



RAPPORT DE STAGE

European Robotics League : Emergency Robots

Yacine BENHNINI

Option Robotique, Promotion 2018, ENSTA Bretagne

encadré par
Fabrice LE BARS

De Juin à Septembre 2017

Résumé / Abstract

English version below

La préparation à la compétition ERL Emergency Robots, anciennement nommée euRathlon, est présentée dans ce rapport, ainsi que les résultats obtenus jusqu'aux épreuves effectuées en Italie à Piombino du 15 au 23 Septembre.

J'ai principalement travaillé sur la partie drone aérien, leur conception, ainsi que les stratégies à adopter pour effectuer les différentes tâches à effectuer lors des épreuves. Une grande part de temps à aussi été investie dans la mise en place des algorithmes et architectures ROS pour l'intégration de différents capteurs les tâches principales étant la cartographie 2D/3D d'une zone, et la détection d'un mannequin.

Le but final était de fournir des drones aériens fonctionnels pour les équipes de l'ENSTA Bretagne et ENSTA Team (collaboration entre ENSTA Bretagne et ENSTA Paris Tech), qui devront collaborer avec des robots terrestres et sous-marins pour remplir les missions imposés dans un temps impartis.

Le concours ERL Emergency Robots consiste en une simulation d'intervention de robots dans une zone sinistrée. Le concours s'est inspiré de l'accident de Fukushima et propose des défis de où des robots aériens, terrestres, et marins doivent collaborer pour mener à bien une mission dans une zone où il serait dangereux d'envoyer des humains. Le concours définit trois approches différentes : mode manuel, les robots sont contrôlés depuis une station de contrôle, mode semi-autonome, et mode autonome. Les approches retenues ont été : autonome pour la partie aérienne et sous marine, et manuel pour les robots terrestres.

Les résultats ont été assez décevants, avec au mieux une 4ème place pour le défi air-terre. Les drones aériens n'ont pas été retenus lors des tests de sécurité, sauf le dernier qui était trop petit pour effectuer l'intégralité des tâches prévues.

This report covers the preparation for the ERL Emergency Robots challenge, previously euRathlon, and the results throughout the different stages until the competition itself, held from the 15-23 September 2017, in Piombino, Italy.

My main tasks were related to AUV. I had to design them and put strategies in place in order to perform several tasks depending on the challenge faced. An other big part of my work was to create algorithms and a ROS architecture for the use of different sensors, focused on producing a 2D/3D map and locating a doll.

The goal consisted in providing aerial robots that would be delivered to ENSTA Bretagne and ENSTA Team (ENSTA Bretagne and ENSTA Paris Tech) teams. The drones have to collaborate with submarines and ground robots.

ERL Emergency Robots is an outdoor multi-domain robotic competition, with challenges inspired by the Fukushima accident. It challenges aerial, ground and marine robots to collaborate and perform tasks in a hazardous area. The mission can be done either manually, semi-

autonomously, or autonomously. We chose to do autonomous aerial and marine robots, and a manual or semi-auto ground robot.

At the end of the competition the results were a bit disappointing considering our best performance was 4th in aerial-land challenge. The drones did not pass the safety checks, apart of the last one which was too small to perform all the tasks.

Table des matières

1	Présentation du contexte	5
2	Thème, objectifs et enjeux du stage	6
3	Description de l'activité et des résultats du stage	7
3.1	Algorithmes utilisés	7
3.1.1	La cartographie	7
3.1.2	La reconnaissance du mannequin	10
3.2	Fabrication des drones	13
3.3	La compétition	15
3.3.1	Le challenge air-terre	15
3.3.2	Le challenge air-mer	18
4	Apport du stage au projet professionnel	20

Table des figures

3.1	Descriptif rapide des capteurs	8
3.2	Carte 2D de la salle robotique (tirée du rapport IQX [6])	9
3.3	Rendu 3D en salle robotique	9
3.4	Matériel utilisé pour la cartographie	11
3.5	Reconnaissance d'un mannequin	12
3.6	Exemple d'un log lors d'un vol durant la compétition	13
3.7	Pièce carbone, procédé d'infusion	14
3.8	Usinage d'un quart de drone dans de la mousse airex	15
3.9	Parcours du drone sur Google Earth	16
3.10	Les points d'intérêts enregistrés lors du vol	16
3.11	Détection du mannequin après découpage en vignettes	17
3.12	Résultat de la carte 2D, au niveau de la plage	18
3.13	Prise de vue des pipelines par le drone	19

