



PROJET CYBATHLON M1

RAPPORT

Détection d'objets pour aider des personnes en situation de handicap

Étudiants :
Robin SOARES

Encadré par :
Dominique BÉRÉZIAT
Isabelle BLOCH

Repository GitHub : <https://github.com/srsrb/Cyathlon>

9 juillet 2024

Table des matières

1	Bibliographie	2
2	Introduction	3
2.1	Contexte	3
2.2	Données	3
3	Tâche confiée	4
4	Solutions envisagées	4
4.1	Présentation d'OpenCV	4
4.2	Detection de la bande adhésive	4
4.3	Detection du tapis	5
5	Conclusion	7

1 Bibliographie

Références

- [1] Wang K., Qu L., Chen L., Gu Y. & Zhang X. (2016). “Non-flat Ground Detection Based on A Local Descriptor”.
- [2] I. Paik, J. Oh & H. Kang (2004). “Ground Plane Detection Method using monocular color camera”. ICCAS2004, 588-591.
- [3] Chen, T. W., Chen, Y. L., & Chien, S. Y. (2008, October). Fast image segmentation based on K-Means clustering with histograms in HSV color space. In 2008 IEEE 10th workshop on multimedia signal processing (pp. 322-325). IEEE.
- [4] Hema, D., & Kannan, D. S. (2019). Interactive color image segmentation using HSV color space. Sci. Technol. J, 7(1), 37-41.
- [5] Kar, A., & Deb, K. (2015, May). Moving cast shadow detection and removal from Video based on HSV color space. In 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT) (pp. 1-6). IEEE.
- [6] Bankhele, A., Patil, N., Waghmare, P., & Patil, S. (2021). Shadow Detection and Removal Technique using CNN. International Journal for Research in Applied Science and Engineering Technology, 9(8), 1748-1755.
- [7] Bibliothèque pytt3 : <https://pypi.org/project/pytt3/>

2 Introduction

2.1 Contexte

Le Cybathlon est un projet à but non lucratif de l'EPFZ (École Polytechnique Fédérale de Zurich) qui met au défi des équipes de développeurs d'universités, d'entreprises et d'ONG du monde entier de mettre au point des technologies d'assistance utilisables au quotidien, avec et pour des personnes en situation de handicap. La compétition porte sur huit disciplines au total. Les parcours reproduisent des activités de la vie quotidienne pour ainsi montrer à quel point la technologie peut améliorer la vie des utilisateurs et utilisatrices. Plusieurs tâches peuvent bénéficier d'une aide informatique, qu'il s'agisse de la détection d'objets ou de la planification de trajectoire. Le premier Cybathlon a eu lieu le 8 octobre 2016, à la Swiss Arena de Kloten au nord de Zurich en Suisse. C'était la première compétition internationale de ce type et 66 participants de 25 nations se sont affrontés devant 4 600 spectateurs.

Notre projet porte essentiellement sur la partie Vision de cette compétition (*Vision Assistance Race*, abrégée en VIS). La technologie d'assistance à la vision intelligente développée dans cet ensemble d'épreuves vise à améliorer la qualité de vie et l'autonomie des personnes souffrant d'une déficience visuelle grave ou d'une perte totale de vision. S'il existe sur le marché une large gamme d'aides à la vision basées sur des approches variées (vision par ordinateur, intelligence artificielle, etc.), leurs fonctions sont généralement restreintes à des domaines précis (par exemple lire un texte à voix haute ou identifier une couleur) et sont parfois peu pratiques à utiliser.

2.2 Données

Les données expérimentales sont acquises avec une webcam Logi HD 1080p fournie par l'ISIR et un téléphone portable. Notre premier choix s'était porté sur une Kinect mais nous avons appris par la suite que l'équipe de l'ISIR utilisait un plastron pour la compétition de cette année. Celui-ci, qui pèse un poids déjà conséquent, comporte uniquement des petites caméras car plus légères et par conséquent plus facilement transportables. L'athlète dispose aussi d'un téléphone portable nous permettant d'avoir une meilleure qualité pour le traitement d'image. Afin de reproduire au mieux les conditions de la compétition, nous utilisons donc la webcam dont la bonne résolution aide à avoir un rendu facilement exploitable et un téléphone portable. Nous travaillons avec des données provenant d'une vidéo couleur capturée en temps réel, dont nous détaillerons l'acquisition dans une prochaine section (cf section 3).

