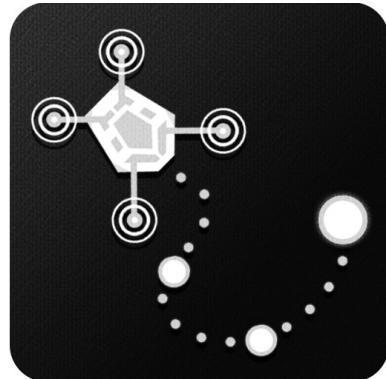


Programmation d'un drone pour la photogrammétrie

Rapport du projet

Miassa Abboute
Melissa Assam
Alexis Breton
Gaspard Coulet
Jean-Maurice Raboude



Département Informatique
Université de Montpellier
Mai 2019

Nous tenons à remercier toutes les personnes qui ont d'une manière ou d'une autre contribué au bon déroulement de ce projet. Nous tenons tout particulièrement à témoigner notre reconnaissance à notre encadrante, Mme Hinde Bouziane, qui nous a guidés tout au long de ce projet, a été à l'écoute, a pris le temps de relire ce rapport et a toujours été de très bon conseil.

Table des matières

1	Introduction	3
1.1	Objectif	3
1.2	Domaine du projet	3
2	Organisation	3
2.1	Planning	4
2.2	Outils de gestion	4
3	Fonctionnement du drone	5
3.1	Principes de base	5
3.1.1	Quels sont les mouvements que peut effectuer un drone?	5
3.2	Vol autonome	6
3.2.1	Points de passage (Waypoints)	6
4	Élaboration d'un plan de vol en fonction du sujet	6
4.1	Etude bibliographique et algorithmique	7
4.2	Notre méthode de planification de trajectoire	7
4.3	Implémentation de l'algorithme	8
5	Réalisation d'une application implémentant le DJI Mobile SDK	8
5.1	Fonctionnalités clef	9
5.2	Implémentation	9
5.3	La carte	10
6	Intégration et tests	11
6.1	Génération de la mission du drone	11
6.2	Tests expérimentaux	12
7	Conclusion	12
8	Bibliographie	14
A	Manuel d'utilisation	14
A.1	Paramétrage	14
A.2	Jonction avec le drone	14
A.3	Démarrage de la mission	14

1 Introduction

La photogrammétrie regroupe un ensemble de techniques permettant d'obtenir des informations sur un sujet physique (personne, objet, bâtiment...) à partir d'un nombre conséquent d'images de ce dernier. Pour obtenir un maximum d'informations sur ce sujet, il est important d'avoir des images de ce dernier sous des angles différents, accompagnées d'informations sur les prises de vue, comme les coordonnées depuis lesquelles la photo a été prise. Un des outils majoritairement utilisé pour la photogrammétrie en extérieur est le drone, qui permet de recueillir facilement des prises de vue aérienne ou terrestre.

1.1 Objectif

Étant donnée une scène et un sujet, l'objectif est de produire une solution permettant d'automatiser un vol de drone autour du sujet afin de prendre un nombre suffisant de photographies afin que ces dernières soient exploitables à des fins de photogrammétrie, les images doivent ensuite pouvoir être traitées par un programme issu d'un autre projet afin d'obtenir une représentation en 3D du sujet.

Cet objectif peut être divisé en deux parties : la première, l'élaboration d'un plan de vol, en fonction des informations sur le sujet, et la seconde, faire exécuter un plan de vol quelconque par un drone.

Nous disposons pour cela d'un drone DJI Phantom 3SE mis à notre disposition par le département informatique, ainsi que d'un DJI Mavic Air, appartenant à un des membres du groupe.

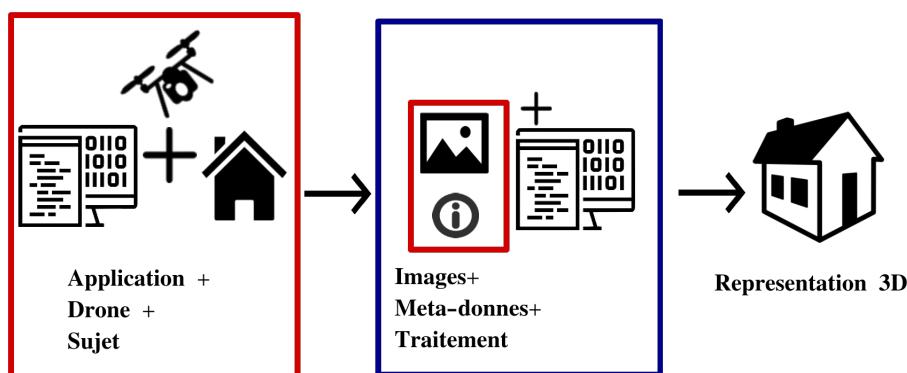


FIGURE 1 – Schéma récapitulatif - En rouge : Réalisé dans ce projet et en bleu : Réalisé lors d'un autre projet

