

**Rapport final de projet**  
**Polytech Informatique 5**

Samuel BAMBA  
Florian CUZIN  
Théo ECHEVET  
Raphaël MANGER

**Aide aux déficients visuels**  
**(avec Intel Movidius et Lidar 2D)**



<b>I) Rappel du sujet</b>	<b>3</b>
Besoin	3
Cahier des charges	3
Projet idéal	4
<b>II) Technologies utilisées</b>	<b>5</b>
<b>III) Architecture</b>	<b>6</b>
<b>IV) Réalisations techniques</b>	<b>6</b>
<b>V) Gestion de projet</b>	<b>7</b>
<b>VI) Outils et métriques</b>	<b>8</b>
<b>Conclusion</b>	<b>9</b>
<b>Bibliographie</b>	<b>10</b>

# I) Rappel du sujet

## 1) Besoin

Les personnes atteintes de déficience visuelle possèdent déjà des équipements pour les aider à se déplacer et à se repérer dans leur environnement (cane blanche), mais les alternatives technologiques se retrouvent bien souvent très chères. Ainsi, les avancées technologiques au niveau du traitement visuel peuvent permettre à ces personnes d'avoir une meilleure connaissance de leur environnement avec des prix beaucoup plus abordables.

Ainsi, l'objectif de notre projet est de réaliser un prototype, décrivant les obstacles présents dans l'environnement de l'utilisateur, afin d'aider ces personnes souffrant de déficience visuelle à mieux appréhender leurs déplacements.

## 2) Cahier des charges

Le but de ce projet est d'utiliser des technologies récentes en terme de détection visuelle, ainsi que des technologies permettant de détecter la distance des obstacles présents sur le chemin de l'utilisateur.

Les deux technologies majeures que l'on devait utiliser pour notre prototype étaient donc un Lidar 2D (Hokuyo URG), nous permettant de détecter un objet dans un rayon de 5 mètres de l'utilisateur, et nous renvoyant l'angle et la distance de cet objet, ainsi qu'une clé Intel Movidius afin de définir la nature des objets présents dans une image à l'aide d'un algorithme de reconnaissance d'objets.

Ces différents éléments seront combinés sur une Raspberry Pi 3, afin de permettre la mobilité de l'utilisateur lors de l'utilisation du prototype.

## 3) Projet idéal

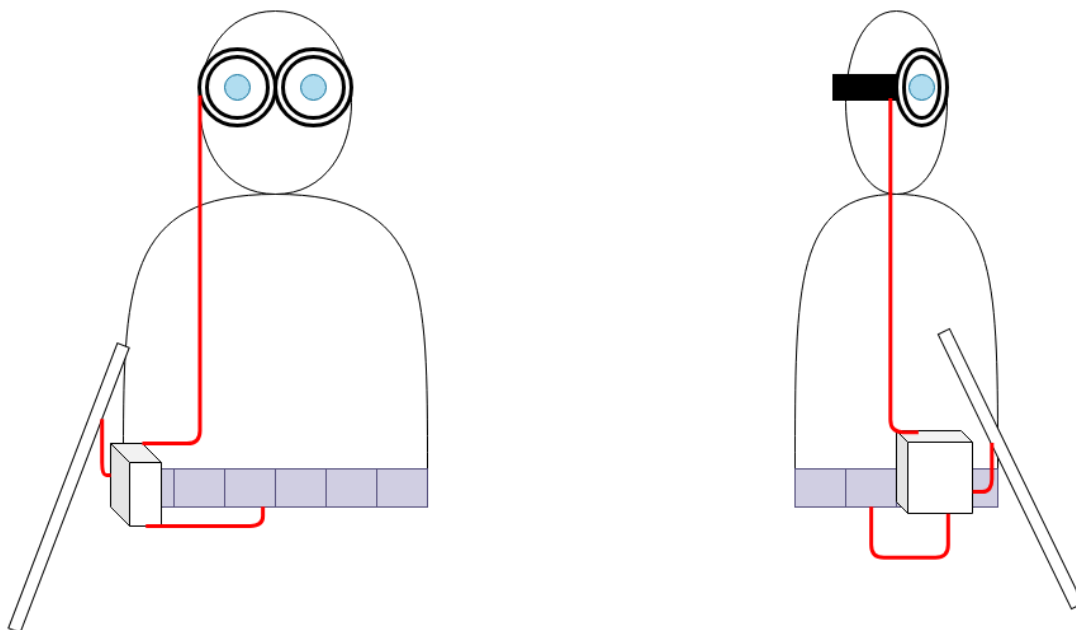
Au début du projet, nous avons tout d'abord préféré imaginer un dispositif où toutes les technologies étaient déjà existantes, et disponibles au FabLab. Ainsi, notre projet idéal devait combiner une caméra V2, un Lidar 3D miniaturisé et des lunettes pour optimiser la mobilité de l'utilisateur.