3 Tâche confiée

Le concours Cybathlon se déroule sur plusieurs tapis de gymnastique bleus mis bout à bout en longueur, sur lesquels sont disposées les différentes épreuves. L'athlète pouvant se déplacer du centre du tapis au fur et à mesure, les chercheurs de l'ISIR nous ont donc demandé de fabriquer un algorithme permettant de replacer l'athlète pour qu'il soit face à la prochaine épreuve et au milieu du tapis. Pour nous aider, des bandes adhésives noires sont placées sur les tapis pour délimiter chaque épreuve. En premier lieu nous pensions qu'il était nécessaire de détecter ces bandes adhésives via la caméra qui vise le sol pour placer l'athlète face aux épreuves suivantes. Cependant, nous avons appris qu'une corde était disposée entre chaque épreuve sur les tapis, ce qui permettra à notre athlète de se réorienter et donc de s'affranchir de cette étape. À l'aide de la caméra placée sur le torse de l'athlète, notre défi est de réussir à détecter le tapis et à calculer le déplacement nécessaire de l'athlète pour qu'il soit bien au milieu de celui-ci.

4 Solutions envisagées

Nous avons décidé d'implémenter tous nos algorithmes en Python pour ce projet, en intégrant les bibliothèques OpenCV et NumPy. C'est un langage facile à prendre en main, qui comporte de nombreuses bibliothèques et fonctions intégrées, ce qui facilite grandement la tâche par rapport à des langages plus bas niveau. Ces derniers (comme le C, par exemple) sont certes bien plus efficaces, mais l'efficacité n'est pas notre but premier et Python 3 est suffisamment rapide pour les tâches que nous avons à effectuer.

4.1 Présentation d'OpenCV

OpenCV (acronyme de Open Source Computer Vision Library) est une bibliothèque open source dédiée à la vision par ordinateur et au traitement d'images. Développée en C++ mais avec des interfaces disponibles pour de nombreux langages dont Python, cette librairie présente une multitude d'outils puissants pour la manipulation, l'analyse et la compréhension d'images et de vidéos. Pour les webcams, OpenCV utilise la classe VideoCapture afin d'accéder à la caméra et récupérer les *frame*. Chaque *frame* capturée par la webcam est essentiellement un tableau NumPy 2D (pour les images en niveaux de gris) ou 3D (pour les images couleurs, notre cas ici) contenant des valeurs de pixels. Celles-ci représentent l'intensité de la lumière à chaque point de l'image. Le format des données d'image dépend du format de couleur et de la profondeur des images capturées par la webcam. Les formats de couleur courants incluent RGB et BGR (valeurs arrangées dans l'ordre inverse du format RGB), et les types de données courants pour les valeurs de pixels incluent des entiers non signés sur 8 bits (uint8) ou 16 bits (uint16), en fonction des paramètres de la caméra.

4.2 Détection de la bande adhésive

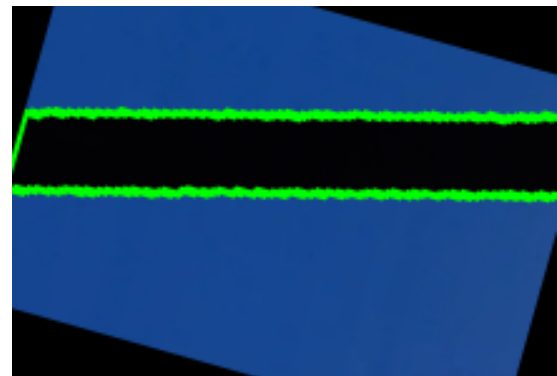
Pour détecter la bande adhésive noire, nous utilisons donc OpenCV et Python. Le défi est de différencier la couleur noire du bleu et d'informer l'athlète de la rotation à faire pour que la bande soit parallèle à la caméra.

Pour cela, nous avons simplement appliqué un masque modifiant la valeur des pixels à 0 (noir) lorsqu'ils sont de couleur noire et à 255 (blanc) pour le reste. Ensuite, il suffit de détecter les contours dans ce masque et à calculer l'angle de rotation pour que la bande soit horizontale. Ce code se trouve dans le fichier *bande.py*.

Nous avons commencé par tester cette approche sur une simple image qui imite la situation.



(a) Image d'une bande adhésive sur un tapis de gymnastique bleu



(b) Figure (a) qui a subi une rotation via l'algorithme

FIGURE 1 – Exemple sur lequel appliquer notre algorithme représentant en fond le tapis de gymnastique bleu et la bande adhésive noire

Comme dit plus tôt, nous n'avons pas poussé cette étape plus loin sachant qu'elle ne serait pas utile à la compétition.

4.3 Détection du tapis

Premièrement, nous avons essayé d'implémenter des méthodes provenant d'articles de recherche.