Chapitre 1

Présentation du contexte

Dans le cadre d'un sinistre, il est parfois dangereux d'envoyer des équipes sur place comme cela a été le cas pour l'accident de Fukushima. Les radiations sont un risque parmi d'autres qui peuvent être présent sur une zone sinistrée. Pour la sécurité des individus il est alors intéressant de songer à envoyer des robots qui agissent alors en tant qu'éclaireur et peuvent d'ores et déjà récupérer des informations qui permettent une meilleure planification de l'intervention par des humains. L'utilisation d'avions ou multicopter est déjà répandu dans certains domaines comme la surveillance de zones forestières ou de champs agricoles. Il s'agit maintenant de développer des robots capables d'intervenir en plus de renvoyer des informations.

La collaboration entre robots prend alors tout son sens. Si des robots parviennent à effectuer des missions comme le repérage de blessé et l'assistance, via l'apport d'un kit de soin par exemple, de manière totalement autonome, il devient possible d'en envoyer une multitude dans une zone sinistrée sans avoir besoin de beaucoup de personnes dans la station de contrôle. On obtient une plus grande vitesse de déploiement sur zone, et on augmente l'efficacité de la prise en charge de l'accident.

C'est dans ce cadre là que l'ERL Emergency Robots souhaite plonger les équipes participantes, privilégiant ceux qui auront fait les missions en autonome plutôt qu'autrement, au travers de la grille de points. L'ENSTA Bretagne et l'ENSTA Paris Tech ont alors formé deux équipes, produisant les robots pour chaque domaine (ENSTA Paris Tech s'est occupé de la partie terrestre pour l'ENSTA Team). Il a fallu les construire de zéro, sauf pour les sous-marins.

De plus, l'ERL est une organisation qui met en place d'autres compétitions, chacune axée sur des domaines différents, sous la tutelle de euRobotics, une association à but non lucratif qui cherche à promouvoir la robotique en Europe.

Chapitre 2

Thème, objectifs et enjeux du stage

Mon stage avait pour but initial de principalement utiliser les drones développés pendant les projets industriels de deuxième année SPID et de les adapter aux pré-requis des épreuves. Il s'agissait donc de reprendre en particulier un des drones sur lequel j'avais déjà travaillé auparavant. Il s'est cependant avéré que les drones en question n'étaient plus en état de voler au début de mon stage ce qui a demandé par la suite de les remettre en état. De plus, comme plusieurs plateformes étaient nécessaires pour deux équipes, il a fallu en fabriquer de nouvelles, qui suivent des exigences changeantes. En effet les règlements de la compétition continuaient de changer et parfois cela avait un impact non négligeable sur le dimensionnement des plateformes par rapport au premier cahier des charges.

Ce projet était aussi l'occasion pour moi de terminer et mettre en oeuvre les algorithmes sur lesquels je travaillais durant l'année sur une situation réelle et non plus en simulation ou simple test sans but réel. L'utilisation de plateformes différentes impliquait aussi l'utilisation de capteurs différents et je prévoyais donc de rendre l'algorithme plus modulaire quant à l'utilisation de supports matériel multiples.

La stratégie adoptée a été de fabriquer nous même les châssis des nouveaux drones. Malheureusement, l'accès aux installations était compliqué, car on était dépendant de la disponibilité des personnes ayant l'accès, ce qui nous a fait accumuler du retard, notamment pour les tests de vol des drones. Pour effectuer ces tests, nous avons eu la chance de pouvoir accéder au terrain d'aéromodélisme de Milizac, proche de Brest.

Finalement, après les aspects techniques où j'ai pu à la fois travailler sur la partie logicielle et la partie physique des drones, la participation à la compétition a été une expérience très intéressante et enrichissante. C'était l'occasion de voir d'autres solutions aux problèmes techniques abordés, et de découvrir d'autres projets, d'autres façons d'envisager et de gérer les choses.

Chapitre 3

Description de l'activité et des résultats du stage

3.1 Algorithmes utilisés

Les différentes épreuves de la compétition nécessitent principalement deux fonctionnalités : la cartographie d'une zone, la reconnaissance d'un mannequin (le "missing worker").

