

# Acquisition photogrammétrique autonome avec le robot Warthog

Enzo-Loïd Mohamed ESSONO AUBAME

2 octobre 2021



Maitre de stage : Luc Jaulin

Tuteurs : Fabrice Le BARS et Benoit ZERR

Adresse : ENSTA Bretagne, Brest, France

# Table des matières

|                                                    |           |
|----------------------------------------------------|-----------|
| <b>Remerciements</b>                               | <b>4</b>  |
| <b>1 Introduction</b>                              | <b>5</b>  |
| <b>2 PRESENTATION</b>                              | <b>6</b>  |
| 2.1 Présentation du Warthog . . . . .              | 6         |
| 2.2 Présentation du matériel . . . . .             | 6         |
| 2.3 Apprentissage de la photogrammétrie . . . . .  | 8         |
| <b>3 CONFIGURATION DE LA MISSION DU WARTHOG</b>    | <b>10</b> |
| 3.1 Contrôle du Warthog . . . . .                  | 10        |
| 3.2 Configuration de la Raspberry Pi . . . . .     | 11        |
| 3.3 Mission autonome du Warthog . . . . .          | 11        |
| 3.4 IHM . . . . .                                  | 12        |
| <b>4 PHOTOGRAMMETRIE</b>                           | <b>16</b> |
| 4.1 Traitement photogrammétrique . . . . .         | 16        |
| 4.2 Reconstruction manuelle de structure . . . . . | 17        |
| 4.3 Analyse des résultats . . . . .                | 18        |
| <b>5 CONCLUSION</b>                                | <b>20</b> |

# Table des figures

|    |                                                                                                                                                                                                                    |    |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1  | Equipement initial du robot Warthog . . . . .                                                                                                                                                                      | 6  |
| 2  | Télécommande de contrôle et commande d'arrêt d'urgence manuelle .                                                                                                                                                  | 7  |
| 3  | Dispositif utilisé par Marie PONCHART pour faire de la photogrammétrie sous-marine. . . . .                                                                                                                        | 7  |
| 4  | Installation du matériel de photogrammétrie . . . . .                                                                                                                                                              | 8  |
| 5  | Principe de la reconstruction d'un objet 3D par photogrammétrie (sources : A.Abadie, Sea(e)scape) . . . . .                                                                                                        | 8  |
| 6  | Logiciel MicMac utilisé par Marie PONCHART pour faire de la photogrammétrie sous-marine . . . . .                                                                                                                  | 9  |
| 7  | Simulation du parcours suivi par Warthog sous python . . . . .                                                                                                                                                     | 11 |
| 8  | Visualisation graphique du mapping sous excel avec en bleu les coordonnées en pixel, orange la transformée des coordonnées GPS et en rouge le garage du laboratoire robotique d'où part Warthog par défaut . . . . | 12 |
| 9  | Différentes idées de IHM développées . . . . .                                                                                                                                                                     | 13 |
| 10 | IHM pour indiquer le chemin à suivre . . . . .                                                                                                                                                                     | 14 |
| 11 | Exemple de tracé de parcours . . . . .                                                                                                                                                                             | 15 |
| 12 | Nuance de point et angles de vue des différentes caméras après traitement sur Colmap . . . . .                                                                                                                     | 17 |
| 13 | Parties d'un bâtiment mal reconstruit visualisées sur meshlab . . . . .                                                                                                                                            | 17 |
| 14 | Reconstruction manuelle sur Cloudcompare . . . . .                                                                                                                                                                 | 18 |
| 15 | Photogrammétrie de deux bâtiments de l'ENSTA Bretagne . . . . .                                                                                                                                                    | 19 |
| 16 | Debut d'une construction de map 3D de l'ENSTA Bretagne . . . . .                                                                                                                                                   | 19 |

# Remerciements

Je tiens tout d'abord à remercier l'ENSTA Bretagne et Luc JAULIN pour avoir rendu ce stage possible.

J'aimerais remercier Marie PONCHART pour ses enseignements et son aide pour tout ce qui attrayait à la photogrammétrie, mais également Robin SANCHEZ qui nous tenait fréquemment compagnie dans le labo-ROB, et qui, connaissant bien le robot Warthog, ayant travaillé dessus l'an dernier, n'hésitait pas à prendre de son temps pour nous donner des explications sur sa conception.

Et bien sûr, nous remercions nos tuteurs de stage Fabrice LE BARS et Benoît ZERR, qui nous ont proposé ce sujet passionnant, ainsi que l'ensemble des professeurs de la filière robotique qui nous ont permis d'accéder à un niveau de savoir suffisant pour pouvoir aborder des thématiques complexes en autonomie.

# 1 Introduction

La photogrammétrie est une technique de reconstitution d'objet sous forme de modèle 3D à partir d'une série de photos prises sous différents axes autour de ce même objet. Cette technique se base sur la reconnaissance de points d'intérêts entre deux images et les fait correspondre afin d'obtenir un objet 3D.

Pour ce stage réalisé avec le Lab-STICC de l'ENSTA Bretagne, en collaboration avec Isaac-Andrei WITT et Maxime LEGEAY, nous devons programmer le robot Warthog afin qu'il puisse parcourir l'ENSTA en autonomie, prendre des photos de l'infrastructure et restituer un modèle 3D de l'établissement.

Ce rapport présente principalement tout ce qu'il faut savoir sur le robot Warthog, comment il réalise ses missions en autonome et enfin comment nous précédons pour réaliser de la photogrammétrie. Vous découvrirez à travers ce document les difficultés que nous avons pu rencontrer et la façon dont nous avons contourner ces mêmes problèmes.

