

ENSTA Bretagne
2 rue François Verny
29806 Brest Cedex 9,
France

WITT Isaac-Andrei
isaac-andrei.witt@ensta-bretagne.org
FISE 2022 – spécialité Robotique

Le 23 septembre 2021

Résumé :

La photogrammétrie est une discipline consistant à reproduire une modélisation d'un objet ou de bâtiments en s'appuyant sur des photos prises préalablement. Malheureusement, reproduire une infrastructure en 3D est quelque chose d'extrêmement chronophage. Aussi le but de ce projet pédagogique était d'automatiser un maximum de tâches de ce processus, notamment à l'aide de Warthog, un char robotisé. Par conséquent, nous avons confié la tâche de photographe au robot et avons développé la reconstruction 3D parallèlement. A l'aide d'un chemin tracé sur une IHM développé sous python, Warthog parcourt l'école et photographie les bâtiments de l'ENSTA Bretagne sous différents angles à l'aide de ses deux GoPro. La façon dont il se déplace varie entre les phases de déplacement pur, où il se meut en ligne droite avec une vitesse moyenne, et de prise de photos, où il se déplace un peu plus lentement en suivant une sinusoïde, afin de maximiser les angles de vues. Par la suite, il rapporte les images qu'il a pris à l'utilisateur afin qu'il puisse procéder au traitement d'image sous Colmap, un logiciel de photogrammétrie. Certes nous avons réussi à automatiser une grande partie du processus, cependant une intervention humaine est tout de même nécessaire pour passer d'un ensemble de photos à une vision 3D de l'établissement. Vous découvrirez les différentes étapes de la conception de ce projet dans ce rapport.

Abstract:

Photogrammetry is a discipline of reproducing a model of an object or buildings based on previously taken photographs. Unfortunately, reproducing an infrastructure in 3D is extremely time consuming. Therefore, the aim of this educational project was to automate as many of the tasks in this process as possible, with the help of Warthog, a robotic tank. Therefore, we gave the robot the task of photography and developed the 3D reconstruction in parallel. Using a path drawn on a GUI developed in Python, Warthog walks around the school and photographs the buildings of ENSTA Bretagne from different angles using its two GoPro. The way he moves varies between pure movement, where he moves in a straight line with an average speed, and taking pictures, where he moves a little slower following a sinusoid, to maximize the angles of view. It then brings the images it has taken back to the user so that he can process them in Colmap, a photogrammetry software. Although we have succeeded in automating a large part of the process, human intervention is still necessary to go from a set of photos to a 3D vision of the establishment. You can read about the different stages of the design of this project in this report.

Table des matières

1. Introduction.....	4
----------------------	---

1.1.	Introduction générale.....	4
1.2.	Remerciement.....	5
2.	Phase de mise en condition.....	5
2.1.	Apprendre comment faire de la photogrammétrie.....	5
2.2.	Présentation du robot Warthog.....	7
3.	Photogrammetrie.....	10
3.1.	Installation du matériel.....	10
3.2.	Trouver un logiciel efficace.....	10
3.3.	Communiquer les photos.....	13
3.4.	Traitement photogrammétrique.....	14
4.	Guider le Warthog.....	15
4.1.	Le guider point par point.....	15
4.2.	Conception d'une IHM.....	15
5.	Résultat.....	20
5.1.	Warthog fait-il de la photogrammétrie en parfaite autonomie ?.....	20
5.2.	Prise de recul.....	21
6.	Conclusion.....	22
7.	Annexes.....	23
7.1.	Liste des figures.....	23
7.2.	Bibliographie.....	24

1. Introduction

1.1. Introduction générale

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 Enzo-Loid Mohamed ESSONO AUBAME 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. Aussi, la problématique principale de ce projet pédagogique est d'automatiser autant que possible la reconstruction 3D de l'établissement. Ce rapport présente principalement mon rôle au sein de cette équipe, mais également les directions vers lesquels mes camarades se sont orientés pour concevoir leur partie. Vous découvrirez à travers ce document les difficultés que nous avons pu rencontrer et la façon dont nous avons contourné ces mêmes problèmes.

1.2. Remerciement

Pour mon stage de deuxième année, il me tenait à cœur de faire quelque chose ayant un lien évident avec le monde de la robotique. C'est pourquoi je tiens tout d'abord à remercier l'ENSTA Bretagne et toute l'équipe robotique pour m'avoir offert cette opportunité.

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.

2. Phase de mise en condition

2.1. Apprendre comment faire 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, un doctorant 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.

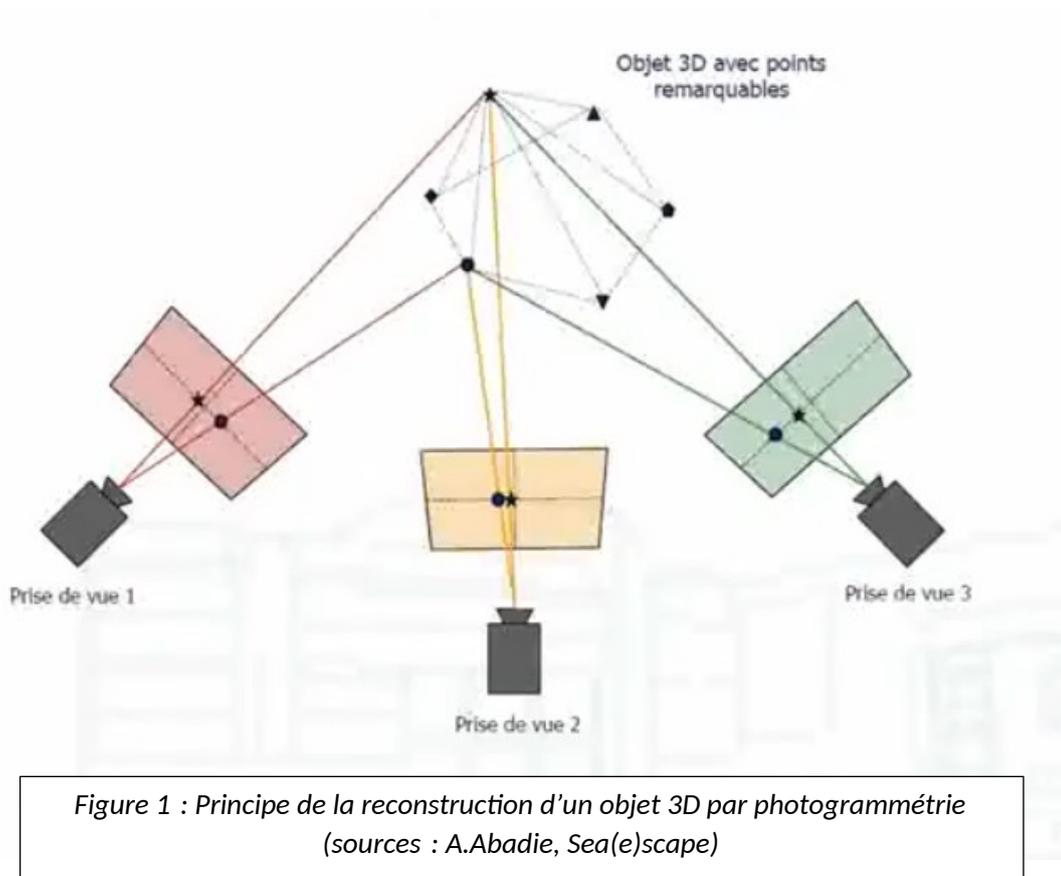


Figure 1 : Principe de la reconstruction d'un objet 3D par photogrammétrie
(sources : A.Abadie, Sea(e)scape)

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.

