Élèves
Maud BLANCHARD
Ludivine VIRATELLE
Clément BEROUNSKY
Basile BONICEL
Loïc DROUARD
Projet encadré par :
Anne-Claire CHENUS

CONCOURS C.GENIAL 2016
Lycée Saint-Charles, Orléans

CanneiSee

Dispositif de suppléance visuelle

Projet réalisé en collaboration avec :

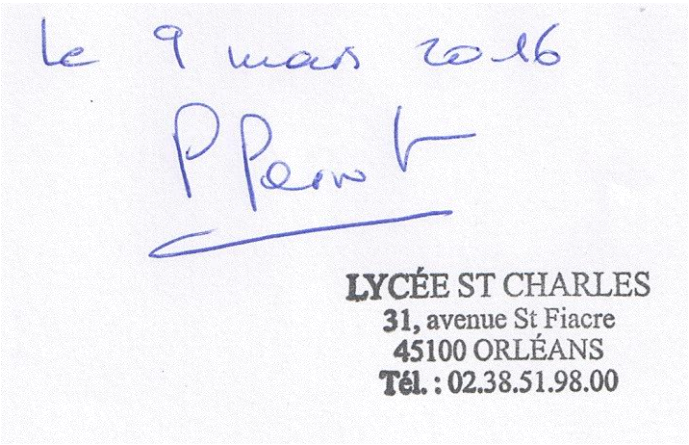
Alan URBAN, Chercheur en Neurosciences et Imagerie cérébrale

CANNE I SEE ?

Problématique : Comment réaliser la suppléance de la vue en combinant l'ouïe et le toucher ?

Résumé

Notre projet consiste en l'élaboration d'une canne blanche électronique à destination des malvoyants. CanneiSee propose des fonctions avancées comme l'évitement d'obstacles, la reconnaissance de proches et la lecture de textes. Le prototype a été réalisé par les élèves en utilisant des composants électroniques grand public comme un microcontrôleur Arduino qui commande deux capteurs à ultrasons et un vibreur afin de communiquer l'information de distance à un obstacle sous forme de vibration. Un ordinateur miniature Raspberry Pi2 équipé de 2 caméras commande quant à lui l'acquisition et le traitement d'images en temps réel. Ainsi, les informations sonores (bips) sont transmises par un casque audio au malvoyant et indiquent le chemin à suivre selon une ligne tracée au sol. La reconnaissance faciale de proches dont les portraits sont stockés dans une base de données, la détection et la lecture de textes dans l'environnement (panneaux, bus…) par synthèse vocale sont également possibles.



REMERCIEMENTS

Ce projet a été effectué au sein du lycée Saint-Charles sous la direction de Madame PERROT et en collaboration avec Alan URBAN, chercheur en neurosciences.

Nous tenons remercier Monsieur Alan URBAN pour sa disponibilité et pour l'aide qu'il nous a apportée dans le montage et la programmation du prototype **CanneiSee** qui a nécessité plus de 200 heures de travail.

Nous remercions également Madame Claire RUELLO, technicienne du laboratoire du lycée pour l'aide apportée et pour la gestion des commandes de matériel.

Nous tenons également à remercier Monsieur Jean-Michel VINCOT pour son soutien et ses conseils dans la mise en place du projet.

Enfin, nous remercions Madame Pascale PERROT, directrice adjointe du lycée Saint-Charles, ainsi que Monsieur Frédéric LE NALIO, directeur du cours Saint Charles, qui nous ont soutenus dans la réalisation du projet.

INTRODUCTION

Il y a dans le monde, selon l'OMS, près de 285 millions de personnes qui présentent une déficience visuelle: 39 millions d'entre elles sont aveugles et 246 millions présentent une baisse de l'acuité visuelle.

On distingue les aveugles des malvoyants profonds pour qui la vision est limitée à la distinction des silhouettes.

Les malvoyants moyens sont dans l'incapacité à reconnaître un visage à 4 mètres, à lire et à écrire (**Figure 1**). La première cause de cécité dans le monde est la cataracte (50% des causes de cécité).

Figure 1 *: Vision d'une personne atteinte de cataracte, de DMLA ou de glaucome.*

Le but de notre projet est d'aider ces personnes (aveugles et malvoyants avancés) dans leur vie quotidienne. Pour mieux cibler nos objectifs, nous avons effectué des recherches pour répondre à la question : **À quelles difficultés font face les aveugles ?**

- Les aveugle et malvoyants ont des difficultés à **se déplacer en ville** : transports en commun, repérage préalable du parcours.
- Difficultés au quotidien : la lecture des textes, reconnaissance d'objets.
- Les interactions sociales sont difficiles car ils ne peuvent reconnaître les personnes.

Quelles solutions techniques existent pour ces personnes en difficulté ?
En dehors des possibilités offertes par la chirurgie, nous avons identifié les solutions suivantes :

- Pour les déplacements : il est possible d'acquérir un dispositif médical comme la canne blanche ou encore le chien d'aveugle. Il existe également des cannes électroniques qui réalisent uniquement la détection d'obstacles comme l'UltraCane, basée sur la technologie des ultrasons et dont le prix est de 600 € (**Figure 2**) ou la canne de Visioptronic, basée sur la détection par infrarouge et par laser (5000 €).

Figure 2 *: Ultracane avec ses détecteurs ultrasonores*

Il existe des applications GPS pour smartphones (comme Google Map) qui peuvent être utilisées par les malvoyants mais qui doivent être programmées par un voyant et qui perdent leur utilité quant aux déplacements dans les bâtiments (non cartographiés).

- **Pour la reconnaissance d'objets et la représentation mentale de l'environnement:**
Il existe actuellement deux approches, **la substitution sensorielle** et **la reconnaissance d'objets et de personnes** par des systèmes embarqués qui ne sont pas développés directement pour les personnes non-voyantes.

Face à cette offre, quels sont les objectifs de notre projet CanneiSee ?
Nous souhaitons développer un dispositif médical innovant et peu coûteux qui permet de guider les malvoyants dans leurs déplacements, de détecter des obstacles et de reconnaître des personnes (profils stockés dans une base de données). Il s'agit d'aider les malvoyants :
- À l'extérieur : détection des obstacles et avertissement pour contourner l'obstacle (mur, poteau, escalier) + reconnaissance de texte simple (numéro de bus, nom de rue, panneau)
- En intérieur : aider à se diriger dans un bâtiment inconnu + permettre la reconnaissance d'objets
- Interactions sociales : reconnaissance des visages

Pour cela, nous avons développé une canne blanche intelligente qui possède des modules électroniques. Les capteurs/récepteurs utilisés sont notamment des webcams, des capteurs à ultrasons pour la détection des obstacles. La scène est analysée en temps réel par un micro-ordinateur embarqué Raspberry Pi2 et des informations sont transmises au malvoyant sous forme de sons et de vibrations. Le système permet la reconnaissance faciale et de caractères. Le passage entre les différents modes se fait à l'aide d'une télécommande fixée sur le manche.