## 2 PRESENTATION

### 2.1 Présentation du Warthog

Le Warthog est un grand véhicule terrestre sans pilote tout-terrain capable de se déplacer brièvement dans l'eau. Il peut gérer des environnements difficiles grâce à sa construction robuste, à sa faible pression au sol et à ses pneus de traction, qui lui permettent de se déplacer sans effort dans les sols mous, la végétation, les boues épaisses et les pentes raides.

Les plaques de montage des charges utiles et les ports d'alimentation et de communication accessibles permettent de personnaliser facilement le Warthog avec des capteurs, des manipulateurs et d'autres charges utiles pour répondre à une grande variété d'applications robotiques dans les secteurs de l'exploitation minière, de l'agriculture et de la surveillance environnementale.

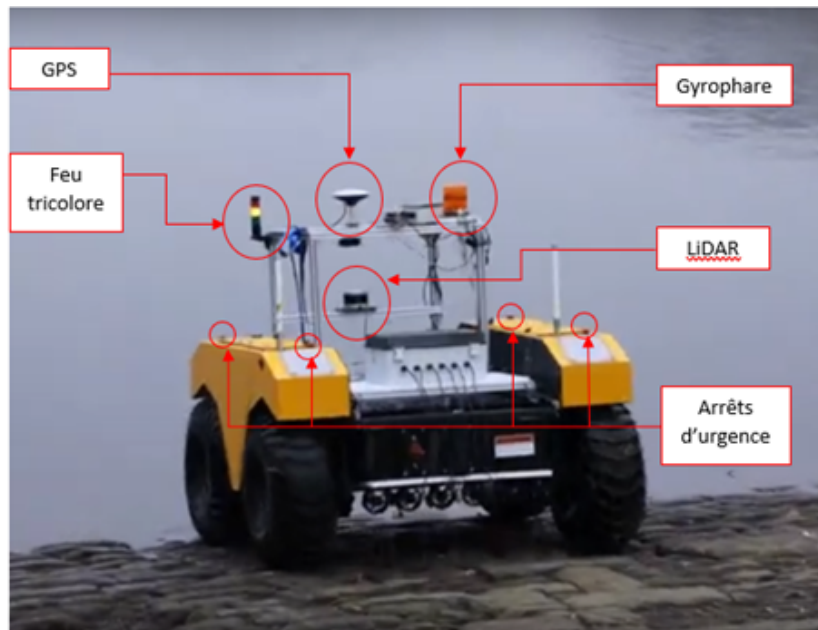


FIGURE 1 – Equipement initial du robot Warthog

En termes de capteurs, le robot est équipé de sonars, d'un GPS, d'un LiDAR et d'une centrale inertielle. Un feu tricolore pour nous montrer les différents états du Warthog. Le vert pour le démarrage des services ou l'arrêt, l'orange pour le commander avec la télécommande et le rouge pour le contrôle en autonome. Le gyrophare permet de savoir si le robot est armé ou non. Enfin, le robot a 4 arrêts d'urgence, la sécurité est très importante sur ce robot.

### 2.2 Présentation du matériel

Le Warthog a du matériel supplémentaire dont sa télécommande afin qu'il puisse être commandé manuellement et une commande d'arrêt d'urgence portable (qui permet aussi d'armer le robot).



FIGURE 2 – Télécommande de contrôle et commande d'arrêt d'urgence manuelle

Afin de pouvoir faire de la photogrammétrie, Marie PONCHARD nous a confié le matériel qu'elle utilisait pour en faire (voir figure 3). Nous avons donc ajouté deux antennes GPS, une retournant le cap et l'autre la position. Son unité de traitement de données GPS est aussi installée. Aussi, nous avons placé sur le côté gauche du robot les trois camera GoPro.

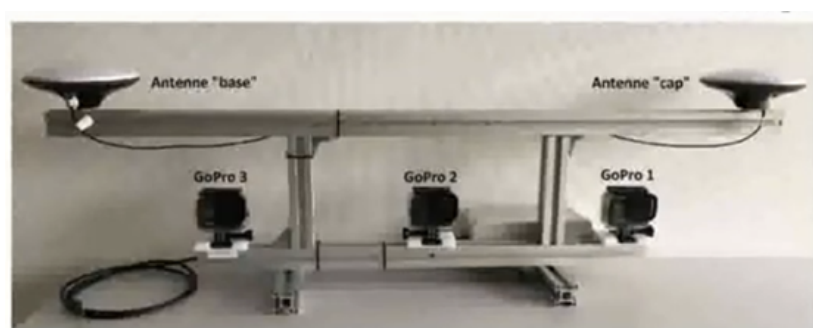


FIGURE 3 – Dispositif utilisé par Marie PONCHART pour faire de la photogrammétrie sous-marine.

Une carte Raspberry pi a été également placée sur le Warthog. Elle permet de communiquer avec l'ordinateur du robot sous ROS. C'est elle qui permet de gérer la prise de photo en autonome et gère le GPS des deux antennes supplémentaires. On peut ainsi avoir des photos synchronisées avec la position du robot.



FIGURE 4 – Installation du matériel de photogrammétrie

## 2.3 Apprentissage de la photogrammétrie

Avant de pouvoir nous mettre à l'ouvrage, nous avons tout d'abord reçu un rapide enseignement de la part de Marie PONCHART, une doctorante ayant travaillé sur la photogrammétrie en fond marin au cours de l'année précédente. Comme dit plus haut, la photogrammétrie est un procédé permettant de passer d'une série de photos prises sous différents angles à une reconstruction 3D de la structure photographiée.