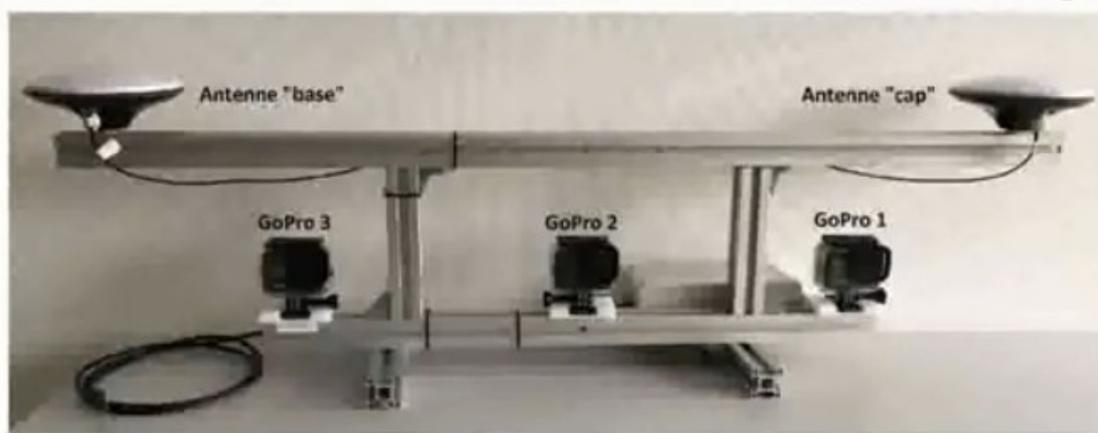


Figure 2 : Dispositif utilisé par Marie PONCHART pour faire de la photogrammétrie sous-marine

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 3 : Logiciel MicMac utilisé par Marie PONCHART pour faire de la photogrammétrie sous-marine

2.2. Présentation du robot Warthog

Après avoir compris en quoi consistait la photogrammétrie, nous devons prendre en main le robot Warthog de l'ENSTA Bretagne. Il s'agit d'un type de robot que nous avons rencontré au cours de l'année - à échelle réduite cependant. En effet, comme les robot Dartv2 que l'on devait programmer afin qu'ils parcourent un labyrinthe en autonomie, le Warthog fonctionne comme un char : il est dénué de roues directionnelles et tourne par différence de rotation entre les roues droites et gauches.



Figure 4 : Arrière du robot Warthog

J'ouvre une parenthèse pour vous dire que ce dernier est techniquement également amphibie, mais ses pompes ont un peu du mal à tenir la cadence. Warthog est équipé de deux télécommandes : une pour pouvoir le commander manuellement et une télécommande de sécurité, qui empêche le démarrage du robot tant qu'elle n'est pas détectée. Le char est également muni de quatre boutons d'arrêt d'urgence situés au niveau des quatre roues. Trois sonars sont disposés à l'avant ainsi qu'un LiDAR et un GPS. La prise en main du char était plutôt intuitive, la seule partie complexe étant le démarrage du robot, mais tout est expliqué dans le tutoriel fournis. Warthog est également muni d'un feu tricolore caractérisant trois états : vert pour le démarrage, orange quand il est prêt et rouge lorsqu'il est en train d'exécuter un code. Il y a également un gyrophare émettant une lumière continue lorsque l'un des boutons d'arrêt est activé, clignotante sinon.

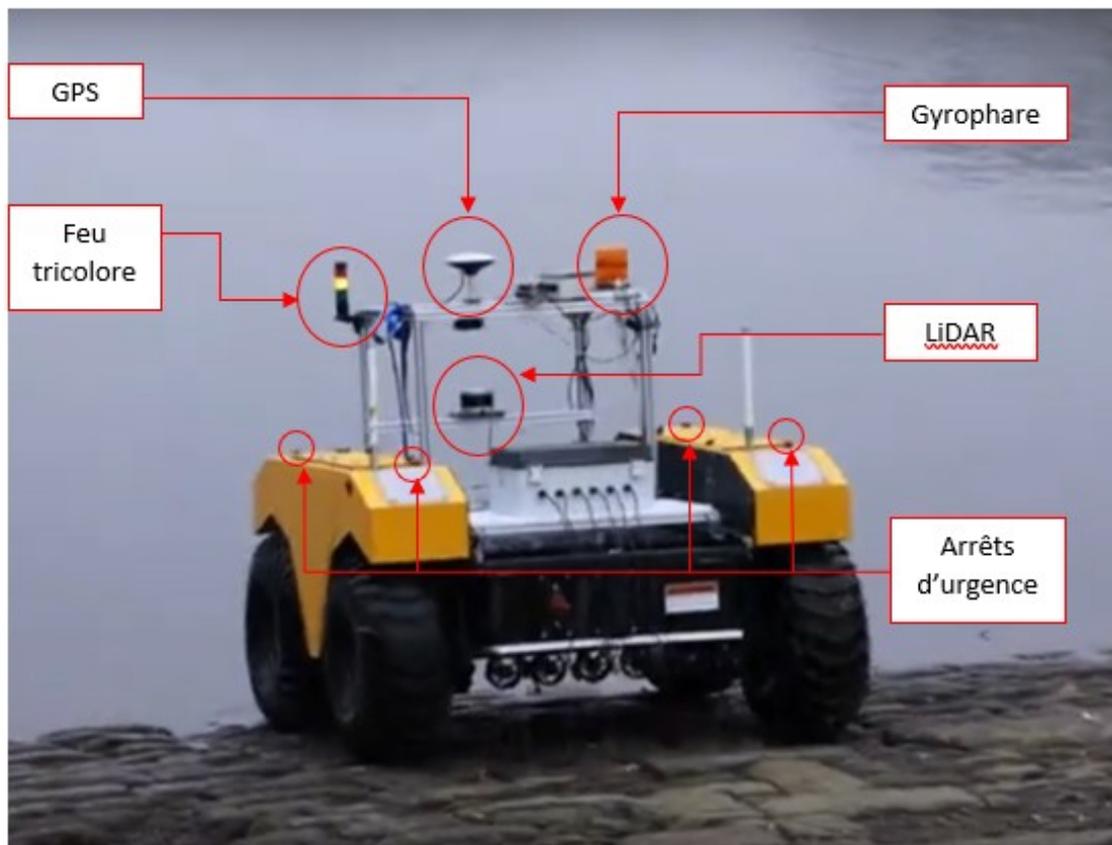


Figure 5 : Equipement initial du robot Warthog



Figure 6 : Télécommande d'arrêt d'urgence



Figure 7 : Télécommande de contrôle manuel

Pour ce qui est de la programmation du Warthog, ce dernier a été conçu sous ROS (en C++). Pour lui envoyer des informations, il suffit de se connecter à son réseau Wifi via la commande « ssh ». Parmi les commandes préinstallées, on trouve y les missions de suivi de ligne, de suivi de cercle et de suivi de « waypoint » (points de passage) donnant des consignes sur la direction à suivre par le robot – retrouvable dans le code « objectif.cpp » du robot.