PARTIE 1 : Choix d'une stratégie de suppléance visuelle

L'algorithme vOICe dédié aux non-voyants

Des chercheurs ont mis au point l'algorithme vOICe (Oh I See) qui convertit l'image (acquise par une webcam) en son binauréal par balayage de gauche à droite (**Figure 3**). La hauteur de l'objet est codée en fréquence et l'intensité des pixels dans l'image (après conversion en noir et blanc) est codée en intensité sonore.

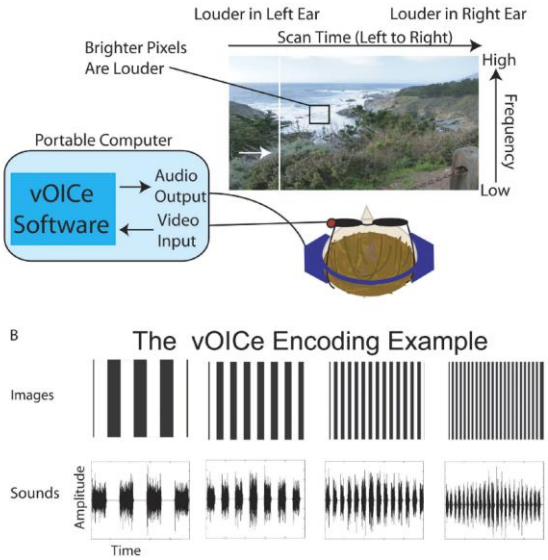


Figure 3 ***:*** *Fonctionnement de l'algorithme The vOICe.*

Cependant, cet algorithme ne permet pas de déterminer la distance des obstacles.

Comment fournir au malvoyant une information de distance ?

Notre choix s'est porté sur un capteur de distance associé à un module qui convertira l'information de distance en vibration. Pour mesurer la distance, nous avons identifié plusieurs solutions présentées et comparée dans le **tableau** de l'annexe 2.

Conclusion : Le laser est couteux et trop directif tandis que le capteur infrarouge est peu fiable sur les longues distances et sensible à la luminosité. Notre choix s'est porté sur les ultrasons dont la fiabilité est satisfaisante, combinés à l'implémentation de l'algorithme the vOICe. Ainsi, notre dispositif devrait ressembler au montage présenté dans la **Figure 4** :

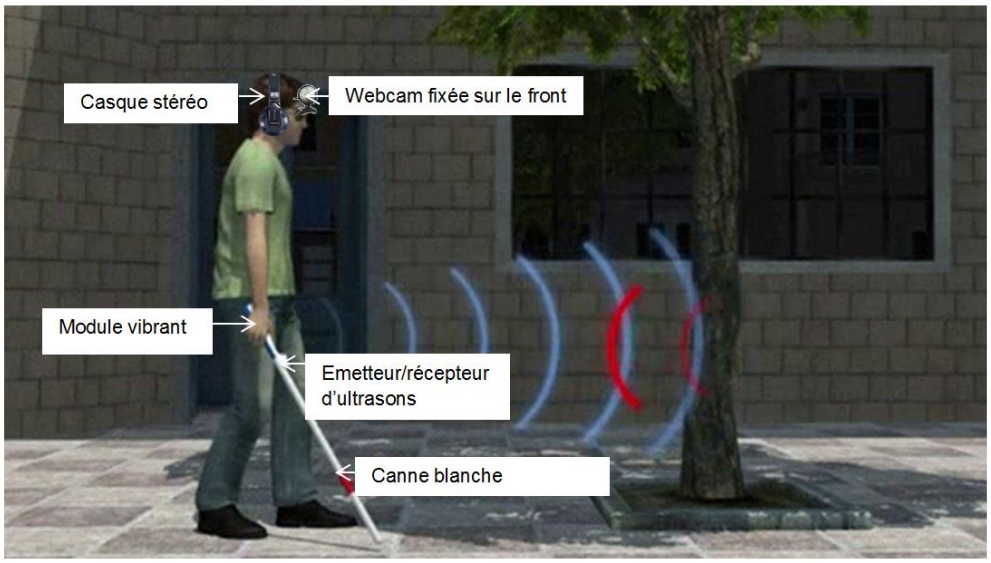


Figure 4 ***:*** *Concept du dispositif de suppléance visuelle CanneiSee.*

PARTIE 2 : Comment mesurer la distance à un obstacle

Principe de l'écholocation

À l'état naturel de nombreux animaux (ex : chauvesouris, dauphins) se déplacent grâce à l'écholocation. Notre dispositif fonctionne de même : une onde ultrasonore est émise sous forme de pulse qui est réfléchie par la cible.

Arduino et capteur à ultrasons

Pour la mesure de la distance, nous utilisons la formule : « $\text{Distance} = c \times \Delta t$ ». Nous considérons que les ondes se propagent à 340 m/s (on néglige l'influence de la température). Nous mesurons le retard Δt et on en déduit la distance parcourue par l'onde qui doit être divisée par 2 (aller-retour).

Nous utilisons pour l'émission des ondes le composant HC-SR04 (ensemble composé d'un émetteur à ultrasons et d'un récepteur - Figure 5) formé de transducteurs piézoélectriques qui vont vibrer lorsqu'une tension est appliquée (émetteur) ou produire une tension lorsqu'une vibration est reçue (récepteur).

Figure 5 : capteur ultrasonore.

Le capteur comporte 4 broches, 2 pour l'alimentation (VCC et GND), une pour l'envoi (TRIG) et une pour la réception (ECHO) des ultrasons qui sont reliées un microcontrôleur Arduino UNO (ATMEGA 328, 16 MHz). L'utilisateur sera averti de la présence d'obstacles grâce à un composant électronique qui vibre.

Le programme de contrôle du capteur

Nous avons écrit le programme qui permet de déterminer la distance (**Figure 6**). Le problème majeur auquel nous avons fait face est qu'au bout de quelques secondes, le programme ne renvoyait que des « 0 ». Nous avons donc inséré une boucle de débuggage qui a pour action l'équivalent d'un reset.

