Mission de fin d'études

Find Your Way

AUBAY

Maître de stage :
Nathan CANTAT

Jury de stage :
Arnaud BANNIER

Auteur :
Nicolas CISTERNAS

28 février - 31 août 2022

Résumé

Vous donnerez une description du sujet et des grandes parties de l'ouvrage. Il s'agit ici de motiver un lecteur potentiel en lui décrivant le contenu du document et les implications que l'on peut en tirer,

Abstract

Abstract in English

Table des matières

1 Remerciements

Je remercie Éric REMILLERET et Nathan CANTAT, mes maîtres de stage, ainsi que Wael NASR, pour m'avoir donné l'opportunité de rejoindre Aubay Innov', qui s'est avérée être une expérience intéressante et instructive, ainsi que pour l'aide à la relecture de ce rapport.

Je remercie Anne-France GALLAND, qui a largement contribué à la phase de suivi et d'organisation du projet, par ses interventions lors des réunions hebdomadaires et pour le temps qu'elle a pris pour la relecture de ce rapport.

Je remercie Jean-Philipe CUNNIET, mon tuteur de stage ainsi qu'Arnaud BANNIER, le jury de cette mission de fin d'études, pour le temps qu'ils ont passé à relire et évaluer ce rapport de stage ainsi que la soutenance de stage.

Je tiens à remercier toute l'équipe avec laquelle j'ai travaillé durant mon stage : Nicolas GUILLERMAIN et Mathieu MONNERET avec qui j'ai travaillé lors de la phase d'état de l'art et de preuve de concept. Mélissa WANG et Jeoffrey MENUDIER avec qui j'ai principalement travaillé lors de la phase projet. Jean-Baptiste CHANIER, Victor CHAVEROT, Jean-Noël CLINK, Ophélie PHONCHAREUN, Miora RASOLOFONERA et Rémi VIDAL avec qui j'ai échangé tout au long du stage concernant leurs parties du projet.

Je tiens également à remercier la direction générale d'Aubay pour leur accueil et leur écoute ainsi qu'Ophélie CHEVALIER, campus manager, pour tous les événements qu'elle a pu organiser et qui ont permis d'établir un climat chaleureux entre les différents groupes.

Enfin, je remercie les autres étudiants d'Aubay Innov' avec lesquels j'ai eu l'occasion d'échanger aussi bien humainement que techniquement.

2 Introduction et contexte

Mon stage s'est déroulé dans une entreprise appelée Aubay, pour une durée de 6 mois (du 28 février 2022 au 31 août 2022). En détail, Aubay est une entreprise de services numériques (ESN) qui a été fondée par Christian Aubert en 1998 et dont le siège social est situé au 13 rue Louis Pasteur à Boulogne-Billancourt. L'entreprise est spécialisée dans les domaines liés à la finance, à l'assurance et à la banque et est également impliquée dans divers marchés, tels que les télécoms, les services, les réseaux, l'énergie et les transports. Aubay accompagne la transformation et la modernisation des systèmes d'information de ses clients. Ils opèrent sur des marchés à forte valeur ajoutée, en France comme en Europe. C'est un acteur référent de la transformation digitale. Son secteur d'activité est centré autour du conseil sur tout type de projet technologique. C'est une ESN cotée en Bourse (SBF 250) et 46% du capital est détenu par les managers. Les chiffres clés concernant Aubay sont présentés dans la Figure 1 ci-après. En 2022, l'entreprise emploie 7306 travailleurs, dont 2728 en France. Selon son site internet, Aubay est implantée dans 7 pays européens, et a réalisée un chiffre d'affaires de 470,6 M€ en 2021. D'ailleurs, ces dernières années, l'entreprise a connu une forte croissance de 10,4% en 2021, qui coïncide avec une augmentation constante des effectifs de l'entreprise, passant de 4600 employés en 2015 à plus de 7000 aujourd'hui.

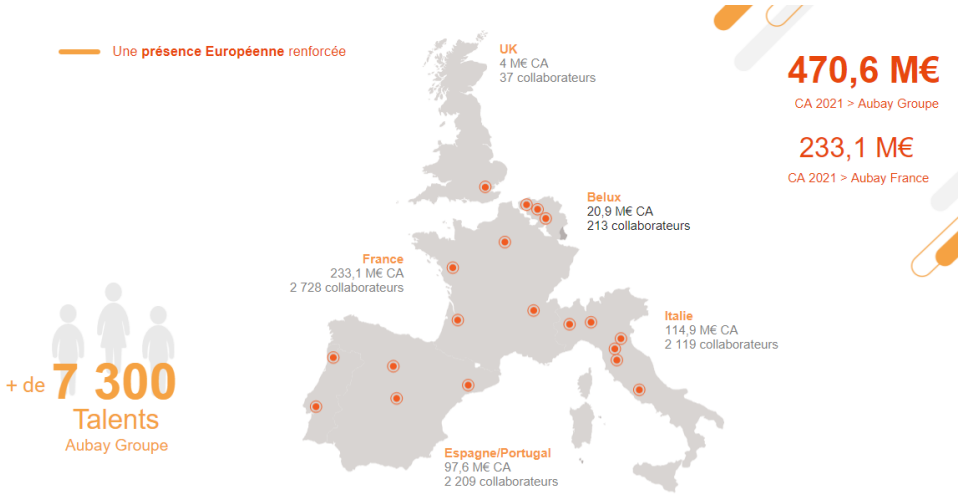

FIGURE 1 – Chiffres clés pour Aubay en 2021.