1.2 Domaine du projet

Étant donné le matériel dont nous disposons, les interactions avec le drone se feront via une application Android que nous devrons implémenter, le projet relève donc en grande partie du développement logiciel pour Android, sur une base de Kit de Développement mobile (mobile SDK) produit et rendu disponible par DJI.

Par ailleurs, l'établissement du plan de vol nécessitera une part d'étude en géométrie dans l'espace, ainsi qu'en calcul de trajectoire.

2 Organisation

Ce projet a été réalisé grâce à une coopération étroite entre chacun des membres du groupe. Les tâches ont été réparties équitablement selon les motivations de chacun.

Nous avons tout d'abord travaillé en parallèle sur les deux premières étapes indépendantes du projet. Avant de réaliser l'intégration de ces deux parties en une application complète.

Automatisation de la création d'un plan de vol : Réalisée par Miassa, Alexis et Melissa, cette sous partie comprenait une étude des algorithmes existants de calcul de trajectoire, et devait aboutir par la production et l'implémentation d'une solution pour générer un plan de vol.

Réalisation d'une application implémentant le DJI SDK : Mise en oeuvre par les autres membres du groupe, cette sous partie portait sur des recherches quant au fonctionnement de la programmation Android, mais aussi sur celui du SDK, le but était d'implémenter une application capable de faire exécuter, par le drone, un plan de vol entré manuellement.

Intégration : Au terme de ces deux étapes, le groupe entier a travaillé ensemble afin de produire l'application finale, en implémentant l'algorithme de création de trajectoire au sein de l'application et en utilisant la sortie de ce dernier comme plan de vol pour le drone.