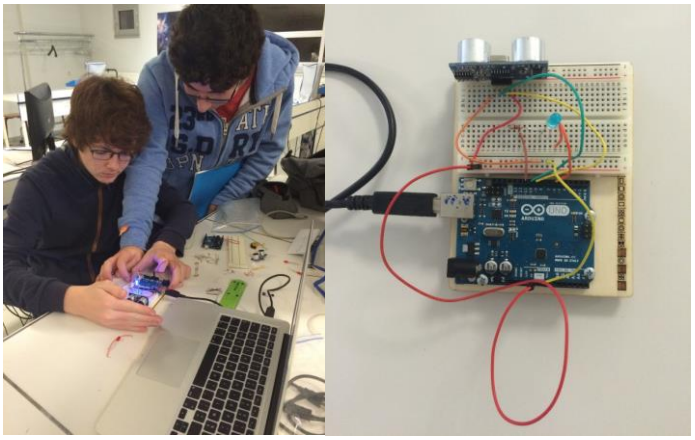


Figure 6 : *Basile et Loic testent le montage*

Pour estimer le nombre de capteurs nécessaires et leur position, nous avons cherché à comprendre le type d'obstacle que le capteur pouvait détecter, et pour un obstacle donné, à quelle distance maximale et à quel angle maximal il était détectable. Cela nous permettait aussi de

connaitre la fiabilité du capteur. Comme prévu par la théorie, les tissus (vêtements) à cause de leur faible impédance, sont mal détectés (1 m dans l'axe). Une assiette est détectée jusqu'à un mètre dans l'axe. Nous avons donc utilisé un écran en métal peint. Sa surface est de 0,5 m2 (surface minimale préconisée par le constructeur). Nous avons effectué 100 mesures aux différents angles (ficelle étirée au sol) et aux différentes distances (**Figures 7 et 8**).

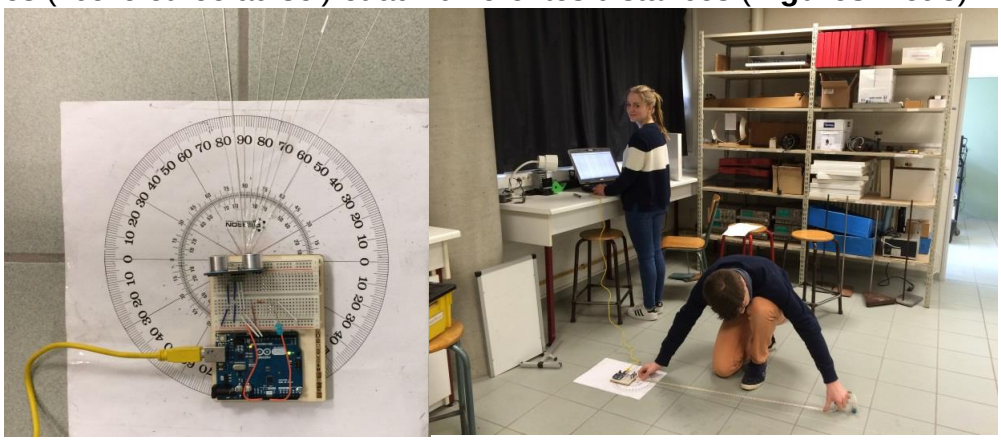


Figure 7 : *Maud et Clément mettent en place le montage.*

Figure 8 : *Loic à l'ordinateur, Ludivine positionne le capteur, Maud approche la cible.*

Le tableau suivant donne dans l'axe la valeur moyenne donnée par le montage, l'écart type, et la valeur mesurée dans l'axe 0° (cible : écran). Toutes les distances sont en cm.

Distance (mesurée au mètre $\pm 0{,}1\ cm$)	20,0	50.0	80.0	100.0	300.0	400.0
Distance moyenne mesurée par le capteur (100 mesures)	20,1	52.7	72.3	103.4	295.1	394.4
Ecart à la distance réelle	0.1	2.7	2.3	3.4	-4.9	-6.4
Ecart type	0,4	0.2	1.1	0.8	0.6	0.4

Le capteur est fiable pour l'utilisation qu'on souhaite en faire (binaire : il y a un obstacle à moins d'un mètre 50 : vibration, sinon : rien).

Le diagramme ci-dessous (**Figure 9**) indique les zones où l'obstacle est détecté avec une fiabilité suffisante. Pour cela, nous comptons le nombre de fois où « out of range » ou une valeur absurde est renvoyée (échec !). Le pourcentage de réussite est donc « (100 – Nombre d'échecs)/100 ».

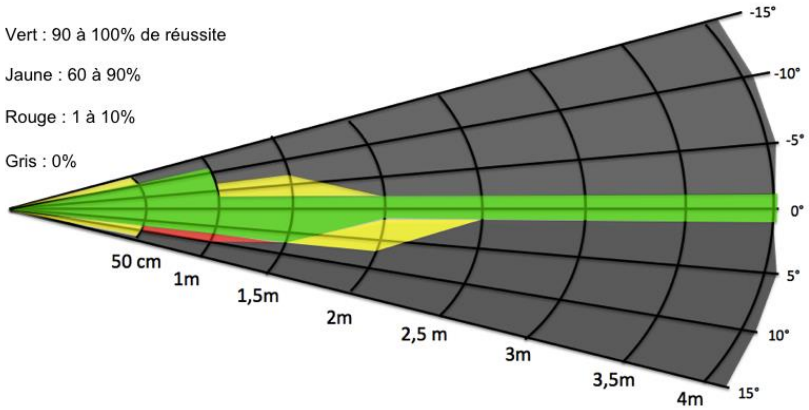


Figure 9 : Diagramme de fiabilité de détection d'un obstacle par le capteur

Le capteur est donc très directif. Le cône est légèrement inférieur à la donnée constructeur (10° au lieu de 15° pour des distances de l'ordre du mètre). La dernière mesure effectuée est la détection sur la verticale. L'objet n'est plus perçu s'il s'écarte de plus de 5° de l'axe ce qui indique qu'il est aussi très directif en hauteur.

Conclusion concernant la position des capteurs sur la canne

Le dispositif doit permettre au malvoyant d'éviter les obstacles sur sa route mais aussi prévenir un trou ou une marche. Puisque les capteurs à ultrason sont très directifs, la canne sera donc équipée de 2 capteurs : un premier dirigé vers le sol et un autre à l'horizontal.