3. Photogrammetrie

3.1. Installation du matériel

Tout d'abord, il nous fallait installer l'équipement donné par Marie PONCHART sur le robot. Ainsi, nous avons démonté les capteurs et le boîtier de l'équipement pour tout replacer sur le haut de la structure du Warthog. Initialement, nous voulions placer les deux cameras sur l'avant du char, mais après mûre réflexion nous avons plutôt opté pour une installation sur son côté gauche ce qui lui permet de continuer son déplacement lorsqu'il prend ses photos, ce qui est plus facile que devoir faire des marches avant puis marche arrière en face du bâtiment et donc de se déplacer latéralement.



Figure 8 : Installation du matériel de photogrammétrie

3.2. Trouver un logiciel efficace

Par la suite, nous avons commencé à suivre le tutoriel de Micmac, le logiciel de reconstruction 3D utilisé par Marie. Alors que les premières étapes se passait très bien, on s'est retrouvé tous les trois bloqué lorsqu'il s'agit de modéliser un bâtiment entier.

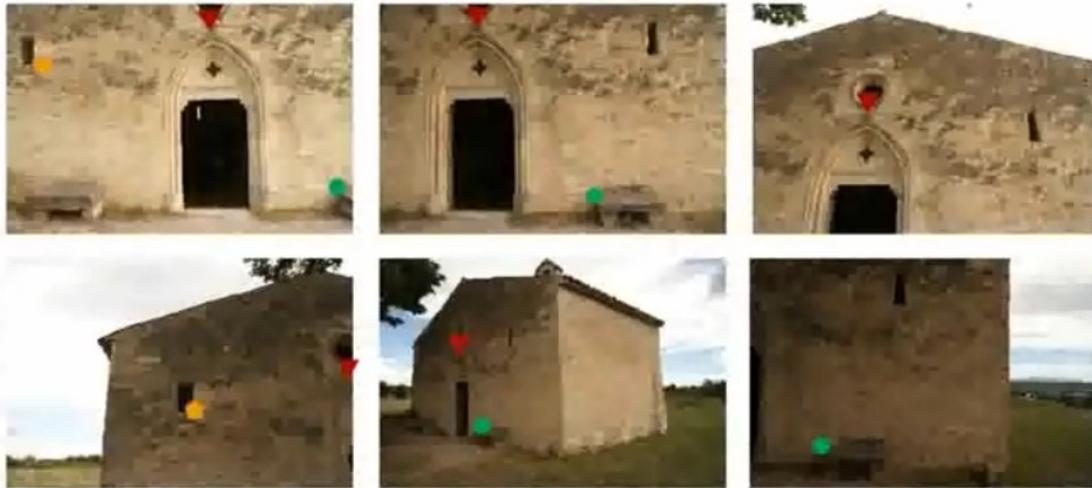


Figure 9 : Jeu de photos d'un bâtiment (les points d'intérêts sont indiqué ici)

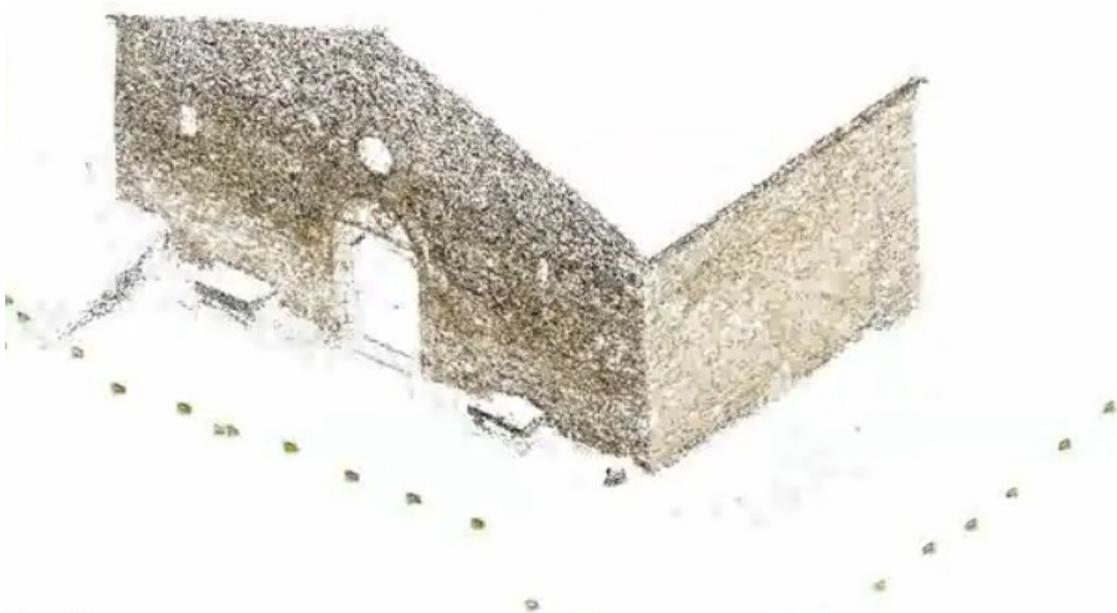


Figure 10 : Traitements des images via le logiciel MicMac



Figure 11 : Rendu que l'on aurait dû obtenir via Micmac

De plus, nous avons discuté avec un doctorant ayant lui-même rencontré les développeurs de ce logiciel qui nous a expliqué que Micmac n'était qu'un simple projet de fin d'étude et n'a jamais eu pour vocation l'utilisation dans le monde de la recherche ou de l'industrie, d'où son développement catastrophique et ses nombreux bugs non corrigés. C'est pourquoi j'ai décidé de regarder ce qui se faisait de mieux en termes de logiciel de photogrammétrie afin d'en trouver un gratuit qui fonctionne sur nos plateformes. Il faut savoir que lancer l'un de ces logiciels est extrêmement chronophage, aussi il faut plusieurs heures pour pouvoir se rendre compte qu'une reconstruction a échoué. De plus, cela nécessite une bonne carte graphique.

Nous avons donc essayé différents logiciels : Zephyr, Visual SFM, Regard 3D, openMVG, Recap - qui n'est pas gratuit, mais l'école ayant une licence Autodesk, nous avons pu l'essayer -, Alice Vision, etc... La plupart de ces logiciels soit étaient trop complexe pour être maîtrisé avec les délais imposés, soit ils ne fonctionnaient tout simplement pas car n'étaient conçus que pour des petits objets. Finalement, nous nous sommes arrêtés sur le logiciel Colmap qui marchait bien sur l'ordinateur d'Enzo qui était muni d'une carte graphique Nvidia. Aussi, Enzo devint le responsable de la partie photogrammétrie.

