

UNIVERSITÉ DE MONTRÉAL

MÉTHODES D'INSPECTION AUTOMATIQUE D'INFRASTRUCTURE PAR ROBOT  
MOBILE

ANDRE PHU-VAN NGUYEN  
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE ÉLECTRIQUE)  
DÉCEMBRE 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MÉTHODES D'INSPECTION AUTOMATIQUE D'INFRASTRUCTURE PAR ROBOT  
MOBILE

présenté par : NGUYEN Andre Phu-Van  
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées  
a été dûment accepté par le jury d'examen constitué de :

M. NOM Prénom, Doct., président

M. NOM Prénom, Ph. D., membre et directeur de recherche

M. NOM Prénom, Ph. D., membre et codirecteur de recherche

M. NOM Prénom, Ph. D., membre

## DÉDICACE

*Je dédie ce travail à la mémoire de Dopey,  
ami, frère, chien.*

## REMERCIEMENTS

Je remercie beaucoup de gens.

## RÉSUMÉ

Le résumé est un bref exposé du sujet traité, des objectifs visés, des hypothèses émises, des méthodes expérimentales utilisées et de l'analyse des résultats obtenus. On y présente également les principales conclusions de la recherche ainsi que ses applications éventuelles. En général, un résumé ne dépasse pas quatre pages.

Le résumé doit donner une idée exacte du contenu du mémoire ou de la thèse. Ce ne peut pas être une simple énumération des parties du document, car il doit faire ressortir l'originalité de la recherche, son aspect créatif et sa contribution au développement de la technologie ou à l'avancement des connaissances en génie et en sciences appliquées. Un résumé ne doit jamais comporter de références ou de figures.

## **ABSTRACT**

Written in English, the abstract is a brief summary similar to the previous section (Résumé). However, this section is not a word for word translation of the French.

## TABLE DES MATIÈRES

|  |      |
|--|------|
| DÉDICACE . . . . .   | iv   |
| REMERCIEMENTS . . . . .  | v    |
| RÉSUMÉ . . . . .   | vi   |
| ABSTRACT . . . . .   | vii  |
| TABLE DES MATIÈRES . . . . .                                       | viii |
| LISTE DES TABLEAUX . . . . .                                       | x    |
| LISTE DES FIGURES . . . . .  | xi   |
| LISTE DES SIGLES ET ABRÉVIATIONS . . . . .                         | xii  |
| LISTE DES ANNEXES . . . . .  | xiv  |
| <br>CHAPITRE 1 INTRODUCTION . . . . .                              | 1    |
| 1.1 Objectif de recherche . . . . .                                | 1    |
| 1.1.1 Inspections au sol . . . . .                                 | 1    |
| 1.1.2 Inspections aériennes . . . . .                              | 2    |
| 1.2 Plan du mémoire . . . . .                                      | 3    |
| <br>CHAPITRE 2 REVUE DE LITTÉRATURE . . . . .                      | 4    |
| 2.1 Méthodes de positionnement . . . . .                           | 4    |
| 2.2 <i>Simultaneous Localization and Mapping</i> . . . . .         | 5    |
| 2.3 Génération de trajectoires et couverture de surfaces . . . . . | 5    |
| 2.4 Méthodes de navigation . . . . .                               | 7    |
| 2.5 Méthodes automatiques d'inspection d'éoliennes . . . . .       | 8    |
| 2.6 Traitement d'images . . . . .                                  | 8    |
| 2.6.1 Modèle de caméra . . . . .                                   | 8    |
| 2.6.2 Vision stereo . . . . .                                      | 11   |
| 2.6.3 Détection de lignes . . . . .                                | 11   |
| <br>CHAPITRE 3 Choix de capteurs . . . . .                         | 12   |

|  |   |    |
|--|---|----|
| 3.1  | Systèmes passifs . . . . .                                | 12 |
| 3.2  | Systèmes actifs . . . . .                                 | 12 |
| 3.2.1  | Time-of-flight . . . . .                                  | 12 |
| 3.2.2  | Projection laser . . . . .                                | 12 |
| 3.2.3  | subsubsection name . . . . .                              | 12 |
| CHAPITRE 4 Inspection d'infrastructure par UGV . . . . . |   | 13 |
| 4.1  | Description du problème . . . . .                         | 13 |
| 4.2  | Exploration du périmètre . . . . .                        | 14 |
| 4.3  | Exploration des cavités . . . . .                         | 14 |
| 4.4  | Résultats . . . . .                                       | 14 |
| 4.5  | Conclusion . . . . .                                      | 14 |
| CHAPITRE 5 Inspection d'éoliennes par UAV . . . . .      |   | 15 |
| 5.1  | Description du problème et méthode d'inspection . . . . . | 15 |
| 5.2  | Approche du rotor . . . . .                               | 18 |
| 5.2.1  | Approche de loin . . . . .                                | 18 |
| 5.2.2  | Approche en remontant la tour . . . . .                   | 23 |
| 5.3  | Suivit des pales . . . . .                                | 26 |
| 5.3.1  | Suivit par nuage de points . . . . .                      | 27 |
| 5.3.2  | Suivit par scanner laser 2D . . . . .                     | 27 |
| 5.4  | Implémentation . . . . .                                  | 29 |
| 5.4.1  | Résultats en simulation . . . . .                         | 30 |
| 5.4.2  | Résultats de tests sur le terrain . . . . .               | 36 |
| 5.5  | Discussion des résultats et travaux futurs . . . . .      | 39 |
| CHAPITRE 6 CONCLUSION . . . . .                          |   | 41 |
| 6.1  | Synthèse des travaux . . . . .                            | 41 |
| 6.2  | Limitations de la solution proposée . . . . .             | 41 |
| 6.3  | Améliorations futures . . . . .                           | 41 |
| RÉFÉRENCES . . . . .                                     |   | 42 |
| ANNEXES . . . . .  |   | 47 |

**LISTE DES TABLEAUX**

|             |   |    |
|-------------|---|----|
| Tableau 4.1 | Repères présents dans le système d'inspection d'éoliennes . . . . . | 14 |
| Tableau 5.1 | Repères présents dans le système d'inspection d'éoliennes . . . . . | 17 |

## LISTE DES FIGURES

|             |  |    |
|-------------|--|----|
| Figure 2.1  | Exemple de distortion introduite par une lentille à grand angle. (A) l'image originale (B) l'image rectifiée . . . . .   | 10 |
| Figure 5.1  | Systèmes de coordonnées sur le md4-1000 . . . . .  | 16 |
| Figure 5.2  | (A) Image originale (B) Image en tons de gris (C) Image floutée par un filtre moyenner (D) Image avec égalisation d'histogramme . . . . .  | 19 |
| Figure 5.3  | Résultat de l'application du filtre de Canny avec $T_b = 27$ et $T_h = 81$ .   | 20 |
| Figure 5.4  | Résultat d'une détection réussis à la fin de la procédure. . . . .   | 22 |
| Figure 5.5  | Principe d'opération de l'approche du rotor par laser. . . . .   | 24 |
| Figure 5.6  | Résultat de la convolution sur un exemple de mesure. . . . .   | 26 |
| Figure 5.7  | Exemple de suivi de pale par nuage de points. . . . .  | 27 |
| Figure 5.8  | Diagramme de classe des différents contrôleurs utilisés par la machine à états. . . . .  | 30 |
| Figure 5.9  | Véhicule md4-1000 en simulation. Les rayons bleus sont les rayons projetés par les scanner lasers LeddarVu8. . . . .   | 30 |
| Figure 5.10 | Environnements de simulation. . . . .  | 31 |
| Figure 5.11 | Détections réussies en simulation. . . . .   | 32 |
| Figure 5.12 | Échec de la résolution de profondeur par correspondance de blocs. .  | 33 |
| Figure 5.13 | Trajectoire suivie pour la montée de la tour. . . . .  | 33 |
| Figure 5.14 | Trajectoire pour le suivi de la pale 1. . . . .  | 34 |
| Figure 5.15 | Trajectoire complète d'inspection. . . . .   | 35 |
| Figure 5.16 | Le md4-1000 en vol inspectant l'une des pales. . . . .   | 36 |
| Figure 5.17 | Véhicule md4-1000 avec les capteurs et l'ordinateur de bord intégré. (A) Caméra ZED et les télémètres lasers LeddarVu8. (B) La charge utile installé sur le robot (sans la ZED), avec la Nvidia TX2 visible. . | 37 |
| Figure 5.18 | Exemples de succès et d'échecs de perception de distance. À gauche les images RGB de la caméra de gauche et à droite les images de profondeur ramenées sur la caméra de gauche. . . . .                        | 38 |
| Figure 5.19 | Trajectoire alternative pour l'inspection d'une pale à la fois sans besoin de COAS. . . . .  | 40 |

## LISTE DES SIGLES ET ABRÉVIATIONS

|      | English                               | Français                                    |
|------|---------------------------------------|---|
| GIS  | Geographical Information System       | Système d'Information Géographique          |
| GNSS | Global Navigation Satellite System    | Système Mondial de Satellites de Navigation |
| GPS  | Global Positioning System             | Système de Positionnement Global            |
| IMU  | Inertial Measurement Unit             | Centrale Inertielle                         |
| INS  | Inertial Navigation System            | Système de Navigation Inertiel              |
| MVS  | Multi-View Stereo Reconstruction      | Reconstruction Stereo par Vues Multiples    |
| SFM  | Structure From Motion                 | Structure par Mouvement                     |
| SLAM | Simultaneous Localization And Mapping | Locatisation et Cartographie Simultanée     |
| UAV  | Unmanned Aerial Vehicle               | Véhicule Aérien Non-Habité                  |
| UGV  | Unmanned Ground Vehicle               | Véhicule Terrestre Non-Habité               |

## Liste des publications

- [Journal] A. Borowczyk, D.-T. Nguyen, **A. Phu-Van Nguyen**, D. Q. Nguyen, D. Saussié and J. Le Ny, *Autonomous Landing of a Quadcopter on a High-Speed Ground Vehicle*. AIAA Journal of Guidance, Control and Dynamics, In Press, 2017.
- [Conférence] A. Borowczyk, D.-T. Nguyen, **A. Phu-Van Nguyen**, D. Q. Nguyen, D. Saussié and J. Le Ny, *Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle*. Proceedings of the IFAC World Congress, Toulouse, France, July 2017.
- [Accepté] M. S. Ramanagopal, **A. P.-V. Nguyen** and J. Le Ny, A Motion Planning Strategy for the Active Vision-Based Mapping of Ground-Level Structures. Accepted by Transactions on Automation Science and Engineering, February 2016.

Le travail des deux premières publications à propos de l'atterrissement d'un quadrirotor sur une voiture en mouvement a été réalisé dans le cadre de la participation du *Mobile Robotics and Autonomous Systems Laboratory* au DJI Challenge lors des sessions d'hiver et d'été 2016. N'étant pas directement liées au mémoire présent, elles sont mentionnées ici car plusieurs des méthodes de contrôle de véhicule aérien et de traitement d'images ont été apprises lors de notre participation à cette compétition.

La dernière publication intitulée *A Motion Planning Strategy for the Active Vision-Based Mapping of Ground-Level Structures* avait initialement été soumise en février 2016 par M. S. Ramanagopal et J. Le Ny au journal *Transactions on Automation Science and Engineering*. Lors du retour de l'évaluation par les pairs, il a été demandé d'ajouter un volet expérimental à l'article à fin de prouver la validité des algorithmes développés dans l'article. C'est à ce moment, lors de la session d'automne 2016 et au début de la session d'hiver 2017 que nous avons ajouté au projet pour faire l'implémentation sur un robot physique. Au moment de l'écriture de la présente, l'article a été accepté pour publication.

**LISTE DES ANNEXES**

|          |   |    |
|----------|---|----|
| Annexe A | Machine à états pour l'inspection d'éoliennes . . . . . | 47 |
|----------|---|----|

## CHAPITRE 1 INTRODUCTION

Le domaine de la robotique s'applique à ce qui est communément appelé les trois *D* de la robotique *Dull*, *Dirty* et *Dangerous*, c'est-à-dire les tâches ennuyantes, sales et dangereuses. L'inspection d'infrastructure civile rentre dans la première et la dernières de celles-cis. Ennuyantes car elle implique la cueillette répétitive de données et dangereuse car elle implique parfois l'obligation de grimper dans des structures hors de portée ou difficiles à atteindre.

En général, le but de ses inspections est multiple, outre la recherche de défauts dans l'infrastructure, elles peuvent aussi servir à suivre de près l'évolution d'un chantier de construction, faire de l'arpentage ou planifier de futurs projets de développement. L'intérêt de faire exécuter l'inspection par un robot autonome est d'une part d'accélérer la collecte de données par un robot plus mobile qu'un humain et d'autre part de réduire les coûts liés à la collecte d'information.

### 1.1 Objectif de recherche

L'objectif principal de la recherche est de développer des méthodes d'inspection d'infrastructure adaptées aux robots terrestres non-habité (UGV) ainsi qu'aux robots aériens non-habité (UAV). Alors que la navigation du robot se fait entièrement de façon autonome, il est attendu qu'un opérateur intervienne seulement lors de la phase d'analyse des données et dans le cas du UAV qu'il intervienne dans la prise de photos.

#### 1.1.1 Inspections au sol

Dans un premier lieu, nous nous attardons au problème de la création de cartes 3D précises de structures au sol. Ces cartes peuvent être utilisées pour une variété d'applications tel que la réalité virtuelle (Google), la navigation de véhicules autonomes (DeepMap) et la surveillance du progrès d'un chantier de construction (Omar and Nehdi, 2018). Cette dernière application, étant bien établie en industrie, se fait habituellement par l'une de trois méthodes : la photogrammétrie ou vidéogrammétrie, des scanner lasers et des images de profondeur. Chacune d'entre elles demandent qu'un opérateur humain parcourt le chantier en cherchant les meilleurs points de vues avec les capteurs appropriés en main pour la collecte de données. En particulier, dû à la relativement courte portée de capteurs d'images de profondeur, elles demandent aux opérateurs de parcourir une plus grande distance et plus de prises de vue.

C'est dans ce contexte que nous proposons un scénario où un UGV fait l'inspection de

la partie visible d'une structure au sol, sans aucune information *a priori*, au moyen d'un capteur de profondeur. Pour commencer, nous nous basons sur des algorithmes de navigation et de localisation existants tel que les champs de potentiel, le Simultaneous Localization and Mapping (SLAM) visuel et la fusion d'odométrie de roues avec une centrale inertielle (IMU) pour effectuer le contrôle du véhicule. L'objectif est d'avoir une méthode génération de trajectoire en temps réel qui, d'une part facilite la précision du système SLAM et d'autre part garantie la couverture entière de la surface de l'édifice.

### 1.1.2 Inspections aériennes

Dans un second lieu, nous considérons le cas où un UAV fait l'inspection de la surface d'une éolienne. Cette partie du projet, réalisé conjointement avec la compagnie Microdrones Canada, s'inscrit dans le cadre d'un plus large projet de développement qui vise à développer une solution complète d'inspection d'éoliennes par UAV autonome permettant de réduire le temps nécessaire à une inspection. Tout d'abord, une inspection sert à repérer des défauts dans la structure d'une éolienne. Laissés seuls, ces défauts peuvent se traduirent en pertent d'efficacité et dans le pire cas, en un bris fatal.