3.1.1 La cartographie

Plusieurs approches étaient possibles pour la cartographie. Tout d'abord il faut choisir si on effectue une carte 3D ou 2D de la zone, puis vient le choix du capteur associé. Pour éviter de se refermer sur une solution, j'ai commencé par tester tout ce qu'il y avait de disponible parmi le matériel présent au laboratoire, et envisagé ce qu'il y avait de disponible sur le marché et à quel prix. Bien sûr, tout ceci était mis en parallèle du cahier des charges de la compétition, en particulier le plan de vol, et les conditions environnementales (visibilité, etc). Un rapide descriptif d'éléments couramment utilisés dans le domaine est présenté sur la figure 3.1. 3.1.

Il convient de prendre en compte le temps de traitement des données autorisé pour chaque épreuves. Il est de 45 minutes pour les épreuves doubles (air-terre, air-mer, terre-mer), et de 1 heure pour l'épreuve "Grand Challenge" (air-terre-mer). Les points à retenir suivant les capteurs sont les suivants :

- caméra classique : étant donné la taille de la caméra Mobius, elle est très facile à intégrer à un drone, même de petite taille comme un Parrot Beebop. Elle permet aussi d'avoir un flux vidéo en direct, via un canal FPV, en même temps qu'elle enregistre en haute résolution pendant le vol. Il est ainsi possible de faire un repérage en temps réel du mannequin, puis une cartographie à la fin de la mission. Malheureusement, le temps de calcul est trop élevé pour effectuer une carte 3D, il est simplement possible d'avoir un rendu 2D à partir de la vidéo via des logiciels comme Microsoft ICE.

- caméra 3D : elle sont assez large pour certains modèles comme la Kinect, et trop lourde pour des petits drones. Il faut aussi les associer à un ordinateur embarqué qui s'occupera de l'acquisition des données, et il faut ajouter une transmission wifi pour pouvoir faire un traitement en temps réel. Ainsi, on a une contrainte de poids élevé qui force à avoir des structures larges, de l'ordre d'un drone comme le DJI M600. De plus leur portée réduites nécessite que le drone puisse se stabiliser parfaitement en altitude, car l'idéal serait de voler à une altitude d'au plus 5 mètres.

Caméra 3D		Modèle: prix entre 500-1000€
transfocuteur		Kinect2: vol., cble. Portée 5m Prix de l'ordre de 1000-2000€
Lidar 2D		RPLidar: portable 150m Hokuyo portable 120m Prix entre 500-1000€
Lidar 3D		Velodyne: portable 20m prix 20,000€

FIGURE 3.1 – Descriptif rapide des capteurs

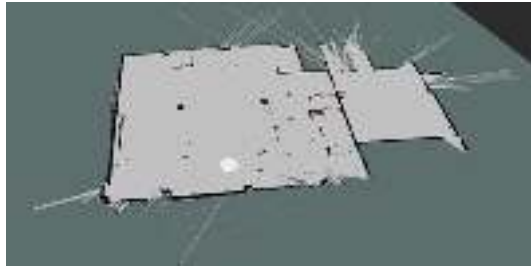


FIGURE 3.2 – Carte 2D de la salle robotique (tirée du rapport IQX [6])



FIGURE 3.3 – Un rendu 3D avec Rtabmap, une kinect, en salle robotique, affiché sous blender.

- lidar 2D : dans le cas présent, il est intéressant de les coupler avec une caméra 3D. Ainsi, la localisation se fait à partir du lidar 2D tandis que la caméra ne s'occupe que de la carte, plutôt que d'utiliser des techniques de slam 3D qui sont gourmandes en calcul et ne fonctionnent pas bien lors de déplacements rapides. Il faut être capable de maintenir une altitude constante pour leur usage aussi, et compenser les variations de roulis/tangage à l'aide de la centrale inertielle du contrôleur de vol.

- lidar 3D : ils sont très chers mais permettent le plus de flexibilité car ont plus de portée et sont plus précis que les rendus 3D à partir de caméra. Il faut par contre une bonne connexion wifi afin de récupérer le plus d'informations du capteur possible, pour le rendu temps réel.

Ma stratégie retenue pour l'utilisation des capteurs a été de faire fonctionner les lidar 2D seuls, les caméras 3D seules, puis de fusionner les capteurs, et enfin d'y ajouter les données du contrôleur de vol, notamment l'altitude et les données renvoyées par la centrale inertielle. Pour ce faire, j'ai utilisé comme base logicielle ROS (Robot Operating System), qui était tout à fait appropriée pour mon cadre d'utilisation par sa modularité, et car je commence à avoir de l'expérience avec.