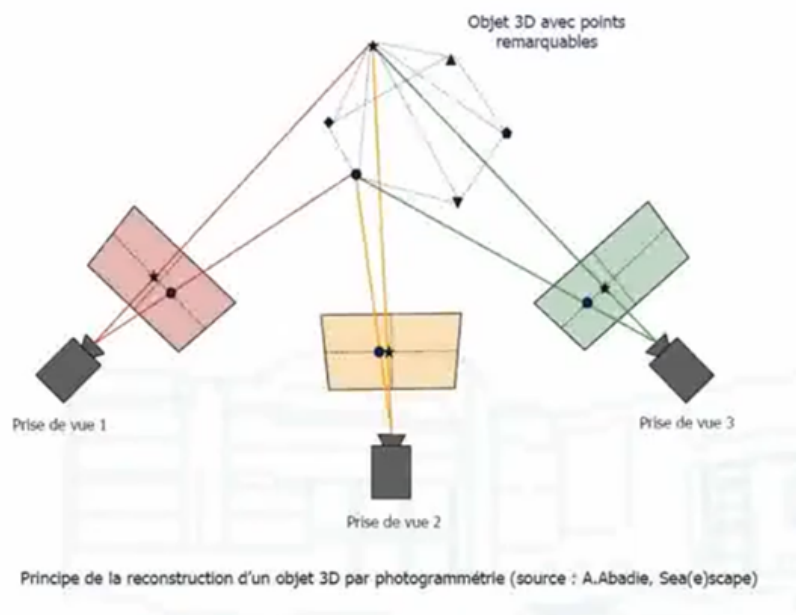


FIGURE 5 – Principe de la reconstruction d'un objet 3D par photogrammétrie (sources : A.Abadie, Sea(e)scape)

Elle nous a donc présenté le dispositif qu'elle avait utilisé qui était composé de deux GPS – un pour la position et l'autre pour l'angle –, trois supports à GoPro et un boîtier

contenant la carte chargée d'interpréter les données communiquées par les GPS.

Marie nous a également fournis les codes python dont elle se servait en parallèle de Micmac. Il s'agit d'un logiciel de reconstruction 3D, gratuit et open source, conçu à l'ENSG (l'Ecole Nationale des Sciences Géographiques) et disposant d'un tutoriel assez complet en libre-service, censé rendre son utilisation intuitive.



FIGURE 6 – Logiciel MicMac utilisé par Marie PONCHART pour faire de la photogrammétrie sous-marine

## 3 CONFIGURATION DE LA MISSION DU WARTHOG

### 3.1 Contrôle du Warthog

Le Warthog peut être contrôlé de deux manières, avec une télécommande ou en autonome. Les codes se trouvant déjà à l'intérieur du Warthog nous ont permis de pouvoir facilement évoluer dans la partie de mise en autonomie du robot. Il y avait notamment un code de suivi de ligne et de point se basant sur les coordonnées GPS.

Le capteur principalement utilisé était le GPS. Des résultats avec une assez bonne précision étaient obtenus dans des zones pouvant facilement capter les satellites. Entre les bâtiments le GPS est moins précis et le Warthog avait tendance à avancer s'il le perd. Maxime a ainsi codé une commande d'arrêt pour y remédier, tant que les données GPS ne sont pas assez précises le robot reste à l'arrêt.

La trajectoire que nous avons trouvée plus judicieuse pour le Warthog est une sinusoïde pour les missions de prise de photos car elle nous permettait de pouvoir imposer au Warthog de prendre plusieurs inclinaisons afin d'avoir plusieurs angles de vue pour un bâtiment filmé. Dans un premier temps nous avons choisi d'utiliser la méthode du point attractif vue en classe pour élaborer la trajectoire du robot. En simulation le robot arrivait à effectuer son parcours sans problème mais sur le terrain, le suivi de point continu n'était pas une bonne idée due aux erreurs de précision du GPS d'une part et d'autre part à cause du rayon de détection des points du robot. Nous avons par la suite décidé de discrétiser la sinusoïde et demandé au robot d'effectuer un suivi de chaque point de la sinusoïde, c'est ainsi que nous avons obtenu une trajectoire qui convenait à nos attentes.

**QUID du LiDAR ?** Nous avons lu le rapport de Robin SANCHEZ afin de savoir comment le LiDAR fonctionne, nous avons eu beaucoup de difficultés à traiter les données de celui-ci. Mais dès que nous avons réussi à avoir des données cohérentes, le second problème que nous avons rencontré était qu'il était presque inutilisable pour faire un suivi de mur à l'ENSTA Bretagne car la majorité des bâtiments sont en verre et donc la lumière aurait traversé la structure. L'idée d'utiliser un couple LiDAR et sonars nous a traversé l'esprit mais avec des soucis de temps, nous n'avons pas pu fouiller en profondeur cette piste.

