healthy person who constantly pays attention to the whole visual field during a walk. This could also eventually help him for neuro-rehabilitation in the long term. Finally, the device should not be intrusive to the normal usage of the wheelchair and its integration to it should be reasonably discreet for the Challenger to avoid attracting to much attention from other people during the walks.

Therefore the aim is to find a way to detect the obstacles and warn the Challenger of them by an auditory feedback, with some reminders. The Challenger will then make their own decision-making process on how to react once the obstacle is acknowledged. Also, the device should be well integrated to the wheelchair while avoiding any disruption to its normal usage and be easy to use.

3 Existing solutions

There are currently no existing solutions commercially available for this specific requirement.
Nevertheless, our baseline was the HackHealth prototype of 2022 which as shown below :

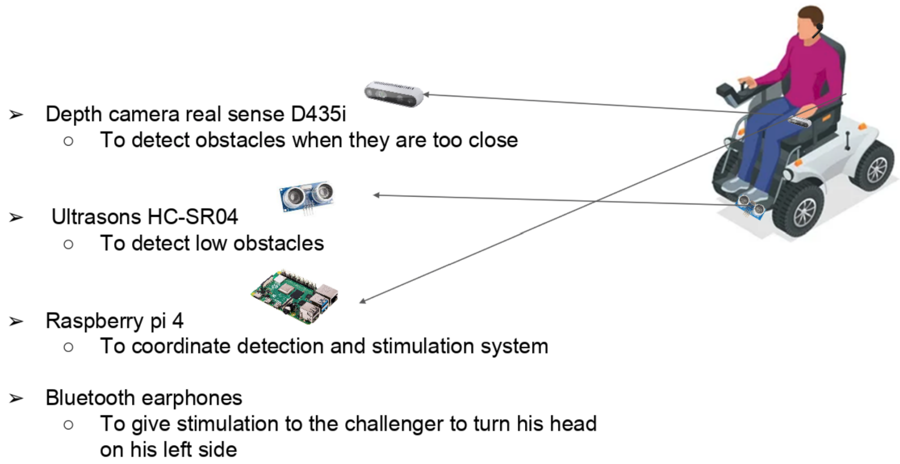

Figure 1
2022 HackHealth prototype [3]

This prototype is similar to the new one developed during this project as the main ideas and components were used. Some improvements on the components were made. The camera was replaced by a similar one but with IP65 rating for water resistance. The Raspberry Pi was replaced by a Jetson Nano for more computation power.
Also, the ultrasonic sensors were not included in the final project for several reasons : 1) the difficulty to fix it to the wheelchair without disturbing the Challenger as they need to be low in front of the wheelchair, 2) the difficulty to make them water resistant as they would be completely exposed for a wide opening of the ultrasound passage, 3) the lack of time as it was limited for this device development; so a choice was made to first focus on the camera detection.

4 User requirements and system specifications (DIR)

As previously stated, our approach to defining, designing, and prototyping our solution was guided by the principles of the V-model. This methodology allowed us to maintain a clear and structured development process, ensuring that each stage was carefully planned and executed. We incorporated a traceability matrix and a risk assessment chart, which can be found in the Appendix. They provide an overview of the project's progression and the mitigation strategies implemented for the identified risks.

5 Final solution and manufacturing

5.1 Components of the system

The final system consists of the following components :

- NVIDIA Jetson Nano
- Aftershockz air slate earphones for auditory feedback
- Realsense D457 depth camera

- Xiaomi Mi 3 Ultra Compact external Battery for power supply
- Button for User Interface
- LED for visual cue of the system's state
- Bluetooth USB dongle 4.0