J'ai eu le privilège d'effectuer mon stage au sein de l'unité "Aubay Innov'" qui est la division d'Aubay France dédiée à la recherche et à l'innovation. Ses membres sont impliqués dans différents projets, chacun lié à la data science, l'analyse de données ou tout autre domaine liés aux nouvelles technologies numériques. L'objectif de l'unité est d'acquérir les connaissances et le savoir-faire pour construire des solutions durables et innovantes adaptées aux besoins futurs des clients. Les projets dirigés dans la cellule innovation d'Aubay ne sont donc soumis à aucun client pendant leur réalisation, ce qui laisse l'opportunité aux stagiaires d'expérimenter autant qu'ils le souhaitent durant leur stage. Chaque année, cette unité donne la chance à des dizaines de stagiaires d'améliorer ou de perfectionner leurs connaissances relatives à la science des données en leur laissant l'opportunité de découvrir et d'expérimenter les technologies les plus innovantes disponibles dans les domaines de la recherche. Par ailleurs, l'unité "Aubay Innov'" est largement considérée comme une source de recrutement pour l'entreprise, qui a souhaité cette année engager environ 800 nouveaux collaborateurs en France.

C'est dans ce cadre que ma mission de fin d'études a débuté. Le projet "Find Your Way" (FYW), qui représente l'expérience que je vais détailler dans ce document, a été lancé en février 2022 et a pour objectif de créer une application embarquée sur des lunettes, basée sur des algorithmes capables de reconnaître les éléments de l'environnement immédiat d'un utilisateur malvoyant pour le guider vers des lieux ou l'avertir d'obstacles présents sur son chemin tels que des chaises ou une personne par exemple. Un objectif majeur de ce projet est également de localiser l'utilisateur dans son environnement afin de lui permettre de retrouver son chemin jusqu'à un endroit précédemment enregistré et de le guider avec des indications de direction et d'orientation dans ses déplacements en intérieur en temps réel, le tout répondant à un système de commandes vocales.

Les systèmes d'aide automatisés au déplacement de personnes malvoyantes existent déjà sur le marché et nécessitent de fournir une carte du bâtiment avec les lieux importants préalablement renseignés afin de rendre le guidage possible. Ce projet vise à s'affranchir de cette contrainte afin de permettre une plus grande polyvalence pour ce genre d'outil. Le projet a été séparé en plusieurs parties délimitées par des dates clés qui sont présentées dans la Figure 2 avec les livrables associés à chaque fin de phase. La prise en main du projet a démarée début février, s'en est suivi la partie concernant les états de l'art (EA sur la Figure 2) qui devait être réalisée entre mi-février et mi-mars en parallèle du Design Thinking qui nous a permis de cadrer

et définir le projet par rapport aux besoins de l'application et les tâches à accomplir. La phase de preuve de concept qui a suivi a duré jusqu'à mi-mai. Une fois ces étapes réalisées nous avons pu passer à la phase projet afin de réunir toutes les preuves de concept et concevoir l'application de démonstration nécessaire à la journée des stagiaires (JDS) du 7 juillet, moment phare pour le pôle innovation de chez Aubay où tous les stagiaires présentent leurs projets lors d'une démonstration auprès des directeurs généraux et directeurs commerciaux de l'entreprise.

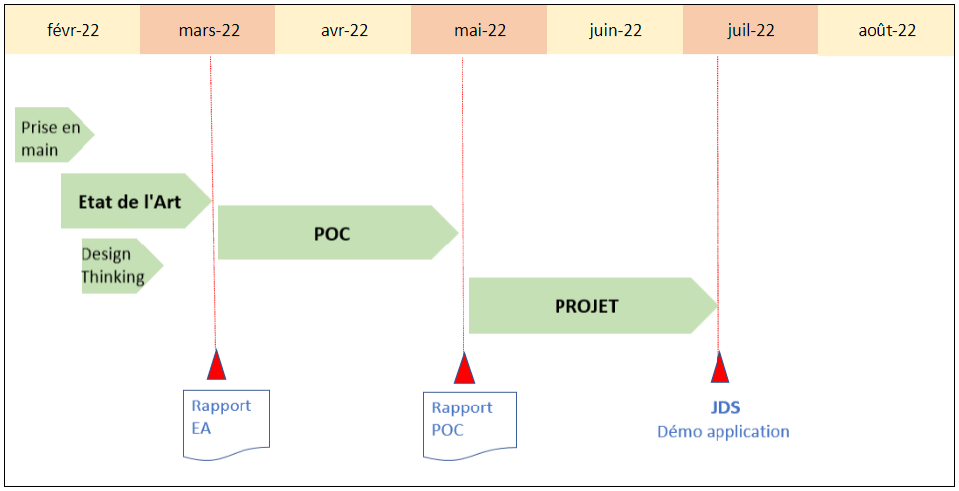

FIGURE 2 – Planning prévisionnel du projet FYW.

J'ai réalisé ce projet dans une équipe de 11 stagiaires, tous en stage de fin d'études. Étant donné les multiples parties concernant le projet, à savoir la détection d'objets, la localisation de l'utilisateur et la gestion des interactions vocales, nous avons dû nous séparer en 3 groupes afin de réaliser les états de l'art et les preuves de concept. J'ai personnellement travaillé sur la partie s'intéressant à la localisation et le guidage de l'utilisateur dans son environnement.