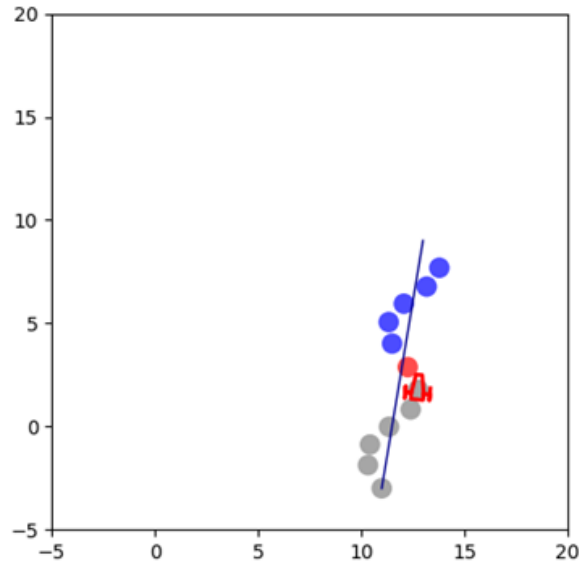


FIGURE 7 – Simulation du parcours suivi par Warthog sous python

### 3.2 Configuration de la Raspberry Pi

La carte raspberry pi est celle qui était chargée d'effectuer la prise de photo. Pour ce faire, il était nécessaire d'installer ROS sur le microprocesseur. Dans un premier temps, nous avons eu des difficultés à recevoir les données de ROS du Warthog sur la raspberry pi. Grâce à M. LEBARS nous avons réussi à résoudre ce problème. Il s'agissait d'un souci d'adresse ip car le robot dispose de deux adresses ip de connexion (192.168.131.1 et 10.42.0.1). Et donc, lors de la connexion il est important de bien configurer le master du robot avec l'adresse ip 192.168.131.1 et d'utiliser la même adresse ip sur carte raspberry ou l'ordinateur utilisé pour communiquer avec le ROS du Warthog.

Par la suite, nous avons eu souci d'importation de bibliothèques python sur ROS dans la raspberry pi. La version kinetic était en Python2 tandis que nos codes et modules étaient en Python3. Nous avons décidé de passer en ROS noetic pour résoudre ces problèmes de compatibilité.

Finalement, il fallait que la raspberry puisse se connecter primordialement au robot s'il est allumé et si le robot est éteint la carte se connecte directement au réseau de l'école. Avec quelques recherches sur le internet, ce problème fut vite résolu.

### 3.3 Mission autonome du Warthog

Dans le document Warthog advanced use and maintenance remis par M. LEBARS, on découvre comment lancer une mission en autonome. Le code gérant donc principalement le lancé des missions est objectif.cpp, c'est donc celui-ci qui a été le plus modifié durant le stage.

Une mission autonome complète est subdivisée en plusieurs sous missions qui sont des missions de positionnements et des missions de prise de photos. Les missions de positionnements sont des simples suivis de points et grâce à elles, ont arrive a dirigé le robot vers les endroit que nous aimerions qu'il effectue une prise de photos. Les missions de prises de photos comme leur nom l'indique, elles permettent de prendre des photos

des batiments desirés. Cependant, cette mission est lancée deux fois. Une première fois sur le robot et qui enclenche le déplacement suivant la sinusoïde mentionnée au **3.1** et une seconde fois sur la raspberry pi pour qu'elle puisse prendre des photos. Nous avons essayé d'effectuer un threading pour pouvoir lancer les deux missions en une seule fois vu que les deux ordinateurs communiquent ensemble via ROS mais comme nos codes étaient écrits en Python il avait énormément de bugs du fait que Python gère très mal les threads.

### 3.4 IHM

Afin de pouvoir indiquer facilement le chemin que devra suivre Warthog, nous avons décidé de programmer une IHM (Interface Homme Machine) adaptée à notre situation – même s'il existait déjà une IHM implémentée dans le char. Pour cela, j'ai commencé par faire du mapping en assimilant les coordonnées en pixel d'une carte de l'ENSTA avec des coordonnées GPS. Isaac a tout d'abord annoté un maximum de point remarquable sur google map, bien que la vision satellite proposée n'était plus à jour. Par la suite, il a travaillé sous tableau Excel et il a recherché une formule mathématique permettant au maximum de faire correspondre ces points particuliers par régression linéaire, rotation, etc... Ainsi il a réussi à faire correspondre ses deux plans avec une erreur maximum de 3m.

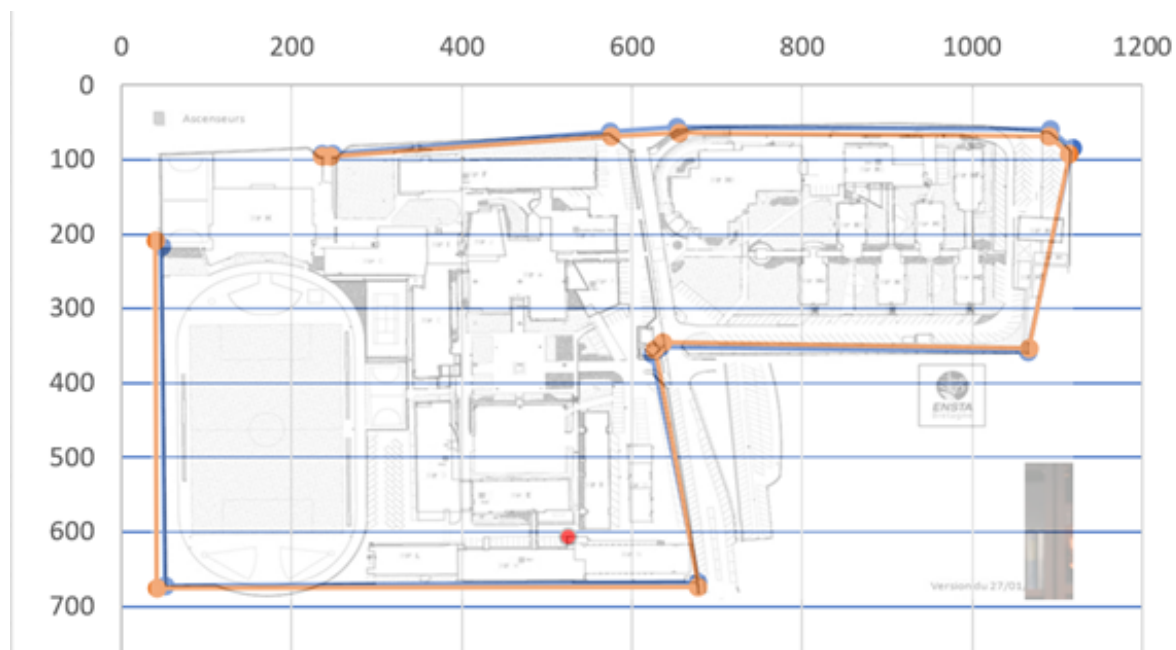


FIGURE 8 – Visualisation graphique du mapping sous excel avec en bleu les coordonnées en pixel, orange la transformée des coordonnées GPS et en rouge le garage du laboratoire robotique d'où part Warthog par défaut