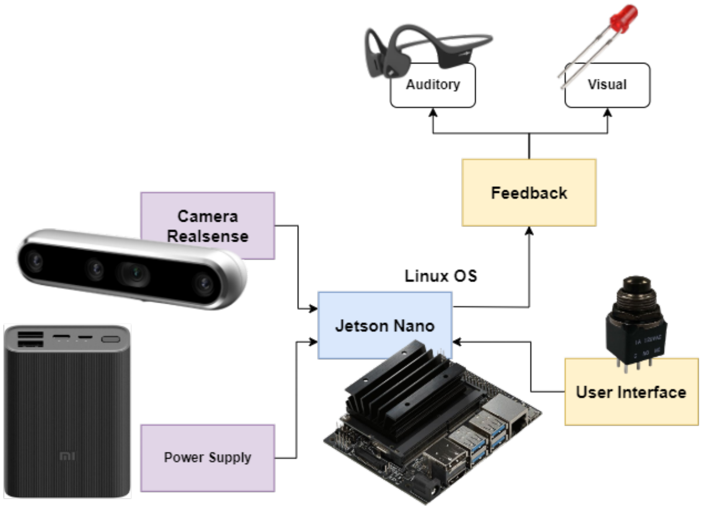

FIGURE 2
Components of the final system

These components need to be encapsulated to be protected from rain/dust and fixed to the wheelchair without being intrusive. They also have to to be well integrated to be sure that they won't affect the challenger daily usage of the wheelchair. To achieve this, the casing and fixation of the parts were designed to be as small as possible with only few cables and by placing them strategically on the wheelchair. This was thoughtfully discussed with the therapist to make sure that the Challenger won't notice the device at all once mounted on the wheelchair. All of this while keeping in mind that the components should stay fixed on the wheelchair without the hussle of having to remove/put back on it.

Finally, compared to the solution developed in 2022, an user interface is added. Indeed, the device should be operable only during the outside walks since the Challenger does not need the feedback when using the wheelchair at home. Also, the Challenger should know in an easy way if the detection is on or not. This is why an user interface that consists of a button and a LED are introduced in this project. The Challenger will only need to press the button to activate/deactivate the system, with the LED helps to remember the current state of the system. Also, a battery is added to make the device easily chargeable and autonomous during the walks.

5.2 POWER SUPPLY

One of the key requirements was to ensure the use of a separate power supply that remains isolated from the wheelchair's battery, preventing any potential interference. A power bank of $27.5Wh$ usable capacity was chosen as it is compact and not expensive. Since the Jetson nano is limited to a power of 5W, the system can theoretically be powered during over 5h which is way more than the initial 1h (mean walk time) requested by the Challenger.
The power bank is from a well known company and has low voltage (5V rated voltage). Electrical hazards are therefore completely mitigated.

5.3 Manufacturing

To simplify manufacturing and provide more design flexibility, it was decided to 3D print the device's casing and fixation system using PETG plastic. All the 3D printed parts and their main dimensions can be found in the Figure 3 with their description in Figure 4.