Après de longs tests de différents paquets ROS concernant la cartographie 3D, mon choix

s'est fait pour RTABMAP [3] qui selon moi donnait de meilleurs résultats que la librairie octomap. L'avantage d'octomap est de pouvoir utiliser des lidars en entrée pour effectuer la carte, mais comme je n'avais pas les lidars adaptés pour obtenir un rendu 3D, je suis passé sur l'autre librairie qui est spécifique à l'usage de caméra 3D. De plus rtabmap supporte beaucoup de matériel différent : les caméras RGB-D (D pour Depth, canal pour l'image des distances aux obstacles), les caméras en stéréo vision comme la zed (on obtient la profondeur suite à un traitement à partir de deux caméras RGB), et globalement toutes les caméras supportées par la librairie OpenNI2. Ainsi il y avait peu de problèmes liés aux drivers, grâce à OpenNI2 qui permet un support de quasiment toutes les caméras RGB-D ou RGB-RGB, voire uniquement D comme pour le structure sensor. Et dans le cas de la zed ou de la kinect il suffisait de passer par des drivers spécifique comme freenect ou freenect2. De plus, la documentation est suffisamment fournie pour s'en sortir en cas de problème, et pas mal d'exemples sont donnés pour l'utilisation conjointe avec ROS.

Concernant octomap et l'usage des lidars, j'ai quand même fait un test avec le leddarOne, qui envoie un faisceau en cône, sur environ 50 mètres, mais le résultat était inutilisable. Non seulement l'installation est fastidieuse, mais le remplacement d'un lidar 3d par un lidar 2d qu'on fait tourner soi même est peu envisageable, ou alors nécessite d'investir suffisamment de temps, au détriment d'autres parties du projet.

Une fois que les capteurs fonctionnaient séparément, j'ai commencé à les fusionner. Il s'agissait donc d'utiliser la localisation du lidar (en l'occurrence un RPLidarA2) sur lequel se repose la génération du nuage de points de la caméra (une kinect 2). En comparant l'utilisation d'une caméra seule, puis l'association caméra-lidar, les points clés à retenir sont :

- une bien meilleure précision au niveau de la localisation lorsque le lidar est utilisé, comme ordre de grandeur, sur un parcours d'environ 30 mètres en revenant au point de départ, le décalage était de l'ordre du centimètre.
- le système de détection de fermeture de boucle [3], permettant d'éviter de dédoubler un objet et de corriger sa position quand on passe deux fois au même endroit, fonctionne un peu mieux avec une caméra seule. Pour augmenter la qualité du rendu, il est possible de faire du post-processing manuellement, en vérifiant soi même les zones où une fermeture de boucle a été repérée, puis en la modifiant ou supprimant, on peut aussi en ajouter d'autres, etc.

3.1.2 La reconnaissance du mannequin

Après maintes recherches, mon choix s'est porté sur l'utilisation du dépôt Github py-faster-rcnn [4]. Un réseau RCNN (Region-based Convolutional Neural Network) a l'avantage de localiser en plus de classifier, ce qui est parfait dans mon cas de figure, car en plus de localiser le mannequin il fallait larguer un kit de soin à proximité. En sachant où se trouve l'entité repérée sur l'image et en sachant la position de la caméra et du kit sur le drone, il est facile de positionner le drone afin que le kit tombe à proximité de la cible.

Durant la période où je travaillais sur cette partie, je n'arrivais pas à trouver de liens encore fonctionnels vers le fichier contenant le réseau déjà entraîné. J'ai donc procédé à l'entraînement moi même à partir d'un réseau pré-entraîné et d'une plus petite banque de donnée, tirée du PASCAL VOC2007 [2]. Ainsi il était possible de détecter 20 classes d'objets dont en particulier les personnes. J'ai donc appliqué cette détection à des photos de mannequins tirées des compétitions précédentes, et ça a très bien fonctionné. J'ai ensuite effectué d'autres tests en prenant des photos de mannequins sur le web, et j'en ai conclu que le principal facteur de détection semblait être les vêtements plutôt que la forme d'une personne. Un mannequin sans gilet n'est pas correctement détecté, et au plus la texture est lisse et brillante, au moins ça marche, ce qui est logique quand on compare avec la peau d'un être humain.



FIGURE 3.4 – Photo de la configuration utilisée pour le test de la cartographie 3D. On voit le contrôleur HK32Pilot à gauche, équipé d'un LidarLite pour obtenir l'altitude (invisible de ce point de vue), le RPLidarA2 au milieu et une kinect à droite.