COLMAP



Figure 12 : Reconstruction de Rome sous le logiciel Colmap

3.3. Communiquer les photos

Pour communiquer les photos à un poste, nous avons utilisé un Raspberry Pi muni de son propre réseau faisant office de relais entre les GoPros et le logiciel de photogrammétrie. Ainsi le Warthog se retrouvait affuté d'un nouvel équipement, qui accumulait les photos jusqu'à rejoindre un poste assez puissant pour effectuer la photogrammétrie. Nous aurions aimé que la carte puisse enclencher le processus par elle-même, mais malheureusement nous n'avons pas réussi à implémenter cela.

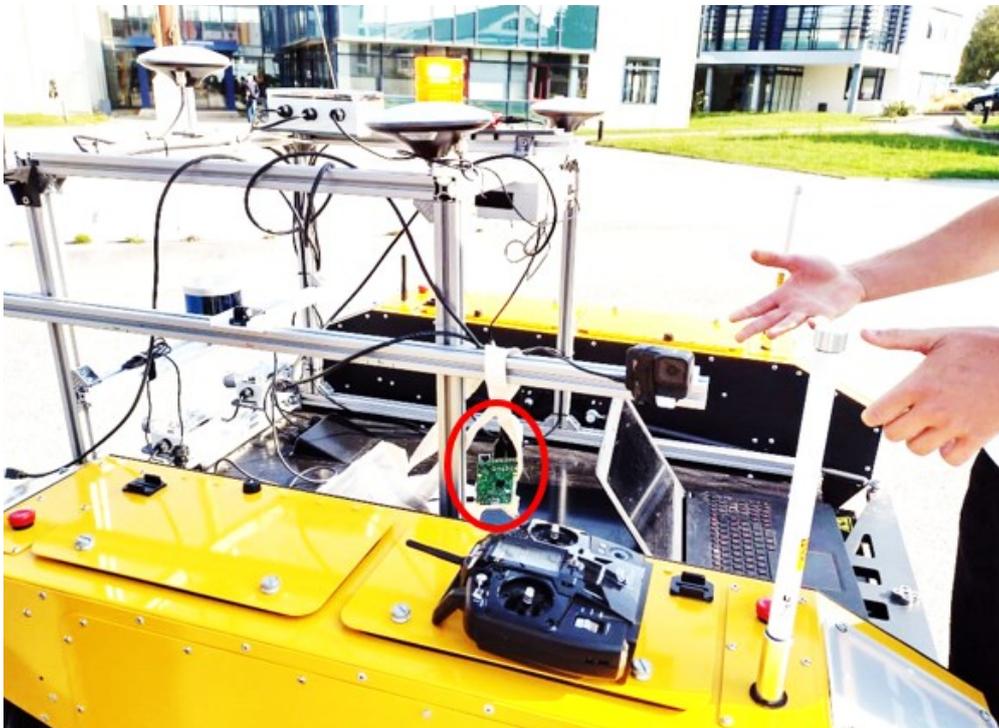


Figure 13 : Traitement des images présente sur le Raspberry Pi

3.4. Traitement photogrammétrique

Avant d'obtenir un modèle 3D, plusieurs étapes sont nécessaires. Premièrement, il faut trier manuellement les photos car il se peut que certaines soient trop floues ou ne montre pas la façade du bâtiment en entière, dans ce cas on doit ajouter de nouvelles photos. De manière général, photographier trop proche des infrastructures n'est pas une bonne chose, à la fois pour la qualité mais aussi parce que les bâtiments empêchent de bien recevoir les données GPS. Cependant, il ne doit pas être trop loin non plus, sinon le logiciel n'arrivera pas à repérer les points d'intérêts.

Parfois la reconstruction restituée peut être incomplète, à cause de la réflexion de la lumière sur les infrastructures. Dans ce cas une reconstruction manuelle peut être nécessaire.