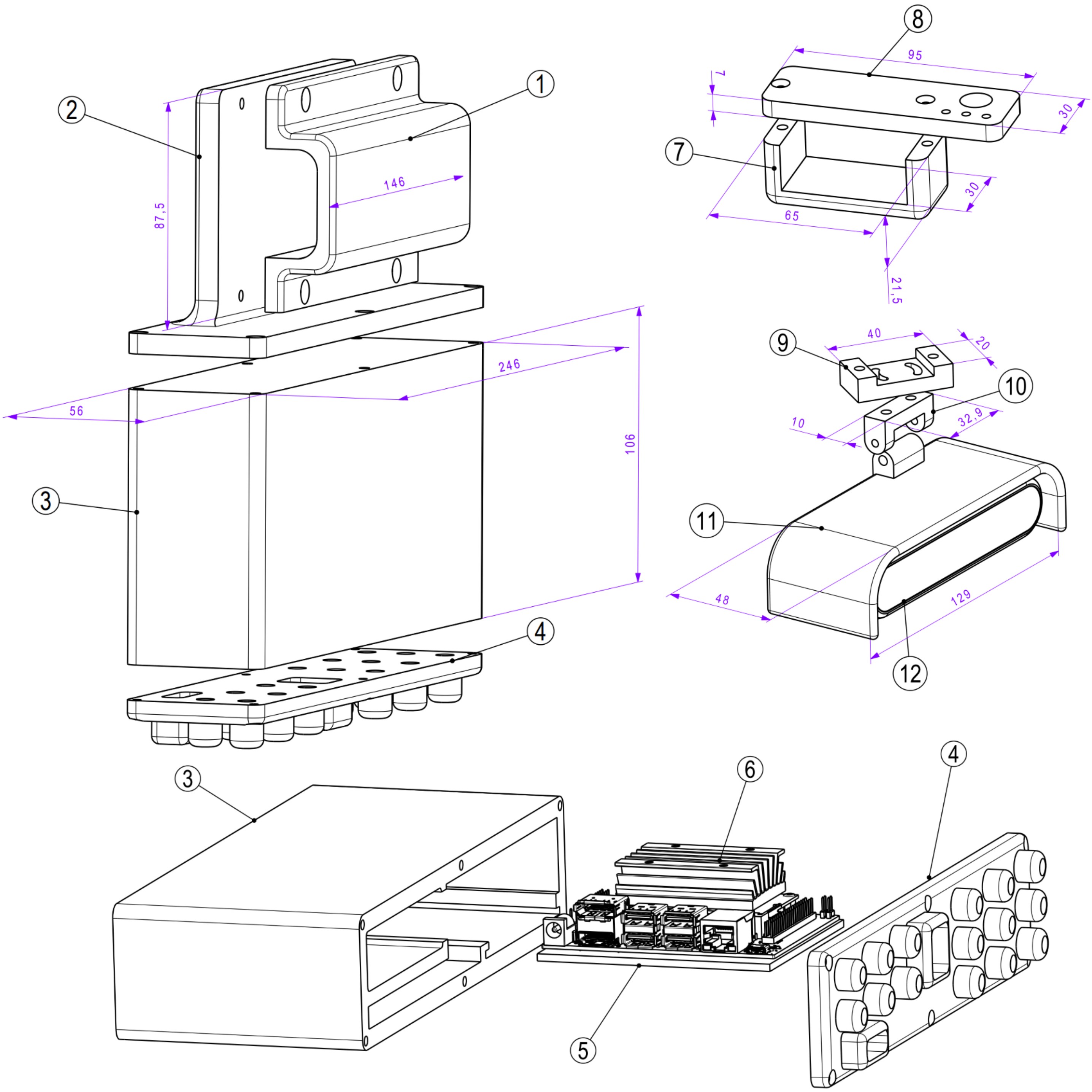

Figure 3
Design and main dimensions of 3D printed parts

Part number	Part name	Part description and specification
1	Box fixation 1	Together with part 2, it fixes the Box to a horizontal bar behind the wheelchair
2	Box fixation 2	Fixed on the top of the Box
3	Box	Contains the Jetson nano, the battery and cables : • Completely closed from the top (for water resistance) • Jetson nano with its support (part 5) can slide inside • Battery is tight in it's compartment (no movement possible)
4	Cover	Closes the Box : • Has holes for ventilation, cables and charger plug • Holes are extruded to keep rain from flowing inside
5	Jetson support	Jetson nano is fixed on top and slides in the Box
6	Jetson nano	Computer for obstacle detection algorithm
7	Button support 1	Together with part 8, fixes the button support to the right armrest
8	Button support 2	Support for button and 3 LED
9	Camera rotation	Is screwed to the left armrest and allows to set different camera rotation angle
10	Camera inclination	Allows to set different camera inclination angle
11	Camera holder	Holds and protects the camera
12	Camera (Realsense D457)	Depth camera

FIGURE 4
3D printed part name and description

All the printed parts on Figure 3 were assembled using standard M3 and M4 screws that can be find easily on the market.
Threaded inserts were used on part part 2,3,7,10 and 11 to allow to screw the parts together. One of this threaded insert can be visualised in Figure 5

FIGURE 5
Threaded insert that are used to screw the parts together

The printing time and plastic use are listed in the Figure 6 below.

Part number	Printing time	Plastic (PETG) used
1	12h17	129 g
2	15h27	155 g
3	~2 days	621 g
4	9h46	80 g
5	1h30	15 g
7;8	2h13	26 g
9;10;11	9h59	102 g

FIGURE 6
Printing time and plastic usage

5.4 INTEGRATION OF THE SYSTEM ON THE WHEELCHAIR

Some pictures were taken in order to better visualize how these components are placed on the wheelchair. Indeed, parts 1-4 (as shown in Figure 3 are mounted together and placed on the back of the wheelchair, using the horizontal bar as shown in Figure 8. Part 3 contains the Jetson Nano and the external battery.

Then, the button and the LED placed on part 8 are fixed on the front below the right arm rest of the wheelchair, with the help of part 7. This location was chosen so that the Challenger can easily control if the system is on when using the wheelchair by a quick visual check. It is also easier to reach to button there since the Challenger is already used to control the wheelchair with the joystick command on the right, as shown in Figure 9. A white multicore cable is mounted along with another already existing wheelchair's cable for the link between the Jetson Nano and the user interface.

Finally, parts 9-12 with the camera are fixed in front below the left arm rest, in a way to optimize the camera angle for the obstacle detection, but also to not disturb the Challenger. Another cable is also mounted against below the arm rest for the link between the Jetson Nano and the camera as shown in Figure 10. The camera angle is easily adjustable by tightening/untightening the screws on part 10.

6 COMPUTER VISION ALGORITHM

6.1 OVERVIEW

The computer vision algorithm uses the depth map of the camera, some image processing and the camera's depth information to detect if an object is less than 3 meters away and potentially in the Challenger's path. If it is, it will send an audio feedback to the Challenger's earbuds reminding him to look on his left. Furthermore, the algorithm periodically sends a reminder to the Challenger every 2 to 3 minutes, regardless of it detecting an obstacle or not. We programmed it in Python, using the OpenCV, Scikit-image, NumPy and PyRealsense2 libraries.