En temps normal, un inspection implique qu'un employé fasse tourner les pales de l'éolienne pour en positionner une vers le bas. L'employé grimpe la tour pour ensuite faire une descente en rappel le long de la pale pour inspecter visuellement et tactilement la surface de la pale. Ce processus devant être répété pour chaque pale, il peut prendre entre 2 et 4 heures dépendamment de la quantité de défauts à documenter. Plus récemment, nous notons un immense essort dans le nombre de compagnies offrant des services d'inspection par UAV qui ne requièrent pas l'ascention de la tour. Étant plus rapide, plus sécuritaire et moins dispendieuse, cette méthode est rapidement adoptée par l'industrie de l'énergie. Un rapport publié en 2015 par la firme d'étude de marché Navigant Research prédit que le revenu global lié aux ventes d'UAV et de services d'inspection atteindra près de 6 milliards USD en 2024 (Navigant Research, 2015).

Le problème présent des ces systèmes est qu'ils demandent habituellement une équipe d'au moins deux personnes dont un pilote habile et un opérateur de caméra pour fonctionner. Le but général du projet est de rendre l'utilisation de l'UAV plus accessible à fin de retirer le besoin d'un pilote et réduire la taille de l'équipe à une seule personne. Il faut noter que le but n'est pas encore de complètement remplacer une inspection manuelle par un humain. En effet, puisque la peinture reposant sur la surface d'une pale est moins flexible que le composite de la structure qui se retrouve en dessous, un défaut dans la structure implique une craque dans la peinture. L'inverse n'étant pas nécessairement vrai, une inspection visuelle par UAV

devra parfois demander une deuxième passe de près par un humain pour confirmer l'existence du défaut.

En somme, le but final du projet est de produire un système autonome plus facile d'utilisation pour l'inspection d'éoliennes, pouvant être déployé par un seul opérateur.

## 1.2 Plan du mémoire

Avant tout, nous commençons par une revue de littérature dans le Chapitre 2. Débutant par les bases d'un véhicule autonome, nous survolons les méthodes d'estimation d'état tant au niveau de l'estimation de l'attitude de notre véhicule qu'au niveau de la position de notre véhicule par rapport à un référentiel quelconque près de la structure à inspecter. Ensuite nous examinons le travail existant relatif à la planification de trajectoire et de couverture de surface. Ces ouvrages nous permettent de mieux situer la contribution exacte du volet UGV de notre projet, dans le corps de travail existant. Puisqu'il doit y avoir un moyen pour nos véhicules d'exécuter les dites trajectoires, nous survolons rapidement certaines méthodes de navigation pour le suivi de trajectoires et le contrôle des véhicules. Finalement, nous terminons la revue par une vue d'ensemble sur les méthodes de cartographie pour la reconstruction 3D.

Par la suite, nous présentons l'essentiel de notre travail en trois volets. Dans le Chapitre 3 nous faisons l'étude des capteurs existants sur le marché avec une comparaison des avantages et des désavantages de ceux-ci par rapport à l'inspection d'infrastructure. Au Chapitre 4 nous présentons notre travail réalisé dans le cadre de nos recherches dans l'inspection d'infrastructure par UGV qui s'est culminé en un article soumis en mars 2016 au journal *Transactions on Automation Science and Engineering*. En dernier lieu, au Chapitre 5 nous présentons notre travail réalisé lors du projet Mitacs Accelerate IT08812 en collaboration avec la compagnie Microdrones à propos de l'inspection d'éoliennes par drone autonome.

## CHAPITRE 2 REVUE DE LITTÉRATURE

Bien que les deux types d'inspections traités dans cet ouvrage sont exécutées sur des véhicules entièrement différents, les concepts utilisées lors de ces opérations se recoupent énormément. Dans ce chapitre nous présentons d'abord dans la section 2.1 diverses méthodes de positionnement couramment utilisées par les robots mobiles pour naviguer leur environnement. Dans le même ordre d'idées nous présentons ensuite dans la section 2.2 diverses façons pour un robot de résoudre le problème de *Simultaneous Localization and Mapping* (SLAM) pour calculer la position relative du robot dans son environnement. La section 2.3 présente l'état de l'art en termes de génération de trajectoires pour la couverture de surfaces et de volumes. Puis la section 2.4 présente quelques algorithmes couramment utilisés pour la navigation de robots mobiles. Finalement dans la section 2.5 nous abordons le sujet spécifique de l'inspection d'éoliennes au moyen de robots autonomes.

### 2.1 Méthodes de positionnement

Selon (Borenstein et al., 1997) les systèmes de positionnement se regroupent dans l'une de deux catégories :

1. Les systèmes à l'estime fonctionnant par l'intégration d'une mesure à travers le temps tel que l'odométrie provenant des roues d'un robot ou la double intégrale d'une accélération pour estimer une position.
2. Les systèmes absolus fonctionnant grâce à de l'information externe tel que les GNSS reposant sur les données de satellites.

Les systèmes de navigation complets vont habituellement se doter d'une combinaison des deux types de capteurs pour obtenir une estimation de la pose (position et orientation) du robot dans l'espace. Les systèmes de positionnement absolus tel que le GPS ayant normalement des fréquences de mise à jour lentes, autour de 1 Hz, ils sont souvent jumelés à une centrale inertie (IMU), c'est-à-dire un combiné de gyroscopes et d'accéléromètres permettant de mesurer la vitesse angulaire et l'accélération d'un corps rigide (Noureddin et al., 2013). Bien que les capteurs inertielles permettent d'estimer la position à de hautes fréquences, parfois même 1 kHz, en prenant l'intégrale de ceux-ci, une accumulation d'erreur peut rapidement faire diverger l'estimation d'état. Ainsi, un système inertiel à l'estime et un système GPS absolu jouent des rôles complémentaires où le système inertiel calcule la position à haute fréquence puis une correction est faite par GPS à basse fréquence.

On appelle tout système reposant sur des capteurs inertIELS un Système de Navigation Inertiel (INS). De nos jours, ceux-ci sont faciles d'accès et peuvent être achetés pour quelques milliers de dollars. La fusion de ces différents capteurs peut se faire au travers de différents moyens tels que les filtres à particules (Carvalho et al., 1997) ou les graphes de facteurs (Indelman et al., 2012) mais l'outil normalement utilisé est le filtre de Kalman (Noureddin et al., 2013).

De base, le filtre de Kalman (KF) est un algorithme récursif permettant de faire une estimation aux moindres carrés d'un vecteur d'états au travers d'une procédure de prédition de l'évolution du vecteur d'état suivant un modèle du système suivi de la correction de celui-ci par les mesures entrantes au filtre. Cet algorithme fonctionne pourvu que la transformation entre l'état et les mesures en entrées soit linéaire. Dans le cas d'un système non-linéaire une linéarisation par un développement de Taylor autour de l'estimé optimal courant du vecteur d'état est effectuée résultant en un filtre de Kalman étendu (EKF) (Chui and Chen, 2017).

Les GNSS n'étant pas la seule façon de corriger l'erreur cumulative d'un INS, certains auteurs proposent des filtres de Kalman modulaires permettant de fusionner une quantité arbitraire de capteurs hétérogènes tels que des IMU, l'odométrie d'encodeurs de roues, l'odométrie visuelle, receveurs GNSS, etc. (Moore and Stouch, 2014) pour estimer la position et l'orientation du robot. Pour le cas spécifique des UAV (Lynen et al., 2013) proposent aussi une architecture modulaire pour un filtre de Kalman étendu itéré qui permet de faire une calibration inter-capteurs en-ligne pour estimer les délais temporels et les différences en échelle des mesures, chose importante dans le cas d'odométrie visuel monoculaire où l'échelle des mesures est arbitraire et peut dériver avec le temps.

## **2.2 *Simultaneous Localization and Mapping***

En

## **2.3 Génération de trajectoires et couverture de surfaces**

En général, le but des générateurs de trajectoire est d'optimiser une certaine métrique permettant de décider à quel endroit il faut placer le véhicule et son capteur pour faire l'inspection de la structure.

Tout d'abord, considérant le cas d'une mission sans information *a priori*. Il s'agit d'une mission d'exploration où la génération de trajectoire doit se faire itérativement en temps réel pour envoyer le robot aux confins de l'espace connu. On tente en fait de répondre à la

question :

Given what you know about the world, where should you move to gain as much new information as possible? (Sachant ce que nous savons à propos du monde, où devrions nous aller pour gagner le plus d'information possible?) (Yamauchi, 1997).

Pour ce faire Yamauchi introduit le concept de *frontier-based exploration* qui cherche à guider un robot vers la frontière entre l'espace connu et libre et l'espace inconnu. Suivant cette idée plusieurs chercheurs proposent des fonctions de coût à minimiser permettant de choisir à quel endroit de la frontière explorer. (Wirth and Pellenz, 2007) proposent de subdiviser une carte 2D en cellules pour lesquelles une fonction de coût est calculée en prenant en compte non seulement la plus courte distance par rapport au robot mais aussi la distance par rapport à l'obstacle le plus proche. Chaque nouvelle position objectif minimise ainsi la distance pour s'y rendre mais aussi le risque encouru par le robot. Au lieu d'utiliser les *frontier* comme objectifs (Dornhege and Kleiner, 2011) proposent une formulation du problème utilisant les *frontier* en tant que candidats possibles d'ouvertures dans les murs qui permettraient à une caméra de cartographier l'espace vide se retrouvant derrière.

(Bircher et al., 2016) proposent une méthode d'exploration 3D basé sur l'algorithme *Rapidly exploring Random Trees* (RRT) où un arbre est grandit dans l'espace ouvert connu. À chaque sommet, l'angle de vue de la caméra est utilisé pour estimer le gain exploratoire de la position. Une fois l'arbre construit, le véhicule exécute la première étape de la branche possédant le plus grand gain total. L'arbre complet est ensuite recalculé en utilisant la branche choisie comme point de départ. En d'autres mots, leur méthode tente de maximiser le gain exploratoire en prenant aussi en compte les gains futurs d'une trajectoire. Outre les gains exploratoires, certains auteurs tentent aussi de prendre en compte l'effet des trajectoires sur leurs systèmes de navigation. Par exemple (Papachristos et al., 2017) améliorent la méthode de Bircher en sélectionnant un trajectoire qui permet aussi de minimiser l'incertitude sur leur système de cartographie et d'odométrie visuelle. (Wirth and Pellenz, 2007) notent d'ailleurs que la proximité d'un obstacle est un danger (de collision) mais l'éloignement des obstacle en est aussi puisque la portée limitée des capteurs pourrait rendre un système de SLAM temporairement aveugle.

Alors que les méthodes précédentes se préoccupent de maximiser le gain d'information, elles ne prennent pas en compte les contraintes de temps liées à l'autonomie des robots ; Un problème qui affecte grandement les véhicules aériens multi-rotors. En combinant une fonction d'entropie calculée dans un voisinage local avec le coût en distance d'une trajectoire, (Wang et al., 2017) proposent une méthode d'exploration se basant sur les *Information Potential*

*Fields* (les champs de potentiels d'information). Similaire à la méthode des champs de potentiels pour l'évitement d'obstacles, le robot fini par être attiré aux régions les plus proches à haut gain d'information. Toutefois, minimiser la longueur de la trajectoire n'assure pas nécessairement une minimisation du temps d'exploration si nous prenons aussi en compte l'accélération ou le jerk requis pour exécuter celle-ci. Pour résoudre ce problème (Cieslewski et al., 2017) proposent une extension de la méthode de (Yamauchi, 1997) où le prochain *frontier* choisi est sélectionné dans le champ de vision du véhicule et de tel sorte qu'il minimise le changement de vitesse requis au robot. Ainsi, la trajectoire exécutée peut être plus longue que les méthodes conventionnelles mais elle permet de maintenir une vitesse de navigation plus élevée.

Dans un autre ordres d'idée, une trajectoire complète et globalement optimale peut être générée au préalable si de l'information *a priori* est disponible. Pour une mission où la géométrie de la structure à inspecter est connue (Bircher et al., 2015) proposent de résoudre le problème en deux temps. En subdivisant le modèle en un maillage de triangles, on obtient pour chaque face un ensemble de points de vues admissibles. À chaque itération, des point de vus sont choisis séquentiellement en résolvant un problème d'optimisation QP où la fonction de coût minimise la distance par rapport aux points de vue voisin et où les contraintes assurent que la surface soit visible par le véhicule. Une fois l'ensemble choisi, une recherche heuristique est effectuée pour résoudre un problème du commis voyageur à travers l'ensemble. Bircher réexécute les deux étapes jusqu'à ce qu'une trajectoire satisfaisante soit trouvée.

Comparativement à l'inspection par exploration (en temps réel) la génération de trajectoire hors-ligne permet de créer des plans plus optimals en termes de temps et de distance à parcourir. Par contre, cette trajectoire peut avoir un risque de collision s'il s'avère que le modèle utilisé diffère de la structure réelle, par exemple dans la cas d'un édifice partiellement endommagé par un désastre naturel ou une variation dans l'angle des pales d'une éolienne. (Hepp et al., 2017) présentent un système faisant une première passe à haute altitude pour construire une carte rudimentaire dans laquelle la seconde inspection de près est planifiée suivant une maximization du gain d'information.

## 2.4 Méthodes de navigation

Champs de potentiel

## 2.5 Méthodes automatiques d'inspection d'éoliennes

L'inspection d'éoliennes par UAV autonome étant une application relativement nouvelle, peu de travail a été publié à ce sujet particulier et à ce jour, la majorité du travail demeure théorique, en simulation seulement ou en environnements à petite échelle. Avant tout, (Zhang and Jackman, 2014) démontrent la faisabilité de repérer automatiquement des craques sur la surface d'une éolienne au moyen des détecteurs de bordures bien connus Canny et Sobel. Du travail préliminaire a été réalisé sur les éléments de base requis pour faire une inspection autonome. Notamment au niveau de l'utilisation du principe du flux optique pour l'estimation de la vitesse relative entre un véhicule aérien et une éolienne (Høglund, 2014) ainsi que l'utilisation de la transformée de Hough linéaire et de traitements géométriques pour la reconnaissance automatique d'éoliennes (Stokkeland et al., 2015). Stokkeland explique qu'une inspection entièrement visuelle est difficile à réaliser dû à la grande quantité de bruit et de sources d'erreurs présentes sur le terrain. Par exemple, la segmentation entre la surface blanche de l'éolienne et les nuages en arrière plan est extrêmement sensible au réglage des paramètres du filtre de couleur. C'est pourquoi (Heggem, 2017) fait usage de vision active au moyen d'un projecteur laser et de caméras stereo pour détecter la distance et la forme de la pale. Avec une matrice de seulement 11x11 points, Heggem parvient ensuite à calculer dans quelle direction son UAV doit se diriger pour poursuivre son inspection.

Du côté commercial, peu de compagnies se sont aventurées dans le domaine<sup>1</sup> mais tous semblent faire usage de radars lasers et de caméras sous une forme ou une autre.

## 2.6 Traitement d'images

### 2.6.1 Modèle de caméra

Il arrive souvent dans des applications de traitement d'image de devoir passer du domaine 3D à 2D et vice-versa. Pour ce faire, il est important de passer en revue les modèles de projection et de distortion d'images couramment utilisés. Tout point 3D  $p_w$  à coordonnées connues peut être projeté sur le plan image à un point en coordonnées en pixels  $\mathbf{x}_s$  par l'équation 2.1.

$$\mathbf{x}_s = K[R|t]p_w = Pp_w \quad (2.1)$$

On nomme la matrice  $3 \times 4 P$  la matrice de caméra, la matrice  $3 \times 3 K$  la matrice des paramètres intrinsèques,  $R$  est une rotation orthogonale et  $t$  est une translation. Dépendamment

---

1. Nous notons la présence de SkySpecs aux États-Unis, Pro-Drones en Italie et Perceptual Robotics en Angleterre.

des auteurs  $K$  peut être posé différemment, l'équation 2.2 présente sa forme générale où  $f_{[x,y]}$  sont la distance focale en pixels,  $(c_x, c_y)$  le centre optique en pixels,  $s$  le *skew* prenant en compte l'angle possible entre les axes du capteur et les axes optiques et finalement  $\alpha$  est le facteur de forme.