Une fois ces tests effectués sur image fixe, j'ai ajouté la prise en charge d'un flux vidéo pour évaluer le comportement en temps réel. Le repérage était correct à une distance de 10 mètres, ce qui était suffisant, mais l'algorithme était un peu trop lent, même avec un flux vidéo en 360p voire moins, et à 10-15 images par secondes. Pour palier à cela, j'ai rapidement essayé de modifier le réseau afin de n'avoir qu'une seule classe à différencier, ce qui diminuait sa taille et donc augmentait la vitesse de calcul. Malheureusement je n'ai pas réussi à l'entraîner correctement, les détections étaient bien moins correctes et je n'avais plus vraiment le temps de m'attarder sur le problème.

Au final, même avec une détection à seulement 10 fps, il est possible de s'adapter en diminuant la vitesse de déplacement du drone lors de sa surveillance pour être certain qu'une image avec le mannequin soit traitée par le programme.



FIGURE 3.5 – Reconnaissance d'un mannequin : on constate que bien qu'il soit beaucoup plus petit, le mannequin habillé est repéré en tant que personne et pas l'autre.

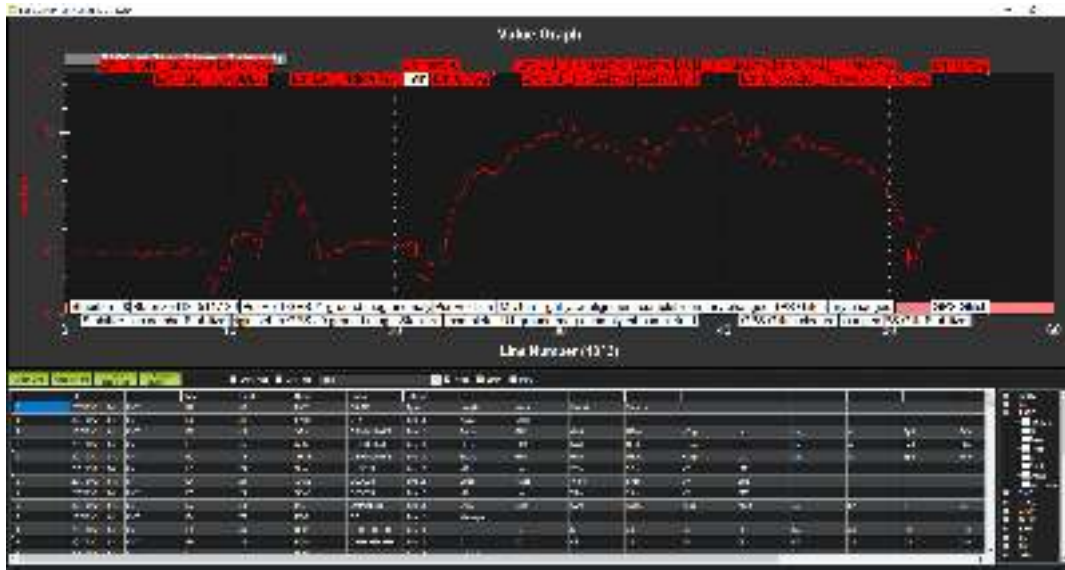




FIGURE 3.7 – Pièce carbone, procédé d'infusion

Après les problèmes mécaniques, des problèmes électroniques ont été identifiés, notamment concernant l'alimentation du contrôleur de vol. Les ESC (Electronic Speed Control) avaient pour la plupart un BEC (Battery Eliminator Circuit) intégré, ce qui ajouté au power module montait à 5-6 le nombre d'alimentations différentes arrivant sur le contrôleur. Or le circuit n'était pas prévu pour prendre en charge ce genre de configurations, le maximum étant une arrivée sur l'entrée "power" correspondant au power module, permettant en plus d'avoir en direct le niveau de tension de la batterie, et une arrivée sur les entrées/sorties principales et auxiliaires, là où les ESC et les capteurs sont branchés. La présence de deux arrivées permet d'être certains que les capteurs puissent être alimentés correctement, grâce à la redondance des sources d'alimentations. Un petit ajout d'un circuit avec diode et condensateur a de plus permis de stabiliser le courant à 5.7 volts comme conseillé par les documentations du firmware arducopter [1].