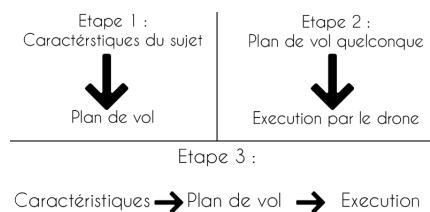


FIGURE 2 – Schéma du déroulement du projet

2.1 Planning

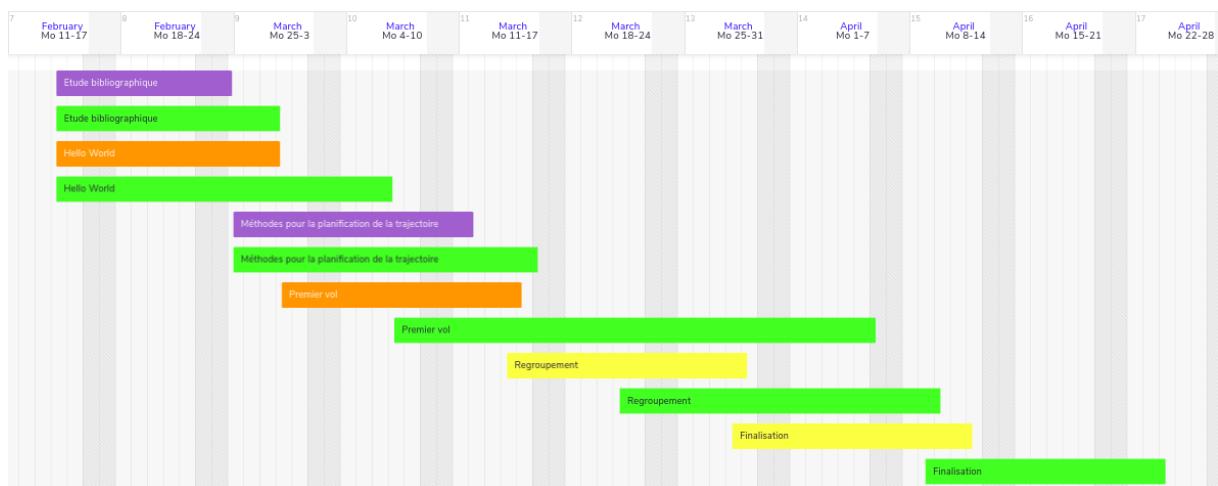


FIGURE 3 – Diagramme de Gantt

Sur le diagramme (Figure 3), nous pouvons voir en mauve, orange et jaune le planning prévisionnel : En mauve l'élaboration du plan de vol, en orange la réalisation de l'application et en jaune l'intégration. Le vert correspond au temps effectif que nous avons demandé chaque tâche. Le diagramme montre que le projet a été terminé en retard par rapport aux estimations initiales, l'évolution au sein de l'environnement Android a notamment été plus difficile que prévu. Cependant, tous les objectifs du projet ont été atteints.

2.2 Outils de gestion

Pour nous aider dans l'organisation, nous avons mis en place :

- Un serveur GitLab : https://gitlab.info-ufr.univ-montp2.fr/Gaspard_C/Drone_Routing

- En dehors de l'université, nous avons utilisé le logiciel propriétaire de communication écrite et vocale Discord, afin de parler du projet, et de partager des informations et des documents à ce sujet.
- Un Google Docs afin d'annoter l'état des tâches à réaliser.

3 Fonctionnement du drone

Un drone désigne un aéronef sans-pilote, donc sans humain à bord (le plus souvent télécommandé), il évolue dans un espace tridimensionnel (XYZ).

La première chose à comprendre est «Comment vole un drone ?», dans notre cas un quadricoptère (engin volant avec quatre hélices).

3.1 Principes de base

Le principe de base du drone quadricoptère est que la rotation des hélices permet au drone de s'élever dans les airs et de s'y maintenir grâce à la force de sustentation.

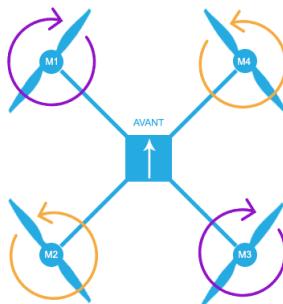


FIGURE 4 – Principes de base du vol d'un drone.

Sur l'image présentée ci-dessus (Figure 4), on distingue différentes parties du drone : les rotors M1, M3 tournent dans le sens des aiguilles d'une montre tandis que les rotors M2, M4 tournent dans le sens inverse. Cela a pour conséquence un « effet couple nul » c'est-à-dire que les forces faisant tourner le drone vers la gauche sont compensées par celles qui le font tourner vers la droite. Chaque rotor étant connecté à son propre moteur individuel, ils peuvent tourner de manière indépendante à des vitesses différentes selon les besoins.

3.1.1 Quels sont les mouvements que peut effectuer un drone ?

Le drone est un engin extrêmement agile qui peut aller dans huit directions différentes : en avant, en arrière, à gauche, à droite et en diagonale (avant-gauche, avant-droit, arrière-gauche, arrière-droit) ; sans oublier les hauteurs. Les mouvements qu'il effectuera se feront via la vitesse des différents rotors. Ainsi on distingue 4 types de mouvements distincts :

- Gaz ou Throttle (monter ou descendre) : ce mouvement s'effectue en augmentant la vitesse des quatre rotors pour monter ou en la diminuant pour descendre.
- Tassage ou Pitch (avancer ou reculer) : il s'effectue en augmentant la vitesse des rotors de derrière et en diminuant celle des rotors de devant, ainsi le drone avancera. Pour le faire reculer, la vitesse des rotors de devant est augmentée et celle des rotors de derrière est diminuée.

- Roulis ou Roll (voler vers la gauche ou vers la droite) : même principe que pour le Pitch mais avec les rotors des cotés. Ainsi la vitesse des rotors de gauche est augmentée et celle des rotors de droite est diminuée pour aller vers la droite et vice-versa.
- Lacet ou Yaw (vol circulaire) : ici la vitesse des rotors (M2, M4) est diminuée et la vitesse des rotors (M1, M3) est augmentée pour faire voler le drone de façon circulaire dans le sens des aiguilles d'une montre et le contraire est fait s'il faut le faire voler dans l'autre sens.