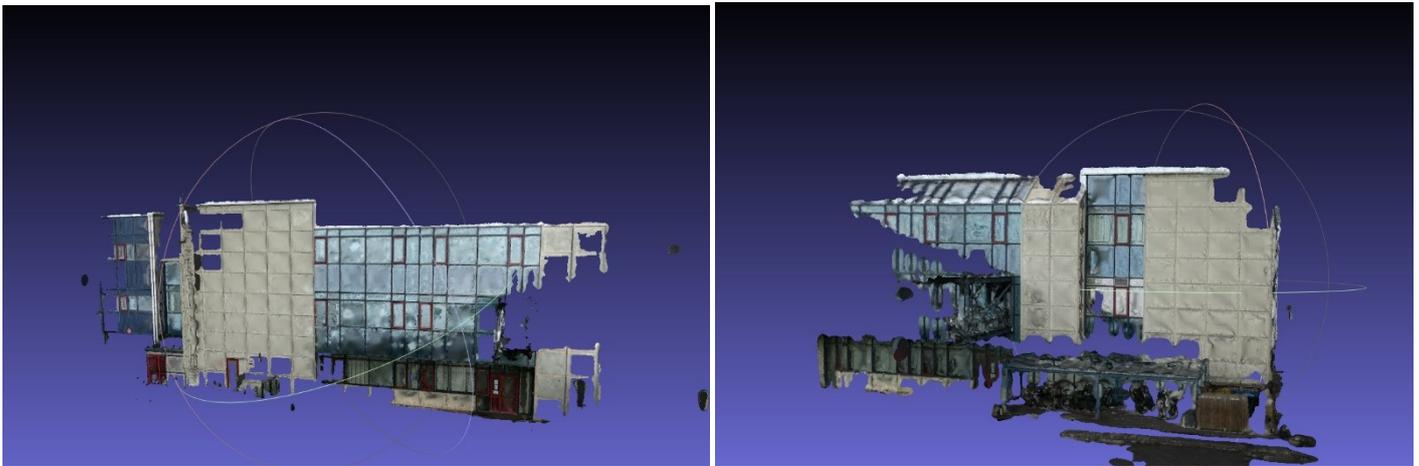


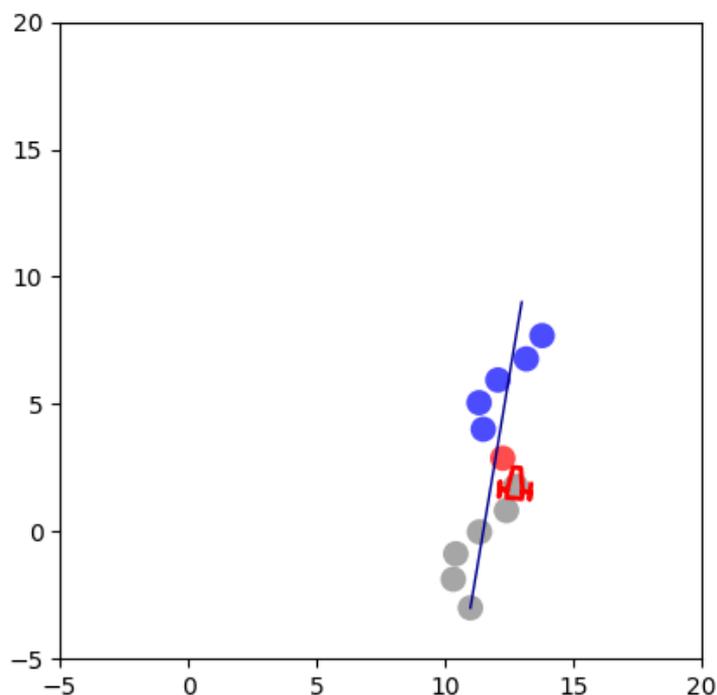
Figure 14 : Reconstruction incomplète

4. Guider le Warthog

4.1. Le guider point par point

Maxime s'est principalement intéressé au déplacement du robot. Nous avons fait la plupart de nos tests sur le terrain de rugby et nous nous répartitions les tâches comme ceci : l'un envoyait les commandes au Warthog avec son ordinateur, un autre manipulait le robot avec la télécommande manuelle pour le placer loin du point d'arrivée et le troisième avait dans les mains la télécommande de secours et prenait des photos. Nous avons testé les différentes fonctions du robot, notamment le suivi de ligne en lui indiquant les coordonnées de début et de fin et le suivi de point en lui indiquant une coordonnée. Le premier problème auquel nous avons souvent été confronté au début était que le char ne comprenait pas l'ordre des missions et se stoppait avant d'atteindre l'objectif. Le second souci était que le char avait du mal à se positionner à l'endroit où l'on fixait l'objectif et bien souvent il ne faisait que labourer le terrain en rond autour du point à atteindre.

Nous avons décidé de distinguer le déplacement du robot en deux phases : une phase déplacement pur et une phase de déplacement avec photo. Lors de cette dernière phase, le Warthog suit une sinusoïde à une vitesse raisonnable, voire même un peu lente, entre deux points GPS désignés préalablement, afin d'avoir des photos dont les points d'intérêts sont reconnaissables.



4.2. Conception d'une IHM

Afin de pouvoir indiquer facilement le chemin que devra suivre Warthog, nous avons décidé de programmer une IHM (Interface Homme Machine) adapté 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. J’ai tout d’abord annoté un maximum de point remarquable sur google map, bien que la vision satellite proposé n’était plus à jour. Par la suite, j’ai travaillé sous tableau Excel et j’ai recherché une formule mathématique me permettant au maximum de faire correspondre ces points particuliers par régression linéaire, rotation, etc... Ainsi j’ai réussi à faire correspondre mes deux plans avec une erreur maximum de 3m.

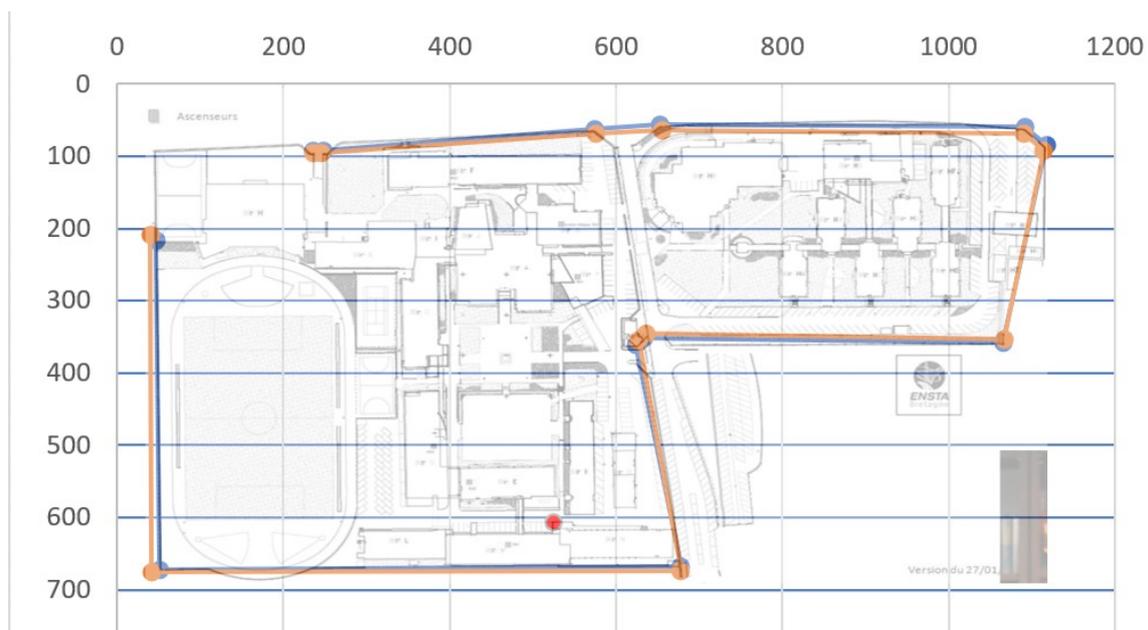
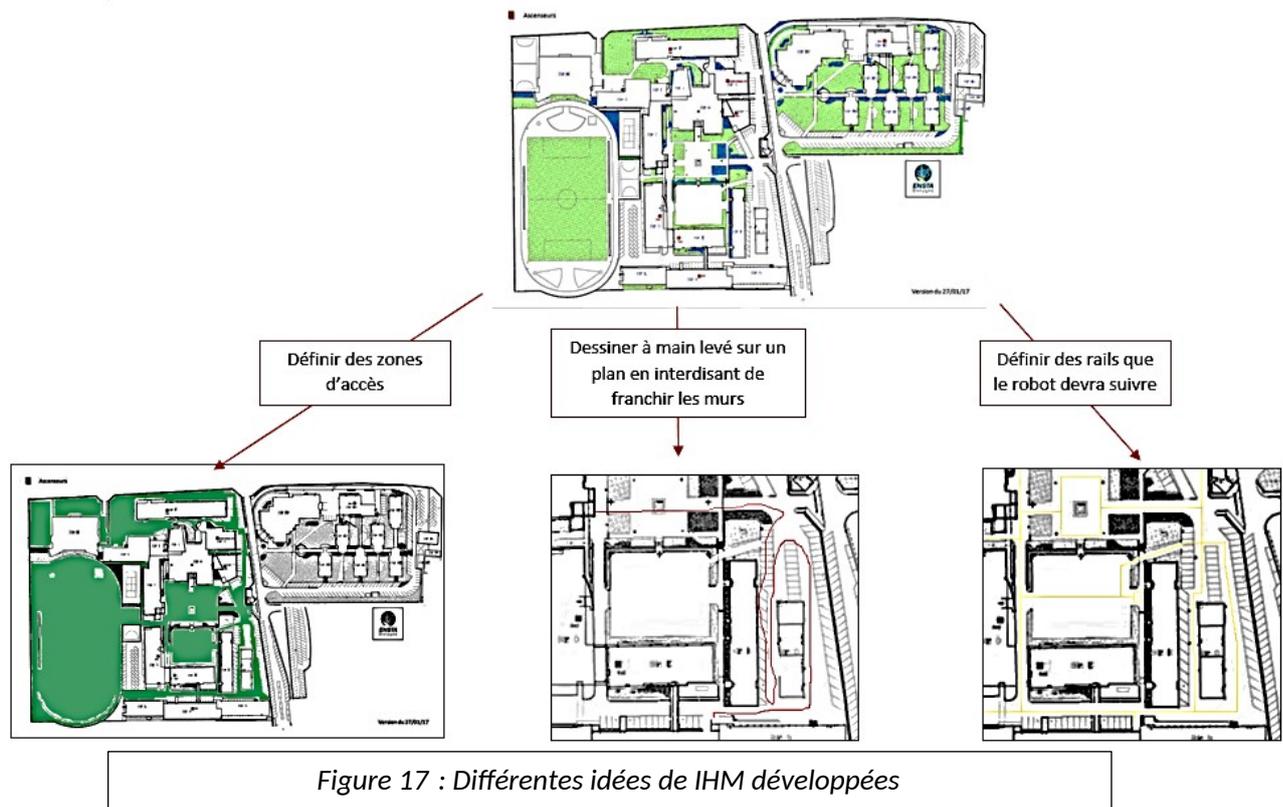