We chose to separate our code into 8 files :

- `main.py` : main loop, file to execute to start the algorithm
- `parameters.py` : contains all the adjustable parameters
- `constants.py` : contains all the non-adjustable parameters
- `filters.py` : implementations of different filters that we used or tested
- `display.py` : if using a computer screen, responsible for displaying relevant information

- `pipeline.py` : fetches the depth map and color frames from the camera and processes them
- `image_processing.py` : processes the depth map and color images to detect obstacles
- `reminder.py` : reminds the Challenger periodically or when an obstacle is detected

The code was made so that it could be used for its main application using the Jetson nano. However, during testing, we also needed to find a way to quickly debug our code by having the camera connect to a computer and display our computations. To accomplish this, we implemented a simple configuration mechanism where switching between modes is as easy as modifying a single variable in the `parameters.py` file. By changing the value of the `using_jetson` variable, we can seamlessly transition between different modes of operation. This helped with making sure what was displayed during testing would be exactly the same as the implementation on the Jetson nano, as we didn't have to change any of the code.

The detection algorithm mainly consists of the following steps, looping indefinitely :

1. Fetch the camera's depth frame and apply frame-level filters (if any)
2. Transform to depth image and apply image-level filters (includes a threshold)
3. Find connected blobs and their bounding boxes
4. Get the distance to each blob
5. If the bounding box intersects the detection zone and the blob is closer than 3 meters away :
   Send a reminder

In the code, you will see that we also computed the same steps in parallel for the color image. This was to test out some other solutions that would include computations on the color image, such as using an object recognition neural network. This also helped us with debugging, so we were able to see both the depth and the color outputs side-by-side on a screen.

You can find a small demo of the obstacle detection here : `https://youtu.be/zHOwot36nm0`

6.2 Obstacle detection

All of the parameters were calibrated with a lot of testing on our side and with direct feedback from the Challenger. The final parameters we used are available in the `parameters.py` file.

The first step to detecting obstacles was to apply a bilateral filter to the depth map. This approach helped reduce the noise generated by the camera, while simultaneously preserving the sharp edges in the captured images. As a result, we achieved a significant reduction in image noise without sacrificing the clarity and detail of the captured content. We then applied an additional small Gaussian blur to try and get rid of some of the artifacts created by the bilateral filter, such as gradient reversal (preserving false edges in the image).

To find the closest points in the scene, we then applied a threshold to the filtered depth image. As the camera always gives the maximum value (255) to the smallest distance in the depth map, our threshold essentially ensured that we were only keeping track of the closest points in front of the Challenger, whether they were close or far away.

We then found the close objects by applying a blob detection algorithm to the thresholded image. This was done by using Scikit-image's `measure.label` function, which labels all the connected pixels in an image with the same number. We then went through each connected blob in the image and only kept the ones that were big enough by pixel count. This ensured we got rid of more noise that would still have been present.

Then, we computed the bounding box for each blob, and checked if it intersected with our detection zone. The detection zone was a strategically positioned bounding box which ensured that only objects within the intended area of interest were detected. By carefully defining this zone, we were able to exclude irrelevant regions from the camera view, such as the sky or areas outside the Challenger's intended path. We also ignored most of the right side of the camera's field-of-view, as the Challenger would be able to see it without the help of our system. This approach enhanced the accuracy and efficiency of the detection system by focusing mainly on the relevant objects within the specified boundaries.

Finally, if a blob intersected the detection zone, we used the camera's depth sensors to get the distance to each of its pixels and averaged them to get the distance to the object. If the object was closer than 3 meters away from the camera, we sent a reminder to the Challenger.

6.3 Reminders

With instructions from his therapist, we implemented a periodic reminder to pay attention to his left side in random intervals of 2 to 3 minutes starting from the last audio feedback. If no detection occurred, the Challenger would still be reminded. This could eventually help him as a neuro-rehabilitation in the long term, but it is also good for safety reasons, as he would pay attention to his left side more often. To do that, we used threads with timers and a callback function. This enabled us to not have to add too much extra computation for something that would rarely occur.

To avoid the audio feedback from becoming too frequent and therefore annoying, we decided to implement a minimum delay of 30 seconds after each reminder. During this delay, no audio could be played, meaning that the Challenger wouldn't be constantly getting reminders if there was an obstacle in front of him for more than a few seconds. It also didn't make sense to remind him too often, as our Challenger could pay attention to his left side for a certain period of time when reminded to look out.

For each detection, we also chose a random audio file to play from 10 different voice reminders to avoid making the experience annoying and repetitive.