3.2 Vol autonome

Les drones de DJI proposent différents modes de vol autonome :

- Follow me : il suffit de demander au drone de suivre un objet et de le filmer automatiquement.
- Return to home : le drone enregistre automatiquement un point de départ en cas de perte de connexion avec la radiocommande ou dans le cas où le niveau de la batterie est faible.
- Waypoints : le drone se dirige vers un point précis et effectue une action une fois arrivé à ce point, il est donc possible de définir plusieurs points GPS ou Waypoints et le drone les utilisera automatiquement pendant que l'utilisateur se concentre sur le contrôle de la caméra.
- Point of Interest : il suffit de définir une cible comme point d'intérêt et l'avion le contournera continuellement pendant que l'utilisateur enregistre les photos

3.2.1 Points de passage (Waypoints)

Les points de passage sont des points sur une carte, plus souvent définis par des coordonnées GPS, par lesquelles le drone devra passer pour prendre des photos. Une des tâches de ce projet sera alors de décider, à partir du sujet, où s'arrêter pour prendre des photos et donc où placer les waypoints.



FIGURE 5 – Exemple d'un plan avec des waypoints application DJI Ground Station.

Dans l'exemple précédent, nous pouvons voir que le drone devra, dans l'ordre, passer par les points 1,2,3,4,5,6 à des altitudes spécifiées.

4 Élaboration d'un plan de vol en fonction du sujet

Afin de réaliser l'objectif du projet : "programmer un drone pour la photogrammétrie", comme avant de se lancer dans tout projet, plusieurs études devaient être effectuées. Cette partie a pris un certain temps, car il fallait d'abord définir une bonne stratégie de recherche relative à la partie algorithmique : l'idée donc était de commencer à faire une étude bibliographique sur les algorithmes qui peuvent être utiles pour notre étude ou même qui peuvent être une source d'inspiration pour notre algorithme final, et ensuite finir avec l'implémentation de ce dernier.

4.1 Etude bibliographique et algorithmique

Comme déclaré ci-dessus, cette étape a été l'une des plus longues. En effet, programmer un drone pour effectuer des tâches de manière quasi autonome ont nécessité des recherches approfondies afin de répondre aux questions posées, telles que : Quelle serait la trajectoire du drone ? Sous quels angles les prises de vue doivent être prises ?

Après maintes recherches, des algorithmes existants ont été trouvés. En voici les deux plus connus pour la génération de la trajectoire d'un point quelconque au sujet étudié :

Algorithme Dijkstra : Il sert à résoudre le problème du plus court chemin entre deux points A et B, sur un graphe pondéré (orienté ou non orienté).

Algorithme A* : Un algorithme de recherche de chemin dans un graphe entre un nœud initial et un nœud final tous deux donnés. Il a été créé pour que la première solution trouvée soit l'une des meilleures. Il s'agit d'une extension du précédent.

Puis vint l'étape où nous devions faire un choix : utiliser des algorithmes déjà prêts, s'en inspirer, ou alors, écrire notre propre code qui répondra à tous les besoins du projet.

4.2 Notre méthode de planification de trajectoire

Ces algorithmes peuvent être utile pour trouver un chemin optimal entre la position initiale du drone et le premier point de la trajectoire tournant autour du sujet. Cependant, des recherches ont permis de comprendre que le drone peut aller de lui-même vers ce premier point. Ces algorithmes n'ont donc pas été utilisés.

Néanmoins, la trajectoire autour du sujet doit être effectuée. Pour cela, nous avons décidé de créer notre propre algorithme. Nous considérons l'environnement comme connu et statique, c'est-à-dire que les obstacles sont invariants dans le temps. De plus, nous négligeons les contraintes de temps imposées par l'autonomie de la batterie du drone.

Algorithme de génération de trajectoire

Pour réaliser cette trajectoire, il a été décidé d'utiliser un chemin en forme d'hélice autour du sujet. Elle paraît être la trajectoire la plus optimale puisqu'elle permet au drone de former des mouvements circulaires tout en augmentant son altitude. Toutefois, la redescense du drone une fois le sommet du sujet atteint peut être utilisée pour effectuer un autre passage autour de lui afin d'optimiser le chemin et profiter de prendre plus de photos du sujet sous différents angles. C'est ainsi qu'une trajectoire en double hélice a été choisie : une première remontant le sujet, et une deuxième le redescendant, symétriquement opposée à la première.