3 État de l'art

3.1 Recherches bibliographiques

Dans cette section je présente les recherches bibliographiques que j'ai pu effectuer avec mon groupe lors de la phase d'état de l'art. Cet état de l'art concerne la partie de localisation et de guidage de l'utilisateur. Nous avons retenu 8 publications, nous les avons confrontées et comparées afin de n'en sélectionner qu'une sur laquelle nous allions nous concentrer lors de la phase de preuve de concept. Je présente ici 4 des publications les plus pertinentes sur le sujet.

3.1.1 Wearable Travel Aid for Environment Perception and Navigation of Visually Impaired People

L'appareil consiste en une caméra grand public rouge, vert, bleu avec de la profondeur (RGB-D pour red, green, blue, depth) et une unité de mesure inertielle (IMU pour Inertial Measurement Unit), c'est un capteur qui consiste généralement en des gyroscopes pour mesurer des vitesses angulaires et des accéléromètres pour mesurer la force [1]. Ces appareils sont montés sur une paire de lunettes et reliés à un téléphone. L'appareil proposé dans cette solution se sert de la continuité de la hauteur du sol entre les images adjacentes poursegmenter le sol avec précision et rapidité pour ensuite chercher la direction du mouvement en fonction du sol. Un réseau de neurones à convolution (CNN pour Convolutional Neural Network) léger est utilisé pour la reconnaissance d'objets (PeelNet avec l'ensemble de données "MS COCO" contenant des images de dimensions 640 x 640). Son schéma de fonctionnement est présenté dans la figure 3. Il permet de récupérer des informations concernant les endroits aux alentours et l'orientation des objets environnants. Le schéma de fonctionnement du système est présenté dans la Figure 4.

Le système de navigation contient un module de localisation intérieur qui va nous intéresser : un algorithme VSLAM (Visual Simultaneous Localization and Mapping) est utilisé. SLAM est un problème de computer vision visant à traquer les mouvements d'un module en se basant sur ce qu'il voit. Il faut traiter les objets dynamiques qui entrent et sortent du champ de vision pour ne pas les prendre en computer dans l'estimation de la position du module. L'estimation se fait en trouvant des points clés sur les images successives. Visual SLAM se sert des informations récuperées pour trianguler la position 3D du module.

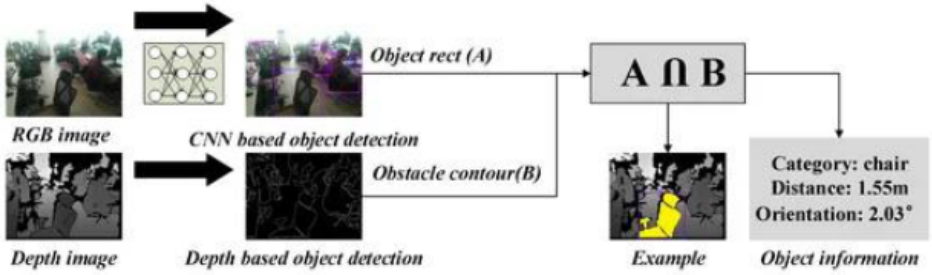

FIGURE 3 – Schéma présentant le fonctionnement du système de reconnaissance d'objets.

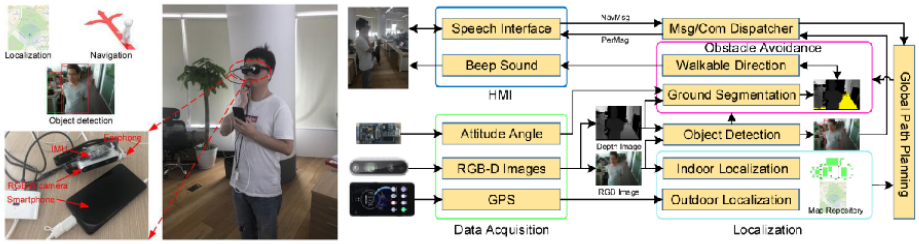

FIGURE 4 – Schéma de fonctionnement du système proposé.

3.1.2 CDSFusion : Dense Semantic SLAM for Indoor Environment Using CPU Computing

La solution CDSFusion [4] se sert d'images RGB comme l'article précédent ainsi qu'un capteur IMU comme paramètres d'entrée et est composée de 3 modules imagés dans la Figure 5 :

3.1.2.1 Le module VIO Le module d'odométrie visuelle-innertielle (VIO pour Visual-Inertial Odometry) se sert des entrées pour estimer avec précision la position afin de proposer une trajectoire. Ce module VIO est basé sur VINS-Mono. VINS-Mono est un framework SLAM en temps réel pour les systèmes visuo-inertiels monoculaires. Il utilise une méthode de fenêtre glissante basée sur des optimisations pour fournir une odométrie visuelle-inertielle de haute précision. Les features FAST (Features from accelerated segment test) ont été adaptées pour accélérer le VIO à la place des feature Shi-Thomas (une manière de détecter les coins sur une image) et la profondeur a été introduite afin d'obtenir une échelle plus précise. Les résultats

expérimentaux montrent que les features FAST augmentent la rapidité du système d'une manière plus conséquente que les features Shi-Tomas et ORB avec la même précision et robustesse. Ce module est composé de 3 parties :

Visual-Inertial Frontend : prend en charge le traitement des données issues des capteurs. Les mesures effectuées par le capteur IMU sont pré-intégrées entre deux images consécutives, le frontend de vision détecte les FAST et les traque entre les images consécutives en utilisant l'algorithme KLT optical flow.