6.4 Shortcomings

Our detection algorithm has some problems, which are mainly to do with the Realsense camera that uses both stereo vision and an infrared emitter/receptor. As described in this paper [4], 3D cameras suffer from a big range of issues. In particular, we suspect 3 of these to be responsible for most of our detection inaccuracies :

- Occlusions : When some parts of one camera are hidden in the other. This creates holes in the depth map, making finding the relevant objects more difficult.
- Specularities : Shiny parts of a surface. Fools stereo vision by interpreting a point on one camera to be in the wrong position on the epipolar line of the other camera. Specular surfaces can also reflect infrared light, which means both of these phenomenons can give wrong values for some points in the depth map.
- Ambient light saturation : Saturation of the infrared sensor due to natural light. This uniformly increases the computed distances, rendering them meaningless, as we need an absolute distance metric.

We also noticed that we ended up having a lot of false negatives. This was in part due to the Challenger's feedback, as he found it too annoying when there was a lot of false positives. Having too many false positives also basically meant that it would have ended up just being a periodic reminder on a timer. We therefore ended up cropping some of the image, adding in the detection zone, and tweaking the parameters to make the detection more precise. Most of the reasons for false negatives came from the distance measures being wrong. We noticed that our image processing was detecting the objects and assigning them the correct bounding box, but the camera's computed distances often ended up longer than in reality.

Often times, we noticed the closest points in the depth map simply ended up being floor. For most objects, there exist multiple points on the floor which are closer to the camera. This means that we usually always have some kind of floor object in our blob detection. Our solution was to just tilt the camera a little upwards, but this is not ideal, as we would still like to detect small objects that are on the ground.

7 JETSON NANO SETUP

The Python script is designed to run on the NVIDIA Jetson Nano SBC (Single Board Computer), known for its superior performance in Computer Vision and Deep Learning applications compared to the Raspberry Pi. Setting up the Jetson Nano is a straightforward process:

1. Begin by following the *Getting Started*[5] guide for the Jetson Nano, which provides step-by-step instructions for initial setup and configuration.
2. Install the Pyrealsense2 library by following the provided guide[6]. This library is essential for accessing the features and functionality of the Intel RealSense camera.
3. Install the required Python packages, including Scikit-Image, Numpy, and CHECK. These packages are necessary for image processing and data manipulation.
4. Install VLC for audio playback if it's not already installed on your system.
5. Clone the code repository containing the script from the provided GitHub link. This allows you to access and use the Python script for your application.
6. Configure the Linux environment to automatically execute the Python script upon boot-up[7]. This ensures that the script runs seamlessly without manual intervention.

The audio playback is done by VLC, so make sure to have it installed. To turn on and off the Jetson Nano with the button instead of it auto-booting each time, connect the 5th and 6th pins on the Button Header [J12] with a jumper, and connect the button to the 11th and 12th pins, also on [J12], as shown in Fig.7.

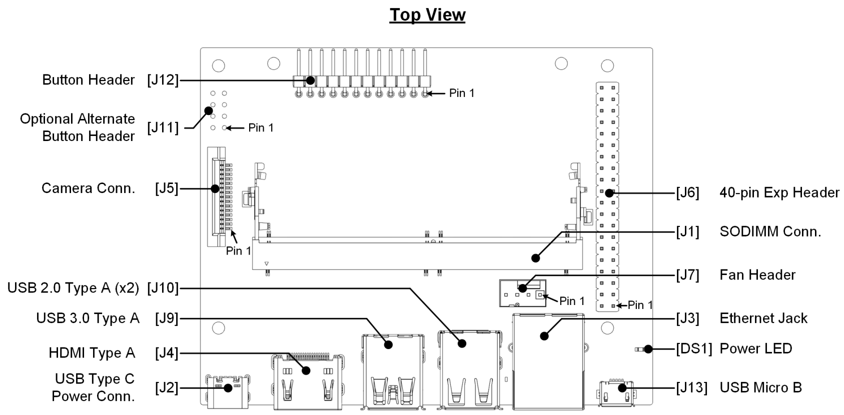

FIGURE 7
Top-view of the Jetson nano[8].

8 FURTHER IMPROVEMENTS

The Challenger wasn't able to use the earphones independently, as the buttons were too small and theirs mobility didn't permit to put them without any help from another person. This was an oversight on our part that would need to be fixed in future work.

Many improvements can still be made to the computer vision algorithm, such as trying to find further solutions to some of our shortcomings described in 6.4 and the inherent issues with RGB-D cameras described in [4]. Furthermore, it would be necessary for a safer experience to find a way of detecting holes and sidewalks. There may also be better ways to improve the image processing, such as introducing machine learning and/or spatio-temporal techniques to segment the objects.