Pour le reste des drones, il a été décidé de faire nous même les châssis, en utilisant de la mousse aircx sur laquelle on fixe des couches de carbone avec de l'époxy. Le choix de la fibre de carbone est justifiée par sa faible densité et une très bonne résistance mécanique. Le poids étant un facteur très limitant dans le cas des drones, il était obligatoire d'utiliser un matériau léger, mais suffisamment résistant pour embarquer les capteurs requis, et résister un minimum aux chocs.

Le procédé utilisé a été l'infusion. On pose les tissus de carbone sur la mousse, puis on y ajoute un tissu d'arrachage puis un film micro-perforé, le tout dans une bâche étanche à l'air. Ensuite viennent les tuyaux, une partie reliée à la pompe pour la mise sous vide, et l'autre reliée à un pot rempli du mélange époxy.

Une fois le temps de séchage terminé, on déballe la pièce, puis on découpe et ponce les bords pour obtenir la forme voulue dans les zones où il n'a pas été possible de replier la feuille. Les patrons en mousse ont été usiné avec une fraiseuse, ou découpé à la main au cutter.

L'équipement des drones a été au maximum homogénéisé afin qu'une panne sur l'un puisse permettre d'anticiper sur les autres. Les même paramètres ont été utilisés au niveau du contrôleur de vol, sauf pour le réglage des PID, qui a été fait avec la fonction d'autotune du firmware. Dans ce mode de vol, le drone va envoyer des accoups selon un axe, et les données renvoyées par la centrale inertielle lui servent à se régler. Globalement le réglage par autotune a rendu les drones assez nerveux, ce qui n'arrangeait pas le pilotage manuel, mais une nette amélioration dans la qualité de vol était constatée pour les vols en mode automatique.

Avec les modules Crossfire, tout a été ramené à une seule télécommande pour contrôler les drones, ainsi il n'y avait plus à s'embêter à chercher la bonne télécommande du bon drone, et la



FIGURE 3.8 – Usinage d’un quart de drone dans de la mousse airex

télémetrie était intégrée, ce qui fait donc un composant de moins à s’occuper. Malheureusement, la connexion en bluetooth a un débit extrêmement lent, entraînant une durée de deux à cinq minutes pour se connecter en télémetrie avec Mission Planner.

3.3 La compétition

3.3.1 Le challenge air-terre

Ce scénario met l’accent sur le fait de devoir faire une reconnaissance d’une zone à l’aide d’un robot aérien et de robots terrestres. Le drone va inspecter l’extérieur du bâtiment et en rapporter le plus d’informations possible tout en repérant le mannequin, tandis que le robot terrestre s’occupera de l’intérieur du bâtiment, dans lequel il doit aussi trouver un mannequin [5].

Il faut aussi être en mesure de déposer un kit de soin auprès de chaque mannequin, et repérer des points d’intérêts, comme un mur endommagé, et référencer le tout dans un fichier kmz. De cette manière on a pour chaque zone intéressante des coordonnées gps correspondantes, et une carte produite géo-référencée.

Les juges de la compétition ont imposé des formats de fichiers tels que le kmz, ou le ply, car ce sont des formats pris en charge par la plupart des logiciels, notamment les logiciels gratuits. Ceci permettait de faciliter le portage des données d’un ordinateur à l’autre et de ne pas se retrouver coincé pour une incompatibilité de format.

Nous avons malheureusement eu des problèmes lors du passage des tests de sécurité de nos drones. Ils étaient pilotés en mode "stabilize", c’est à dire en mode manuel avec stabilisation automatique de la part du contrôleur de vol. Une fois en vol, il y a eu des réactions excessives des moteurs qui mettaient le drone quasiment sur la tranche, provoquant alors un décrochage, avant de se reprendre quelques mètres plus bas. Les erreurs visibles sur les logs étaient liées à la boussole pour chacun des drones, bien qu’on ai mis des contrôleurs neufs pour éviter ce problème précis qui était apparu bien avant pendant nos tests sur le terrain de Milizac.

Plusieurs raisons peuvent justifier un tel comportement. Soit il faut éviter le mode "stabilize" à partir d’une certaine taille de drone, le petit n’ayant pas montré un tel comportement. Soit les contrôleurs ont pris un coup pendant le voyage. Soit le contrôleur lui même n’est pas fiable, selon les juges qui nous ont dit avoir eu de mauvaises expériences sur ces plateformes là.