Back-End : est utilisé pour fusionner les mesures traitées afin d'obtenir l'estimation de la position. Une optimisation non linéaire est utilisée pour re-localiser et optimiser le calcul de la position en fonction de la boucle détectée, en utilisant le solveur Ceres.

Le module de détection de boucle : permet de relocaliser et optimiser le calcul de la position en fonction de la boucle détectée. En effet, une boucle est détectée lorsque l'on a réalisé une boucle dans le parcours d'un chemin, il devient alors inutile de recalculer la position tant que l'utilisateur se situe dans cette boucle, on peut alors facilement optimiser le calcul de la position en se basant sur des calculs précédemment effectués. Cela se base sur la bibliothèque DBoW2 qui est à l'état de l'art de la reconnaissance d'endroits par sac de mots (bag of words approach). De même, lorsqu'une boucle est détectée, une optimisation est possible pour le calcul de la position globale. Cette optimisation est similaire à la méthode VINS-Mono.

3.1.2.2 Le module de segmentation sémantique La segmentation sémantique résulte d'images RGB en entrée qui sont acquises en temps réel en utilisant le module de segmentation PSPNet. Le module de segmentation sémantique traite chaque image RGB et retourne des vecteurs indiquant la probabilité d'appartenance à une classe pour chaque pixel. Ils classifient et colorent chaque pixel en fonction de la plus haute probabilité d'appartenance à une classe. L'image segmentée finale est composée des pixels colorés et est transmise au module de reconstruction 3D.

3.1.2.3 Le module de reconstruction 3D Le nuage sémantique local est généré en utilisant une image sémantique (générée par le module précédent) et une carte de profondeur. Ce nuage local servira à produire un nuage global une fois qu'il sera combiné avec les estimations de position de la caméra depuis le module VIO. Pour construire une carte 3D globale qui

soit précise, un modèle basé sur des voxels est utilisé pour filtrer le bruit et extraire un mesh global. À chaque image importante (Keyframe), la carte de profondeur courante est transformée en nuage de points 3D, ensuite les options Voxblox et FAST sont adaptées, le nuage local de points 3D est transformé en mesh local et ensuite ce mesh local est intégré dans le mesh global. L'ensemble de cette procédure est réalisée en temps réel sur CPU. Voxblox est une bibliothèque de cartographie volumétrique basée principalement sur les TSDF (Truncated Signed Distance Fields).

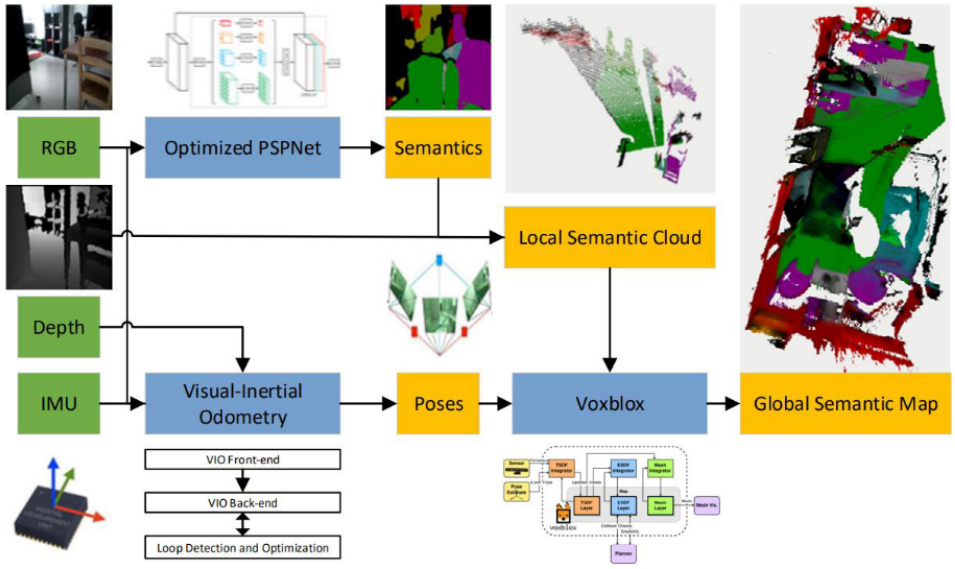

FIGURE 5 – Schéma présentant le fonctionnement du système proposé.

3.1.3 Towards Real-time Semantic RGB-D SLAM in Dynamic Environments

Cet article [3] présente une méthode de SLAM s'inspirant des features ORB (Oriented FAST and Rotated BRIEF). Le module de segmentation sémantique est une adaptation de SegNet, un réseau de neurones léger qui permet un traitement en temps réel. La segmentation se concentre sur des objets dynamiques dont les caractéristiques ne seront pas utilisées dans la création de la carte locale. La segmentation est faite uniquement sur les images clés les plus récentes, accélérant grandement le processus. La segmentation peut être très rapide, mais reconnaît principalement les objets qu'elle connaît déjà et est moins efficace dans des milieux inconnus avec des objets dynamiques nouveaux. Ce module essaye de résoudre ce problème. Pour chaque nouvelle image l'idée est d'utiliser l'algorithme du K-means pour segmenter l'image de profondeur en N clusters. Chaque cluster est considéré comme faisant partie du même objet. Pour chaque cluster, une erreur de reprojection est calculée. Si une des erreurs moyennes est relativement supérieure aux autres, le cluster est considéré comme un objet dynamique et les points caractéristiques de l'objet seront supprimés du traitement. Cette manière de traiter les images permet de réduire le taux de faux positifs. Les comparaisons sont effectuées d'une image clé à une autre puisqu'elles se ressemblent beaucoup, mais aussi entre la première estimation et les cartes locales obtenues.