L'idée derrière cette IHM était de renvoyer à Warthog un fichier texte contenant les coordonnées par lesquels le char devrait passer et s'il était en mode déplacement pur ou en mode photographie. Bien que nous ayons vu en première année d'école l'utilisation du logiciel Qt designer, nous avons préféré tout d'abord opter pour quelque chose de plus simple et plus proche d'un code python. Après quelque recherche, Une bibliothèque intéressante et simple d'utilisation nommée Turtle, permettant de dessiner avec sa souris ou les touches directionnelles du clavier sur une image de fond. Malheureusement, cette application n'était pas adaptée à notre utilisation. En effet, cette dernière faisait remonter dans le code d'importants problèmes d'overflow. Après d'autres recherches Tkinter semblait devenir une meilleure option pour cette implémentation. Initialement, si l'utilisateur trace entièrement le parcours que devrait suivre le robot et que ce dernier lise le parcours tracé notant ainsi une suite de coordonnées à suivre dans l'optique d'utiliser la fonction de suivi de ligne déjà implémenté sur Warthog. Tout d'abord, Isaac a programmé une classe Plan permettant d'indiquer la carte du lieu où l'on travaille. Puis par la suite, il y a eu plusieurs ébauches : il a colorié sur une carte les zones de l'écoles praticables par le char et il était impossible de tracer un chemin en dehors de ces zones, un traitement morphologique était donc nécessaire via l'utilisation de la bibliothèque openCV permettant de détecter les murs, par la suite il fallait définir un système de rails, ... Le problème de ces premières idées était que le tracé était assez incertain et compliqué. Il n'était pas rare que le robot se perde en cours de route ou que le nombre de points à annoter soit trop important.

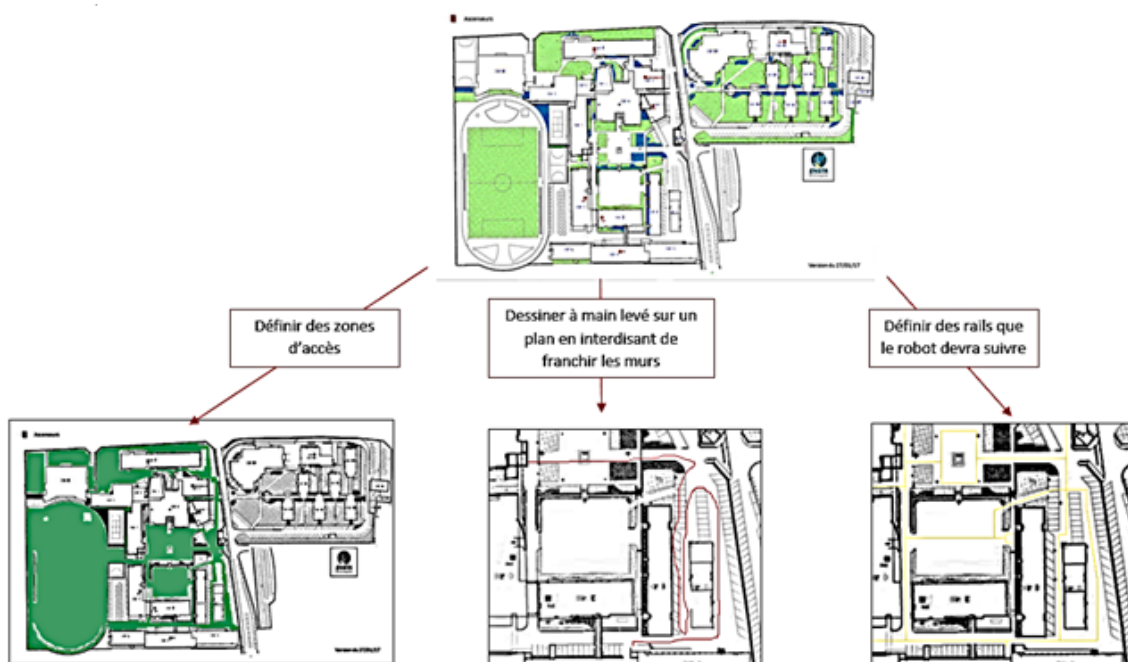


FIGURE 9 – Différentes idées de IHM développées

C'est à ce moment-là que l'objectif c'est précisé. Le robot fera des oscillations entre deux points en s'inspirant du code suivi de waypoint. Aussi, seul deux points suffisent à chaque fois, ce qui fait beaucoup moins de données à enregistrer. L'IHM ci-dessous correspondait plus à ce que qui pouvait ressembler l'attendu pouvant facilité le suivi du parcours du robot.