FIGURE 3.9 – Parcours du drone sur Google Earth



FIGURE 3.10 – Les points d'intérêts enregistrés lors du vol



FIGURE 3.11 – Détection du mannequin après découpage en vignettes

Ainsi nous n'avons utilisé que le petit drone qui est un ancien Parrot Beebop, dont seul le châssis et les moteurs sont d'origine suite à un crash qui a fait bruler son électronique. Il ne nous restait donc plus qu'une dernière solution de secours pour effectuer certaines tâches, à savoir installer une caméra Mobius, qui est suffisamment petite pour être installée sur un drone de cette taille, et utilisé le flux vidéo pour repérer le mannequin, les points d'intérêts, et produire une carte.

Le temps de calcul pour produire une carte 3D étant beaucoup trop long, il a fallu se contenter d'une prise 2D. Pour pouvoir faire une carte 3D, il aurait fallu un recouvrement d'au moins 60% entre les photos pour avoir un rendu un minimum correct, ce qui nécessitait environ 120 photos pour cette zone considérant la focale de la caméra, soit au moins deux heures de calcul sur le logiciel qu'on avait à disposition.

Ensuite, concernant la détection du mannequin en temps réel, il s'est avéré que la cible n'occupait qu'un tout petit espace de l'image totale, donc l'algorithme ne repérait rien car il n'a probablement pas rencontré de cas similaire dans ses données d'entraînement. J'ai essayé de rajouter des découpages, ce qui a fonctionné mais à partir d'un découpage de l'image en 66 vignettes. Autant dire qu'on était à 0.2 fps pour la détection ce qui devenait complètement inutile.

Pour palier à ce problème il aurait suffi de voler plus bas, mais ce n'était pas possible avec ce drone en particulier car son seul capteur d'altitude était un baromètre, contrairement aux autres drones qui avaient en plus un lidar dont les mesures étaient effectives entre 2 centimètres et 40 mètres. Ainsi il était impossible d'assurer un maintien à une altitude stable de l'ordre de 5 mètres sans risquer de crasher le drone, car le baromètre n'est pas assez précis pour ce genre de besoin.

Une autre solution soulevée aurait été de simplement faire un algorithme de détection de



FIGURE 3.12 – Résultat de la carte 2D, au niveau de la plage

couleur. Le gilet orange fluorescent est en effet en net contraste avec le sol, terne et gris au niveau du sable, à peine plus coloré lorsqu'il y a de l'herbe. Cependant les points d'intérêts étaient sujets à modification entre chaque passage d'une équipe. Le mannequin était déplacé, les marqueurs pour indiquer les dégâts sur les murs l'étaient de même.

Et le point important empêchant l'algorithme de fonctionner : le gilet était aussi modifié. On pouvait donc passer de orange à jaune, ou plus ou moins fluorescent, d'une épreuve à l'autre, rendant peu viable une approche par détection de couleur, d'autant plus que selon le moment de la journée et les changements de luminosité, on pouvait être sujet à des fausses alertes, et donc à un largage de kit au mauvais endroit en plus d'un positionnement GPS faux du point d'intérêt.

Au niveau de la collaboration entre robot aérien et terrestre, l'idéal aurait été d'avoir pu cartographier la zone en 3D à l'aide d'une reconnaissance du drone, puis avec ces données de faire naviguer le robot terrestre jusqu'à l'entrée du bâtiment en choisissant un chemin le moins escarpé possible. Ainsi le robot terrestre a plus de chance de rejoindre l'entrée du bâtiment sans encombre, où il peut alors commencer son inspection. Le largage d'un kit de soin depuis le drone vers le robot aurait été relativement simple, un simple échange des coordonnées GPS entre robot aurait suffi.

3.3.2 Le challenge air-mer

Le challenge air-mer se focalise sur la détection de dégâts au niveau de pipeline reliés au bâtiment et leurs embouchures dans la baie un peu plus loin, ainsi que la détection des "missing worker" [5].

Pour le cas du drone, la détection reste sur le même principe que précédemment et la cartographie est réduite à une zone plus petite autour du bâtiment.

Le point changeant est donc le plan de vol, et la coopération entre robot nécessite que le drone aérien trouve le sous-marin au moment où celui-ci fait surface. Le but étant que le drone enregistre sa position GPS au moment de la détection pour l'envoyer au sous-marin qui peut donc se recalculer.

Ce recalage se justifie par le fait que les AUV possédait une bouée mais vide de toute électronique. Il n'était donc pas possible d'intégrer un GPS à la bouée pour recalculer le sous-marin.