3.1.4 ORB-SLAM3 : An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM

ORB-SLAM3 [2] est une amélioration d'ORB-SLAM2, présenté en Figure 6, qui consistait à travailler sur trois tâches simultanément : le suivi, la cartographie locale et la fermeture de boucle. Dans sa partie de suivi, ORB-SLAM2 fait correspondre les caractéristiques image par image et les compare avec une carte locale pour trouver l'emplacement exact de la caméra en temps réel. Il procède à un ajustement du faisceau en fonction du mouvement afin de minimiser l'erreur de reprojection. Vient ensuite la partie cartographie locale dans laquelle ORB-SLAM2 créer des cartes locales et les optimise en utilisant des algorithmes tels que Iterative Closest Point (ICP) et effectue un ajustement local du faisceau afin de calculer la position la plus probable de la caméra. Enfin, il utilise l'optimisation du graphe des positions pour corriger la dérive accumulée et effectuer une fermeture de boucle. Il est nécessaire d'effectuer un ajustement groupé après la fermeture de la boucle, afin que le robot se trouve à l'emplacement le plus probable dans la carte corrigée. Après l'ajout d'une image clé à la carte ou l'exécution d'une fermeture de boucle, ORB-SLAM2 peut démarrer un nouveau fil d'exécution qui effectue un ajustement de l'ensemble de la carte afin que l'emplacement de chaque image clé et des points dans celle-ci obtienne une valeur d'emplacement ajustée.

Pour traiter une image RGB-D, on calcule d'abord les caractéristiques ORB sur l'image RGB, ensuite on estime les coordonnées de l'image de gauche depuis une paire d'images. Un point est associé à "proche" ou "éloigné" en fonction de sa profondeur. Chaque dénomination a des caractéristiques utiles, un point proche sera représentatif pour l'échelle, la translation, la rotation alors qu'un point éloigné sera surtout utile pour la rotation et devra être supporté par un plus grand nombre d'images. À l'initialisation, le système nécessite seulement les premières images clés pour estimer la première carte locale en utilisant les informations de profondeur. L'insertion d'une nouvelle image clé est importante, car elle définit le nouvel environnement sur lequel se base l'estimation du mouvement de la caméra. Elle suit l'idée établie par la première version d'ORB-SLAM, d'insérer régulièrement de nouvelles images clés, même si cela implique de devoir les retirer si elles sont redondantes. L'ajout de points clés proches et éloignés permet un nouveau seuil d'apparaître pour déterminer si une nouvelle image clé est nécessaire. Si le nombre de points proches devient inférieur à un seuil, il sera nécessaire d'insérer une nouvelle image clé comportant au moins un certain

nombre de points proches supérieur à un autre seuil.

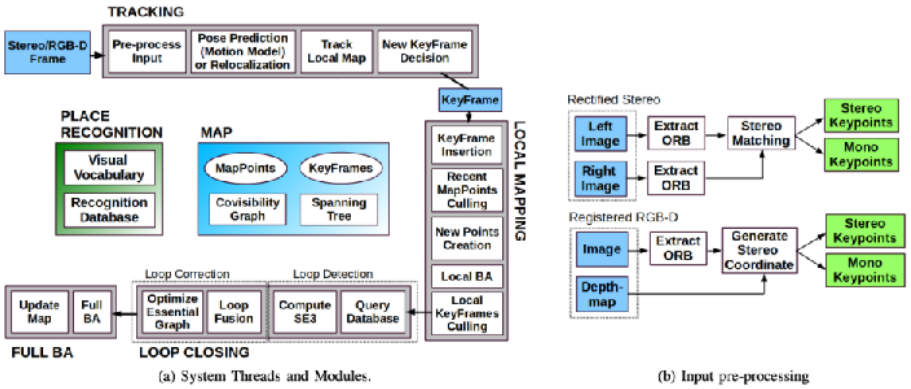

FIGURE 6 – Schéma présentant le fonctionnement d'ORB-SLAM2.

ORB-SLAM3 introduit l'utilisation de mémoire à court, moyen et long terme. En se servant des données que l'on a déjà vues, on peut retrouver facilement les boucles et ainsi réduire les erreurs de dérive. Ce système est présenté en Figure 7

Le thread de tracking traite chaque nouvelle image afin de vérifier si elle représente une nouvelle image clé afin de la localiser dans la carte active de l'atlas. Si le tracking est perdu, le système essaye de se relocaliser en cherchant des repères visuels qu'il a déjà rencontrés par le passé. Si cela ne fonctionne pas, il créera une nouvelle carte qui sera la nouvelle carte active dans l'atlas. On dispose donc d'un atlas de cartes locales, un ensemble de cartes déconnectées qui seront peut-être combinées avec le temps grâce au système de fermeture de boucles.