Figure 16 : Visualisation graphique du mapping sous excel avec en bleu les coordonnées en pixel, orange la transformée des coordonnées GPS et en rouge le garage du laboratoire robotique d’où part Warthog par défaut

L’idée derrière cette IHM était de renvoyer à Warthog un fichier texte comprenant les coordonnées par lesquels le char devrait passer et s’il était en mode déplacement pur ou en mode photographe. Bien que nous ayons vu en première année d’école l’utilisation du logiciel Qt designer, j’ai préféré tout d’abord opter pour quelque chose de plus simple et plus proche d’un code python. Après quelques recherches, je suis tombé sur 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, je me suis vite aperçu que cette application n’était pas adaptée à notre utilisation. En effet, cette dernière faisait remonter dans le code d’importants problèmes d’overflow. J’ai donc repris mes recherches et je me suis rappelé d’une bibliothèque dont je m’étais déjà servi pour animer des images sous python : Tkinter. Initialement, je pensais qu’il serait mieux que l’utilisateur trace entièrement le parcours que devrait suivre le robot et que ce dernier lise le chemin tracé notant ainsi une suite de coordonnées à suivre dans l’optique d’utiliser la fonction suivi de ligne déjà implémenté sur Warthog. Tout d’abord, j’ai défini une classe Plan permettant d’indiquer la carte du lieu où on travaille et de dessiner dessus. Puis par la suite, il y a eu plusieurs ébauches : j’ai colorié sur une carte les zones de l’école praticables par le char et il était impossible de tracer un

chemin en dehors de ces zones, j'ai programmé un traitement morphologique via l'utilisation de la bibliothèque openCV permettant de détecter les murs, j'ai également défini 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.



C'est à ce moment-là que l'objectif s'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. J'ai donc décidé d'implémenter une IHM comme la figure ci-dessous.

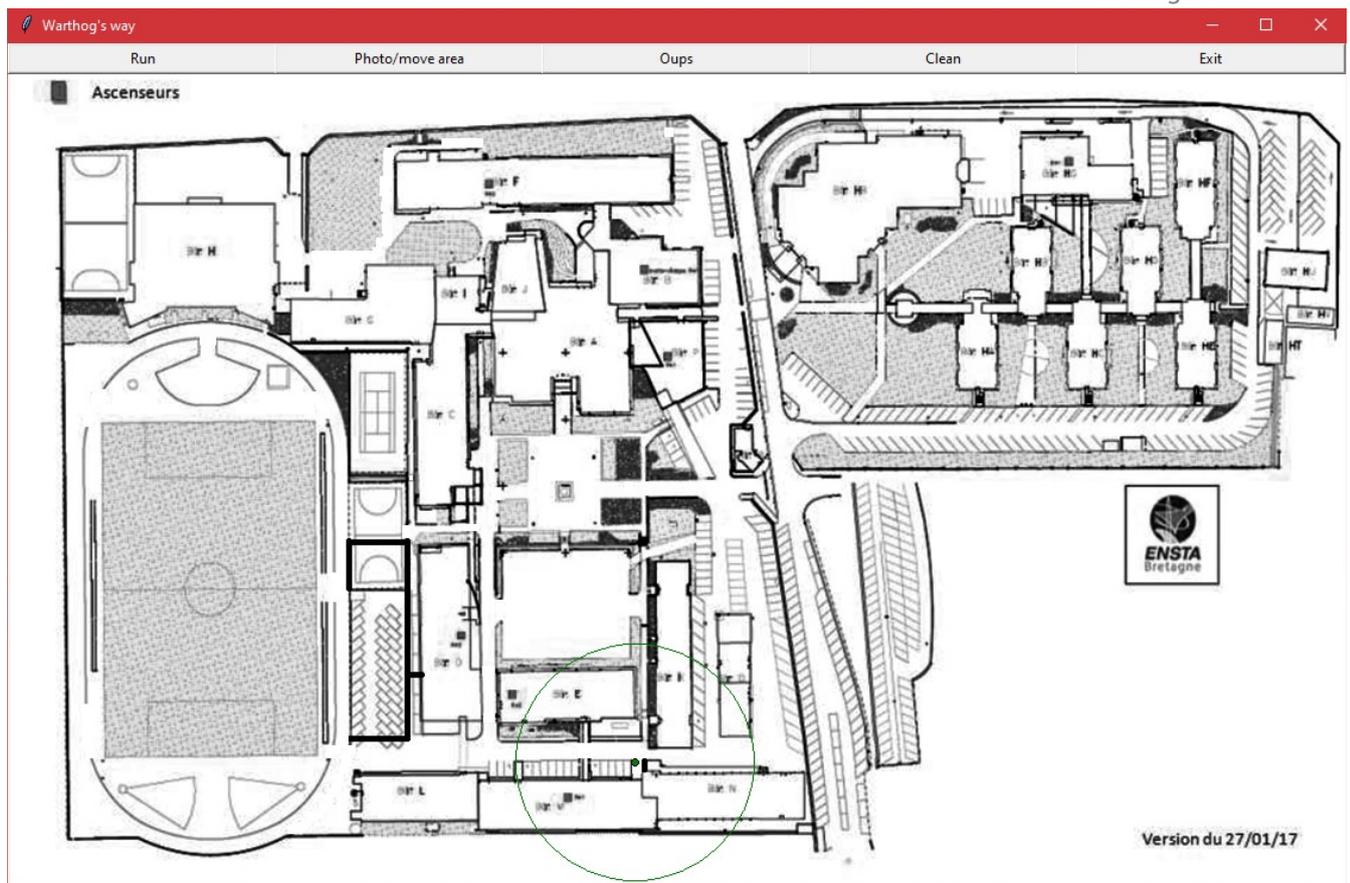


Figure 18 : IHM pour indiquer le chemin à suivre

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 photographe, représenté par un chemin bleu, et le mode déplacement sans photos, représenté par un chemin rouge.
- Oups : 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 en 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. Notez que la dernière position du robot reste en mémoire tant que l'IHM est ouverte.