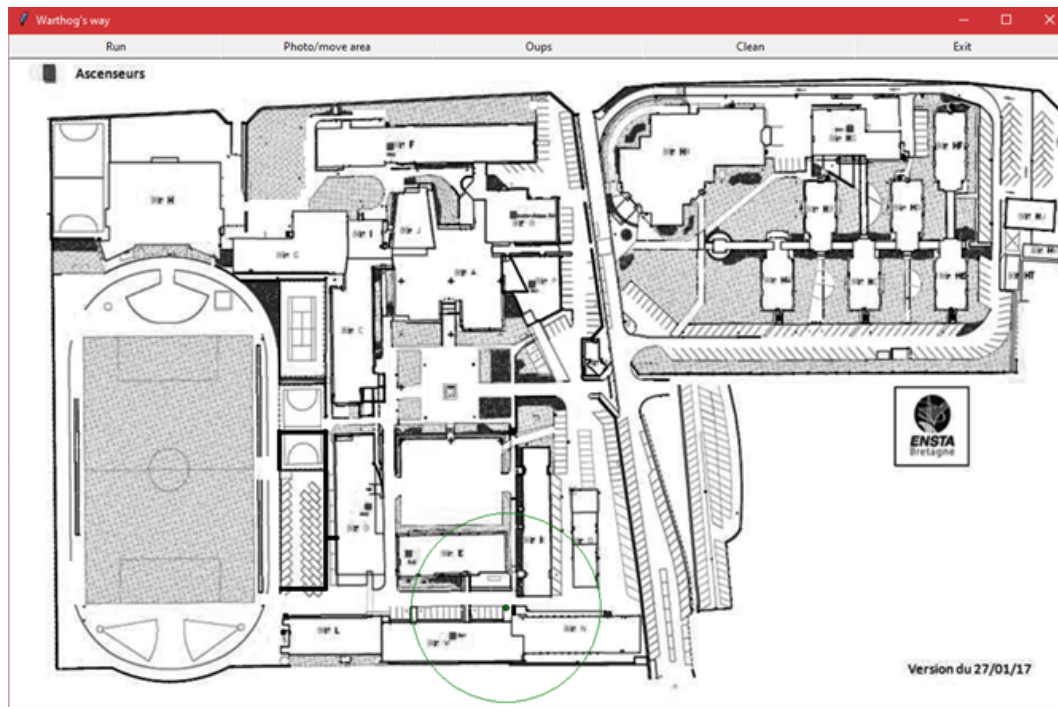


FIGURE 10 – IHM pour indiquer le chemin à suivre

On a ici 5 boutons :

- Run : Confirme le parcours tracé et renvoie un document texte contenant à chaque ligne une liste de 3 éléments [latitude, longitude, mode de déplacement].
- Photo/move areas : Permet de changer le mode de déplacement entre le mode photographie, représenté par un chemin bleu, et le mode déplacement sans photos, représenté par un chemin rouge.
- Oops : Permet de revenir en arrière et de rechoisir le dernier point de passage.
- Clean : Permet d'effacer le chemin et de tout recommencer.
- Exit : Pour quitter la fenêtre.

Ainsi l'utilisateur doit sélectionner les différents points de passage du Warthog et précisant son mode de déplacement. La dernière localisation du char est indiquée par un point vert et l'utilisateur est limité au cercle vert pour choisir la prochaine direction.

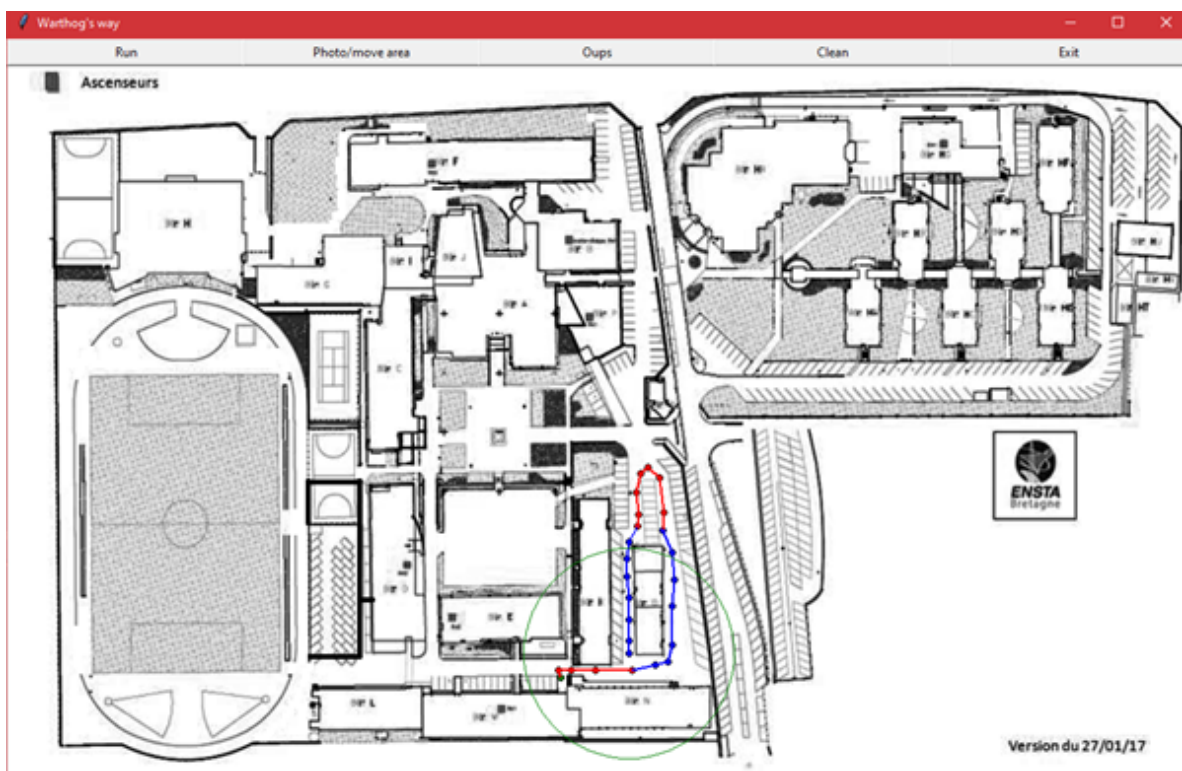


FIGURE 11 – Exemple de tracé de parcours

## 4 PHOTOGRAMMETRIE

Après les missions de prise de photos effectuées, les photos sont envoyées de Raspberry pi vers un ordinateur afin de pouvoir lancer le traitement photogrammétrique via le logiciel Colmap.