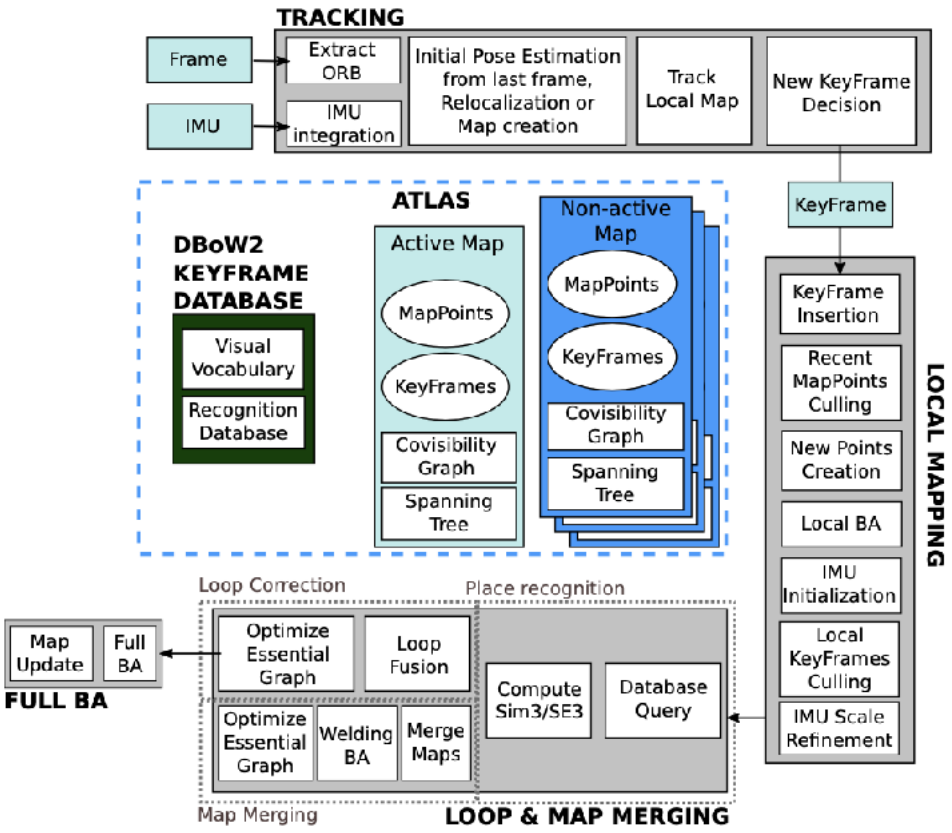

FIGURE 7 – Schéma présentant le fonctionnement d'ORB-SLAM3.

3.2 Synthèse

L'article "Wearable Travel Aid for Environment Perception and Navigation of Visually Impaired People" est très détaillé et demande beaucoup de temps pour saisir chaque subtilité. Au début de l'article, les auteurs listent les solutions qui existent déjà et expliquent leurs limitations respectives. Chaque partie de la solution est expliquée et peut être trouvé dans leur travaux précédents. Cette solution prendrait beaucoup de temps à implémenter puisque les auteurs font surtout une description de leur travail sans mettre à disposition de code source. Il faudrait parcourir tous leurs travaux précédents pour comprendre l'intégralité de leur solution. De plus il faudrait adapter les travaux proposés afin de créer la carte locale en temps-réel plutôt qu'avant, puisque c'est une des contraintes imposées par nos superviseurs.

L'article "CDSFusion : Dense Semantic SLAM for Indoor Environment Using CPU Computing" est très prometeur mais n'est pas assez détaillé pour nous permettre de reproduire leur solution à partir de rien. De plus ils utilisent des capteurs IMU, que nous avons convenu de ne pas utiliser dans le cadre de ce projet.

L'article "Towards Real-time Semantic RGB-D SLAM in Dynamic Environments" apporte une solution de tracking de la localisation d'un module en temps réel, ce qui correspond à ce que l'on recherche dans le cadre de ce projet. Cette solution aide à garder en mémoire les directions prises jusqu'à arriver à un point donné. Cependant les auteurs ne parlent que peu d'ORB-SLAM et se focalisent surtout sur leur valeur ajouté permettant d'améliorer les résultats de ce dernier.

Enfin, l'article "ORB-SLAM3 : An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM" semble être le plus abouti que nous aillons pu lire, il est à l'état de l'art et obtient de meilleurs performances que ses concurants. L'article est assez bien expliqué et fourni un code source disponible sur Github. Leur solution semble réutilisable pour notre problème, cela apporterait un bon moyen de traquer une personne malvoyante dans une pièce. Certains aspects de la publication semblent compliqués à comprendre mais la méthode en elle même nous semble accessible. Il faudrait passer du temps sur leur code source afin de l'appréhender et se rendre compte de la faisabilité de sa mise en place.

3.3 Conclusion

Pour conclure sur cet état de l'art, nous avons découvert plusieurs méthodes permettant de traquer le mouvement d'une caméra dans un milieu intérieur, chacune avec son niveau de complexité et de faisabilité par rapport à nos contraintes qui étaient de n'utiliser aucun autre capteur qu'une camera. Notre but final était de fournir un module permettant de guider une personne d'un point A à un point B avec l'aide de notre module créant une carte locale de l'environnement en se basant sur des points et des lieux d'intérêt, s'emboîtant avec les modules de traîtement du langage naturel, et le module de détection d'objets. L'article "ORB-SLAM3 : An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM" nous a semblé être le plus pertinent pour arriver à notre but. En effet nous pourrions nous baser sur leur solution de tracking et construire par dessus pour répondre au besoin fonctionnel de ce projet. De plus le fait que les auteurs mettent à disposition leur code source donne une première vision de ce qui pourrait être réalisé en terme de performances avec notre propre environnement.