We briefly considered and implemented a pre-trained neural network which recognized and labeled objects, but after discussion with the Challenger and his therapist, we realized that it would be too much information and probably unnecessary. However, there may be a way to track objects and/or help with the object segmentation by using a neural network on the color images.

We still think it would be a good idea to reintroduce the ultrasonic sensor array as a second object detection signal. However some major problems would need to be solved, such as the integration on the wheelchair and the prioritisation of one signal over the other. We also didn't have time to test if our button and LED module were water-resistant.

9 Conclusion

In conclusion, this project has successfully developed a personalized assistive device to aid a Challenger with spatial neglect in navigating their daily environment using a wheelchair. The device has been design to detect potential obstacles and provide auditory feedback to the Challenger. The solution has been developed following the V-model and in close collaboration with the Challenger and their therapist, ensuring that it meets their specific needs and enhances their independence and autonomy.

Despite the success of the project, there are areas for further improvement. The User Interface could be made more accessible, and the computer vision algorithm could be refined to address certain shortcomings and enhance the object detection. The potential integration of an ultrasonic sensor array and the use of machine learning techniques for object segmentation could also be explored in future work.

References

[1] T. J. Schmitz S. B. O'Sullivan (Eds.) 'Cognitive and Perceptual Dysfunction'. In: *Physical Rehabilitation*. Philadelphia, F.A: Davis Company: Unsworth, C. A., 2007, pp. 1149–1185.

[2] Mark D'Esposito. 'Neurological foundations of cognitive neuroscience'. In: ed. by Bradford Books. MIT Press, 2003, pp. 1–19.

[3] HackHealth 2022 team. *Presentation slides from HackHealth prototype*. 2022.

[4] Ayush Bhandari Achuta Kadambi and Ramesh Raskar. *3D Depth Cameras in Vision: Benefits and Limitations of the Hardware*. 2014. URL: `https://web.media.mit.edu/~achoo/tr/3d_benefits_limits.pdf`.

[5] NVIDIA. *Getting Started With Jetson Nano Developer Kit*. `https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit`. Accessed: 2023-06-08. 2023.

[6] Lieu Zhenghong. *How to install librealsense and pyrealsense2 on the Jetson NX*. `https://www.lieuzhenghong.com/how_to_install_librealsense_on_the_jetson_nx/`. Accessed: 2023-06-08. 2020.

[7] Linux Hint. *How to Configure Services to Start Automatically at Boot*. `https://linuxhint.com/lad-configure-services-to-start-automatically-at-boot/`. Accessed: 2023-06-08. 2023.

[8] NVIDIA. *Jetson Nano 2GB Developer Kit User Guide*. Accessed: 2023-06-08. 2020. URL: `https://developer.nvidia.com/embedded/learn/jetson-nano-2gb-devkit-user-guide#id-.JetsonNano2GBDeveloperKitUserGuidevbatuu_v1.0-12pin`.

Appendix

Obstacle detection demo : `https://youtu.be/zHOwot36nm0`

Short video about our project : `https://youtu.be/V3-m3lgVwH4`

GitHub repository : `https://github.com/Bwananah/ATC`

FIGURE 8
Back of the wheelchair with the parts 1-4 encapsulating the Jetson Nano and the power Supply

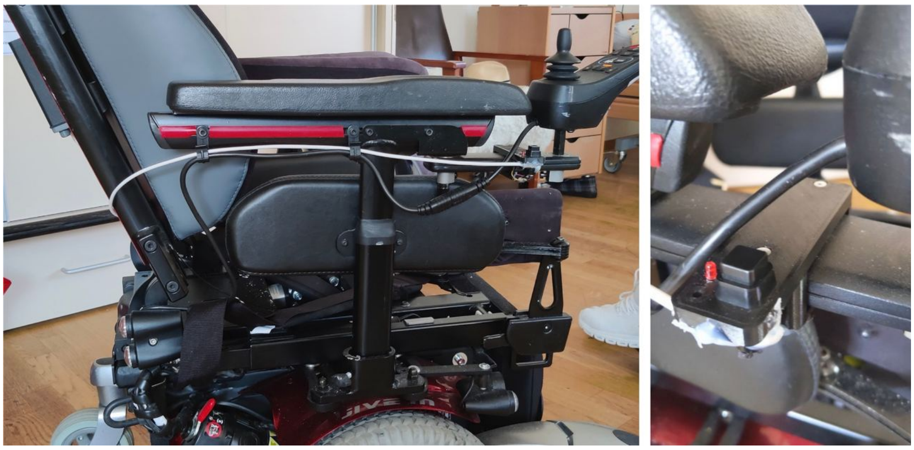

FIGURE 9
Right arm rest of the wheelchair with the white cable for support for button and LED and parts 7-8

FIGURE 10
Left arm rest of the wheelchair with the camera fixed on the front below of the arm rest with the help of the parts 9-12

Note: Orange text is to be replaced or deleted!