L'article "Non-flat Ground Detection Based on A Local Descriptor" (cf [1]) présente une méthode pour détecter le sol via la vision stéréo. La vision stéréo nous permet d'obtenir une carte de disparité. L'article décrit un descripteur appliqué à cette carte permettant d'obtenir une image distinguant efficacement le sol des murs ou autres. Puis, l'idée est de segmenter cette image en régions de superpixels à l'aide de l'algorithme SLIC (*Simple Linear Iterative Clustering*). Enfin, un réseau de neurones est utilisé pour classifier ces régions de superpixels comme étant du sol ou non.

Ou encore l'article scientifique "Ground Plane Detection Method using monocular color camera" (cf [2]) présente une méthode utilisant l'IPD (*Identical Pixel Detection*) qui utilise une zone de référence pour décider si un pixel appartient au sol ou non en les comparant aux pixels de la zone de référence. Les auteurs proposent d'utiliser la méthode de Prewitt pour détecter les bords. Il semble nécessaire de l'utiliser pour compenser les limitations de l'IPD.

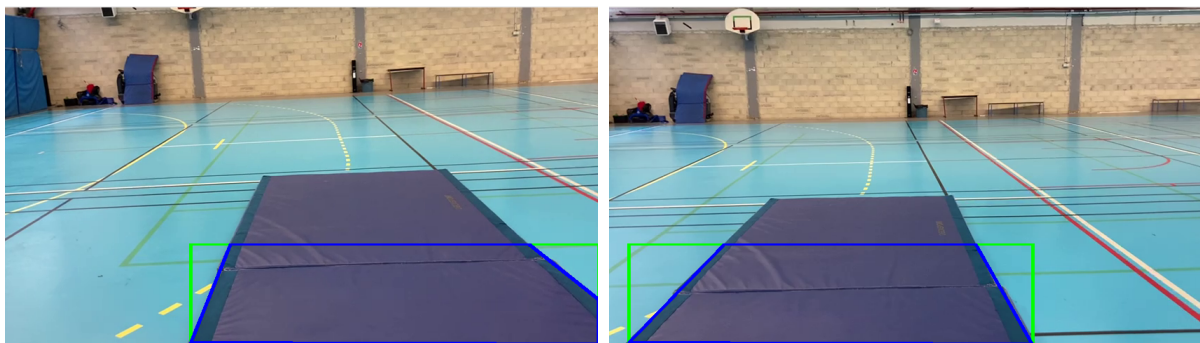
Cependant, ces méthodes furent assez complexes à implémenter à notre projet et les résultats n'étant pas fructueux nous avons décidé d'utiliser un algorithme plus basique.

Le code pour détecter le matelas se trouve dans le fichier *replacement.py*. Premièrement, nous délimitons une région d'intérêt pour trouver notre matelas. Nous divisons la hauteur de notre *frame* par $\frac{10}{7}$ pour ne récupérer que le bas de celle-ci. Ensuite, nous

filtrons les pixels de couleur bleue dans cette région d'intérêt et donnons les contours dans le masque obtenu. Puis, nous dessinons un rectangle autour du plus grand contour trouvé. Ce rectangle nous permet finalement de calculer le centroïde pondéré des contours et d'ajuster la position de l'athlète en fonction de ce centroïde. Quatre possibilités peuvent s'afficher sur le terminal :

- “Décalage vers la droite”
- “Décalage vers la gauche”
- “Au centre”
- “Pas de matelas détecté”

Nous avons tenté de fournir ces messages oralement en utilisant des bibliothèques telles que pyttsx3 (cf. [7]), cependant, leur performance s'est avérée insuffisante, surtout lorsqu'il s'agit de traiter des données en temps réel.



(a) Décalage vers la droite - Détection d'une situation où l'utilisateur doit se déplacer vers la droite.

(b) Décalage vers la gauche - Détection d'une situation où l'utilisateur doit se déplacer vers la gauche.



(c) Au centre - L'image Détection d'une situation où l'utilisateur est au centre du matelas.

FIGURE 2 – Démonstration de l'algorithme de remplacement avec une vidéo prise avec un téléphone portable

5 Conclusion

Malgré certaines difficultés rencontrées, notamment dans la mise en œuvre des techniques de détection du sol, nous avons pu concevoir un algorithme robuste de détection de matelas, qui servira certainement à réorienter l'athlète correctement lors de la compétition. Cependant, des améliorations restent possibles, notamment en ce qui concerne la précision de la détection. De plus, des tests plus approfondis avec le groupe de recherche pour le Cybathlon doivent être effectués pour s'assurer de l'efficacité de celui-ci.