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & \alpha f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

En pratique la majorité des application tel quel la librairie de traitement d'images OpenCV vont simplifier  $K$  au moyen de  $s = 0$  et  $\alpha = 1$  (Itseez, 2017). Il est aussi utile de passer aux coordonnées homogènes au moyen d'une matrice  $\tilde{\mathbf{P}}$   $4 \times 4$  en ajoutant une ligne à  $P$ ,

$$\tilde{\mathbf{P}} = \begin{bmatrix} K & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} R & t \\ \mathbf{0} & 1 \end{bmatrix} = \tilde{\mathbf{K}}\mathbf{E} \quad (2.3)$$

où  $\mathbf{E}$  est une matrice de transformée 3D. La projection se fait de la même façon qu'à l'équation 2.1 avec  $\bar{\mathbf{p}}_w = [p_w^\top \ 0]^\top$  et  $\mathbf{x}_s = [x_s \ y_s \ 1 \ d]^\top$

$$\mathbf{x}_s \sim \tilde{\mathbf{P}}\bar{\mathbf{p}}_w \quad (2.4)$$

où le symbole  $\sim$  indique une égalité à l'échelle et  $\mathbf{x}_s$  doit être normalisé après la multiplication pour que la troisième composante soit 1 (Szeliski, 2011).

Ces formules sont valides dans le cas de projections entièrement linéaires, par contre les systèmes optiques vont habituellement introduire une certaine distortion de l'image. Par exemple dans la Figure les lentilles de la caméra Zed avec un angle de vue de 110° (D) introduisent beaucoup de distortion courbant ainsi des lignes qui devraient normalement être droites.

Plusieurs modèles de distortion ont été proposés à travers le temps mais le plus répandu est celui radial-tangential, radial puisqu'il dépend de la distance du pixel par rapport au centre optique et tangentiel pour le déplacement des rayons tangentiel au cercle autour du centre optique. La distortion radiale-tangentielle peut-être exprimé par un polynôme permettant de mettre en correspondance les pixels de l'image originale  $(x_c, y_c)$  en coordonnées normalisées aux pixels de l'image rectifiée  $(u, v)$ .



Figure 2.1 Exemple de distortion introduite par une lentille à grand angle. (A) l'image originale (B) l'image rectifiée

$$\begin{aligned}\hat{x}_c &= x_c \left( \frac{1 + \kappa_1 r_c^2 + \kappa_2 r_c^4 + \kappa_3 r_c^6}{1 + \kappa_4 r_c^2 + \kappa_5 r_c^4 + \kappa_6 r_c^6} \right) + 2p_1 x_c y_c + p_2(r_c^2 + 2x_c^2) \\ \hat{y}_c &= y_c \left( \frac{1 + \kappa_1 r_c^2 + \kappa_2 r_c^4 + \kappa_3 r_c^6}{1 + \kappa_4 r_c^2 + \kappa_5 r_c^4 + \kappa_6 r_c^6} \right) + p_1(r_c^2 + 2y_c^2) + 2p_2 x_c y_c \\ r_c^2 &= x_c^2 + y_c^2\end{aligned}\tag{2.5}$$

$$\begin{aligned}u &= f_x \hat{x}_c + c_x \\ v &= f_y \hat{y}_c + c_y\end{aligned}\tag{2.6}$$

Les paramètres  $\kappa_{[1-6]}$  sont les coefficients de distortion radial et  $p_{[1,2]}$  les coefficients de distortion tangentiel. L'équation 2.6 permet de mettre à l'échelle les coordonnées normalisées en coordonnées en pixels. Les paramètres de distortion et intrinsèques peuvent être estimés conjointement suivant la résolution d'un problème des moindres carrés non-linéaire (Zhang, 2000).

Bien que le modèle radial-tangentiel soit le plus populaire, entre autres par sa présence dans la librairie de traitement d'images OpenCV, plusieurs auteurs proposent de nouveaux modèles avec divers avantages. Par exemple le modèle de distortion équidistant proposé par (Kannala and Brandt, 2006) est apte à modéliser la distortion à la fois de lentilles normales et de lentilles ultra-grand angle et le modèle par fonction rationnelle de (Claus and Fitzgibbon, 2005) adapté aux lentilles grand-angle et catadioptriques.

### 2.6.2 Vision stereo

Ajouter la théorie sur la géométrie épipolaire pour la résolution de profondeur par caméra stereo.

### 2.6.3 Détection de lignes

1. canny
2. HED
3. Transformée de hough
4. edlines
5. lsd

## CHAPITRE 3 Choix de capteurs

Intégrer ce chapitre à la revue de littérature ? Ou faire un chapitre très court.

3.1 Systèmes passifs

3.2 Systèmes actifs

3.2.1 Time-of-flight

3.2.2 Projection laser

3.2.3 subsubsection name

## CHAPITRE 4 Inspection d'infrastructure par UGV

Dans cette section nous présentons notre travail réalisé dans le cadre de l'article *Motion Planning Strategy for the Active Vision-Based Mapping of Ground-Level Structures*.

### 4.1 Description du problème

Une composante importante de tout système de SLAM est sa capacité de fermeture de boucle, c'est-à-dire d'être capable de reconnaître quand le robot retourne à un endroit déjà visité, particulièrement en l'absence de systèmes de positionnement absolus. Ceci peut être fait localement sur un sous-ensemble local des observations courantes ou globalement sur toutes les observations faites jusqu'au présent. Ceci peut être réalisé d'une variété de façons dépendamment des capteurs utilisés par l'algorithme de SLAM. (Hess et al., 2016) proposent Google Cartographer une approche hybride fonctionnant par scanner laser, où les nouveaux scans sont insérés et appariés par rapport à une sous-carte locale alors que la recherche de fermeture de boucle globale est exécutée en arrière plan par un algorithme de séparation et évaluation progressive (*branch-and-bound*). En revanche RTAB-MAP (Labbé and Michaud, 2014) utilise une approche par sac-de-mots (*bag-of-words*) appliquée à des descripteurs extraits d'une image couleur. Pour assurer une opération en temps réel les noeuds du graphe de pose, contenant aussi les mots visuels extraits à chaque endroit, sont séparés dans différentes mémoires : la mémoire court terme, la mémoire de travail et la mémoire long terme. Seuls les noeuds dans la mémoire de travail sont considérés pour la fermeture de boucle. Dans les deux cas de Google Cartographer et RTAB-MAP il est donc possible que la détection de la fermeture de boucle échoue si la pose estimée courante du robot a trop dérivé au point que les algorithmes ne cherchent plus à apparier l'environnement courant avec les éléments de la carte connue.

Le but de la fermeture de boucle est de réduire les erreurs de localisation en imposant certaines contraintes sur la carte en cours de construction. Cette minimisation d'erreure peut-être réalisée de plusieurs façons, notamment par optimisation de graphe de poses (Carlone et al., 2016), par compensation par faisceaux (*Bundle Adjustment*) (Mei et al., 2011) ou tel que proposé dans ORB-SLAM2 une combinaison des deux (Mur-Artal and Tardós, 2017).

C'est dans ce contexte que nous proposons une méthode de planification de trajectoire pour l'inspection de structures au sol cherchant à fermer la boucle le plus tôt possible pour d'une part augmenter les chances de succès de la détection de la fermeture de boucle et d'autre

part minimiser les erreurs de localisation. Plus spécifiquement étant donné une structure de taille finie et un robot équipé d'un capteur de profondeur mobile, nous cherchons à optimiser le positionnement de ce capteur

| Symbol | Nom           | Description  |
|--------|---------------|--|
| G      | <i>Global</i> | Origine au point de départ du robot  |
| R      | <i>Robot</i>  | Repère centré sur le robot avec les axes avant-gauche-haut.  |
| C      | <i>Camera</i> | Repère centré sur la caméra, la transformée est rigide par rapport à R sauf pour l'angle de lacet. |

Tableau 4.1 Repères présents dans le système d'inspection d'éoliennes

## 4.2 Exploration du périmètre

## 4.3 Exploration des cavités

## 4.4 Résultats

## 4.5 Conclusion

## CHAPITRE 5 Inspection d'éoliennes par UAV

Dans ce chapitre nous traitons du projet de l'automatisation d'inspection d'éoliennes par un véhicule aérien autonome réalisé conjointement avec la compagnie Microdrones Canada Inc.

### 5.1 Description du problème et méthode d'inspection

Les éoliennes doivent régulièrement être inspectées pour détecter et réparer le plus tôt possible des failles structurelles sur leurs pales qui peuvent apparaître avec le temps dû aux intempéries ou même quand elles sont frappées par la foudre. La peinture recouvrant celles-ci étant moins élastique que le composite de fibre de verre dont elles sont construites, un défaut structurel se manifeste inévitablement à la surface de la pale sous la forme de craques ou de trous. C'est pourquoi une inspection visuelle doit être régulièrement faite pour déceler et réparer toute irrégularité dans la structure avant qu'un bris catastrophique ne se produise.

En temps normal, une inspection se fait par une équipe d'une ou deux personnes devant grimper l'éolienne. D'abord, l'équipe attend que le vent fasse tourner le rotor jusqu'à ce que la pale devant être examinée pointe vers le sol. À ce moment, le frein est enclenché et l'angle d'inclinaison des pales est ajusté pour que le vent ne crée pas de couple sur le rotor. L'équipe grimpe ensuite la tour jusqu'au sommet, ce qui peut prendre une dizaine de minutes si l'ascenseur est non fonctionnel, chose qui arrive fréquemment. Une fois rendu au rotor, un frein mécanique secondaire est activé pour manuellement bloquer la rotation des pales. L'équipe fait ensuite une descente en rappel le long de la pale pour faire leur inspection visuelle et tactile de la surface. Une fois l'inspection terminée le processus est recommencé pour les deux autres pales. En tout, inspecter une éolienne peut prendre entre deux et quatre heures dépendamment du nombre de bris à documenter.

Certains opérateurs d'UAV offrent maintenant des services d'inspection d'éoliennes. Une caméra haute définition est montée sur un cardan stabilisateur à trois axes qui est à son tour monté sur le véhicule. Ce dernier est desfois aussi augmenté d'un télémètre permettant au pilote de savoir à quelle distance il est de la structure, chose difficile à estimer lorsque le véhicule est à 100 mètres d'altitude. Ces inspections peuvent prendre jusqu'à trois employés dont un pilote, un opérateur pour la caméra et un observateur sous l'éolienne.

Le but du projet est d'explorer les solutions possibles pour réaliser un système capable de naviguer un quadrioptère autour des pales d'une éolienne à l'aide de capteurs peu coûteux avec seulement le pilote de sécurité de présent et le moins d'information *a priori* possible.

Ce projet peut aussi servir de tremplin vers un projet futur dans lequel aucun humain n'interviendrait, par exemple pour une flotte d'UAV chargée d'inspecter régulièrement un parc éolien entier. Puisque manœuvrer précisément autour des pales demanderait l'usage de capteurs dépassant le budget du projet et nécessiterait que l'UAV soit équipé de capteurs de position haute précision tel qu'un GPS différentiel, nous simplifions le problème en un problème à deux dimensions où le véhicule cherche à suivre le bord d'attaque (ou le bord de fuite) des pales. Deux approches ont été développées, l'une entièrement au moyen de scanners lasers et l'autre par caméras stereo.

La mission se divise en deux phases : l'approche du rotor et le suivi de ses pales qui peuvent être gérées par une simple machine à états finis. Pour l'approche du rotor nous comparons une approche à longue distance au moyen de traitement d'images à une approche à tâtons où l'UAV remonte la tour au moyen de scanner lasers. Pour le suivi des pales, nous comparons encore une méthode par traitement d'images à une méthode de suivi à tâtons. Puisque ce projet est réalisé conjointement avec la compagnie Microdrones, le quadrioptère md4-1000 a été choisi sur lequel nous installons deux scanners lasers LeddarTech Vu8, l'un horizontal et l'autre vertical et puis une caméra stereo ZED. Il est à noter que ces capteurs ne servent qu'à la navigation de l'UAV alors que la caméra haute définition sur stabilisateur resterait le moyen par lequel prendre des photos de défauts à la surface des pales.



Figure 5.1 Systèmes de coordonnées sur le md4-1000

| Symbole   | Nom                 | Description  |
|-----------|---------------------|--|
| $W$       | <i>World</i>        | Suivant la convention <i>East-North-Up</i> (ENU), centré au point de décollage du véhicule avec l'axe $x$ pointant vers l'est, $y$ pointant vers le nord et $z$ pointant vers le haut.   |
| $B$       | <i>Body</i>         | Encore suivant la convention ENU, l'axe $x$ pointant vers la droite, $y$ vers l'avant et $z$ vers le haut.   |
| $BT$      | <i>Body Tangent</i> | Repère centré sur $B$ mais compensé pour ses angles de roulis et de tangage (mais pas de lacet). En d'autres mots, le repère $BT$ est donc tangent à la surface de la Terre et centré sur $B$ .  |
| $C[L, R]$ | <i>Camera</i>       | Suivant la convention de représentation d'images, nous avons l'axe $x$ vers la droite, $y$ vers le bas et $z$ sortant de la caméra. Les suffixes $L$ et $R$ indiquent la caméra gauche et droite respectivement. Par souci de brièveté si le repère est écrit $C$ sans préciser la caméra gauche ou droite, il indique la caméra droite. |
| $L[H, V]$ | <i>Laser</i>        | Les lasers aussi suivent un repère main droite avec $x$ vers l'avant, $y$ à gauche et $z$ vers le haut. Les données de distance du laser se situe tous sur son plan $XY$ . Les suffixes $H$ et $V$ indique la direction du scan, horizontal et vertical, par rapport au repère $B$ .   |
| $T$       | <i>Turbine</i>      | Le système de coordonnées de la turbine est centré sur le rotor avec $x$ vers la droite (vu de devant), $y$ vers l'intérieur de la nacelle et $z$ vers le haut.  |

Tableau 5.1 Repères présents dans le système d'inspection d'éoliennes

## 5.2 Approche du rotor

### 5.2.1 Approche de loin

L'approche à longue distance comporte plusieurs avantages dont l'absence du besoin d'information *a priori* à propos du placement des pales. Avec une vue d'ensemble de l'éolienne à inspecter, il devient possible de mesurer l'angle des pales, une information cruciale qui sera exploitée lorsque l'UAV devra choisir où se diriger lorsqu'il est près du rotor. Le deuxième avantage est qu'il est possible de mesurer l'angle de lacet relatif entre l'UAV et l'éolienne, permettant ainsi de placer le véhicule perpendiculairement à la surface des pales. Cette méthode requiert tout de même la présence d'un détecteur de proximité avec une portée d'au moins 10 mètres, par contre ça reste beaucoup moins dispendieux qu'un scanner laser.

Dans cette section nous détaillons une méthode légèrement modifiée de celle de (Stokkeland et al., 2015) pour détecter le centre du rotor d'une éolienne à partir d'une image. En somme les étapes restent les mêmes mais de la robustesse additionnelle est introduite au moyen d'algorithmes de regroupement, une représentation différente des pales détectées est utilisée et un filtre de Kalman non-linéaire (au lieu d'un filtre de Kalman linéaire) est proposé.

La méthode de détection du rotor se divise en 6 étapes et repose majoritairement sur des raisonnements géométriques.