Les deux hélices (Figure 7, en rouge et bleu) forment deux brins enroulés symétriquement l'un autour de l'autre (comme la double hélice de la molécule d'ADN). Le but est de récupérer les coordonnées cartésiens des waypoints et ensuite les convertir en coordonnées GPS qui vont servir pour l'application Android du drone.

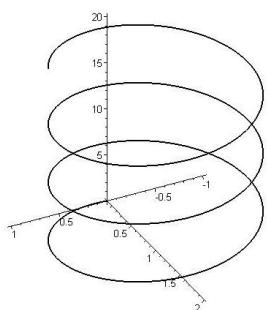


FIGURE 6 – Simple hélice

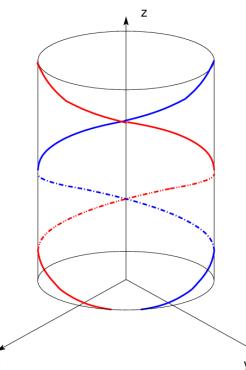


FIGURE 7 – Double hélices

Le sujet pouvant être de forme rectangulaire, il convient de donner également à notre hélice une forme elliptique afin que la trajectoire soit le plus proche possible du sujet.

La formule mathématique ci-dessous permet de calculer les coordonnées cartésiennes (x, y, z) du drone dans l'espace sur cette trajectoire.

$$\begin{cases} x = \text{lengthSubject} * \cos(\Theta) * \cos(\text{angleSubject}) * \text{widthSubject} * \sin(\Theta) * \sin(\text{angleSubject}) \\ y = \text{lengthSubject} * \cos(\Theta) * \sin(\text{angleSubject}) * \text{widthSubject} * \sin(\Theta) * \cos(\text{angleSubject}) \\ z = z + \text{heightBetweenWaypoint} \end{cases}$$

L'algorithme suivant permet de calculer la première hélice, c'est-à-dire l'hélice montante schématisée ci-dessus (Figure 7).

Les deux fonctions `xtoLat` et `ytoLon` servent à transformer un point qui est en coordonnées cartésiennes, en un point qui est en coordonnées géographiques.

Algorithm 1 WaypointsGenerator

Result: List of Waypoints

Input : `latitudeSubject`, `longitudeSubject`, `lengthSubject`, `widthSubject`, `heightSubject` : **Float**
`numberOfWaypoints`, `coefHelix` : **Int**

Variables : `Waypoints` : **Array** , `theta`, `delta`, `heightWaypoint`, `heightBetweenWaypoints`, `angleSubject` : **Float**

```

Waypoints ← []
heightWaypoint ← 0
theta ← 0
angleSubject ← (-1*angleSubject)
delta ← ((2π*coefHelix)/(numberOfWaypoints−1))
heightBetweenWaypoints ← (heightSubject/(numberOfWaypoints−1))
for i ← 0 to numberOfWaypoints do
    latitudeWaypoint ← xtoLat(latitudeSubject, (lengthSubject*cos(theta)      *
        cos(angleSubject)−widthSubject*sin(theta) * sin(angleSubject)))
    longitudeWaypoint ← ytoLon(longitudeSubject, (lengthSubject*cos(theta)      *
        sin(angleSubject)+widthSubject*sin(theta) * cos(angleSubject)))
    Waypoints.add(latitudeWaypoint, longitudeWaypoint, heightWaypoint, true)
    heightWaypoint ← (heightWaypoint+heightBetweenWaypoints)
    theta ← (theta+delta)
end
return Waypoints

```

L'hélice miroir redescendant le sujet est quant à elle calculée de la même manière en inversant uniquement l'altitude de chaque point, partant cette fois-ci du sommet du sujet vers la base (hauteur 0).

4.3 Implémentation de l'algorithme

Afin d'implémenter l'algorithme précédent, nous avons décidé d'utiliser le langage de programmation Java : Premièrement, c'est l'un des langages communs à tous les membres du groupe, il a donc été choisi unanimement. La deuxième raison est due au fait que les applications sur Android sont presque essentiellement codées en Java, et comme cet algorithme est la base de l'application Android développée par l'autre partie du groupe, il était donc important de connaître ce langage et de l'utiliser afin de faciliter le fonctionnement des différentes parties.

5 Réalisation d'une application implémentant le DJI Mobile SDK

Afin de concevoir une application capable d'interagir avec le drone, nous n'avons eu d'autres choix que d'utiliser un kit de développement sobrement intitulé "Mobile SDK" produit par DJI et étant utilisable

sous la forme d'une application Android ou iOS. N'ayant que peu de matériel Apple, nous avons choisi de travailler sur un application Android.