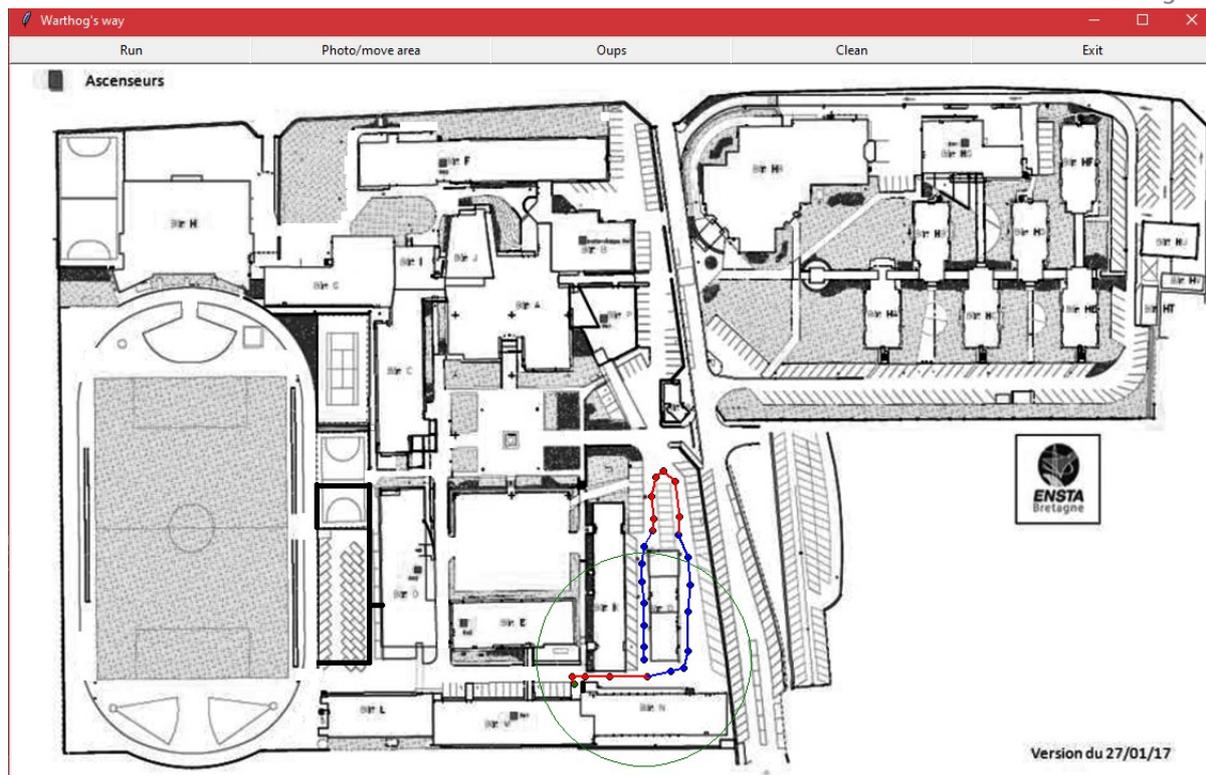


Figure 19 : Exemple de tracé de parcours

5. Résultat

5.1. Warthog fait-il de la photogrammétrie en parfaite autonomie ?

Dans l'état actuel des choses, Warthog ne peut pas faire de la photogrammétrie en parfaite autonomie. Certes, il est capable de parcourir un chemin que l'on lui a préalablement renseigné en différenciant les phases de photographie et de déplacement pur, il prend alors une série de photos des bâtiments que l'on aimerait reproduire et les conserve sur le Raspberry.



Figure 20 : Exemple de mission, Warthog en train de photographier les bâtiments

Cependant, la présence humaine est encore nécessaire pour extraire les données de la carte et les proposer au logiciel Colmap. Un autre problème est que, comme dit plus tôt la photogrammétrie étant très chronophage et exigeante en puissance, plus l'on a de photos prises, plus le processus prend du temps. Après quelques missions et un peu plus de 16 heures de traitement, on peut obtenir des objets similaires aux images suivantes.