#### Étape 1 : Prétraitement de l'image

Avant de commencer la détection, quelques corrections à l'image doivent être apportées. Puisque la procédure ne requiert pas de données de couleur, une conversion RGB à échelle de gris est effectuée au moyen de la transformée

$$Y \leftarrow 0.299R + 0.587G + 0.114B \quad (5.1)$$

où  $Y$  est le niveau de gris du pixel entre 0 et 255 ce qui donne comme résultat l'image de la Figure 5.2 (B). Pour éliminer la présence de bruit et atténuer les textures à haute fréquence pouvant donner lieu à une sur détection de lignes lors de la transformée de Hough, nous appliquons un filtre moyenneur 3x3 par convolution. Pour l'image floutée  $F$  nous avons donc

$$F = \left( \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) * H \quad (5.2)$$

avec le résultat visible à la Figure 5.2 (C) Pour les convolutions sur les bords de l'image nous répétons la valeur du pixel du bord pour les données manquantes dans l'opération de convolution. Pour augmenter le contraste et corriger les erreurs d'exposition de la caméra, une égalisation d'histogramme est appliquée. Pour l'image originale  $I$  et l'image égalisée  $H$  que l'on peut voir dans la Figure 5.2 (D), nous avons

$$p_n = \frac{\# \text{ de pixels d'intensité } n}{\# \text{ de pixels total}}, \quad n = 0, \dots, 255 \quad (5.3)$$

$$H_{i,j} = \text{floor}(255 \sum_{n=0}^{I_{i,j}} p_n) \quad (5.4)$$

Où floor est l'opérateur arondissant à la baisse à l'entier le plus proche.

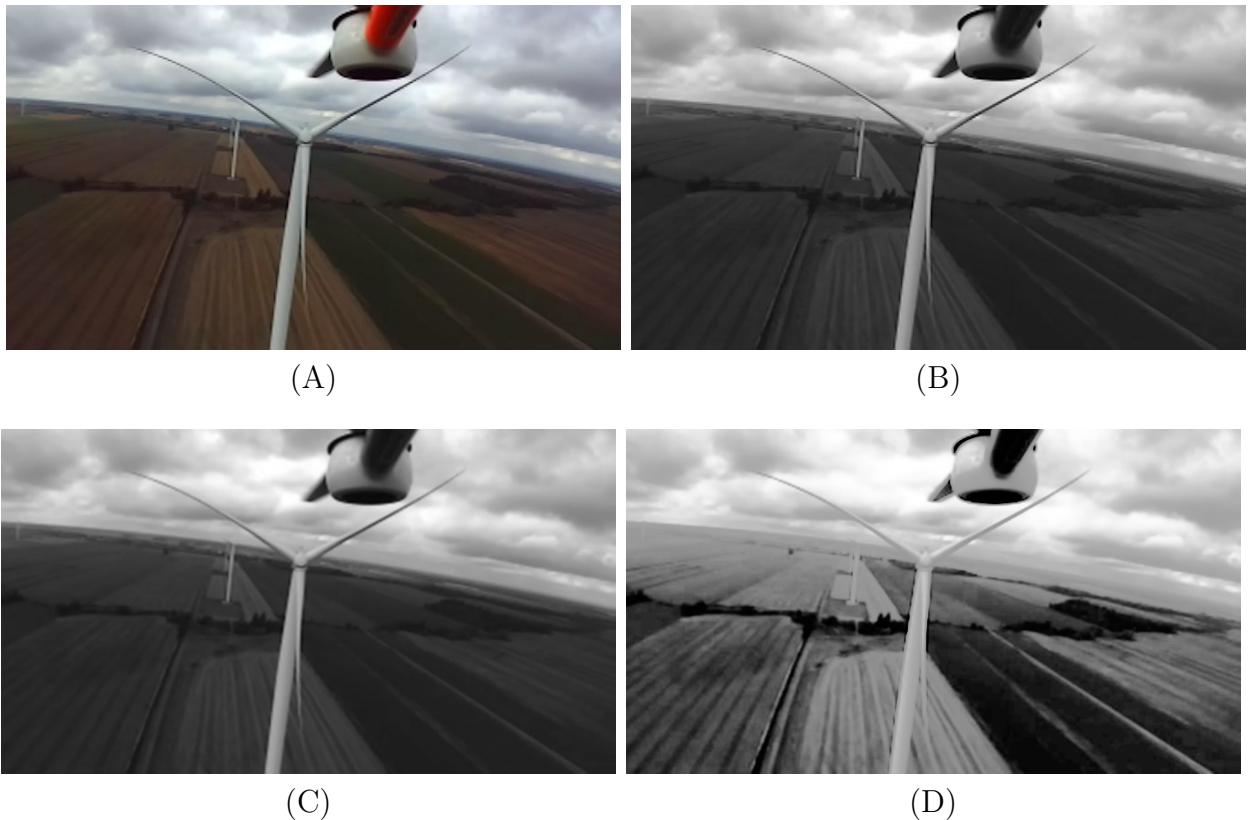


Figure 5.2 (A) Image originale (B) Image en tons de gris (C) Image floutée par un filtre moyenneur (D) Image avec égalisation d'histogramme

## Étape 2 : Détection de contours

La détection de contours se fait au moyen de l'algorithme de (Canny, 1986) et résultat de la procédure est visible dans la Figure 5.3.



Figure 5.3 Résultat de l'application du filtre de Canny avec  $T_b = 27$  et  $T_h = 81$

Outre la méthode de Canny, il existe des méthodes modernes plus performantes que celle de Canny, par exemple ("Xie and Tu, 2015) proposent un réseau de neurones entièrement convolutionnel capable de détecter beaucoup mieux les contours, et ce, sans avoir à manuellement ajuster des paramètres. Par contre dans nos tests, la haute précision de la méthode retournait trop de données et il devenait difficile de passer aux prochaines étapes de la détection.

### **Étape 3 : Détection de lignes**

Une fois les contours trouvés, nous pouvons les utiliser pour détecter des segments de droite dans l'image. Dans le cas où des parties du véhicule sont visibles dans l'image, il suffit d'appliquer un masque avant de débuter cette étape. Pour trouver les segments, nous appliquons la transformée de Hough probabiliste (Mat, 2000) pour détecter des segments de droites dans l'image de contours.

### **Étape 4 : Recherche de la tour**

Stokkeland explique que la partie la plus facile à détecter est la tour puisqu'elle est toujours verticale et répond fortement à la détection de lignes. Pour la trouver, il suffit de chercher une ligne verticale (compensée pour l'angle de roulis de l'UAV) dont l'un des sommets repose dans le base de l'image. Puisqu'il arrive parfois que la détection de ligne défait la tour en plusieurs segments, une fois le segment initial choisi on recherche aussi des prolongements de ce segment à une certaine distance du plus haut sommet.

## Étape 5 : Recherche des pales

À la fin de l'étape 4 nous avons un ensemble de segments appartenant à la tour. À partir du plus haut point de ces segments nous recherchons tout autre segment ayant un sommet dans un certain rayon du point. Ces segments sont considérés en tant que candidats à être des segments d'une pale.

Nous divergeons de la méthode proposée par Stokkeland où il tente de regrouper les segments selon l'angle par rapport à la tour ; Au travers d'une procédure vorace simple où un nouveau groupe est créé lorsqu'un segment est à un angle au dessus d'un certain seuil du groupe courant. Nous notons que ceci fonctionne acceptablement sur le jeu de données à Stokkeland où le ciel est clair et son éolienne est sur une colline près de la mer, mettant donc l'horizon bien en dessous du rotor. Lors de nos essais tant en simulation que sur le terrain, nous avons noté que l'horizon était très proche de rotor donnant lieu à une grande présence de segments près du rotor faussant ainsi les résultats de la recherche de pales. C'est pourquoi nous implémentons l'étape 5 avec un algorithme de groupage formel qui inclut la résistance au bruit de mesure nommé *Density-based spatial clustering of applications with noise* (DBSCAN) proposé par (Ester et al., 1996).

La particularité de DBSCAN par rapport à d'autres algorithmes de partitionnage tel que *k*-means est que nous n'avons pas à fournir le nombre *k* de partitions recherchées. À la place, nous fournissons que deux paramètres  $\epsilon$  (eps) la distance d'un point à son entourage et MinPts le nombre minimum de points pour qu'une partition soit formée. Pour chaque segment, nous projettons leur angle par rapport à la tour sur un cercle unitaire. Ceci permet d'exécuter DBSCAN dans un espace euclidien pour regrouper les segments adjacents.

Une fois les groupes formés, Stokkeland utilise une stratégie de vote pour éliminer les fausses détections. La moyenne des angles est calculée et une procédure de vote débute s'il y a plus de 2 groupes. Nous choisissons 2 au lieu de 3 tel que proposé par Stokkeland pour prendre en compte le cas où 1 pale est vis-à-vis la tour. Sachant que les éoliennes ont toujours 3 pales espacées de 120 degrés, chaque groupe calcule sa distance angulaire par rapport aux autres et vote pour les groupes n'étant pas à 120 degrés de eux-mêmes. Dans le cas où DBSCAN ressort que 2 groupes, la direction de la troisième pale est approximée à 120 degrés des deux autres pales.

## Étape 6 : Recherche du rotor et filtrage

L'étape 5 nous a donc fournis 3 groupes de segments représentant chacuns les pales de l'éolienne. Le centre du rotor peut donc être calculé en moyennant chaque groupe puis en

tenant la moyenne de l'intersection des prolongements de chaque segment.

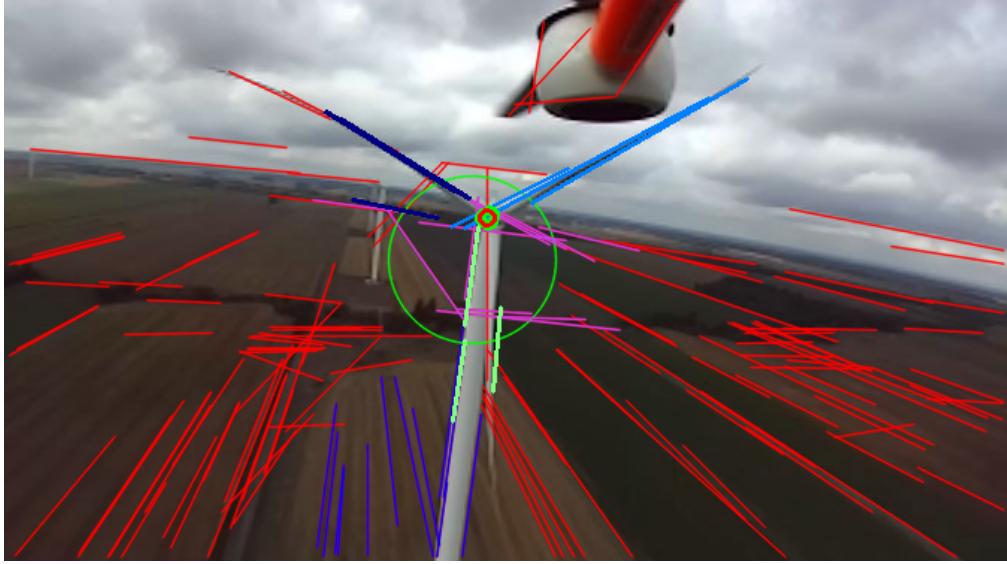


Figure 5.4 Résultat d'une détection réussie à la fin de la procédure.

Dans la Figure 5.4 nous pouvons observer les étapes 3 à 6 en opération. Toutes les lignes sont le résultat de la transformée de Hough appliquée à l'image des bordures suite à l'application de l'algorithme de Canny. Au bas de l'image, les lignes mauves sont les candidats à être des lignes de la tour. Elles se font rejettées cas elle ne s'étendent pas assez haut dans l'image. Le grand cercle vert est celui tracé au plus haut point de la tour dans lequel les segments de pales sont recherchées. Les segments roses sont les candidats de pales rejettés par l'algorithme de partitionnage DBSCAN et la procédure de vote. Les pales finales sont indiquées par les lignes bleus, bleu foncé et vert pâle. Finalement, avec l'application des étapes 5 et 6 on peut voir que l'intersection des pales croise bien sur le centre du rotor.

Le point détecté est ensuite entré dans un filtre de Kalman linéaire permettant ainsi de garder un estimé de la position du rotor dans l'image lorsque la détection échoue. Le vecteur d'état  $\mathbf{x}_k$  contient la position et la vitesse de déplacement en coordonnées image du centre du rotor. La matrice de transition  $\mathbf{F}_k$  contient les termes  $\gamma_x, \gamma_y \in [0, 1]$  faisant décroître la vitesse exponentiellement avec le temps. Ceci empêche l'estimé de diverger à l'infini quand

aucune mesure n'entre dans le filtre pendant de longues périodes de temps.

$$\begin{bmatrix} u \\ v \\ \dot{u} \\ \dot{v} \end{bmatrix}_{\boldsymbol{x}_k} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & \gamma_x & 0 \\ 0 & 0 & 0 & \gamma_y \end{bmatrix}_{\boldsymbol{F}_k} \begin{bmatrix} u \\ v \\ \dot{u} \\ \dot{v} \end{bmatrix}_{\boldsymbol{x}_{k-1}} + \boldsymbol{w}_k \quad (5.5)$$

où  $\boldsymbol{w}_k$  est un bruit blanc Gaussien. Le modèle de mesure inclut seulement la détection du point dans l'image.

$$\begin{bmatrix} z_x \\ z_y \end{bmatrix}_{\boldsymbol{z}_k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}_{\boldsymbol{H}_k} \begin{bmatrix} u \\ v \\ \dot{u} \\ \dot{v} \end{bmatrix}_{\boldsymbol{x}_k} + \boldsymbol{v}_k \quad (5.6)$$

Le terme d'entrée de commande  $\boldsymbol{u}_k$  pourrait être utilisé pour prendre en compte les vitesses angulaires commandées au robot, par contre sachant qu'il ne seront pas disponibles par télémétrie dans le md4-1000 nous ne les prenons pas en compte. La meilleure façon d'améliorer le filtre serait d'ajouter une mesure de  $\dot{u}$  et  $\dot{v}$ , chose qui pourrait être faite par des méthodes de calcul de flux optique.

Une fois le point  $(u, v)$  dans l'image trouvé, il suffit d'appliquer le modèle de caméra sténopé de la section 2.6.1 et sa matrice de paramètre intrinsèques  $K$  pour obtenir un vecteur unitaire  $u_C$  dans le repère caméra pointant vers le rotor. Puisque la distance entre l'UAV et le rotor est beaucoup plus grande que la distance entre la caméra ( $C$ ) et le centre du véhicule ( $B$ ) nous pouvons la négliger. Ce qui implique que nous pouvons appliquer quelques rotations pour remettre ce vecteur unitaire dans le système d'axes  $W$  (centré sur  $B$ ) à partir de  $C$  pour obtenir  $u_W$ .

Pour se rendre au rotor, il suffit donc de commander une vitesse dans la direction  $u_W$ . Puisque nous n'avons pas de mesure de distance entre l'éolienne et l'UAV la vitesse demeure constante, elle peut être régulée que lorsque la surface entre dans la portée du capteur de proximité.

### 5.2.2 Approche en remontant la tour

Dans cette méthode le véhicule est placé à proximité de la tour de telle sorte que le scanner laser puisse capter la forme de la tour et de la discontinuité introduite à l'endroit où la tour rencontre le rotor. Cette méthode est plus simple et plus fiable que l'approche visuelle mais

elle repose sur deux suppositions :

1. Le véhicule est bien placé par l'opérateur devant l'éolienne de tel sorte les capteurs seront perpendiculaires au plan sur lequel les pales reposent.
2. Aucune pale n'est placée ni proche, ni devant la tour et l'angle de chaque pale est connu d'avance.