### 4.1 Traitement photogrammétrique

- Pré-traitement

Avant de passer à la phase de calcul dans le logiciel colmap, deux grandes étapes doivent d'abord être effectuées.

La première étape est le tri des photos. Lors de la mission de prise de photos, le robot prend certaines inclinaisons dues à sa position sur la sinusoïde qui lui font parfois ratées son objectif. Ces photos pouvant créer des discontinuités de l'objectifs et qui peuvent ajouter des données supplémentaires à analyser inutilement sont donc à retirer du dossier de photos final.

La seconde partie, contraire à la première, consiste quant à elle à ajouter des photos prises à la main dans le dossier de photos objectif. En effet, le Warthog prend des photos suivi le champ de vision des cameras GoPro, mais parfois si le bâtiment filmé est très grand, il est possible que le robot n'arrive pas à filmer la fin du bâtiment (dans le sens de la hauteur). Il peut arriver également si le robot est très éloigné de sa cible (ce qui est souvent le cas car on évite que le robot soit très proche d'un grand bâtiment sinon possibilité de perdre le gps) qu'il n'arrive pas récupérer toutes les informations essentielles d'une photo à cause du nombre de mégapixel des cameras qui peuvent faire défaut à une certaine distance. Dans ces deux cas, l'ajout de photos complémentaires aide à faciliter le traitement.

- Traitement

La création de l'objet 3D se fait sur colmap. Le principe est simple, il suffit de cliquer `reconstruction`, `reconstruction automatique`, créer un Workspace pour le traitement puis le sélectionner, sélectionner ensuite le dossier contenant les images que l'on souhaite transformer en objet 3D puis lancer le calcul.

Le temps du calcul dépend de plusieurs variables. Les plus importantes sont sans doute la nature du matériel composant l'ordinateur, le nombre de photos et la qualité du mesh final.

- Composants de l'ordinateur : Il existe plusieurs versions de colmap, mais les plus efficaces sont celles fonctionnant avec CUDA (logiciel de Nvidia). Ce logiciel permet d'utiliser les cœurs cuda d'un gpu nvidia afin d'améliorer la vitesse de calcul de colmap. Il est donc évident qu'il faut que l'ordinateur soit équipé d'une carte graphique nvidia. La vitesse de calcul dépend également de processeur, plus le processeur est récent et a de nombreux cœurs plus vite il pourra gérer les tâches dans le traitement.
- Le temps de calcul est proportionnel au nombre de photos. Plus il y a des photos plus le traitement sera long et inversement.
- Plus l'objet 3D aura des détails plus de celui-ci est long. On peut choisir la qualité des détails directement sur colmap.

Le calcul est très gourmand en ressource. Un traitement de 650 photos en qualité high, il faut réserver une centaine de gigas de stockage dans l'ordinateur afin d'obtenir un objet 3D qui ne pèse que quelques mégas. Et son temps de calcul est environ de 17h.

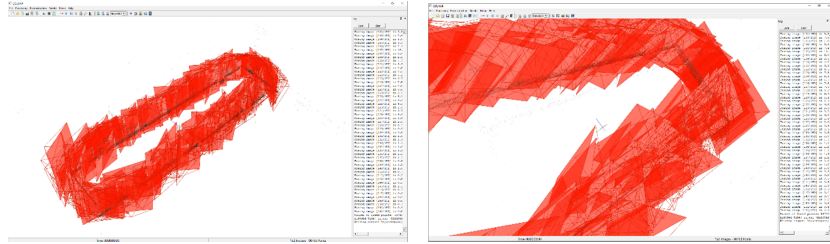


FIGURE 12 – Nuage de point et angles de vue des différentes caméras après traitement sur Colmap

## 4.2 Reconstruction manuelle de structure

La photogrammétrie a besoin d'un éclairage uniforme, que l'objet filmé ne puisse refléter la lumière et la suite de photo soit le possible discontinue. Quand ces conditions ne sont pas réunies, dans le logiciel colmap l'objet est divisé en plus parties qui peuvent parfois être utilisable après certaines étapes de traitements manuelles. Ces étapes de traitements permettent souvent de reconstruire l'objet souhaité. Plusieurs bâtiments de l'ENSTA Bretagne reflètent la lumière et nous avons dû faire certaines reconstructions manuelles.