User Requirement Specifications (Columns due 16/03/23)			Design Input Requirements (Columns due 06/04/23)		Verification (Columns due 18/05/23)				Validation (Columns due 25/05/23)			
URS ID	URS	Importance [MUST / NICE TO HAVE]	DIR ID	DIR	Plan/method	Report result	Pass? [Y/N]	Comment	Plan / method	Report / result	Pass ? [Y/N]	Comment
URS-01	User wants the device to be water-resistant	MUST	DIR-01	The prototype shall be protected from rain by its casing and therefore IP65 protected (at least from the top).	Use water resistant hardware and test the casing by putting it under a strong shower stream for several minutes. (The stream hits the top and sides of the casing but not the bottom)	The camera has already a IP65 rating. The casing did resist the shower test, no water went inside, however as it cannot be splashed from under, it can only be considered IP65 protected from the top. The main components are rain resistant.	Y/N	Since the casing is completely closed from the top, no dust can penetrate from there. No water or dust test was done on the start button.	No tests on the whole device have been done.	No tests on the whole device have been done.	N	The only test was to put the casing under the shower (see verification).
URS-02	User wants the device to have good cabling	MUST	DIR-02	The prototype shall have the least amount of cables as possible and be attached together	All the internal cabling is hidden inside the main casing. The two external components (Camera and User Interface) have only one cable each that are attached to the chair.	Prototype is non intrusive cable-wise. Cables are tied to the chair an do not stick out.	Y	It is not possible to have even less external cables.	Observe if the cables seem to fit in the wheelchair, and does not bother the challenger during his daily routine.	Challenger and therapist were very satisfied on how the cables were fitting on the wheelchair.	Y	The therapist observed that the cables did not disrupt challenger's daily activities.
URS-03	User wants the feedback to be non-visual	MUST	DIR-03	The prototype's feedback shall be auditory, using earbuds.	Use the same auditory system as last year.	Prototype has bluetooth compatibility issues, it works with normal headphones but not yet with the vibratory earbuds of last year.	Open	Should be ready for final presentation.	Test if feedback is auditif and works.	Feedback works with the wanted earbuds in the final prototype.	Y	Therapist comment : Good balance between reminder and obstacle detection.
URS-04	User wants enough battery life for his daily outings	MUST	DIR-04	The prototype's power supply shall last at least 1h. The battery capacity should be at least 5 Wh (main computer consumes 5W).	Use the main computer in Power Efficiency mode on a challenger's daily walk to test the battery usage.	The battery capacity is 27.5 Wh, estimated autonomy = 5h. Prototype survived more than an hour on without any issue.	Y	Battery capacity is more than enough for the normal daily usage.	Test battery life with challenger over an hour.	There was still a lot of battery life after the test.	Y	Theorically, it should last 5 hours.
URS-05	User wants the feedback to be on his/her right.	MUST	DIR-05	The auditory feedback shall be on his right ear.	Test if feedback comes out of earphones.	The auditory feedback comes out on both sides of the earphones.	Y	User requested after discussion to use both earbuds.	Auditive test	Feedback comes out of both ears.	Y	Challenger finally prefers to have feedback on both ears. Therapist comment : The only issue is that the challenger is not autonomous to put on the earbuds (needs
URS-06	The prototype shall not be intrusive with the wheelchair	MUST	DIR-06	The prototype's cabling and casing shall not interfere with the wheelchair's nominal operations.	Prototype has only two external cables, one main casing behind the chair and a camera and a button that are attached on the arm sides.	The casing, cables, camera and button did not interfere with the wheelchair or its use	Y		Observe if the whole device fits in the wheelchair and don't go outside the wheelchair's limit.	Device does not bother the challenger at all.	Y	Therapist comment : It was not intrusive to the daily usage of the wheelchair. The wheelchair control joystick was still accessible.
URS-07	The prototype must be powered independently from the wheelchair	MUST	DIR-07	The prototype shall have it's independant and sage power supply.	Test main computer with a battery, implement the battery inside the main casing and verify overheating.	The device worked perfectly while being supplied from the battery. The battery is easy to charge, small, does not overheat and has no inflammable risk (low voltage and power).	Y		No test was done since the device has its own battery.		Y	