Le principe opérationnel est simple et illustré dans la Figure 5.5, le scanner laser horizontal est utilisé pour centrer l'UAV devant la tour et garder une distance de sécurité alors que le scanner vertical est utilisé pour détecter le rotor. Pour ce faire il faut tout d'abord projeter le vecteur de distances à un nuage de points 3D au moyen de l'équation 5.7 où  $r_i$  est une distance,  $\theta_i$  est l'angle correspondant à la mesure  $i$  et  $n$  est le nombre de mesures retournées par le laser.

$$p_{Li} = \begin{bmatrix} p_{Lix} \\ p_{Liy} \\ p_{Liz} \end{bmatrix} = \begin{bmatrix} r_i \cos \theta_i \\ r_i \sin \theta_i \\ 0 \end{bmatrix} \text{ où } i = 1, \dots, n \quad (5.7)$$

L'équation 5.7 nous donne des points dans le repère du laser correspondant, puisque les

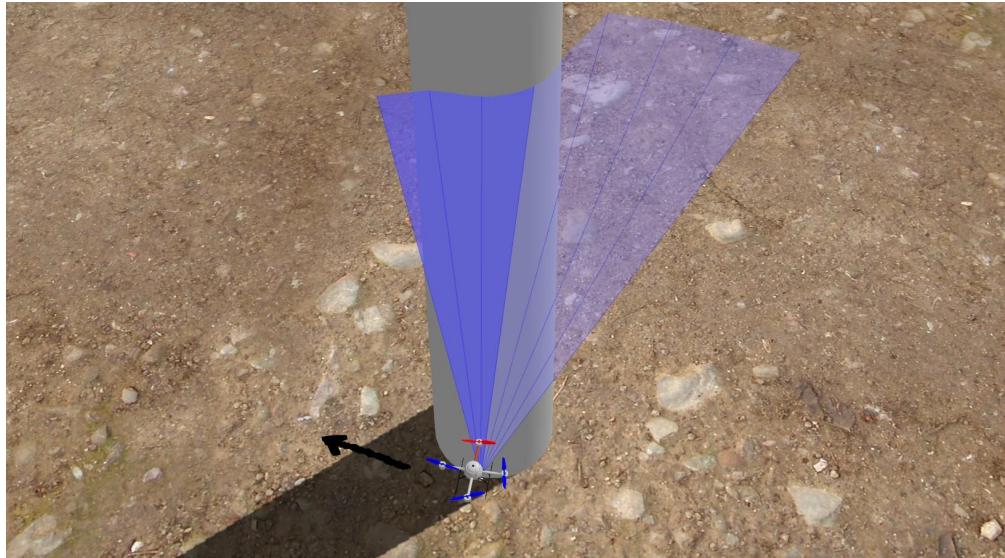


Figure 5.5 Principe d'opération de l'approche du rotor par laser.

matrices de transformées homogènes  $T_{LH}^B$ ,  $T_{LV}^B$  ainsi que la rotation  $R_B^{BT}$  sont connues, il suffit de transformer chaque  $p_L$  au repère  $BT$  suivant l'équation 5.8.

$$\begin{bmatrix} p_{BTi} \\ 1 \end{bmatrix} = \begin{bmatrix} R_B^{BT} & 0 \\ 0 & 1 \end{bmatrix} T_L^B \begin{bmatrix} p_{Li} \\ 1 \end{bmatrix} \quad (5.8)$$

Une fois ces mesures transformées dans le bon repère nous pouvons appliquer une loi de contrôle PID sur les axes  $x$  et  $y$  en plus d'une vitesse constante  $v_z$  en  $z$  pour calculer la commande de vitesse  $u(t)$  dans le repère  $BT$  à envoyer à l'UAV tel que présenté dans l'équation 5.9.

$$u(t) = \begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = \begin{bmatrix} K_{p_x} e_x(t) \\ K_{p_y} e_y(t) \\ v_z \end{bmatrix} + \begin{bmatrix} K_{d_x} \dot{e}_x(t) \\ K_{d_y} \dot{e}_y(t) \\ 0 \end{bmatrix} + \begin{bmatrix} K_{i_x} \int e_x dt \\ K_{i_y} \int e_y dt \\ 0 \end{bmatrix} \quad (5.9)$$

$$e_x = \frac{1}{n} \sum_{i=1}^n p_{BTi_x} \quad (5.10)$$

$$e_y = d - \min(p_{BTi_y}) \quad (5.11)$$

L'équation 5.10 présente le terme d'erreur  $e_x$  calculé en moyennant la composante  $x$  des points  $p_{BT_i}$  provenant du laser horizontal pour déterminer si l'UAV doit bouger vers sa gauche ou sa droite pour se centrer devant la tour. L'équation 5.11 présente le terme d'erreur  $e_y$  qui prend la plus petite mesure en  $y$  pour savoir la distance par rapport à la surface de la tour et  $d$  est la distance de sécurité à garder par rapport à la tour.

La détection du rotor se fait au moyen d'une détection de discontinuité en faisant une convolution (voir l'équation 5.13) de la composante  $y$  des points du laser vertical avec un filtre à dérivée de Gaussienne  $\Gamma(x)$  discrétilisé sur  $n$  points pour  $x = [-5\sigma, 5\sigma]$  généré par l'équation 5.12. Une discontinuité est détectée si la réponse  $r$  du filtre est plus haut qu'un certain seuil à déterminer de façon empirique.

$$\Gamma(x) = -\frac{x}{\sigma^3 \sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma^2}\right) \quad (5.12)$$

$$r = p_{BTy} * \Gamma(x) \quad (5.13)$$

Pour contrer le bruit de mesure et s'assurer que la détection est valide, le véhicule s'arrête que si la détection s'active plusieurs fois dans un certain lapse de temps. Un exemple de résultat en simulation est démontré dans la Figure 5.6. Le masque de convolution est de taille  $1 \times 8$  et  $\sigma = 0.5$ .

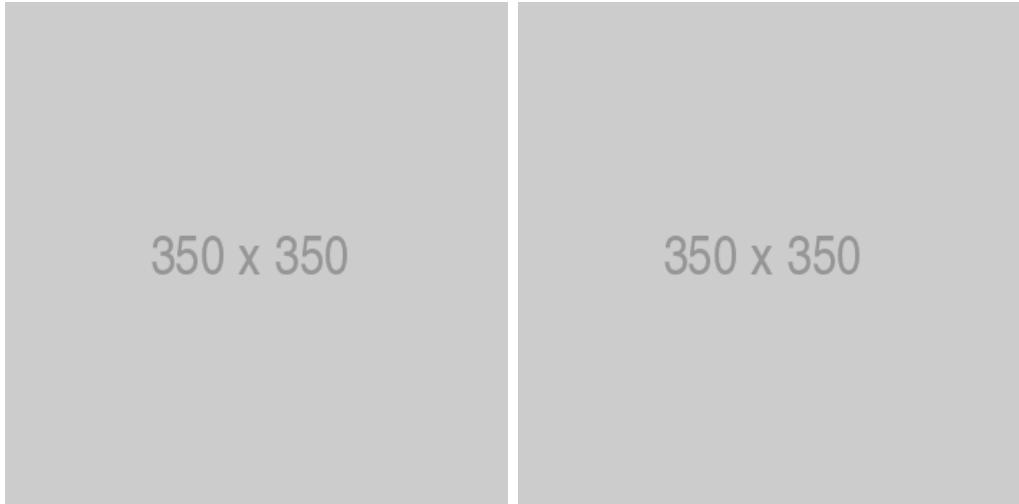


Figure 5.6 Résultat de la convolution sur un exemple de mesure.

### 5.3 Suivit des pales

Une fois devant le rotor le véhicule utilise l’information sur l’angle des pales acquise au préalable (visuellement ou par entrée utilisateur) pour savoir dans quelle direction se diriger. Il est important à cette étape de garder une grande distance de sécurité ( $d_{grand} = 8m$  ou plus) du rotor puisque celui-ci, constitué de multiples lignes électriques hautes tension et d’une grande quantité de métal, peut causer de l’interférence magnétique avec le magnétomètre de l’UAV qui peut potentiellement résulter en une perte de l’orientation. Cette distance peut être diminuée à  $d_{petit} = 4m$  lors du suivit une fois que le véhicule commence à s’éloigner du rotor.

Dans l’étape de suivit des pales, le principe est similaire à la montée de la tour, le robot cherche à suivre la pale à une vitesse constante tout en restant à une distance désiré de la pale. Nous présentons deux approches possibles, la première dans la Section 5.3.1 par nuage de point applicable à des scanner lasers 3D ou un capteur par vision active, la deuxième dans la Section 5.3.2 usant que de scanner laser 2D. Suivant la machine à états de la mission, une fois la fin de la pale détectée, le robot revient aux coordonnées GPS où le rotor a été détecté initialement. Puisque les pales d’une éolienne sont droites ou courbent vers l’avant, une trajectoire en ligne droite entre la fin de la pale et le devant du rotor est garantie d’être libre d’obstacles.

### 5.3.1 Suivit par nuage de points

Dans cette méthode nous reprenons l'algorithme proposé dans le Chapitre 4 pour le suivit de mur par UGV. L'algorithme est d'ailleurs plus simple puisqu'il n'y aucun sol à filtrer du nuage de points, cependant l'avantage majeur de cette méthode est que avec le calcul de la normale de la surface il devient possible de toujour placer le véhicule perpendiculaire à celle-ci, améliorant ainsi la qualité des images prises pendant l'inspection.

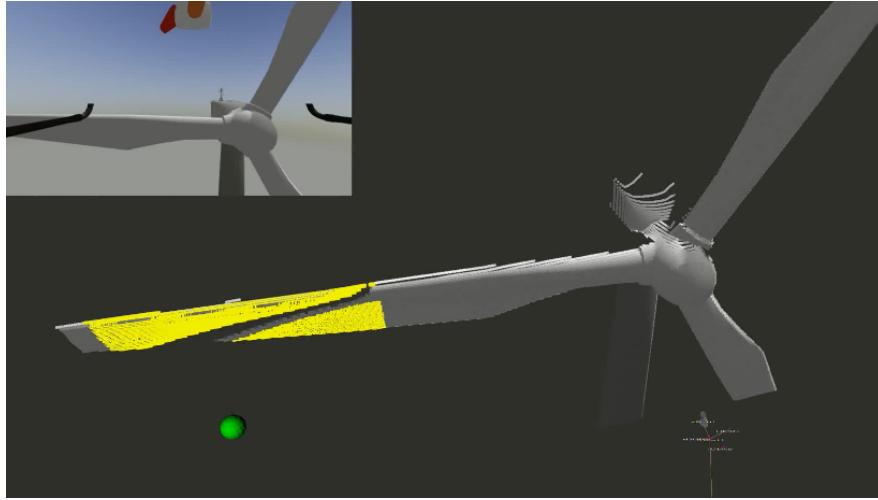


Figure 5.7 Exemple de suivi de pale par nuage de points.

La fin de la pale est détecté de la même façon qu'un mur à angle aigü est détecté dans la section **XXX**. À mesure que le véhicule avance vers la fin de la pale, le nuage de points  $\mathcal{S}$  devient de plus en plus petit, ainsi il devient possible de détecter la fin une fois que  $\mathcal{S}$  passe un certain seuil. Dans la Figure 5.7 les points blancs sont le nuage de point perçu par l'UAV alors que les points jaunes représentent la tranche sélectionnée  $\mathcal{S}$ . La boule verte est la commande de position envoyée à l'UAV, perpendiculaire à  $\mathcal{S}$ .

### 5.3.2 Suivit par scanner laser 2D

Le suivit par scanner laser se fait de la même façon que la montée de la tour au moyen de deux scanner lasers perpendiculaires l'un à l'autre. Suivant les mêmes formules de projection 2D à 3D présentées dans la section 5.2.2 le véhicule tente de se centrer par rapport à la pale en moyennant les points du nuage résultant tout en appliquant une vitesse constante  $V_C$  dans la direction estimée de suivit. Dépendamment de l'angle de la pale seulement l'un des deux lasers est utilisé pour le suivit.

La loi de controle est donc encore un PID où les termes d'erreurs sont calculés à partir de la

projection laser et la sortie est une commande de vitesse dans le repère  $BT$ .

$$u(t) = V_C(t) + K_P e(t) + K_D \dot{e}(t) + K_I \int e(t) dt \quad (5.14)$$

$e(t)$  est calculé différemment selon la direction de la pale, en pratique, on peut utiliser la moyenne du laser horizontal si la pale est à  $\pm 15^\circ$  de la verticale et le laser vertical dans tous les autres cas. Pour une pale horizontale nous aurions par exemple

$$e(t) = \begin{bmatrix} e_x(t) \\ e_y(t) \\ e_z(t) \end{bmatrix} = \begin{bmatrix} 0 \\ d - \min(p_{BT_{iy}}) \\ \frac{1}{n} \sum_{i=1}^n p_{BT_{iz}} \end{bmatrix} \text{ avec } i = 1, \dots, n$$

Dans le cas où une erreur de capteur survient, par exemple si la pale devient assez mince pour passer entre deux rayons du laser, on répète la commande de vitesse précédente  $u(t) = u(t-1)$ . Ceci implique qu'une fois que le véhicule passe la fin de la pale, il continuera à voler dans le vide. La détection de fin de pale se fait donc aisément en comptant le nombre de lectures vides dans un certain lapse de temps avec un seuil à ajuster en prenant en compte que des lectures vides peuvent survenir pendant le suivit de la pale.

## 5.4 Implémentation

Pour l'implémentation la majorité de l'effort de développement a été dédié à l'approche purement par scanner laser suivant les sections 5.2.2 et 5.3.2 puisqu'elles s'avèrent à être les plus fiables, dépendent moins de paramètres à ajuster manuellement et sont peut coûteuses en termes de matériel comparé à une approche par scanner laser 3D. Le suivi par nuage de point de la section 5.3.1 a aussi été implémenté séparément pour nous permettre de vérifier la possibilité d'utiliser un capteur de profondeur. Cependant, nous posons l'hypothèse que cette méthode ne fonctionnerait pas avec une caméra stereo dû à la surface sans texture de l'éolienne ni avec un capteur fonctionnant par *Time of Flight* de par leur portée limitée. De plus, nous implémentons aussi la méthode de Stokkeland modifiée décrite à la section 5.2.1 pour la détection du rotor mais en vu des mauvais résultats obtenus nous n'avons pas poursuivi l'implémentation du contrôleur correspondant.

La simulation est donc développée en prévision d'un quadricoptère md4-1000 de Microdrones équipé de deux lidar LeddarVu8 de LeddarTech, une camera stereo Zed de ZedLabs et un ordinateur de bord Nvidia TX2. Les lidars Vu8 ont été sélectionnés avec un angle de vu de 48° et ont la propriété particulière d'émettre des segments au lieu de faisceaux collimatés de la même manière que les lidars traditionnels. Par conséquent il devient improbable en pratique qu'une pale d'éolienne puisse se perdre entre deux rayons du lidar.

L'entièreté du logiciel a été développé en C++ à l'aide de l'interlogiciel Robot Operating System<sup>1</sup> (ROS) incluant les pilotes pour les divers capteurs et la communication par UART au md4-1000. Le traitement d'images repose sur la librairie OpenCV (Itseez, 2017) et la machine à états faisant la gestion de la mission, présentée dans l'annexe A, a été écrite à l'aide de la librairie Boost *Meta State Machine* (Schling, 2011). En tout temps la machine, que l'on nomme **Commander** garde un pointeur vers un objet implémentant l'interface **ControllerInterface**. Ces contrôleurs sont instanciés et détruits dynamiquement à chaque changement d'état, nous permettant ainsi d'avoir accès à une grande quantité de contrôleurs spécifiques tout en gardant l'utilisation de la mémoire vive à un minimum. Le **Commander** calcule les commandes de vitesse à une fréquence de 50Hz, la même que la fréquence maximale de l'odométrie. Quand une mise à jour d'odométrie ou des lidars est reçue elle est mise dans un tampon en attente du prochain cycle de calcul.