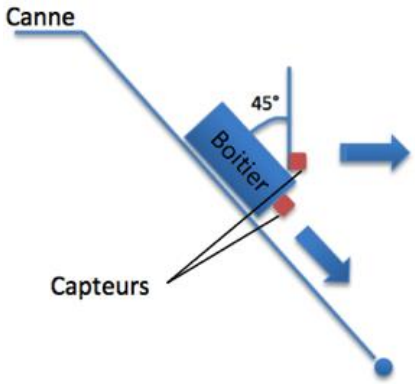


Figure 10 *: Positions des capteurs sur la canne.*

L'information sera codée sur deux vibrations : une vibration continue pour les dangers au sol, une vibration par secousses pour les obstacles dans le champ de déplacement. Nous avons réalisé un programme avec deux boucles pilotant chacune un capteur.

PARTIE 3 : Comment aider le malvoyant à se déplacer en environnement urbain et à l'intérieur

Échec de l'algorithme vOICe

Notre idée initiale était d'utiliser l'algorithme vOICe mais après une série de tests nous nous sommes rendus compte que son utilisation nécessitait déjà un long apprentissage pour distinguer un carré d'un rond blanc sur fond noir ! Il était donc inenvisageable de l'utiliser en paysage urbain beaucoup plus complexe ! Nous avons éliminé cette option au profit d'une technologie qui ne nécessite pas d'apprentissage particulier et guide le malvoyant de manière efficace. Nous avons aussi changé notre problématique puisque nous n'effectuons plus une substitution sensorielle mais une **suppléance** sensorielle. (problématique originale : comment réaliser la *substitution* de la vue par l'ouïe et le toucher ?)

Analyse d'image en temps réel

L'analyse d'image en temps réel utilise un micro-ordinateur Raspberry Pi 2 (RPi2) (processeur ARM 800MHz, 1 Gb RAM, SD 16 Go) équipé du système d'exploitation Rasbian Jessie (une variante de Debian) sur carte microSD 16 Go Class 10. Le RPi2 est équipé d'une caméra embarquée (type Picamera) et d'une webcam qui seront utilisées pour les modules 1 (guidage par ligne), 2 (reconnaissance de visage) et 3 (reconnaissance de texte). Un module 4 donnant l'heure sera également implémenté. Le RPi2 sera équipé d'un casque audio et alimenté par une batterie.
Toute cette partie de notre travail s'appuie sur le langage de programmation Python version 3 (complémenté par le C++) et utilise un ensemble de logiciels/scripts libres de droit (ou sous licence CC) qui seront adaptés pour les besoins du projet :

- Logiciel pour la vision par ordinateur : OpenCV (Computer Vision) qui offre une bibliothèque complète et open-source d'outils pour la vision par ordinateur
- Logiciel de synthèse vocale (Text-to-speech) TTS crée par Google
- Logiciel de reconnaissance de caractères (OCR) – Tesserach crée par Google

Canneisee offre 4 modes de fonctionnement

Toutes les opérations d'analyse des données d'images par la webcam sont très couteuses en puissance de calcul, si bien qu'il est impossible de les effectuer en même temps. Pour résoudre ce problème, nous avons mis au point une commande à distance située sur le manche de la canne qui permet le déclenchement de chaque mode :

- Mode 1 NAVIGATION = Suivi de ligne colorée (3 couleurs correspondant à 3 itinéraires sont programmés Rouge, Jaune, Vert)
- Mode 2 RECONNAISSANCE = Détection des visages
- Mode 3 ANALYSE DE TEXTE = Lecture des textes
- Mode 4 TEMPS = Donne l'heure !

Mode 1 : Navigation/guidage par suivi de ligne colorée

Le porteur du dispositif suivra à l'aide d'indications sonores émises en temps réel une « ligne de guidage » tracée au sol. L'objectif étant de convertir l'image enregistrée par la caméra en informations sonores permettant grâce à l'émission de bips dans un casque audio d'indiquer la direction et les déplacements latéraux permettant de suivre la ligne (**Figure 11**). Cela permet ainsi de guider le malvoyant dans un bâtiment, en imaginant que dans un futur proche, des lignes colorées seront tracées au sol (par exemple, à l'hôpital, la ligne rouge va à l'accueil, la ligne bleue, à la salle d'attente, la ligne noire aux toilettes etc).

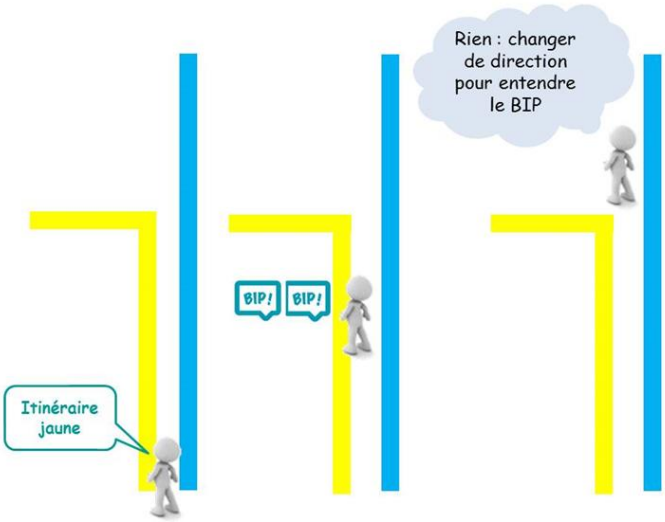


Figure 11 *: Principe du guidage par suivi de ligne colorée.*

Ici, il est en effet nécessaire de détecter une ligne au sol et de reconnaître sa couleur pour suivre l'itinéraire de son choix. Le codage RVB de la couleur (ROUGE par exemple R255 V0 B0) dépend des conditions de luminosité qui affectent le codage de V et B également, ce qui rend compliquée sa détection.

a) Intérêt du codage HSV par rapport au codage RVB

Les valeurs RVB varient en fonction de l'éclairage, des ombres etc ce qui compromet la reconnaissance. Pour s'affranchir cette dépendance en luminosité, il existe un codage plus adapté appelé HSV pour Hue-Saturation-Value (teinte, saturation, valeur). La représentation du codage HSV est donnée dans la **Figure 12**. H, S et V prennent des valeurs comprises entre 0 et 255 sauf H (de 0 à 180). La saturation correspond à l'aspect plus ou moins gris de la teinte. Value correspond à la luminosité du pixel.