Les premières semaines de travail furent presque exclusivement dédiées à la lecture de la documentation du SDK et d'Android, ainsi qu'à des tests. Cette phase de découverte fut assez efficace, DJI fournissant en exemple une "Sample App" implémentant toutes les fonctions du SDK au sein d'une application, fort utile pour comprendre comment implémenter les fonctions et classes de DJI.

Une fois que suffisamment de connaissances ont été accumulées, nous avons dû faire un choix : nous pouvions soit alléger l'application d'exemple pour ne garder que les parties nous étant utiles, ou bien recréer une application de zéro pour n'implémenter que les fonctionnalités que nous prévoyons d'utiliser. Face au challenge que représentait le fait de retravailler exclusivement sur du code écrit par une entreprise, qui ne ressemblait en rien aux exemples android trouvés lors de notre étude, nous avons donc choisi la deuxième option, ce choix à permis de réduire grandement l'espace occupée par l'application.

5.1 Fonctionnalités clef

Après une phase de recherche, une liste des fonctionnalités jugées nécessaires a été élaborée :

Saisie : Permettre de saisir les caractéristiques du sujet.

Contrôle : Offrir une interface de contrôle simple vis à vis du drone et du plan de vol : lancer le vol, le mettre en pause, l'arrêter.

Retour : Afficher un retour vidéo de la caméra principale du drone dans l'application, représenter le plan de vol, la position de l'opérateur et la position de l'appareil en temps réel sur un fond de carte.

5.2 Implémentation

Ces fonctionnalités ont donc été intégrées à l'application, mais aussi d'autres, mineures, telles que l'affichage du niveau de batterie de l'appareil, de sa vitesse, des messages relatifs au déroulement de la mission...

Android est un environnement assez particulier en terme de développement logiciel, ainsi, chaque vue (ou écran) de l'application est un objet différent de type Activité, ces objets peuvent interagir directement avec leur vues associées, en affichant ou en récupérant des informations à l'écran, ces Classes sont donc des points d'entrée du programme. L'application est composée de trois activités, une activité d'accueil qui attend la connexion d'un appareil, et deux autres plus intéressantes présentées ci-dessous.

Tout d'abord sur la figure 8, l'activité Input, qui sert d'interface de paramétrage du vol, l'utilisateur y est invité à rentrer les caractéristiques du sujet.

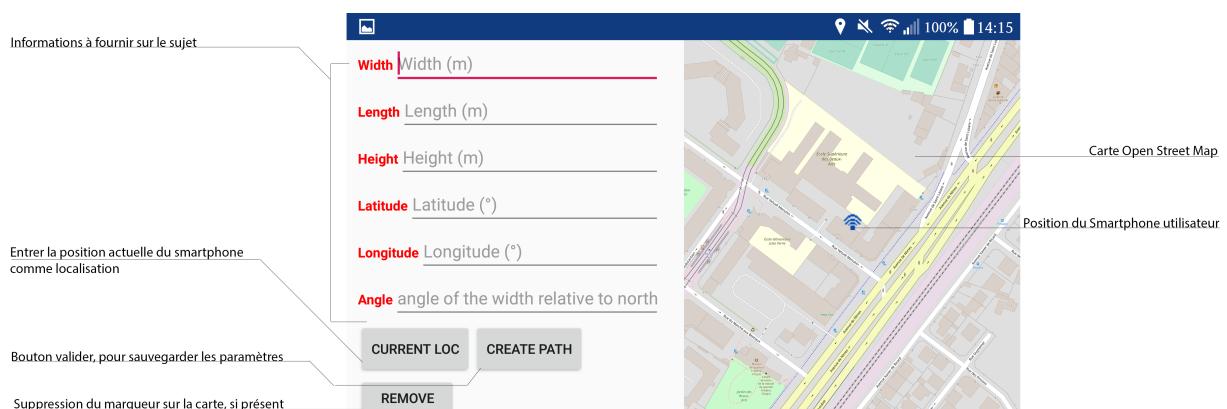


FIGURE 8 – InputActivity, l'activité de paramétrage

Ensuite figure 9, l'activité Working, qui travaille elle, avec le drone, et qui permet le chargement et la réalisation du vol selon les paramètres entrés dans InputActivity.