La Figure 5.8 présente les contrôleurs utilisés dans le logiciel. Nous avons en particulier le **PositionController** pour la phase de retour au rotor, le **LidarBladeFollower** pour le suivi de pale et le **TowerAscentController** pour la montée de la tour. le **DirectionController**

---

1. <http://www.ros.org/>

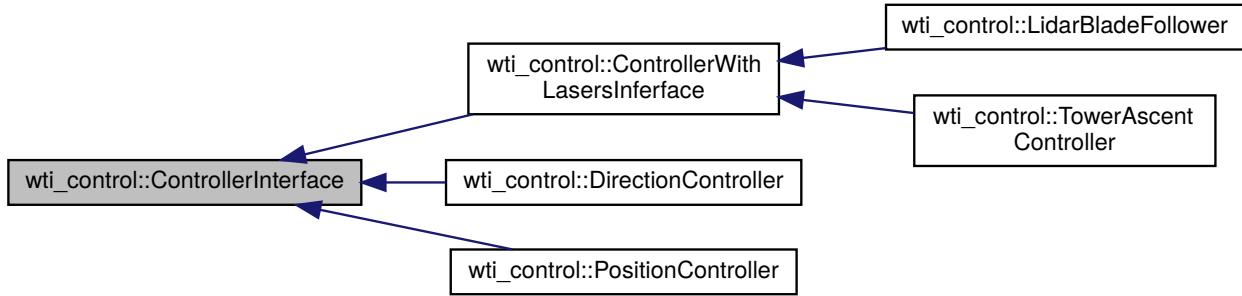


Figure 5.8 Diagramme de classe des différents contrôleurs utilisés par la machine à états.

n'est utilisé que pendant de courts instants de contrôle en boucle ouverte par exemple la manœuvre pour éviter la zone d'interférence magnétique devant la nacelle.

#### 5.4.1 Résultats en simulation

La simulation se base sur celle pour véhicules multi-rotors nommée RotorS développée par (Furrer et al., 2016) dans l'environnement Gazebo<sup>2</sup>. Puisque le micrologiciel effectuant le contrôle de vol dans les véhicules de Microdrones demeurent un secret industriel la simulation a été développé au moyen de véhicules déjà présents dans RotorS. Le but n'est donc pas de réaliser une simulation précise de la physique du véhicule. On se concentre plutôt sur la simulation des capteurs et des comportements autonomes de haut niveau.

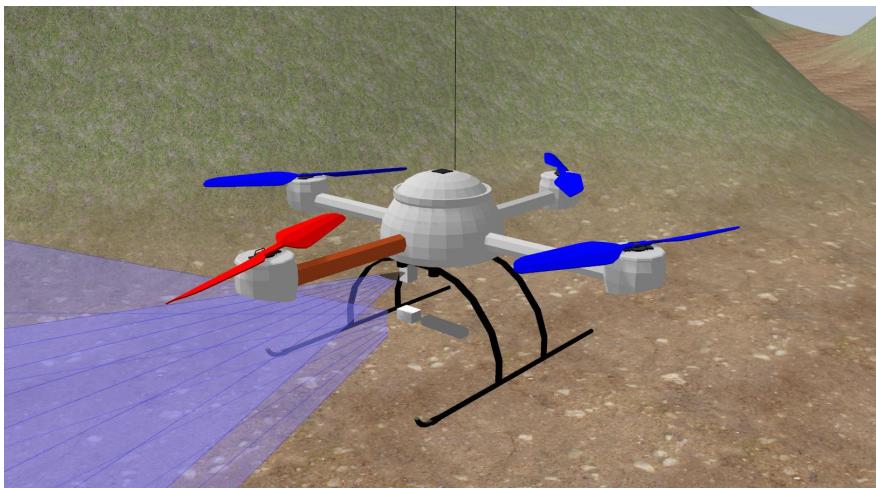


Figure 5.9 Véhicule md4-1000 en simulation. Les rayons bleus sont les rayons projetés par les scanner lasers LeddarVu8.

---

2. <http://gazebosim.org>

Plusieurs environnements ont été implémentés incluant un monde simulant un parc éolien avec plusieurs éoliennes alignées (Figure 5.10 (B)) et un monde avec une seule éolienne avec une montage au loin (Figure 5.10 (A)). La simulation suit le pire cas possible d'un parc éolien respectant les consignes du Département de l'Environnement du Royaume-Uni demandant un minimum de trois diamètres de rotor d'espacement entre chaque éolienne (Department of Environment, 2009). Il faut noter que suite à des discussions avec des travailleurs de l'industrie, l'un des défauts de la simulation est que les pales des éoliennes sont normalement inclinées vers l'arrière de tel sorte qu'elles sont parallèles au sens du vent. Ceci permet au rotor de rester stationnaire lors des inspections.

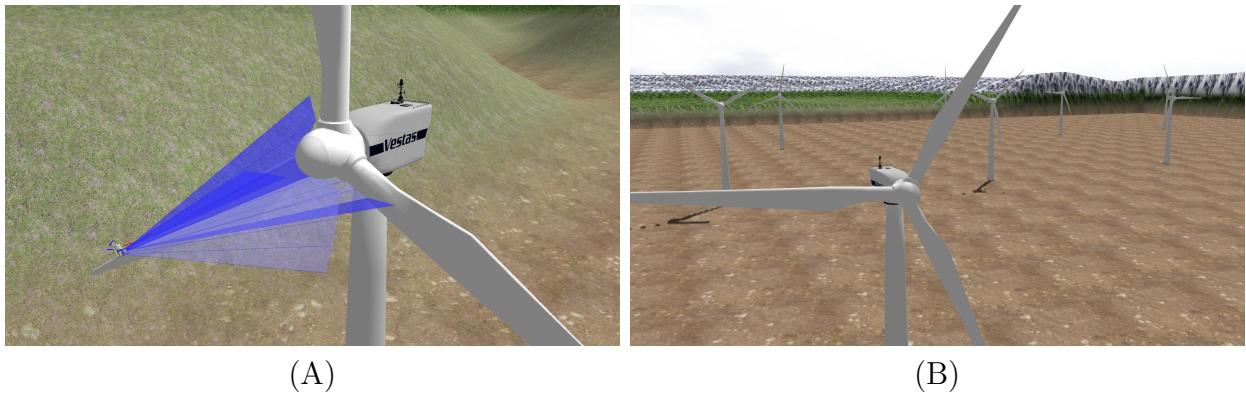


Figure 5.10 Environnements de simulation.

### Résultats par traitement d'images

Tout d'abord nous testons en simulation la méthode de Stokkeland modifiée. Nous pouvons observer dans la Figure 5.11 (A) une détection réussie en présence d'un angle de roulis et dans la Figure 5.11 (B) une détection lorsque l'horizon est présent près du rotor. Le grand cercle vert représente le rayon de recherche autour duquel les segments de pale sont recherchés, les petits cercles verts sont les intersections des projections des pales, le cercle rouge est la moyenne des intersections (le centre du rotor) et le cercle rose est la sortie du filtre de Kalman.

Suite à beaucoup d'expérimentation, il devient difficile d'apprécier cette méthode et nous démontrons qu'une approche entièrement visuelle sans information au préalable n'est pas pratique. Dans nos tests en simulant une trajectoire d'avance manuellement à l'aide d'une manette, le taux de détection varie grandement entre 10% et 60% et dépend fortement du réglage des divers paramètres au niveau des filtres Gaussiens, de la détection des contours et des lignes et des divers paramètres de distance de recherche. De plus, une fois les paramètres

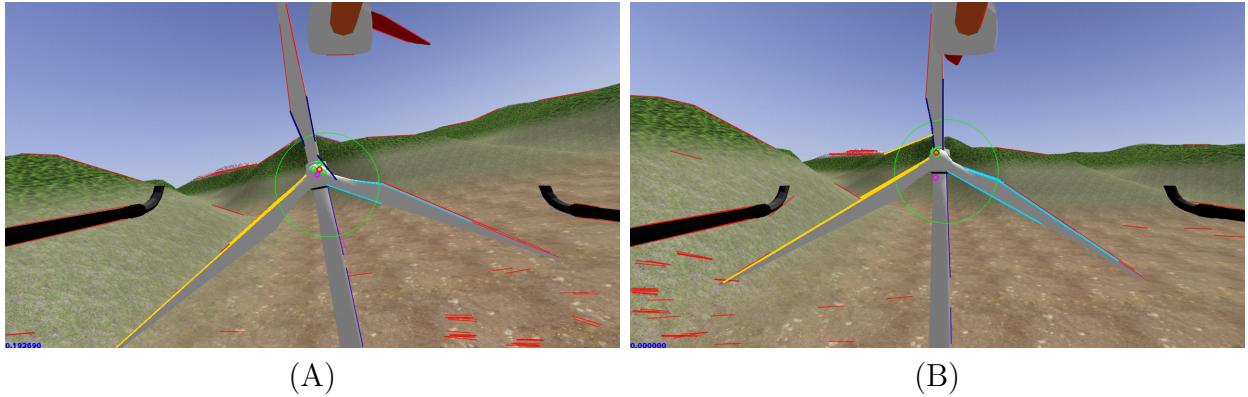


Figure 5.11 Détections réussies en simulation.

régler pour augmenter le taux de détection, ceux-ci deviennent invalides lorsque l'on modifie les paramètres de l'environnement de simulation tel que l'angle des pales, la texture du sol, les nuages dans le ciel et la position de la source de lumière. Un autre des problèmes rencontrés est la difficulté de choisir les paramètres du flou Gaussien tel que la texture de l'arrière plan est atténué tout en préservant les contours de l'éolienne pour l'algorithme de Canny. En somme, il est impossible de régler les paramètres au préalable pour accomoder toutes les conditions possibles qui seront rencontrées sur le terrain.

Nous profitons de la présence de la caméra stereo ZED simulée pour tester le fonctionnement de la résolution de la profondeur sur la surface de l'éolienne. Nous pouvons observer dans la Figure 5.12 que la surface blanche de l'éolienne ne comporte pas assez de texture pour bien calculer la résolution de blocs. Résultat innatendu, le calcul de profondeur fonctionne parfois près de la bordure de l'objet où la transition entre la pale blanche et l'arrière plan donne une caractéristique pouvant être mis en correspondance dans l'image de droite. Par contre, lorsque la pale est horizontale, elle devient quasi-invisible du fait que la bordure est parallèle à la géométrie épipolaire. Ce problème est bien connu par les chercheurs utilisant la vision stéréoscopique pour le calcul de profondeur, certains tel que (Meier et al., 2017) ont récemment proposé une solution possible à l'aide de caméras stereo orthogonales, cependant la résolution de ce problème sort du cadre de ce projet.

### Résultats du suivi par laser

La mission d'inspection entièrement par laser se déroule avec succès dans une grande variété de scénarios. Dans cette section nous présentons et analysons l'un des vols effectués sur une éolienne avec pour seule information *a priori* que l'une des pales a été placée vers le haut. Dans la Figure 5.15 on peut observer la trajectoire entière exécutée par l'UAV avec les

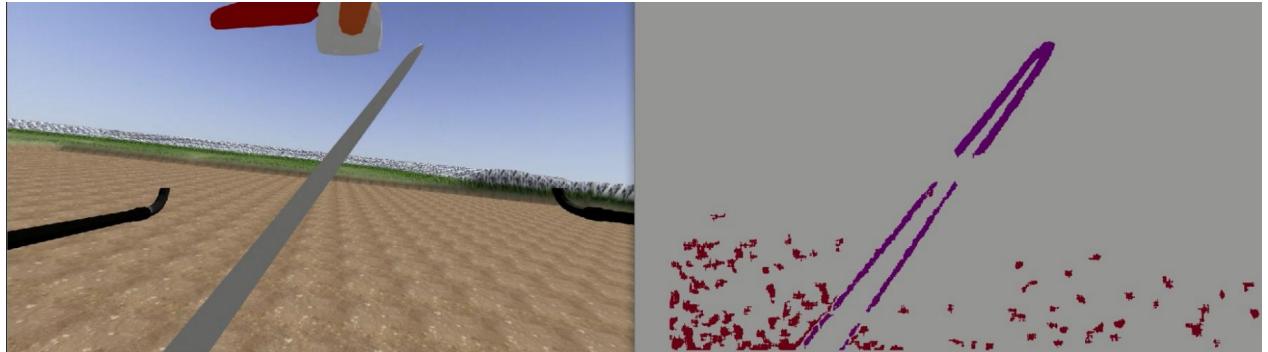


Figure 5.12 Échec de la résolution de profondeur par correspondance de blocs.

différentes phases séparées par couleur. Outre une petite discontinuité dans la trajectoire de la pale 3, la majorité du vol s'exécute de façon lisse.

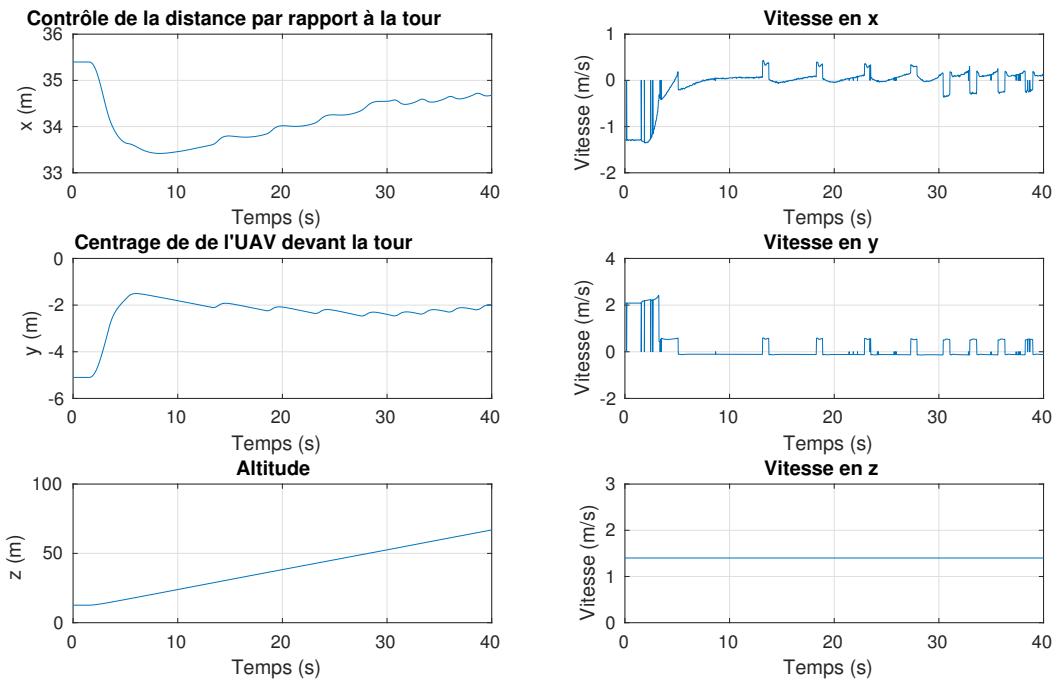


Figure 5.13 Trajectoire suivie pour la montée de la tour.