URS-08	The prototype must not disturb the Challenger's normal usage of the wheelchair	MUST	DIR-08	The prototype's module placements shall not exceed the original dimensions of the wheelchair or interfere the users movements.	Test of the placement of each component on the user's wheelchair and measure by how much they exceed the wheelchair's original dimensions.	The camera exceeds the wheelchair dimensions by 2-3cm on the left. All other components are well hidden and do not exceed the wheelchair's dimensions.	Y/N	Only camera sticks out a little bit.	Challenger tested the device during a walk outside in his wheelchair.	It does not bother the challenger at all during his routine.	Y	Therapist and challenger were very satisfied with the implementation of the device. Going in and out was as easy as before. The challenger could also sit at the table as before.
URS-09	The user wants to be reminded of the obstacles on the front and left sides	MUST	DIR-09	The prototype shall detect the closest object and warn the user when it is less than 2 m away.	Test the detection outside with a wheelchair using the software and the camera.	The computer vision algorithm has a non-negligeable amount of False Positives, but never False Negatives.	Y/N	The CV algorithm can be further improved, it has problems with shiny surfaces as well as textures that repeat at a very high frequency (known problem with stereo cameras).	Test during the walk outside as mentionned above. Challenger warns us each time he hears the feedback.	Works on static obstacles quite well, although at a closer distances than expected. Moving obstacles are detected if coming from in front or from behind, but the CV algorithm struggles when coming from the sides. False positives and negatives can happen.	Y/N	The challenger is happy with the periodic reminders. Therapist comment : There is a good compromise between reminder and obstacle detection, although the detection still needs improvement.
URS-10	The feedback must be able to adapt to the environment (volume, etc...)	NICE TO HAVE	DIR-10	The volume of the auditory feedback must adapt to the environment's noise volume automatically.	NOT DONE	NOT DONE	N	Not implemented in the algorithm. But the user can ajdust the volume of the feedback on the earphones.	Not implemented	-	N	Volume can be changed manually with the buttons on the earbuds. But the buttons are quite small and difficult to reach for the challenger.

Note: Everytime you make significant changes to this document (e.g. for traceability matrix deliverables) you should increase the versioning of the file from v1.0 to v2.0 etc. For drafts for a deliverable you could use v0.1 to v0.2. Discuss the best approach with your team.

Note: Orange text is to be replaced or deleted!

Risk identification, evaluation and control (Columns due 06/04/23)							Risk verification (Columns due 25/05/23)	
Risk ID	**Hazard**	**Risk**	**Severity ("S")**	**Occurrence ("O")**	**Risk ("R")**	**Risk control measure**	**mitigate**	**Comment**
R1	Electrical energy	Prototype has its own power source. Risk of electrical hazard	3	1	3	- Only low voltage power supply (5V), encapsulated and protected wires and electronics. - No Homemade battery. Only standard of the shelf component.	Yes	Battery was commercially available. The whole system has a very low voltage. Only 2 protected cables are outside (one for the camera, the other for the on/off button)
R2	Prototype Overheat	Computational intensive algorithm can lead to overheat of the main computer.	1	3	3	-Add a heatksink and expose it to the environment -Limit the Power consumption of the main computer - Place the main computer out of the user's reach during nominal operations	Yes	Power consumption of main computer was limited. During the tests, no heating was observed.
R3	Slow Detection	The CV algorithm is too slow and cannot detect moving obstacles.	2	3	6	- Optimize the CV algorithm - Use better hardware, capable of RT object detection	Open	Detection is not 100% accurate. Using better algorithms and hardwares could improve the detection.
R4	False Positive Obstacle Detection	The CV Algorithm falsely warns the user of an inexisting obstacle	1	2	2	-Improve the CV algorithm -Use high resolution sensors such as depth cameras.	Open	False positives didn't happen with our upgraded algorithm during testing. But it could still potentially happen. Computer vision is very complex when applied to real world.
R5	False Negative Obstacle Detection	The CV Algorithm fails to detect an existing obstacle in the challenger's path	3	2	6	- Add a lower resolution sensor array as a fail-safe, such as ultrasonics. - Use better hardware for the obstacle detection. -Improve the CV algorithm	Open	False negative happen. Same comment as R3

R6	Prototype goes dark	The obstacle detection stops signaling obstacles (either due to hardware or software errors)	3	1	3	Indicate the nominal operations of the prototype with a visual indication, such as a simple LED turned on during nominal operations, turned off otherwise.	Open	The LED turns ON/OFF depending on if the Jetson is ON/OFF. The LED should turn off if the program crashes. But we didn't have time to test it correctly.

Note: Everytime you make significant changes to this document (e.g. for deliverables) you should increase the versioning of the file from v1.0 to v2.0 etc. For drafts for a deliverable you could use v0.1 to v0.2.