De plus, un lecteur braille miniature avait été envisagé et aurait pu être présent sur la membrane de ces lunettes, afin de ne pas gêner l'utilisateur au niveau du sens de

l'ouïe, relativement importante pour une personne souffrant de déficience visuelle. Après une étude plus approfondie, il s'est avéré que cette modalité n'était pas pratique car elle aurait mobilisé une main de l'utilisateur, et nous avons préféré choisir un autre moyen pour transmettre la nature de l'objet.

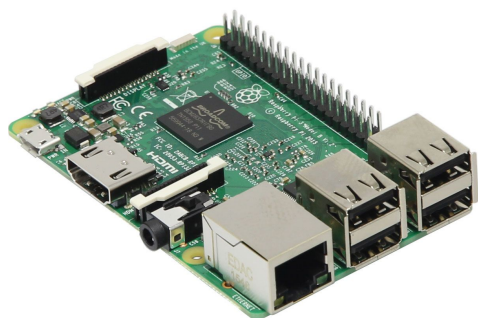
Ensuite, une reconnaissance visuelle de la canne blanche de l'utilisateur aurait pu permettre d'ajouter des commandes gestuelle, telle que l'indication de la direction de l'attention du dispositif. Nous avons été mis en garde sur l'importance de cet outil pour les personnes déficientes visuelles et sur leur réticence à voir cet objet modifié ou alourdi. Nous avons donc exclu toute commande manuelle sur la canne blanche, comme par exemple des boutons.

Ce dispositif aurait été aussi composé d'une ceinture munie de vibrateurs Piezo, afin de pouvoir encoder l'angle et la distance d'un obstacle. Ainsi qu'un boîtier fixé à la ceinture contenant la Raspberry 3.



## II) Technologies utilisées

Notre projet étant de mettre au point un prototype, nous avons utilisé plusieurs technologies pour parvenir à notre résultat final. Notre prototype se compose principalement de 3 parties: une carte Raspberry Pi 3 + caméra, une clef Intel Movidius et un Lidar 2D.



### **Raspberry Pi 3:**

Cette carte est un ordinateur miniature, il nous permet de faire le lien entre les autres technologies que l'on utilise tout en restant transportable. C'est la Pi 3 qui nous permet de faire tourner notre programme tout en utilisant le lidar 2D, la clef movidius et la caméra Raspberry.



### **Intel Movidius :**

Cette clef n'est pas faite pour le stockage, elle nous permet d'avoir une puissance de calcul suffisante pour pouvoir reconnaître des objets. L'Intel Movidius contient un VPU (Visual Processing Unit), c'est un processeur spécial qui permet d'effectuer des calculs sur des images comme la reconnaissance d'objet ou bien la classification d'image plus rapidement que sur un CPU.



### **Lidar 2D Hokuyo :**

Ce dispositif est particulièrement utilisé dans la robotique et l'automatisation. Il comporte plusieurs parties, un laser pour mesurer une distance et un moteur pour faire bouger ce laser dans un axe. Grâce à lui on peut "scanner" les alentours (un plan 2D) et donc récupérer la distance entre le Lidar et un objet pour plusieurs angles autour du Lidar.

### III) Architecture

Notre prototype repose sur une architecture logicielle minimale. Etant donné un contexte mobile et afin d'éviter une surcharge cognitive, nous avons choisi une approche d'épuration des actions utilisateur.

L'interface utilisateur est donc entièrement intégrée à son environnement et permet les actions suivantes:

- Détection et évitement d'obstacle imminent
- Déplacement et orientation en direction d'objet pour identification

L'architecture matérielle repose sur 5 composants communicants entre eux.

Dans un premier temps, les "sens" de notre système, le Lidar et la caméra, vont récupérer les informations nécessaires à une compréhension de l'environnement. On obtient ainsi, une information synchronisée sur la distance et aspect des objets à proximité de l'utilisateur.