Dans la Figure 5.13 nous pouvons voir la décomposition de la trajectoire suivie pour monter la tour. L'UAV commence légèrement décentré par rapport à la tour et tente immédiatement de se centrer. La vitesse commandée en  $z$  est constante et on voit que l'altitude augmente de

façon lisse. Par contre on peut voir que la trajectoire en  $x$  et en  $y$  est légèrement saccadée. En  $x$  la raison est une combinaison de gains PID légèrement trop aggressifs et du fait que le diamètre de la tour change au fur et à mesure qu'elle monte. En  $y$  le problème provient plutôt de la faible résolution du scanner laser. Avec seulement 8 points, la bordure de la tour se retrouve entre deux rayons du laser, les différentes bosses dans la courbe  $vy$  sont les instants où l'un des rayons adjacent au centre frappe la tour. À cet instant le terme d'erreur devient non nul et le robot bouge latéralement.

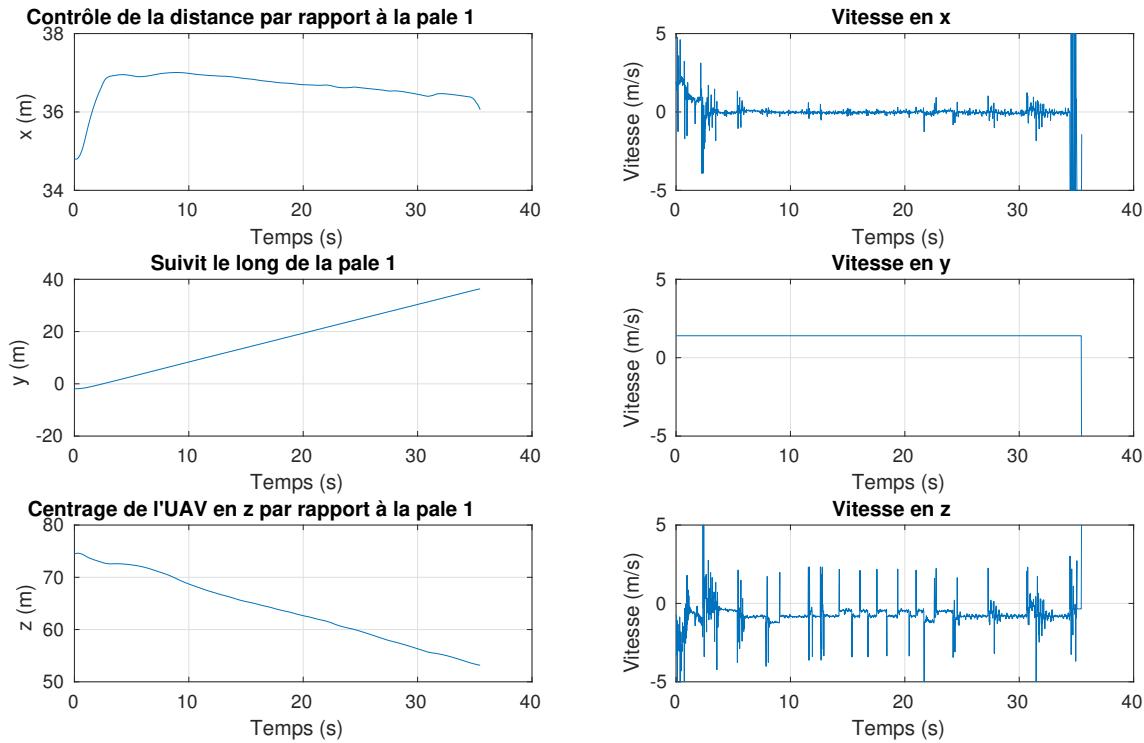


Figure 5.14 Trajectoire pour le suivit de la pale 1.

La Figure 5.14 présente la trajectoire et les commandes de vitesses pour le suivit de la pale 1. Nous voyons que les vitesses en  $z$  semblent saccadées mais ceci est encore dû au petit nombre de rayons du scanner laser. Chaque petit pic est en fait un instant où un rayon supplémentaire du laser parvient à toucher la surface de la pale et influencer le calcul de son centre. Vers la fin de la courbe  $vx$  nous pouvons observer une oscillation entre la commande 0, +5 et -5 mètres par secondes. Ceci est dû au fait que la fin de la pale est assez petite pour passer entre deux rayons du scanner laser, combiné au mouvement de tangage du véhicule pour avancer et reculer le laser parvient momentanément à toucher la pale. En réalité ceci ne devrait pas

être un problème car le LeddarVu8 ne souffre pas des angles morts entre ses rayons. Malgré cela, le robot poursuit son chemin jusqu'à ce que la fin de la pale soit officiellement détecté par le compteur de lectures vides.

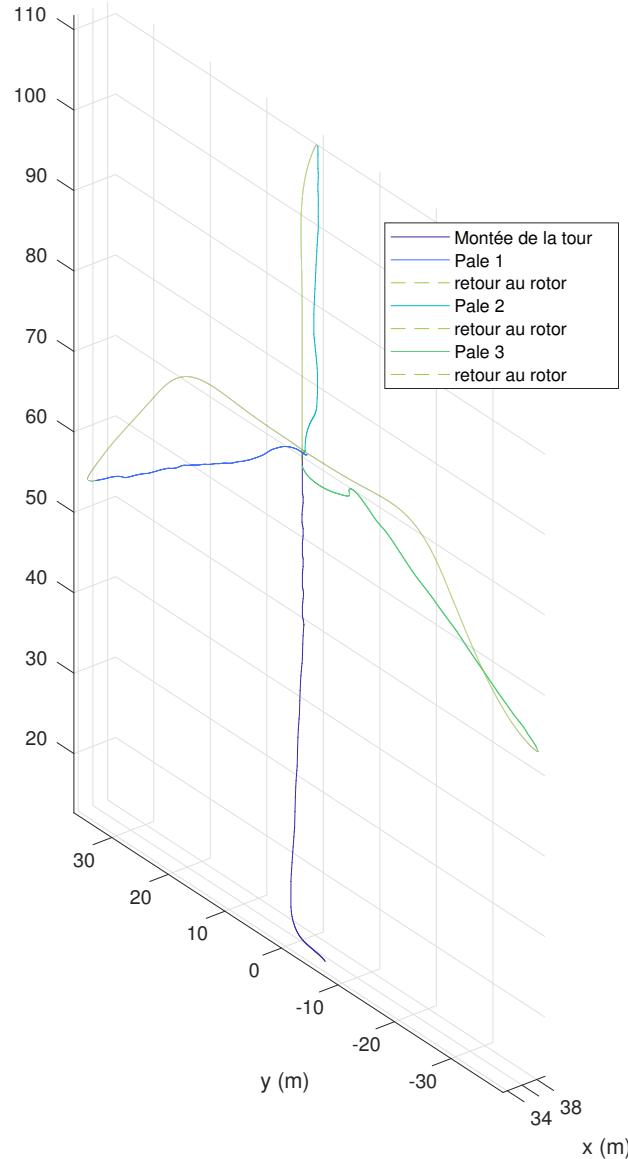


Figure 5.15 Trajectoire complète d'inspection.

### 5.4.2 Résultats de tests sur le terrain



Figure 5.16 Le md4-1000 en vol inspectant l'une des pales.

Suivant les spécifications initiales, l'intégration des capteurs s'est faite sur un md4-1000 pris en photo dans la Figure 5.17 pour faire un test de vol manuel nous permettant de confirmer certaines de nos hypothèses au niveau de la vision stereo. Le système d'exploitation Ubuntu 16.04 a été installé sur l'ordinateur Nvidia TX2 avec des modifications apportées au noyau Linux fourni par Nvidia à fin de permettre au TX2 d'interfacer avec les différents périphériques à bord de la planche de support AuvIdea J140. Les LeddarVu8 et la caméra ZED sont branchés par USB3 alors que la communication au md4-1000 se fait par UART.

Les tests de vol ont été effectués au parc éolien Pierre-de-Saurel où la hauteur de la tour est de 100 mètres et le diamètre du rotor est de 92.5 mètres. La Figure 5.16 nous donne un ordre de grandeur de la structure par rapport à l'UAV.

La caméra stereo s'avère à avoir une performance extrêmement variable à travers le vol tel que l'on peut voir dans la Figure 5.18. Il est difficile de corrélérer exactement les caractéristiques de la scène au succès ou à l'échec de la résolution de profondeur. Dans certains cas une pale diagonale est invisible alors que dans le dernier exemple elle est parfaitement visible. On voit aussi que le ciel clair est parfois résolu par erreur. Par conséquent, nous prouvons notre hypothèse initiale et en concluons qu'une caméra stereo est inutilisable dans le contexte de

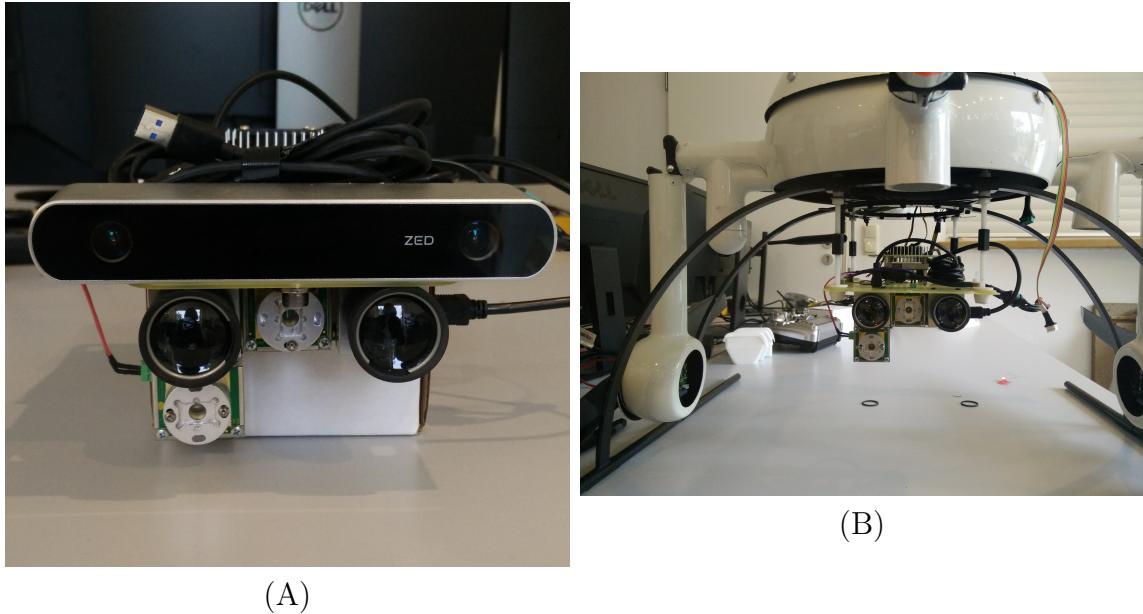


Figure 5.17 Véhicule md4-1000 avec les capteurs et l'ordinateur de bord intégré. (A) Caméra ZED et les télémètres lasers LeddarVu8. (B) La charge utile installé sur le robot (sans la ZED), avec la Nvidia TX2 visible.

l’inspection d’éoliennes.

En ce qui concerne l’approche par laser il n’a pas été possible de tester le système sur le terrain en raison de différentes circonstances (ils ont pas mit le frein alors l’éolienne bougeait, le gars en allemagne a fait une erreure dans le driver du laser etc...). Plus de temps serait requis pour faire l’intégration et le réglage des gains PID sur le vrai véhicule. Par contre, puisque l’entièreté du logiciel est déjà écrit en C++ et prêt à installer sur le véhicule, il ne suffirait que de changer quelques branchements au niveau des flux de données pour faire les tests sur le terrain.

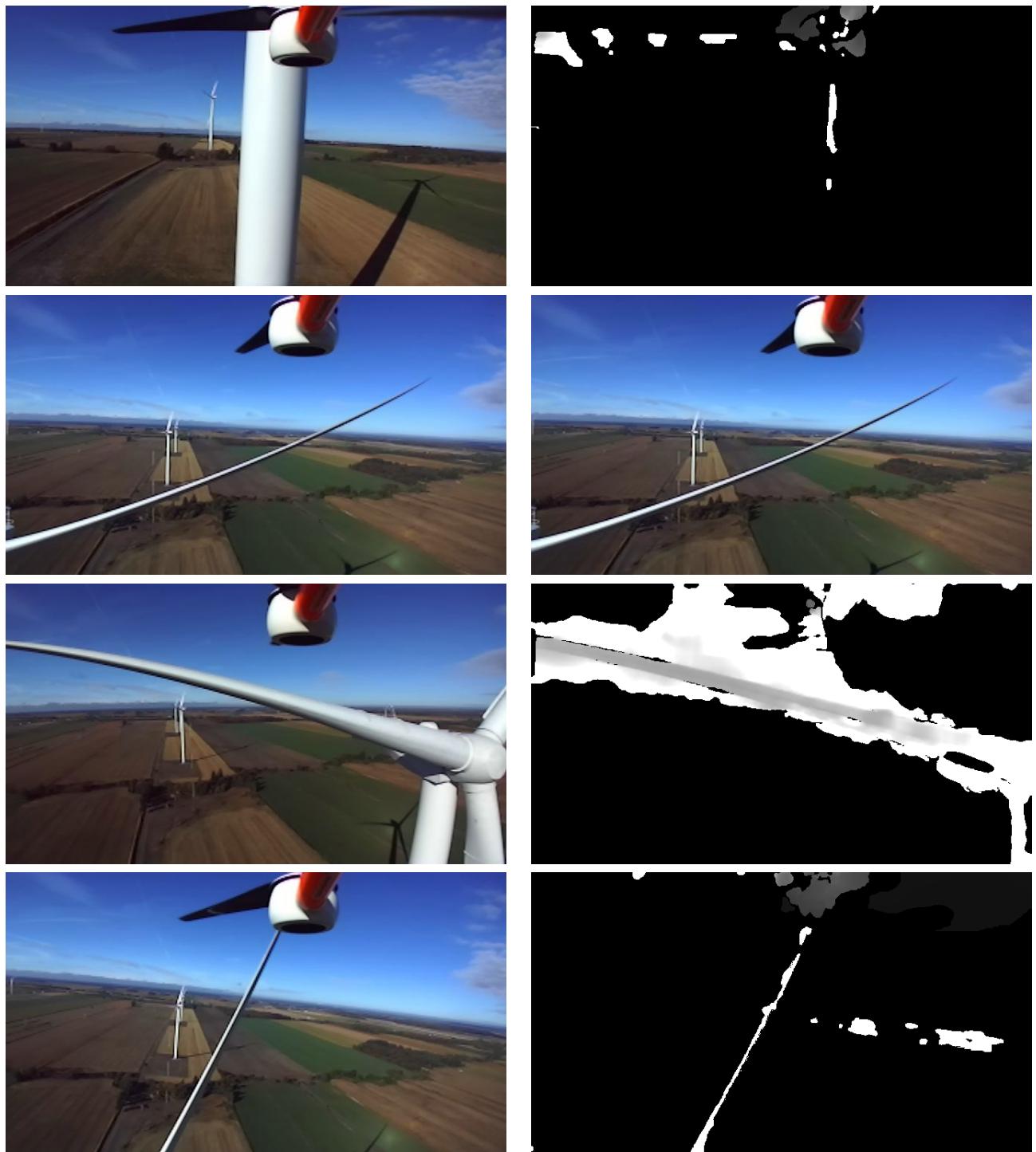


Figure 5.18 Exemples de succès et d'échecs de perception de distance. À gauche les images RGB de la caméra de gauche et à droite les images de profondeur ramenées sur la caméra de gauche.