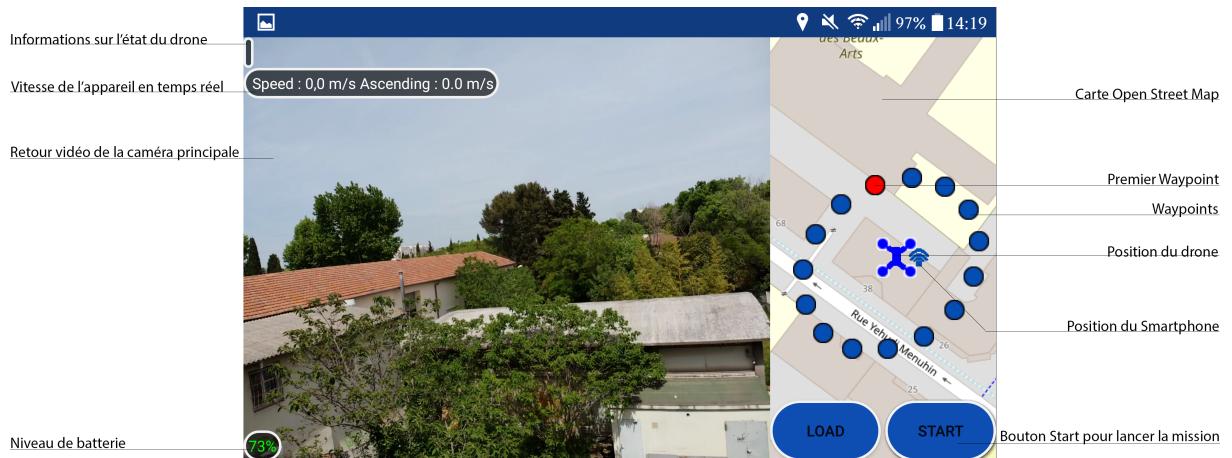


FIGURE 9 – WorkingActivity, l'activité de travail

Aussi, il est intéressant de mentionner la quantité de rétrospection dont nous avons dû faire preuve, étant donné que nous avons débuté ce projet sans avoir jamais touché à Android, nous n'avons eu de cesse de réécrire des pans de programme qui, s'ils semblaient légitimes au moment de leur écriture, ne l'étaient plus du tout quelques jours et quelques pages de documentation plus tard.

5.3 La carte

Afin de choisir le sujet et de visualiser le chemin généré par l'algorithme, il nous fallait une carte. En terme de carte sur android, nous avions plusieurs choix possibles : Google Maps, Microsoft Bing Maps, OpenLayers, OpenStreetMap,...

Nous avons choisi d'utiliser l'API de OpenStreetMaps pour android (OSMdroid) car elle est libre, bien documentée et a une grande communauté de développeurs.

Une fois la carte intégrée à l'application sur Android studio, il était question de pouvoir contrôler cette dernière. Les fonctionnalités de zoom et de centrage autour d'un point donné ont été ajoutées.

La carte est aussi en charge de récupérer la géo-localisation du smartphone, cette tâche nécessitant un lot de permissions supplémentaires que doit accorder l'utilisateur à l'application.

Afin d'afficher un point sur la carte, OSMdroid utilise des "Marker" qui reposent sur une structure "GeoPoint" représentant leurs coordonnées.

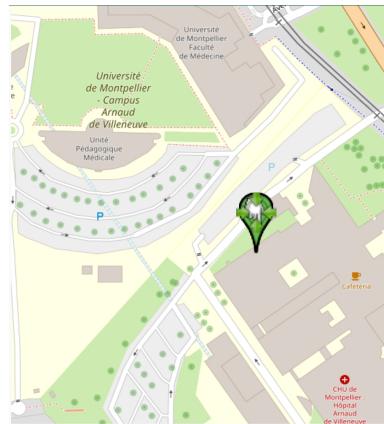


FIGURE 10 – La carte avec un marker pour montrer la position actuelle

Une fois le placement des marqueurs maîtrisé, la représentation du plan de vol complet sur la carte fut assez simple, en effet le plan de vol est composé de points de passage aux coordonnées données, il suffit donc de parcourir la liste des Waypoints en associant à chacun d'entre eux un marqueur. Pour des raisons de praticité, l'affichage d'un Waypoint est différent si ce dernier est le prochain point de passage du drone. (Visible sur la figure 9 mais pas sur la 11).