Ensuite, la partie pensante de notre système est constituée du duo Raspberry Pi et Intel Movidius. Les données vont être formatées par notre Raspberry Pi et l'image obtenue va être envoyée à l'Intel Movidius pour une reconnaissance des objets présents.

Enfin, un dernier traitement va permettre d'unifier et simplifier cette connaissance de l'environnement en un texte puis un son synthétisé pour notre dernier composant, communiquant avec l'utilisateur, le haut parleur.

### IV) Réalisations techniques

Tout d'abord, notre prototype va essayer de détecter un objet dans un rayon de 4 mètres de l'utilisateur, dans un angle de 62 degrés devant celui-ci, à l'aide du Lidar 2D. Une fois qu'un obstacle a été détecté, le prototype prends alors une photo, que l'on va traiter avec un algorithme<sup>0</sup> de détection d'objets Intel.

Cet algorithme, utilisant la Movidius, permet de détecter la nature des objets présents dans une image. On récupère alors la nature de tous les obstacles

présents dans l'image, ainsi que leur position sur l'image afin de les comparer avec l'angle de l'obstacle que le Lidar a détecté.

Ainsi, on décide d'annoncer à l'utilisateur la nature des obstacles présents dans l'angle donné par le Lidar, ainsi que la distance arrondi au mètre, en sortie sonore à l'aide du Haut-parleur branché à la Raspberry.

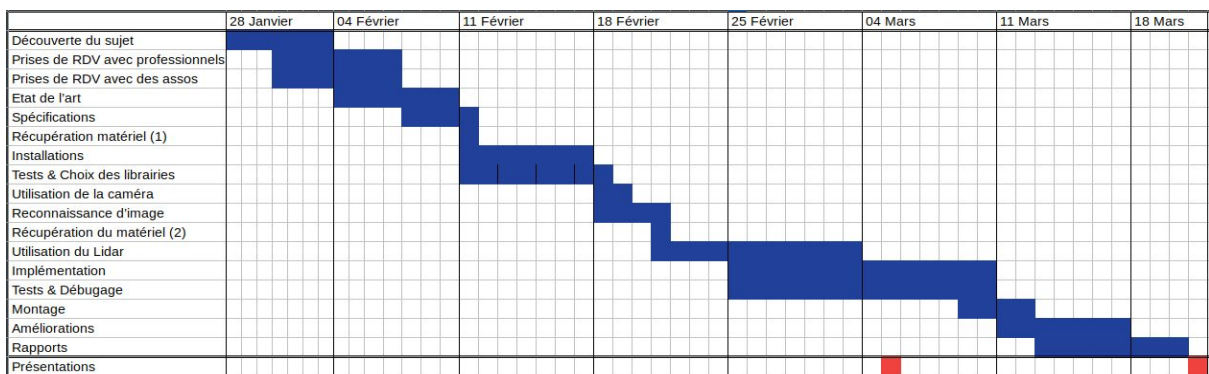
## V) Gestion de projet

Nous avons organisé notre projet selon différentes étapes successives. Nous avons tout d'abord étudié les objectifs de ce projet, les besoins et l'offre déjà disponible, ainsi que ce qui a déjà été fait au sein de Polytech par nos camarades.

Nous avons dû également pour cela prendre contact avec des professionnels. Nous sommes ensuite passés à la réalisation du prototype, de ses spécifications à son implémentation finale. Il existe des écarts entre ce que nous avons planifié et ce que nous avons finalement pu faire. Les figures suivantes représentent respectivement les diagrammes de Gantt prévisionnel et définitif.

Pour la réalisation du prototype, nous avons fait face à de nombreuses contraintes et problèmes techniques qui ont eu un impact majeur sur notre organisation. Nous avons fait le choix d'organiser nos séances de travail chez l'un d'entre nous pour les raisons suivantes :

- nous avons beaucoup de matériel à transporter
- nous avons sous notre responsabilité du matériel coûteux
- nous avons besoin d'un accès physique à une box Internet



	28 Janvier	04 Février	11 Février	18 Février	25 Février	04 Mars	11 Mars	18 Mars
Découverte du sujet	■							
Prises de RDV avec professionnels	■	■						
Etat de l'art	■	■	■					
Spécifications		■	■					
Récupération matériel (1)			■	■				
Résolution problèmes matériels (1)			■	■				
Récupération matériel (2)				■	■			
Résolution problèmes matériels (2)				■	■	■		
Installations				■	■	■	■	
Tests & Choix des librairies					■	■	■	
Utilisation de la caméra						■	■	
Reconnaissance d'image							■	■
Récupération du matériel (3)							■	■
Utilisation du Lidar							■	■
Implémentation							■	■
Tests & Débugage							■	■
Montage							■	■
Rapports								■
Présentations								■