## 5.5 Discussion des résultats et travaux futurs

Bien que nous n'avons pas pu valider notre approche sur le terrain, les résultats en simulation demeurent prometteurs et l'inspection d'éoliennes reste un projet important à poursuivre dans le futur. Par contre, suite à nos tests de vol manuels sur le terrain nous réalisons maintenant que la méthode d'inspection bénéficierait de plusieurs changements.

Tout d'abord, pour qu'un UAV entre 1 et 25 kg utilisé à des fins non-récréatives soit exempt du besoin d'un Certificat d'Opération Aériennes Spécialisées (COAS) par Transport Canada, il doit demeurer en dessous d'une altitude de 300 pieds (91 mètres) (Transports Canada, 2016). Puisque les grandes éoliennes ont leur nacelle à une altitude de 100 mètres ou plus, ceci implique que chaque fois qu'un inspecteur voudrait aller travailler il faudrait appliquer pour un COAS à l'avance ce qui peut prendre beaucoup de temps à être approuvé. Sans quoi, l'UAV n'aurait pas le droit d'inspecter les pales autres que celles pointant vers le bas.

Il reste néanmoins une alternative possible qui comporterait plusieurs avantages sur le système proposé à condition de remettre en question les modalités de la participation de l'inspecteur et du budget alloué. Le système pourrait faire une inspection en 3 dimensions d'une seule pale à la fois pointée vers le bas suivant une trajectoire présenté dans la Figure 5.19. En commençant sur le côté de la pale l'UAV longerait la pale vers bas. Il pivoterait ensuite pour remonter le bord d'attaque avant de pivoter une dernière fois pour longer le côté droit. Ceci demanderait des capteurs différents et que l'inspecteur fasse tourner les pales entre chaque vol mais aurait l'avantage d'avoir de bien meilleures images puisque la caméra serait pointée perpendiculairement à la surface à inspecter.

Pour en revenir à la méthode que nous proposons, l'UAV bénéficierait grandement de l'utilisation de D-GPS ou de GPS-RTK à deux antennes. Le premier avantage serait que avec un positionnement plus précis il serait possible de se rapprocher beaucoup plus de la structure et obtenir de meilleures images. Les distances utilisées en simulation prennent en compte que la précision du positionnement du md4-100 est typiquement entre  $\sigma = 1.0$  et  $\sigma = 1.5$  mètres. De plus, avec les systèmes à deux receveurs, il n'y aurait plus d'inquiétudes à propos de la zone d'interférence magnétique devant la nacelle.

Bien que ce soit rare, il arrive parfois que la mission échoue lorsque l'UAV perd de vue la pale qu'il est en train de suivre. La cause est habituellement une instabilité lorsque le dernier faisceau du lidar est près de la bordure de la pale. L'UAV tangue pour s'approcher ou reculer et le lidar perd et regagne de vue la pale. Ce mouvement peut se répéter jusqu'à ce que l'oscillation fasse perdre la pale complètement. Ce problème peut être palié en introduisant une meilleure gestion du cas où le lidar tombe en faute, une procédure de recherche lorsque

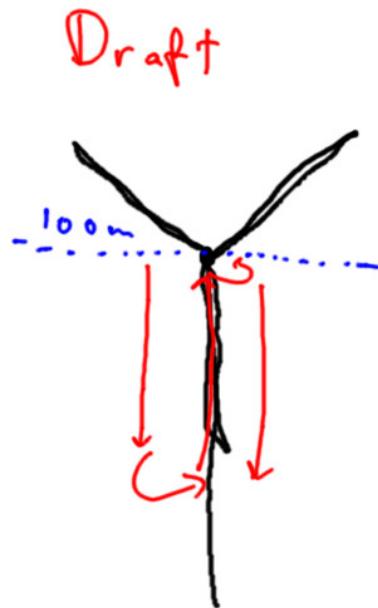


Figure 5.19 Trajectoire alternative pour l'inspection d'une pale à la fois sans besoin de COAS.

la pale est perdue et une limite sur l'accélération à la sortie des contrôleurs.

En ce qui concerne les résultats du traitement visuel, il est difficile d'imaginer que ça vaille réellement la peine de poursuivre dans cette direction. Certaines possibilités pourraient être étudiées par exemple après une détection réussie, les descripteurs de caractéristiques tel que *Oriented FAST and Rotated BRIEF* (Rublee et al., 2011) pourraient être extraits et suivis par un algorithme de flux optique (Lucas and Kanade, 1981). Ceci donnerait une deuxième mesure au filtre de Kalman potentiellement plus fiable. Une autre possibilité serait de remplacer la procédure de détection par un réseau de neurones entièrement convolutionnel qui calculerait conjointement la segmentation et l'orientation des pales.

## CHAPITRE 6 CONCLUSION

Texte.

### 6.1 Synthèse des travaux

Texte.

### 6.2 Limitations de la solution proposée

### 6.3 Améliorations futures

Texte.

## RÉFÉRENCES

“Robust Detection of Lines Using the Progressive Probabilistic Hough Transform”, *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119–137, 2000. DOI : 10.1006/cviu.1999.0831. En ligne : <http://linkinghub.elsevier.com/retrieve/pii/S1077314299908317>

A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, et R. Siegwart, “Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics”, dans *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 6423–6430. DOI : 10.1109/ICRA.2015.7140101. En ligne : <http://ieeexplore.ieee.org/document/7140101/>

A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, et R. Siegwart, “Receding Horizon "Next-Best-View" Planner for 3D Exploration”, dans *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2016, pp. 1462–1468. DOI : 10.1109/ICRA.2016.7487281. En ligne : <http://ieeexplore.ieee.org/document/7487281/>

J. Borenstein, H. R. Everett, L. Feng, et D. Wehe, “Mobile robot positioning : Sensors and techniques”, *Journal of Robotic Systems*, vol. 14, no. 4, pp. 231–249, 1997. DOI : 10.1002/(SICI)1097-4563(199704)14:4<231::AID-ROB2>3.0.CO;2-R. En ligne : [http://dx.doi.org/10.1002/\(SICI\)1097-4563\(199704\)14:4<231::AID-ROB2>3.0.CO;2-R](http://dx.doi.org/10.1002/(SICI)1097-4563(199704)14:4<231::AID-ROB2>3.0.CO;2-R)

J. Canny, “A computational approach to edge detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986. DOI : 10.1109/TPAMI.1986.4767851

L. Carlone, G. C. Calafiore, C. Tommolillo, et F. Dellaert, “Planar pose graph optimization : Duality, optimal solutions, and verification”, *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 545–565, June 2016. DOI : 10.1109/TRO.2016.2544304

H. Carvalho, P. D. Moral, A. Monin, et G. Salut, “Optimal nonlinear filtering in gps/ins integration”, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 835–850, July 1997. DOI : 10.1109/7.599254

C. K. Chui et G. Chen, *Kalman Filter : An Elementary Approach*. Cham : Springer International Publishing, 2017, pp. 19–31. DOI : 10.1007/978-3-319-47612-4<sub>2</sub>. En ligne :

[https://doi.org/10.1007/978-3-319-47612-4\\_2](https://doi.org/10.1007/978-3-319-47612-4_2)

T. Cieslewski, E. Kaufmann, et D. Scaramuzza, “Rapid Exploration with Multi-Rotors : A Frontier Selection Method for High Speed Flight”, dans *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, 2017. En ligne : <http://rpg.ifi.uzh.ch/docs/IROS17{ }Cieslewski.pdf>

D. Claus et A. W. Fitzgibbon, “A rational function lens distortion model for general cameras”, dans *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 213–219 vol. 1. DOI : 10.1109/CVPR.2005.43

DeepMap, “Hd maps for autonomous vehicles”. En ligne : <https://deepmap.ai/>

Department of Environment, “Best practice guidance to planning policy statement 18 ‘renewable energy’”, [https://www.planningni.gov.uk/index/policy/planning\\_statements\\_and\\_supplementary\\_planning\\_guidance/planning\\_policy\\_statement\\_18\\_\\_renewable\\_energy\\_\\_best\\_practice\\_guidance.pdf](https://www.planningni.gov.uk/index/policy/planning_statements_and_supplementary_planning_guidance/planning_policy_statement_18__renewable_energy__best_practice_guidance.pdf), August 2009, (Accessed on 10/23/2017).

C. Dornhege et A. Kleiner, “A frontier-void-based approach for autonomous exploration in 3d”, dans *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, nov 2011, pp. 351–356. DOI : 10.1109/SSRR.2011.6106778. En ligne : <http://ieeexplore.ieee.org/document/6106778/>

M. Ester, H.-P. Kriegel, J. Sander, et X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise”. AAAI Press, 1996, pp. 226–231.

F. Furrer, M. Burri, M. Achtelik, et R. Siegwart, “Robot operating system (ros)”, *Studies Comp. Intelligence Volume Number :625*, vol. The Complete Reference (Volume 1), no. 978-3-319-26052-5, p. Chapter 23, 2016, ISBN :978-3-319-26052-5.

Google, “Google earth vr”. En ligne : <https://vr.google.com/earth/>

H. E. Heggem, “Autonomous Wind Blade Inspection”, Master Thesis, Norges teknisk-naturvitenskapelige universitet, 2017.

B. Hepp, M. Nießner, et O. Hilliges, “Plan3D : Viewpoint and Trajectory Optimization for Aerial Multi-View Stereo Reconstruction”, may 2017. En ligne : <http://arxiv.org/abs/1705.09314>

W. Hess, D. Kohler, H. Rapp, et D. Andor, “Real-time loop closure in 2d lidar slam”, dans *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1271–1278. DOI : 10.1109/ICRA.2016.7487258

S. Høglund, “Autonomous Inspection of Wind Turbines and Buildings using an UAV”, no. June, 2014. En ligne : <http://hdl.handle.net/11250/261287>

V. Indelman, S. Williams, M. Kaess, et F. Dellaert, “Factor graph based incremental smoothing in inertial navigation systems”, dans *2012 15th International Conference on Information Fusion*, July 2012, pp. 2154–2161.

Itseez, “Open source computer vision library”, <https://github.com/itseez/opencv>, 2017.

J. Kannala et S. S. Brandt, “A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335–1340, Aug 2006. DOI : 10.1109/TPAMI.2006.153

M. Labbé et F. Michaud, “Online global loop closure detection for large-scale multi-session graph-based slam”, dans *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2661–2666. DOI : 10.1109/IROS.2014.6942926

B. D. Lucas et T. Kanade, “An iterative image registration technique with an application to stereo vision”, dans *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, série IJCAI’81. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1981, pp. 674–679. En ligne : <http://dl.acm.org/citation.cfm?id=1623264.1623280>

S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, et R. Siegwart, “A robust and modular multi-sensor fusion approach applied to mav navigation”, dans *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 3923–3929. DOI : 10.1109/IROS.2013.6696917

C. Mei, G. Sibley, M. Cummins, P. Newman, et I. Reid, “RSLAM : A System for Large-Scale Mapping in Constant-Time Using Stereo”, *International Journal of Computer Vision*, vol. 94, no. 2, pp. 198–214, sep 2011. DOI : 10.1007/s11263-010-0361-7. En ligne : <https://doi.org/10.1007/s11263-010-0361-7>

L. Meier, D. Honegger, V. Vilhjalmsson, et M. Pollefeys, “Real-time stereo matching failure prediction and resolution using orthogonal stereo setups”, dans *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017.

T. Moore et D. Stouch, “A generalized extended kalman filter implementation for the robot operating system”, dans *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

R. Mur-Artal et J. D. Tardós, “Orb-slam2 : An open-source slam system for monocular, stereo, and rgb-d cameras”, *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct 2017. DOI : 10.1109/TRO.2017.2705103

Navigant Research, “Revenue for wind turbine unmanned aerial vehicles sales is expected to total nearly 6 billion by 2024 navi-gant research”, <https://www.navigantresearch.com/newsroom/revenue-for-wind-turbine-unmanned-aerial-vehicles-sales-is-expected-to-total-nearly-6> Juil. 2015, (Accessed on 08/17/2017).

A. Noureldin, T. B. Karamat, et J. Georgy, *Introduction*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013. DOI : 10.1007/978-3-642-30466-8\_1. En ligne : [https://doi.org/10.1007/978-3-642-30466-8\\_1](https://doi.org/10.1007/978-3-642-30466-8_1)

T. Omar et M. L. Nehdi, *Automated Data Collection for Progress Tracking Purposes : A Review of Related Techniques*. Cham : Springer International Publishing, 2018, pp. 391–405. DOI : 10.1007/978-3-319-61914-9\_30. En ligne : [https://doi.org/10.1007/978-3-319-61914-9\\_30](https://doi.org/10.1007/978-3-319-61914-9_30)

C. Papachristos, S. Khattak, et K. Alexis, “Uncertainty-aware receding horizon exploration and mapping using aerial robots”, dans *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2017, pp. 4568–4575. DOI : 10.1109/ICRA.2017.7989531. En ligne : <http://ieeexplore.ieee.org/document/7989531/>

E. Rublee, V. Rabaud, K. Konolige, et G. Bradski, “Orb : An efficient alternative to sift or surf”, dans *Proceedings of the 2011 International Conference on Computer Vision*, série ICCV ’11. Washington, DC, USA : IEEE Computer Society, 2011, pp. 2564–2571. DOI : 10.1109/ICCV.2011.6126544. En ligne : <http://dx.doi.org/10.1109/ICCV.2011.6126544>

B. Schling, *The Boost C++ Libraries*. XML Press, 2011.

M. Stokkeland, K. Klausen, et T. A. Johansen, “Autonomous visual navigation of Unmanned Aerial Vehicle for wind turbine inspection”, *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pp. 998–1007, 2015. DOI : 10.1109/ICUAS.2015.7152389

R. Szeliski, *Computer vision : algorithms and applications*. Springer, 2011.

Transports Canada, “Exemption de l’application des articles 602.41 et 603.66 du règlement de l’aviation canadien”, Dec 2016. En ligne : <http://www.tc.gc.ca/AviationCivile/Servreg/Affaires/exemptions/docs/fr/2879.htm>

C. Wang, L. Meng, T. Li, C. W. De Silva, et M. Q.-H. Meng, “Towards autonomous exploration with information potential field in 3D environments”, dans *2017 18th International Conference on Advanced Robotics (ICAR)*. IEEE, jul 2017, pp. 340–345. DOI : 10.1109/ICAR.2017.8023630. En ligne : <http://ieeexplore.ieee.org/document/8023630/>

S. Wirth et J. Pellenz, “Exploration Transform : A stable exploring algorithm for robots in rescue environments”, dans *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*. Rome : IEEE, sep 2007, pp. 1–5. DOI : 10.1109/SSRR.2007.4381274. En ligne : <http://ieeexplore.ieee.org/document/4381274/>

S. "Xie et Z. Tu, “Holistically-nested edge detection”, dans *Proceedings of IEEE International Conference on Computer Vision*, 2015.

B. Yamauchi, “A frontier-based approach for autonomous exploration”, dans *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*. IEEE Comput. Soc. Press, 1997, pp. 146–151. DOI : 10.1109/CIRA.1997.613851. En ligne : <http://ieeexplore.ieee.org/document/613851/>

H. Zhang et J. Jackman, “Feasibility of Automatic Detection of Surface Cracks in Wind Turbine Blades”, *Wind Engineering*, vol. 38, no. 6, pp. 575–586, dec 2014. DOI : 10.1260/0309-524X.38.6.575. En ligne : <http://journals.sagepub.com/doi/10.1260/0309-524X.38.6.575>

Z. Zhang, “A flexible new technique for camera calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov 2000. DOI : 10.1109/34.888718

## ANNEXE A Machine à états pour l'inspection d'éoliennes