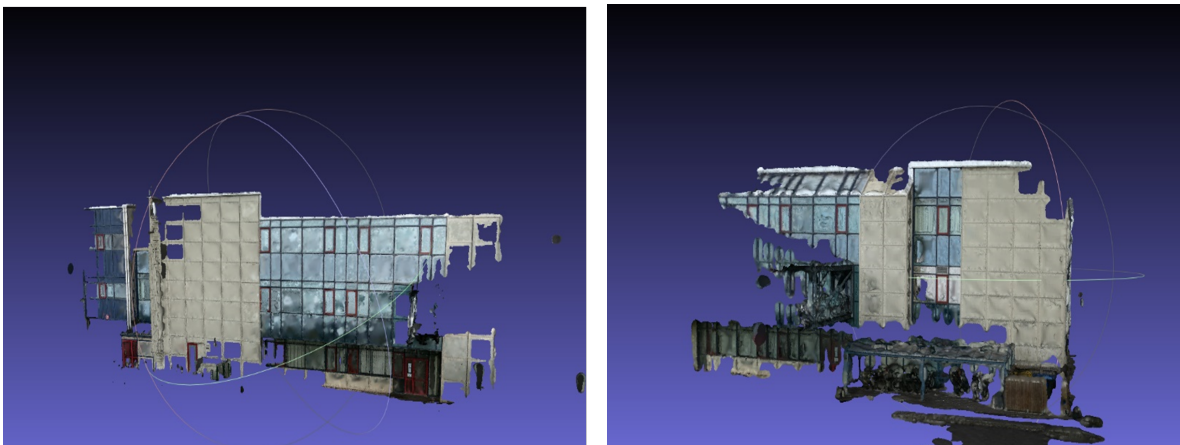


FIGURE 13 – Parties d'un bâtiment mal reconstruit visualisées sur meshlab



FIGURE 14 – Reconstruction manuelle sur Cloudcompare

La reconstruction manuelle se fait en plusieurs étapes. Premièrement grâce au logiciel Meshlab on visualise les différentes parties que l'on souhaite reconstruire puis on retire les résidus du décor inutiles et enfin on peut réaliser un filtre pour un lissage si les parties en ont vraiment besoin. Par la suite, les parties de l'objet sont exportées dans un autre logiciel appelé Cloudcompare qui servira à l'assemblage. Avant de passer à l'assemblage il est important de redimensionner chacune des pièces afin de les rendre compatibles plus tard. Une fois cette phase de redimensionnement terminer, on peut maintenant passer à la reconstruction. Tout d'abord, il faut sélectionner les différentes parties de l'objet que l'on souhaite assembler, puis cliquer sur ce bouton qui ouvre une fenêtre demandant de choisir laquelle des parties devrait se fixer sur l'autre. Pour finir il faut choisir des points de correspondance sur chaque objet puis valider la sélection.

### 4.3 Analyse des résultats

Il n'est encore possible de faire une acquisition photogrammétrie complètement autonome avec le robot Warthog, une grande partie manuelle est encore nécessaire pour obtenir des résultats acceptables. Cependant, les résultats obtenus sont prometteurs, certains bâtiments n'ont pas eu besoin de reconstructions manuelles et une seule mission de prise de photos était nécessaire pour avoir un bon objet 3D satisfaisant.

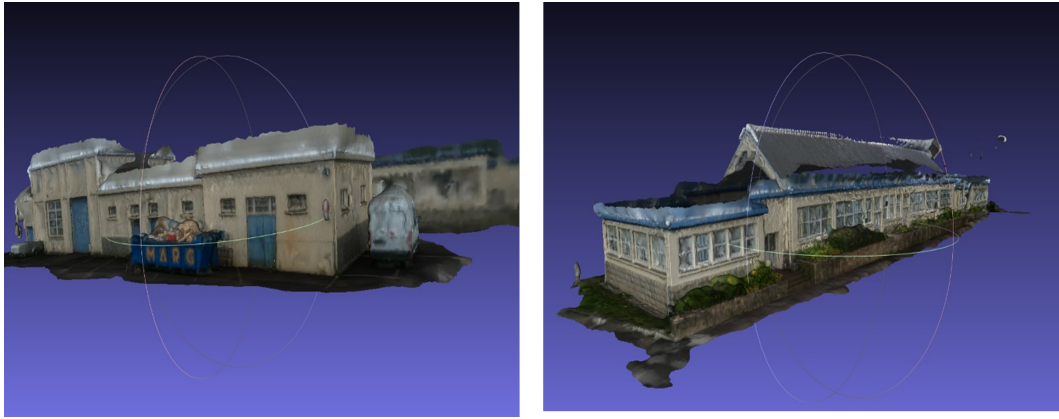


FIGURE 15 – Photogrammétrie de deux batiments de l'ENSTA Bretagne

Grace à Cloudcompare, on a pu commencer à réaliser un montage 3D des différents bâtiments qu'on avait en notre autre possession mais le positionnement dans l'espace n'est pas toujours évident. Mais l'idéal serait de pouvoir exporter les objets 3D dans un logiciel comme VREP qui permet de mieux gérer le positionnement relatif ou absolu d'un objet dans l'espace. Cependant, cette piste n'a pas pu être exploitée dans la durée du stage



FIGURE 16 – Debut d'une construction de map 3D de l'ENSTA Bretagne

## 5 CONCLUSION

Le but final de stage était de pouvoir réaliser une carte 3D de l'ENSTA Bretagne à l'aide la photogrammétrie et du robot Warthog en autonome. Cet objectif n'a pas pu être atteint mais pour pouvoir arriver à cette fin il y a avait plusieurs phases intermédiaires comme pouvoir faire de la photogrammétrie ou rendre autonome le robot à notre convenance qu'il fallait réussir. Et je suis assez satisfait des resultats obtenus pour ces parties. J'aurais souhaité travailler plus de temps sur les algorithmes utilisant le LiDAR 3D et explorer plus en profondeur le couplage LiDAR-sonars.

Ce stage fut pour moi très enrichissant. En effet, j'ai pu toucher à de nombreux domaines de la robotique, de la théorie vue en cours, aux réalités du terrain. Faire de la photogrammétrie n'est peut-être pas ce qu'on attend d'un roboticien mais cela m'a permis d'apprendre beaucoup et de réappliquer toutes les notions vues en cours d'année. Aussi j'ai compris que la robotique n'a pas un domaine d'application, c'est nous qui mettons de la robotique dans ce qui nous entoure.