Nous avons prévu de prendre des rendez-vous avec des professionnels, puis sur leur conseils, de rentrer en contact avec des associations afin de rencontrer et d'échanger avec des personnes malvoyantes pour mieux comprendre leur handicap. Nous avons mis plus de temps que prévu à obtenir certains rendez-vous et certains professionnels n'ont jamais répondu à nos sollicitations.

Nous avons également rencontré des problèmes liés au matériel que nous utilisons et nous avons dû en changer à plusieurs reprises. Pour cela, nous nous sommes rendus au FabLab ou bien nous en avons commandé par nous-même sur Internet, ce qui engendra des délais supplémentaires durant lesquels nous ne pouvions toujours pas commencer l'implémentation.

Nous avons mis un temps particulièrement long à installer et à faire fonctionner les différentes librairies logicielles sur notre carte Raspberry Pi. Nous prévoyions de réaliser cette tâche en une semaine tout au plus, mais compte tenu de la succession de problèmes rencontrés et des temps d'installation, c'est plus de deux semaines qu'il auront été nécessaires.

Ce retard a toutefois pu être compensé. En effet, nous avons surestimé les temps nécessaire à faire fonctionner le Lidar 2D et à implémenter le prototype. Nous avons trouvé une solution "clé en main" permettant de faire fonctionner le Lidar 2D parfaitement, aussi bien sur nos ordinateurs que sur la Raspberry Pi. De plus, il s'est avéré que nous avons moins de code source à produire que ce que nous supposions.

Au final, nous avons prévu une semaine pour réaliser des améliorations sur le prototype. Elle nous aura finalement permis de finaliser notre projet.



## VI) Outils et métriques

Pour une synchronisation de nos idées, ressources et fiche de suivi, nous avons utilisé Google Drive. Ensuite, nous avons utilisé Git et GitHub pour la coordination de notre code.

Le code réalisé en Python 3 est constitué d'environ 200 lignes de codes réparties dans les fichiers `main.py` et `object_detector.py`. De plus, nous avons utilisé les bibliothèques `hokuyo`<sup>1</sup> et `espeak`<sup>2</sup>.

D'un point de vue performance notre prototype a un temps de démarrage moyen de 7 secondes et un intervalle arbitraire de 3 secondes a été défini entre deux annonces vocales.

## Conclusion

Ce projet de fin d'études fut très enrichissant pour tous les membres du groupe sur différents aspects du travail en équipe. En effet, ce projet comprenait les différentes étapes de réalisation d'un nouveau produit, tant au niveau de la créativité afin d'imaginer un dispositif répondant à des besoins spécifiques, qu'au niveau du management du temps afin de pouvoir livrer le meilleur produit possible à la date prévue.

Même si les prévisions réalisées en début de projet n'ont pas pu être tenues, ce projet a pu nous montrer les points faibles (comme la perte de temps pour la rencontre avec les clients et les professionnels) mais aussi les points forts de nos méthodes. De plus, le travail livré répondant principalement aux spécifications demandés, nous sommes contents de notre travail, même si nous n'avons pas pu développer tout ce que l'on souhaitait.

Ainsi, ce dernier projet nous aura permis d'acquérir encore un peu plus d'expérience concrète juste avant de nous retrouver dans le monde du travail.

# Bibliographie

- [0] Detection objets, <https://github.com/movidius/ncappzoo/tree/master/apps/object-detector>
- [1] Librairie laser hokuyo, <https://github.com/pasuder/hokuyo-python-lib>
- [2] Librairie synthèse vocale, <http://espeak.sourceforge.net/>
- [3] Caméra, <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/5>
- [4] Librairie Intel Movidius, <https://github.com/movidius/ncsdk>
- [5] Documentation du module skimage, <http://scikit-image.org/docs/dev/index.html>
- [6] Téléchargement de Raspbian, <https://www.raspberrypi.org/downloads/raspbian/>
- [7] Raspberry 'headless', <https://raspbian-france.fr/raspberry-pi-sans-ecran-sans-clavier/>