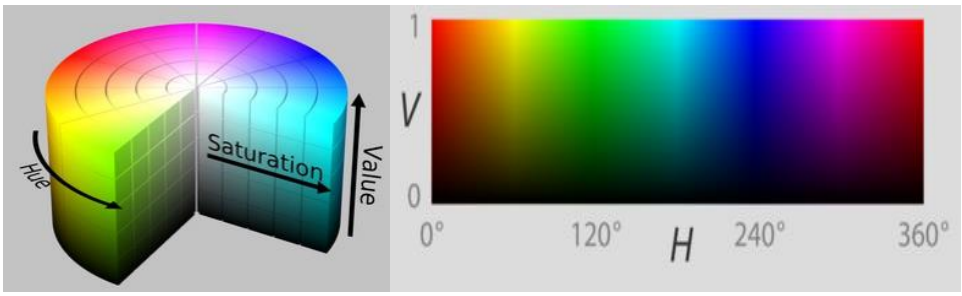


Figure 12 *: Principe du codage HSV en coordonnées cylindriques (à gauche). A droite : diagramme pour S=255.*

b) Les étapes du suivi de ligne

Étape 1 : convertir l'image en HSV
Étape 2 : définition de la couleur à traquer en HSV. On crée un intervalle qui correspond au rouge pour avoir une certaine tolérance sinon on ne détecte que quelques pixels !
Étape 3 : on crée un masque : on rend noir les pixels qui n'appartiennent pas à cet intervalle, et blancs ceux qui lui appartiennent.
Étape 4 : On élimine les pixels blancs éparpillés qui ne représentent pas l'objet à traquer.
Étape 5 : pour les gros objets : on calcule le barycentre en additionnant les coordonnées de tous les pixels blancs et en divisant par la taille.

Étape 6 : on délimite la taille qui nous intéresse depuis le barycentre : on définit un cercle dans lequel on cherche si il y a encore la couleur qui nous intéresse. Si oui, on a bien repéré la ligne et on commande l'émission d'un BIP. Sinon, on n'émet rien.
Étape 7 : on vide les données de la mémoire pour ne pas encombrer le système.

c) Éclairage de la ligne à suivre au sol
Étant donnée que la détection se base sur la couleur, il est très important d'avoir de bonnes conditions d'éclairage pour repérer la ligne. Pour pallier ce problème en cas de conditions non optimales (nuit, faible luminosité) nous avons ajouté 2 DELs blanches au pied de la canne. L'allumage des DELs est contrôlé par à une photorésistance grâce au même module Arduino qui contrôle les capteurs à ultrasons.

Mode 2 : Reconnaissance faciale

Le mode de reconnaissance faciale permet également de fournir des informations dans l'environnement immédiat du porteur comme la présence de personnes connues (amis, famille…) à partir d'une base de données préenregistrées dans le système (série de 20 portraits à différentes luminosités et différentes expressions).
Notre problème est : comment reconnaître un visage ?
Il s'agit d'un problème complexe et auquel la recherche s'intéresse, aucun algorithme actuellement ne pouvant ni reconnaître un visage en temps réel, ni avec une très haute fiabilité.
Le programme agit en deux temps :
1- La détection
2- La reconnaissance

a) Principe de la détection des visages

Il y a 2 méthodes pour traquer des objets avec OpenCV : soit repérer des détails caractéristiques de l'objet (ce qu'on utilise ici !) soit on détecte une couleur (comme vu précédemment)
L'algorithme convertit l'image en niveaux de gris puis la décompose de façon vectorielle et identifie les visages parmi des milliers de petits blocs.

b) Principe de la reconnaissance faciale

- ***Étape 1 : Mise en forme de la zone à reconnaître : redimensionnement + égalisation***

La reconnaissance du visage dépend des conditions de luminosité et de beaucoup de paramètres qui peuvent compromettre l'efficacité de l'algorithme (taille du visage dans l'image, tête tournée par rapport à l'axe, cheveux, maquillage, expressions émotionnelles). C'est pourquoi il est nécessaire d'éliminer les pixels non utilisés autour du visage grâce à un masque elliptique puis de filtrer l'image en réalisant une égalisation de l'intensité lumineuse.

- ***Étape 2 : Algorithme Eigenface***

Il nécessite une base de photos (training set) des personnes que l'on souhaite reconnaître. Chaque photo de la base de donnée doit être traitée par l'étape 1.
L'algorithme Eigenfaces va chercher dans la base de données l'image moyenne qui représente le visage puis calcule pour chaque pixel (x,y) de chaque photo la différence entre ce pixel du visage moyen avec le même pixel (x,y) du visage de chaque photo.

- ***Étape 3 : Détection***

Le RPi2 extrait une image du flux webcam et la compare à la base de donnée comprenant de l'image moyenne, un tableau avec les images Eigenfaces (20 si on a une base de données de 20 photos) et une matrice des valeurs (eigenface ratios) de chaque image d'entraînement. L'écart entre l'image d'entrée et l'image d'entraînement la plus similaire donne la probabilité de reconnaissance (entre 0 et 1). Si le visage est détecté correspond à plus de 75% à un visage de la base de donnée, alors l'ordinateur indique par synthèse vocal quelle est la personne détectée.

À cause de la faible puissance du raspberry Pi2, le temps de calcul est long (10 secondes) et nécessite des bonnes conditions (personne de face, luminosité suffisante) ce qui rend difficile son utilisation en temps réel.

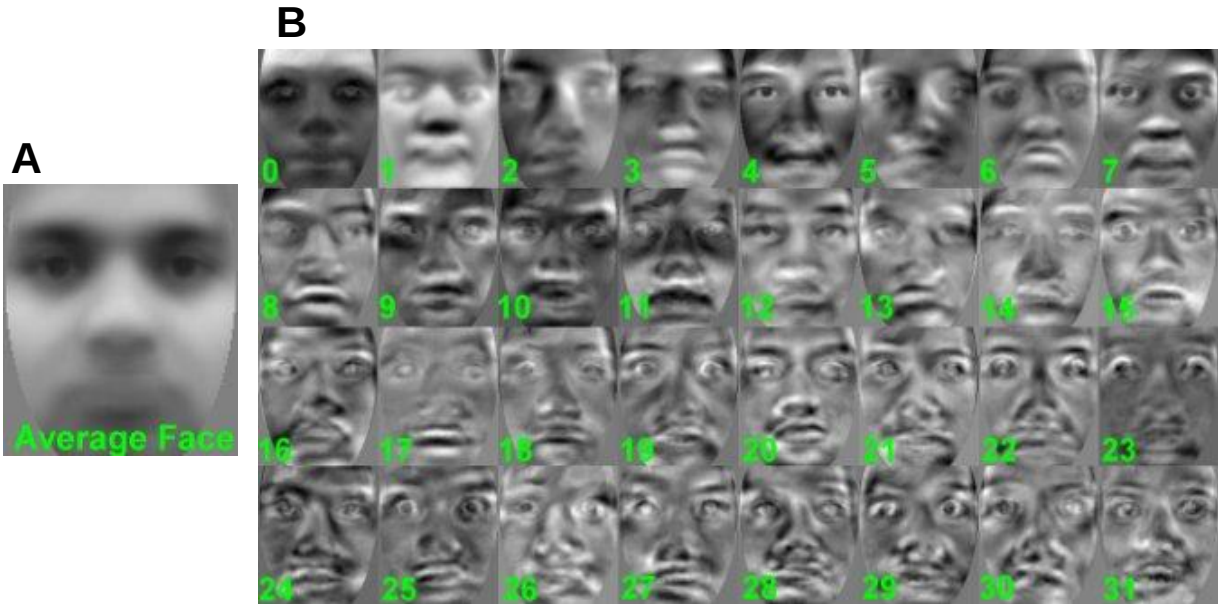


Figure 13 *: Principe de la reconnaissance faciale. (A) Image moyenne du visage à détecter. (B) Recherche des composantes principales des images dans la base de donnée.*