En soi, il s'agissait du challenge le moins "collaboratif", les deux robots faisant leurs tâches dans leur coin tout au long de la mission sauf pour ce passage de recalage. L'utilité de la détection prenait tout son sens lors du Grand Challenge, où l'information du pipeline endommagé est envoyé au robot terrestre qui doit alors fermer la bonne vanne.



FIGURE 3.13 – Prise de vue des pipelines 1 et 3 par le drone

Chapitre 4

Apport du stage au projet professionnel

Ce stage m'a permis d'approfondir mes connaissances dans le domaine des drones que j'avais acquises durant l'année pendant les projets industriels. J'ai pu en apprendre plus sur les différents problèmes que l'on peut rencontrer et les attentes qu'ont des professionnels à ce sujet lors de la compétition. Je pense aussi que je ne travaillerai pas dans ce domaine plus tard car bien que j'ai aimer travailler dessus, je ne me vois pas en faire ma carrière professionnelle.

Pour ce qui est de l'utilisation des capteurs et des algorithmes, je pense m'être beaucoup améliorer. A la fois dans le choix des capteurs et leur utilisation que dans les algorithmes à mettre en uvre. Je pense que cette partie de mon stage me servira beaucoup dans d'autres domaines, que ce soit de la robotique mobile ou autre, car les systèmes de visions pour robot deviennent de plus en plus performant et donc utilisé, notamment pour les voitures autonomes, un autre domaine qui m'intéresse.

De plus c'était l'occasion de voir les différentes approches proposées par d'autres équipes à travers l'Europe et de se rendre compte de ce qui pouvait se faire, à la fois en milieu universitaire comme en entreprise, car certaines équipes étaient issues d'entreprises. Non seulement les approches techniques étaient intéressantes mais aussi les différentes façons de s'organiser pour un tel évènement.

Pour finir, cette compétition m'a beaucoup motivé à essayer d'autres domaines et projets différents, car j'ai été très content de voir des robots un peu exotique tel que le robot chien de l'équipe de l'ETH Zurich fonctionner devant moi plutôt qu'en vidéo. Tout ceci m'a permis d'élargir mon champ de vision sur différents domaines de la robotique.

Conclusion

Ces trois mois de stages m'ont permis de travailler à la fois sur la partie logicielle et la partie physique du développement de drones. J'ai progressé dans ma compréhension et mon usage de ROS ce qui sera très certainement utile pour mon futur. De plus j'ai pu me familiariser avec différentes techniques de détections, localisation et mapping. Ceci me permet d'élargir mes champs de vision et de compétences que je pourrai ensuite appliquer à de futurs projets. J'ai bien entendu pu utiliser mes acquis développés durant l'année pendant les projets industriels et les cours.

Le point noir a été que je n'ai pas eu le temps de faire certaines choses que j'aurai voulu entreprendre, en particulier sur la partie software, où j'ai plus eu l'impression de faire de l'assemblage d'outils déjà existant en les forçant à faire ce pour quoi ils n'étaient pas vraiment prévu plutôt que d'en faire une partie moi même afin d'éviter les problèmes où on se retrouve face à une boîte noire impossible à déboguer sans documentation. Cependant l'usage intensif de ROS m'a permis de bien m'améliorer dans ma compréhension de ce système, et j'aurai très certainement bien moins de mal à me servir de futurs capteurs et paquets ROS à l'avenir, ce qui est très bénéfique.

L'objectif de faire des drones totalement opérationnels pour la compétition n'a pas été rempli à cause des problèmes rencontrés sur place mais j'ai confiance dans le fait que les drones auraient pu faire les tâches souhaitées si ils pouvaient voler. Ainsi, même si c'est une déception de n'avoir pas pu participer au Grand Challenge ou de réaliser plus de tâches, je suis tout de même content d'avoir pu travailler sur ce projet. Il aura manqué la finition.

Bibliographie

- [1] DIY Drones. Site de arducopter. <http://copter.ardupilot.org>.
- [2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [3] M. Labbe and F. Michaud. Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2661–2666, Sept 2014.
- [4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN : Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.
- [5] ERL Staff. Erl emergency robots, rule book. <http://www.robotics-league.eu>.
- [6] Benhnini Yacine Veylon Nicolas, Tuton Sophie, Mayssal Mathieu Tillet Joris, and Ouadrhiri Mohamed Amine. Projet iqx.