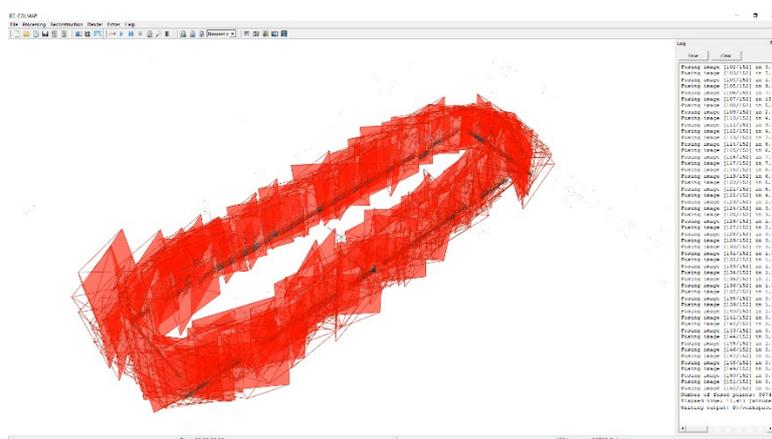


Figure 21 : Phase de traitement des images autour d'un bâtiment par Colmap

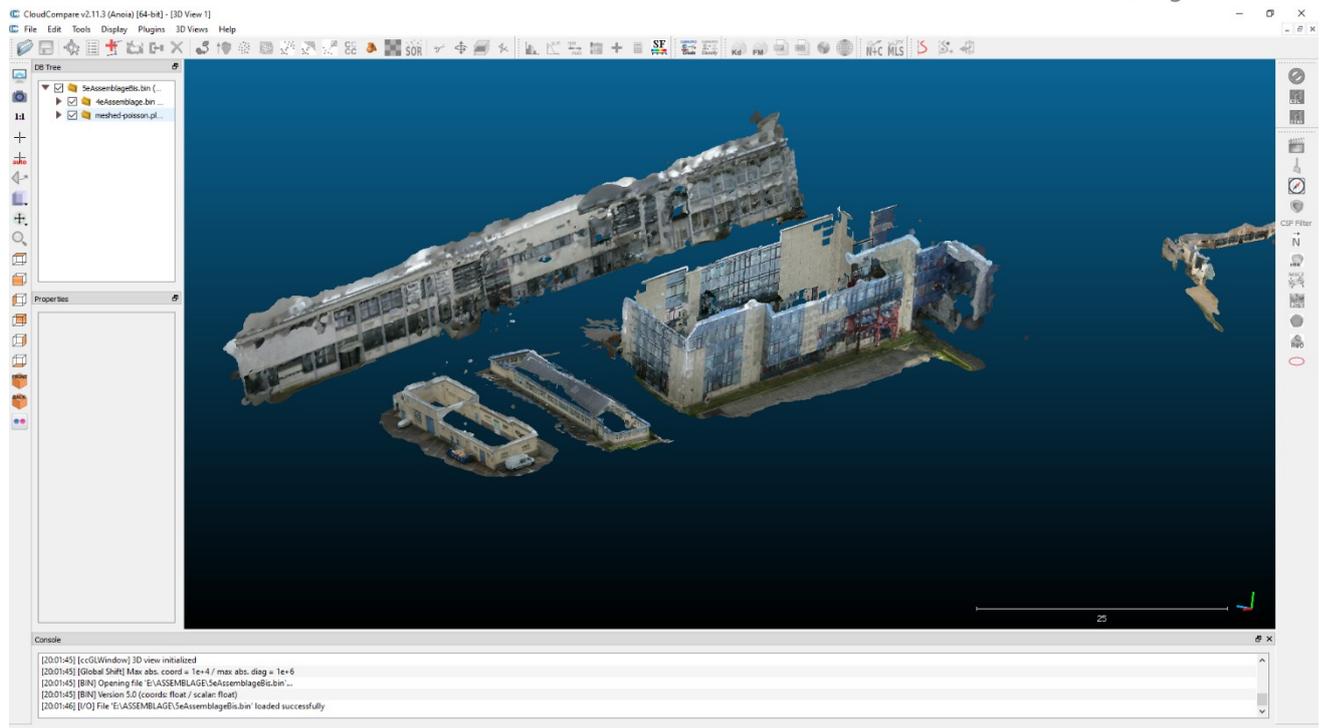


Figure 22 : Reconstruction 3D d'une partie de l'ENSTA par Colmap grâce à Warthog

5.2. Prise de recul

Il reste une partie que nous n'avons pas pu aborder lors du stage : c'est tout ce qui attrait à l'implémentation d'une « vision » sur le robot pour la sécurité et l'auto-localisations. En effet, dans l'état actuel des choses, Warthog est aveugle et sa seule façon de s'orienter est de s'appuyer sur son GPS. Aussi, il est incapable de s'arrêter avant un obstacle. De même, il est plus difficile de lui confirmer qu'il a bien effectué sa mission, alors qu'avec un système de balise, l'on pourrait travailler avec des tubes et d'autres outils visant à prédire son déplacement, ce qui permettrait de palier aux problèmes de perte du signal GPS lorsqu'on est trop proche ou dans des infrastructures. Nous avons bien essayé d'utiliser le LiDAR, mais nous n'avons pas trouvé comment traduire et utiliser les données que le capteur nous renvoyait.

6. Conclusion

Finalement, l'objectif est atteint mais que partiellement. Warthog nécessite un soutien humain dépassant le cadre de l'IHM pour obtenir une modélisation 3D de l'environnement. Cependant, il est tout de même capable de se mouvoir seul et d'effectuer des missions à travers l'école. En tant que photographe, il récoltera les données pour l'utilisateur et lui ramènera afin qu'il puisse terminer la photogrammétrie. Même si en l'absence d'une sécurité automatisée, il reste inconcevable de laisser Warthog se balader sans surveillance.

En ouverture, j'aimerais vous parler d'un logiciel - que j'ai découvert avec mon contrat de professionnalisation - nommé Sikulix. Il s'agit d'un logiciel d'automatisation programmable sous python capable de reconnaître des onglets et d'effectuer des actions tel que cliquer, entrer du texte, taper sur des touches du clavier, etc... J'utilise en ce moment ce logiciel pour faire de l'automatisation de test, et avec du recul, je me dis que cela aurait pu être une bonne alternative à notre problème d'intervention humaine entre les étapes. Même si cela pose toujours un problème pour la détection d'images floues, incomplètes ou pour la reconstruction manuelle.

Ce stage fut très enrichissant, notamment pour nous qui aspirons à devenir roboticien. Certes, il ne nous a pas vraiment permis de découvrir le monde de l'entreprise, mais il nous a entrouvert une porte vers le domaine de la recherche. En effet, l'utilisation d'un tel robot pour faire de la modélisation 3D est quelque chose de très ingénieux permettant d'économiser un temps conséquent en reproduisant une tâche fastidieuse que tout être humain rebuterait à effectuer. Ce stage nous a permis de développer notre gestion du travail d'équipe et demandais beaucoup d'autonomie, ce qui correspond en tout point au cadre favorisé dans le domaine de la Recherche et du Développement de système intelligent.

7. Annexes

7.1. Liste des figures

Figure 1 : *Principe de la reconstruction d'un objet 3D par photogrammétrie (sources : A.Abadie, Sea(e)scape)*

Figure 2 : *Dispositif utilisé par Marie PONCHART pour faire de la photogrammétrie sous-marine*

Figure 3 : *Logiciel MicMac utilisé par Marie PONCHART pour faire de la photogrammétrie sous-marine*

Figure 4 : *Arrière du robot Warthog*

Figure 5 : *Equipement initial du robot Warthog*

Figure 6 : *Télécommande d'arrêt d'urgence*

Figure 7 : *Télécommande de contrôle manuel*

Figure 8 : *Installation du matériel de photogrammétrie*

Figure 9 : *Jeu de photos d'un bâtiment*

Figure 10 : *Traitements des images via le logiciel MicMac*

Figure 11 : *Rendu que l'on aurait dû obtenir via Micmac*

Figure 12 : *Reconstruction de Rome sous le logiciel Colmap*

Figure 13 : *Traitement des images présente sur le Raspberry Pi*

Figure 14 : *Reconstruction incomplète*

Figure 15 : *Simulation du parcours suivi par Warthog sous python*

Figure 16 : *Visualisation graphique du mapping*

Figure 17 : *Différentes idées de IHM développées*

Figure 18 : *IHM pour indiquer le chemin à suivre*

Figure 19 : *Exemple de tracé de parcours*

Figure 20 : *Exemple de mission, Warthog en train de photographier les bâtiments*

Figure 21 : *Phase de traitement des images autour d'un bâtiment par Colmap*

Figure 22 : *Reconstruction 3D d'une partie de l'ENSTA par Colmap grâce à Warthog*

7.2. Bibliographie

- ❖ Tutoriel du logiciel de modélisation 3D MicMac, dernière visite le 26 septembre 2021, <https://micmac.engg.eu/index.php/Accueil>
- ❖ Conférence sur la photogrammétrie sous-marine datant du 12 mai 2021, par Marie PONCHART, [Marie Ponchart : Photogrammétrie sous-marine pour la cartographie haute résolution - YouTube](#)
- ❖ Page de présentation de Colmap, dernière visite le 26 septembre 2021, <https://colmap.github.io/>
- ❖ Fiche d'utilisation du robot Warthog présent sur le site de l'ENSTA Bretagne
- ❖ Rapport de stage assistant ingénieur de Robin SANCHEZ, datant de 2020, [Rapport_SANCHEZ_Robin_2020.pdf \(ensta-bretagne.fr\)](#)

- ❖ Démonstration du robot amphibie Warthog (UGV), ENSTA Bretagne, Brest, publiée le 10 février 2021, [Robot amphibie Warthog \(UGV\), ENSTA Bretagne, Brest - YouTube](#)