Mode 3 : Reconnaissance de texte

Pour cette fonction, nous utilisons également la bibliothèque OpenCv couplé au logiciel Tesseract de reconnaissance de caractère développé par Google. Cette partie du programme a été réalisée en grande partie par le chercheur Alan URBAN et nous ne la détaillerons pas dans le rapport faute de place.

Mode 4 : Temps

Cette fonction utilise la bibliothèque de synthèse vocale pour donner l'heure sous forme d'un message vocal.

Le prototype CanneiSee regroupera donc des fonctions d'analyse d'image et de détecteur à ultrason comme schématisé dans la **Figure 14**.

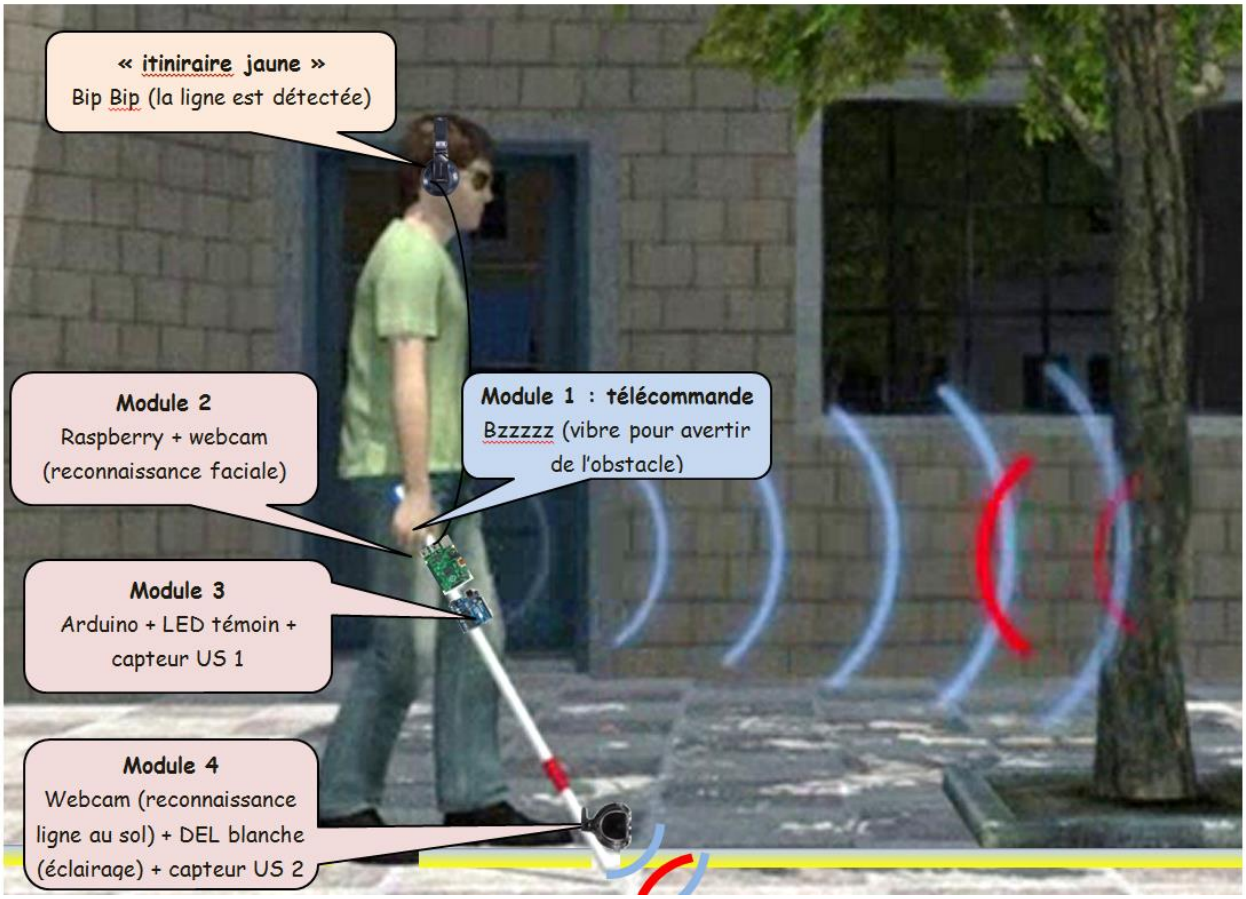


Figure 14 *: Schéma général du prototype CanneiSee V1.0 (détails des modules en* **annexe 1***)*

PARTIE 4 : CONCLUSION GÉNÉRALE

Après une analyse détaillée des solutions disponibles dans le marché comme les cannes blanches électroniques, nous avons démontré qu'il est possible d'assembler un prototype de canne réunissant les avantages des meilleurs systèmes pour un coût bien inférieur (les photos du montage en **annexe 3** !). Ceci a été possible grâce à la disponibilité de cartes électronique grand public et d'ordinateurs à très faible coût. Avec l'aide de nos encadrants, nous avons appris tant sur le plan théorique (en physique et en informatique) que pratique (en programmation, en électronique). Nous envisageons maintenant la prochaine étape qui consiste à prendre contact avec une association de malvoyants comme l'UNADEV (Union Nationale des Aveugles et Déficients Visuels) pour tester le prototype auprès de malvoyants. Ensuite, nous devrons travailler sur la stabilisation du code et aussi sur la miniaturisation des composants pour réduire le poids et la consommation électrique. Tout cela dans l'espoir que notre canne soit peut être un jour commercialisée !