4 Les dimensions techniques du projet (8 à 12 p)

Description des objectifs/tâches qui vous ont été confiés, vous exposerez les difficultés rencontrées et la manière dont vous les avez abordées. Vous mettrez également en valeur l'originalité éventuelle de votre approche, les choix et décisions que vous aurez prises. Vous mettrez vos travaux et réalisations en perspective par rapport à l'ensemble du projet et son historique.

4.1 Phase de preuve de concept

Le projet FYW vise à aider les personnes malvoyantes à se repérer en milieu intérieur. La preuve de concept (POC) sur laquelle j'ai travaillée avec mon équipe se concentrait sur la création d'une carte locale de l'environnement en nous basant sur des points d'intérêts (POI) entre les images vues successivement, c'est la méthode SLAM.

4.1.1 Installation

Il a d'abord fallu se pencher sur l'installation de la solution que nous avions retenue, afin de construire notre module par dessus. Nous avons été capable de produire une documentation d'installation de ORB-SLAM3 sur le Windows sub-system for Linux (WSL2) et sur une machine virtuelle (VM). Ces documentations d'installation appartiennent à Aubay et ne peuvent être divulguées dans ce document. Ensuite il a fallu calibrer la caméra que nous utilisions pour fonctionner avec ORB-SLAM3. La calibration de la caméra assure que l'estimation de la position de la caméra est la plus proche de la position réelle de la caméra. Enfin, l'idée était de développer un algorithme de recherche du plus court chemin entre deux points donnés, par exemple entre la position de l'utilisateur et une destination précédemment sauvegardée.

4.1.1.1 General Notre méthode se base sur ORB-SLAM3, une solution open-source crée dans une université en Espagne. L'idée était simple, nous voulions l'installer et la faire fonctionner puis ajouter les fonctionnalitées nécéssaires à notre POC. Ce projet open-source a été développé en C++ et se base sur de nombreuses bibliothèques de code telles que Eigen, Pangolin, Sophus, DBoW2, g2o, Kalibr et ROS. Avant d'installer ORB-SLAM3 nous avons dû installer Eigen et Pangolin. Sophus, DBoW2 et g2o étant déjà inclus dans leur dépôt Github.

Nous avons essayé d'installer ORB-SLAM3 sur Windows 10, Mac OSX, Ubuntu 20.04 sur VirtualBox et enfin WSL2. L'installation sous Windows 10

a rapidement été déclarée impossible puisqu'ORB-SLAM3 a été développé pour un système Linux en grande majorité. L'installation sur Mac OSX n'a été que temporaire, le temps que l'on obtienne l'autorisation de travailler avec WSL2. Nous avons donc réalisé l'installation d'ORB-SLAM3 sur WSL2 et réussi à lancer le SLAM sur la vidéo d'exemple fournie. Cependant notre POC devait fonctionner avec un flux vidéo en temps réel provenant d'une caméra, et non une vidéo préenregistrée. Le problème était que bien que wsl2 ai accès aux appareils USB (en prenant en compte que pour cela il était nécéssaire de créer un noveau noyau Linux contenant les bons pilotes) et que la caméra soit visible dans les pilotes USB, le sous-système n'était pas capable de créer la vidéo dans /dev/video, rendant le flux vidéo innaccessible à notre application. Nous avons choisi de basculer sur un sysème Ubuntu 20.04 sur VM afin de disposer des fonctionnalités de base nécéssaires à notre avancée.

4.1.1.2 Eigen Eigen is a C++ template library for linear algebra : matrices, vectors, numerical solvers, and related algorithms. It is required to have an efficient ORB-SLAM3. To install Eigen, we had to search for the right version of this library. We have tried several versions which did not work for our installation of ORB-SLAM3. We eventually got to the 3.3.1 version. It suited all our needs.

4.1.1.3 Pangolin Pangolin is a set of lightweight and portable utility libraries for prototyping 3D, numeric or video-based programs and algorithms. It is used to remove platform-specific boilerplate and make it easy to visualize data. To install Pangolin, we had to look around on their GitHub. As a matter of fact, their latest version didn't suit the installation of ORB-SLAM3. We have tried several versions which did not work for our installation of ORB-SLAM3. We eventually got to the version 0.6 on their GitHub.

4.1.1.4 ORB-SLAM3 To install ORB-SLAM3, we have first cloned the repository coming with the v1.0 of the software. It came along with some requirements that did not fit our environment. At least the requirements seemed to conflict. We managed to get around by telling the software to look for our versions of all the softwares. The install of ORB-SLAM3 starts first with the thirdparties available in their repository. DBoW2, g2o and Sophus. The versions asked for OpenCV could not be satisfied, so we changed the file so as the libraries would be looking for the version 4.2

4.1.1.5 ROS ROS stands for Robot Operating System. It is through this package that ORB-SLAM3 is able to access the camera stream and process it on the run. It was quite difficult to understand fully the installation procedure of this package. It was supposed to be very straight forward. Requiring only a sudo apt install of all different parts of the package. However, it so happened to be difficult for some versions of the package on the Virtual Machine, telling us there were some broken packages and attempting to fix it ended in a loop of broken files. We tried it on WSL2 and it worked like a charm. Eventually we got the installation to work on the Virtual Machine by creating a whole new machine

4.1.2 Calibration

Comme dit précédemment, la calibration de la caméra était une phase importante afin d'obtenir une bonne estimation de la position de la caméra utilisée.