ANNEXE :

Annexe 1 : description des modules

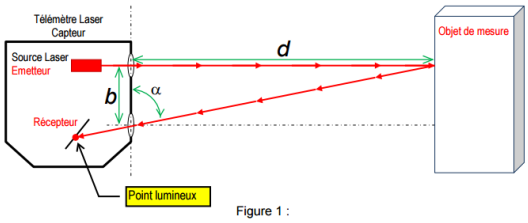
Module 1 : Manette utilisateur	
Télécommande (4 interrupteurs pour switcher d'un mode à l'autre) ON/OFF	Vibreur

Module 2 : Raspberry Pi 2 + Caméra (reconnaissance faciale et reconnaissance de texte)		
	Alimentation Raspberry Pi 2	Picam

Module 3 : Arduino Uno			
	Alimentation de l'Arduino	 Capteur pour les obstacles frontaux	

Module 4 : Caméra guidage			
Webcam 	 2 DEL blanches (illuminer la ligne)	 Capteur pour les obstacles au sol (marche-trottoir)	 Photorésistance

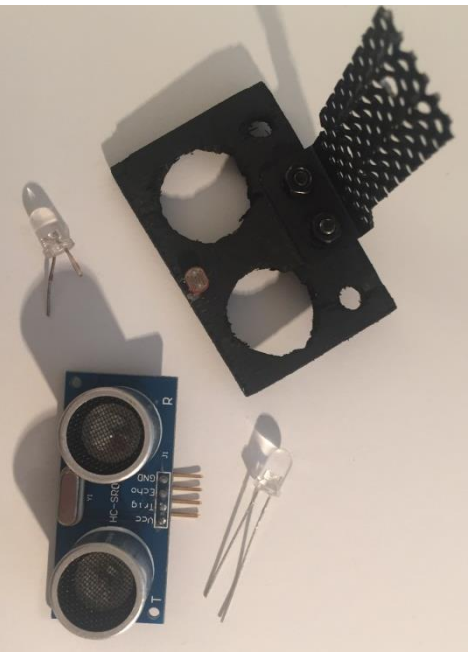
Annexe 2 : comparaison ultrasons, laser, infrarouge

	Ultrasons (*en italique : modèle HCSR04*) 	Infrarouge (*en italique modèle Grove Sharp GP2Y0A21YK*) 	Laser (en italique modèle *LIDAR Lite – laser pulsé à 905 nm*)
Principe physique	Mesure du retard entre l'émission et la réception de l'onde ultrasonore (pulse).	Par triangulation : mesure de l'angle de réflexion d'une émission d'infrarouge modulée grâce à une rangée de récepteur  Figure 1 : • *d* : *formule de la distance mesurée de l'objet de mesure par rapport au télémètre:* $d = b \times \tan(\alpha)$	Par triangulation comme les infrarouges ou en mesurant le retard entre l'émission et la réception de l'onde.
Étendue de mesures	1 cm à quelques mètres *2 cm à 4 mètres*	*De 5 cm à 1 m* *(10 à 80 cm)*	*Plusieurs mètres à plusieurs dizaines de mètres selon les modèles.* *(jusqu'à 40 m)*
Directivité	Cône d'environ 30°	Cône d'environ 5°	Les plus directifs (de l'ordre du degré, voire du demi-degré) 0.5°
Précision	Relativement précis mais la précision diminue avec la distance, l'angle de mesure et les conditions de température et de pression. *Environ 3 cm.*	Relativement précis mais la précision diminue avec la distance.	Sont précis avec un bruit de quelques centimètres sur des mesures de plusieurs mètres. *2,5 cm*
Coût	* *(3.89 euros)*	** *(19 euros)*	*** *116 euros*
Sensibilité aux interférences perturbations extérieurs	Sensible à la température et à la pression. Certains matériaux absorbent complètement les ondes US.	Sont sensibles aux fortes sources de lumière qui contiennent un fort rayonnement infrarouge. Sont également sensibles à la couleur et à la nature des obstacles.	Mesures faussées pour les objets réfléchissants (vitres, objets chromés,…) Brouillé par le brouillard, la pluie, fort ensoleillement. Très directif, nécessite de balayer la zone. Contraignant pour les obstacles proches.

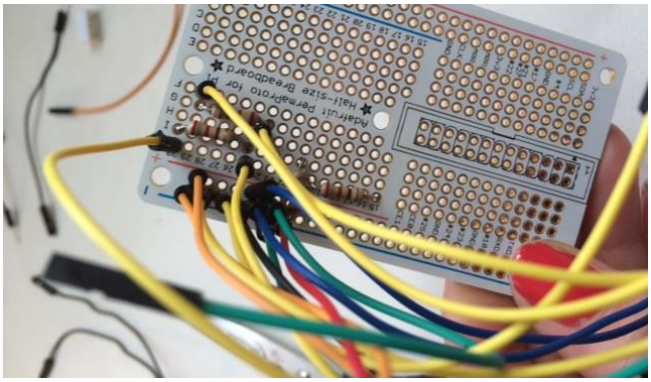
Annexe 3 : montage du prototype

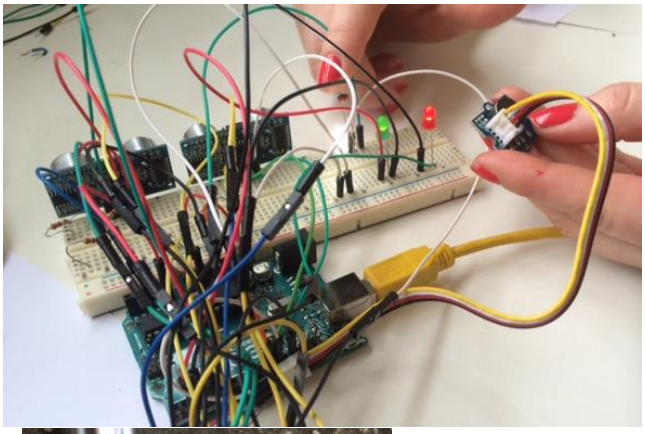
Matériel

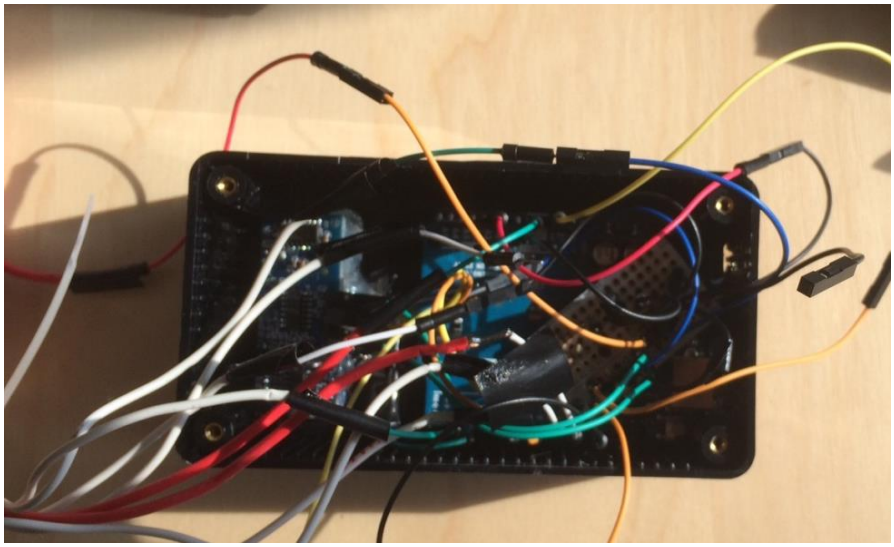
- **Assemblage bas de la canne (Capteur Ultrasons, LED, Photorésistance, webcam)**



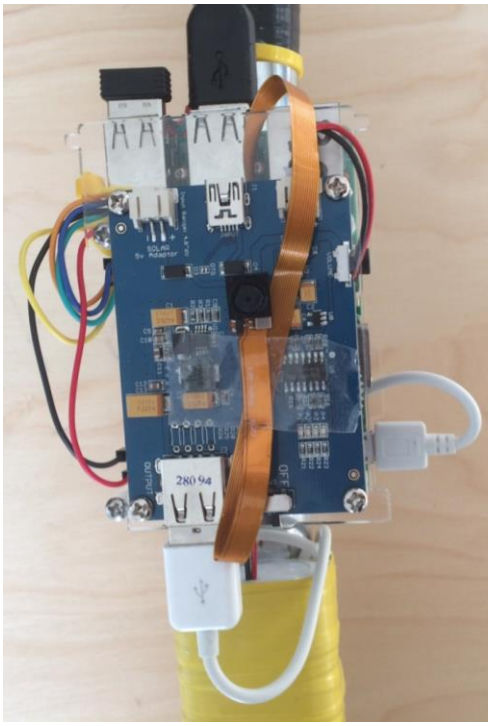
- **Assemblage milieu de la canne (Arduino, Capteurs Ultrasons)**

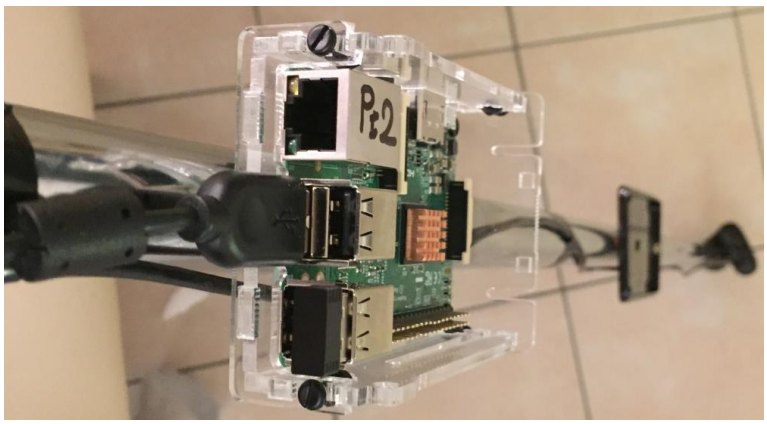




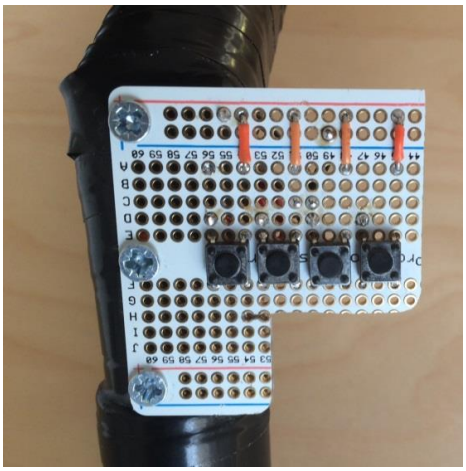


- **Assemblage haut de la canne (Raspberry+alimentation) :**





- **Etage télécommande :**



- **Canne**

BIBLIOGRAPHIE :

http://www2.univ-paris8.fr/ingenierie-cognition/master-handi/etudiant/projets/site_sam/aveugles/universite_av.htm
http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5333499
https://books.google.fr/books?id=bzxZkEcLo1kC&pg=PA919&lpg=PA919&dq=navigation+visually+impaired+people+line+cane&source=bl&ots=W_QKzSsRcD&sig=kaWQgnhWx6dtUFAB21vavD_iqKo&hl=fr&sa=X&ved=0ahUKEwjb5JyP63KAhUC8RQKHUpZCDAQ6AEIJjAB#v=onepage&q=navigation%20visually%20impaired%20people%20line%20cane&f=false
http://www.bcu.ac.uk/news-events/news/a-pioneering-facial-recognition-cane-for-the-blind
http://www.sciencedaily.com/releases/2015/05/150508082625.htm
http://www.visioptronic.fr/fr
https://urbanful.org/2015/03/25/9-advances-helping-visually-impaired-people-navigate-cities/
http://kamali.e-monsite.com/pages/avantages-et-inconvenient.html
http://www.geckogeek.fr/tutorial-opencv-isoler-et-traquer-une-couleur.html
http://www.shervinemami.info/introToOpenCV.html
http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html
https://realpython.com/blog/python/face-recognition-with-python/
http://www.digitaltrends.com/computing/arduino-vs-raspberry-pi/#ixzz4236KE8NU
http://nivea.psycho.univ-paris5.fr/PourLaScience/AuvrayORegan.pdf
http://sti.discipline.ac-lille.fr/Ens-Scienc-et-Techno/enseignements-d-exploration/enseignement-d2019exploration-creation-et-innovation-technologique/ressources-cit/Bruay/CIT%20DISPOSITIF%20DE%20SECURITE%20AUTOMOBILE/Documents%20ressources/pdf/Le%20telemetre%20Laser%20a%20balayage.pdf
https://facmed.univ-rennes1.fr/wkf/stock/RENNES20110331095543cpiszkorUS_PACES_2010-2011.pdf
https://www.arduino.cc/en/Tutorial/Calibration
https://itechnofrance.wordpress.com/2013/05/09/utilisation-dune-photo-rsistance-avec-larduino/
http://www.manuel-esteban.com/arduino-capteur-de-luminosite/