4.1.3 Recherche de chemin

4.1.3.1 Matrice de coordonnées

4.1.3.2 Structures de données définies

4.1.3.2.1 Coord

4.1.3.2.2 AbsoluteCoord

4.1.3.2.3 Path

4.1.3.3 Instances utilisées

4.1.3.3.1 LocalPath

4.1.3.3.2 Absolute2DCoordinates

4.1.3.3.3 PoiList

4.1.3.4 L'algorithme A*

4.1.3.5 Retourner à un point d'intérêt

4.1.3.6 FYWBack

4.1.4 Test et résultats

4.1.5 Difficultés rencontrées

problèmes d'installation sur Windows Switch sur VM Switch sur WSL2

We have spent a major part of the time given for the POC trying to figure out an installation procedure for ORB-SLAM3. We have attempted multiple installations on multiple Operating Systems (OS), such as Windows, Mac, The Windows Subsystem for Linux (WSL2) and a virtual machine (VM).

The tricky part of the installation is the fact that the versions required seemed to conflict with each other. We had to investigate which version would suit all the libraries. For OpenCV for instance we had to install the version 4.2. Having done this, we had to change parts of all the CMakeLists.txt, so as to allow the software to find the needed third parties.

We eventually turned our faces towards a Linux virtual machine. Ubuntu 20.04 on VirtualBox. This virtual Machine has enabled us to work on a complete Linux distribution without any problems with the installation, but rather problems with the virtual machine itself. It is a very nice technology but it comes with its share of troubles, whether that be the latency, the links with keyboards, the mouse that does not work sometimes, and even a corrupted system emerging out of nowhere. But it is the best solution we have found.

4.1.6 Conclusion

4.2 Phase projet

Une fois la phase de POC terminée, nous devions ensuite passer à la phase projet, nécéssitant d'associer nos POC afin de créer une application de démonstration pour la JDS du 7 juillet 2022. Nous avons réfléchis à la création d'Une interface graphique.

4.2.1 Développement de l'application

Développement du Frontend en Flutter Se rendre compte de l'impossibilité technique, beau mais pas possible Passage à PyQt5 Protocole de communication entre les POC Réalisation des liens entre Frontend et Backend avec un objet partagé

5 Les dimensions humaines et managériales (3 à 5 p)

Les dimensions humaines et managériales internes à l'organisme d'accueil. Cette partie consiste en une présentation analytique des processus d'entreprise. Selon les cas, elle portera sur la conduite de projet, les aspects organisationnels, la gestion du changement, le travail en groupe, l'énoncé des objectifs individuels et de l'équipe, la contribution à l'atteinte des objectifs, les difficultés rencontrées, les aides reçues, etc. Les difficultés propres à l'entreprise ou au service dans lequel la mission a été effectuée doivent être abordées de façon professionnelle pour que le jury ait une appréciation réaliste des conditions du travail réalisé.

6 Conclusion (2 à 3 p)

La conclusion générale, de quelques pages, porte sur l'ensemble de votre expérience technique et humaine. Une première partie correspond au bilan et une deuxième partie expose les possibilités d'évolution du projet, du produit. Enfin, vous présenterez vos perspectives d'évolution par rapport à votre projet professionnel initial. Vous préciserez les compétences que vous avez développées en école d'ingénieurs et durant votre mission de fin d'études et préciserez vos axes d'amélioration.

7 Bibliographie

1. Jinqiang Bai et al. « Wearable Travel Aid for Environment Perception and Navigation of Visually Impaired People ». In : *Electronics* 8.6 (juin 2019), p. 697. DOI : 10.3390/electronics8060697.
2. Carlos Campos et al. « ORB-SLAM3 : An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM ». In : *IEEE Transactions on Robotics* 37.6 (déc. 2021), p. 1874-1890. DOI : 10.1109/TRO.2021.3075644.
3. Tete Ji, Chen Wang et Lihua Xie. « Towards Real-time Semantic RGB-D SLAM in Dynamic Environments ». In : *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Mai 2021, p. 11175-11181. DOI : 10.1109/ICRA48506.2021.9561743. arXiv : 2104.01316 [cs].
4. Sheng Wang et al. « CDSFusion : Dense Semantic SLAM for Indoor Environment Using CPU Computing ». In : *Remote Sensing* 14.4 (jan. 2022), p. 979. DOI : 10.3390/rs14040979.

8 Annexes

Acronymes

BRIEF Binary Robust Independent Elementary Features. 12

CNN Convolutional Neural Network. 8

CPU Central Processing Unit. 11, 16

EA État de l'art. 6

ESN Entreprise de Services Numériques. 5

FAST Features from accelerated segment test. 9, 10, 11, 12, 26

FYW Find Your Way. 6, 7, 18

ICP Iterative Closest Point. 13

IMU Inertial Measurement Unit. 8, 9, 10, 16

JDS Journée des stagiaires. 7, 21

ORB Oriented FAST and Rotated BRIEF. 10, 12, 13, 14, 15, 16, 17, 18, 19

POC Proof of Concept. 18, 19, 21

POI Point of Interest. 18

RGB-D Red, Green, Blue and Depth. 8, 13, 16

SLAM Simultaneous Localization and Mapping. 8, 9, 12, 13, 16, 17, 18, 19

TSDF Truncated Signed Distance Fields. 11

VIO Visual-Inertial Odometry. 9, 10

VM Virtual Machine. 18

WSL2 Windows Subsystem for Linux 2. 18, 19