FIGURE 11 – La carte avec la liste de Waypoints

Une classe Map a été créée afin de mieux gérer la carte et les fonctions (désormais des méthodes de classe) qui ont été réalisées. Une de ces méthodes permet d'ajouter un point sur la carte manuellement et de récupérer ses coordonnées, afin de localiser le sujet plus facilement.

Il y a cependant eu quelques soucis car, une fois le drone connecté, l'application n'a plus accès à internet ; donc on ne peut plus charger la carte. Pour y remédier, l'application charge la carte dès quelle a les paramètres de l'"InputActivity". Les données sont sauvegardées dans le cache de l'application, ce qui lui permet de charger la carte même quand il n'y a plus de connexion internet.

6 Intégration et tests

6.1 Génération de la mission du drone

Tout d'abord, l'application doit générer des waypoints de la SDK du drone utilisables par celui-ci. Ces waypoints, avec leur coordonnées GPS, leur altitude et d'autres attributs tels que l'ordre de prendre une photo, sont générés par une méthode qui implémente l'algorithme en double hélice (vu plus haut). Cette méthode crée en premier, une liste d'objets d'une classe de waypoints que nous avons nous-mêmes créée,

permettant d'enregistrer les attributs de chaque waypoints, puis elle les convertie en waypoints du SDK, compréhensibles par le drone. Chaque waypoints est créé tous les "x" mètres d'altitude sur le chemin des hélices, où "x" est définie en fonction du nombre de waypoints voulus.

Cependant, cet algorithme a besoin de connaître les caractéristiques du sujet pour fonctionner. Celles-ci sont entrées par l'utilisateur dans l'InputActivity (figure 8). Nous avons décidé d'implémenter la génération des waypoints et de la mission dans l'activité WorkingActivity, puisque cette dernière manipule directement la mission au travers des boutons Load, Start, Pause et Stop (figure 9). Or, il est impossible dans Android de partager directement des objets en mémoire entre activités, rendant donc impossible le partage des caractéristiques du sujet vers la méthode générant les waypoints. Après plusieurs recherches, nous avons trouvé une solution conseillée par Android pour pallier à ce problème qui consiste à sérialiser l'objet Java sous forme de chaîne de caractères et envoyer cette dernière à l'activité voulue. Ainsi, les paramètres du sujet sont mis sous forme de chaînes de caractères et envoyées à la WorkingActivity.

Une fois les waypoints générés, l'application crée ensuite une mission pour le drone qui n'est autre qu'un regroupement de waypoints mis dans un certain ordre. Les boutons de la WorkingActivity peuvent finalement utiliser des méthodes de cette mission pour la charger, la lancer, la mettre en pause ou l'arrêter.

6.2 Tests expérimentaux

Étant donnée la législation relative aux vols de drone de loisir, il a été assez complexe de réaliser des tests dans le cadre de ce projet, il est en effet interdit de faire voler un drone sans autorisation spécifique en ville, et dans de nombreuses zones protégées. Par ailleurs, la faible autonomie de l'appareil rendait encore plus complexes les procédures de tests. Un certain nombre de tests a cependant été réalisé au cours de ce projet. Une majorité d'entre eux ont été réalisés sur un DJI Mavic Air, en mode simulateur. Lorsque cette option est activée, les moteurs sont désactivés, et les capteurs de l'appareil sont simulés via une application sur ordinateur, cette dernière offre une vue basique en 3D du drone simulée ainsi que de son environnement (un terrain plat). Nous pouvons spécifier via l'ordinateur les coordonnées auxquelles se situent l'appareil au cours de sa simulation, ou encore simuler la présence de vent. Le simulateur fournit alors une multitude d'informations sur les capteurs du drone. Cependant, cet environnement ne permet pas de simuler un sujet, ni obtenir des informations pratiques sur la trajectoire suivie par l'appareil, cette solution a été utilisée afin de vérifier que le drone effectue bien un plan de vol donné, mais n'a pas permis de vérifier la validité des prises de vues. D'autres solutions de simulation existent, notamment une éditée par DJI, qui aurait peut être été plus efficace, mais qui ne rentrait pas dans notre budget.

Par ailleurs, afin de disposer d'une vidéo de démonstration pour la présentation publique de ce projet, nous prévoyons de réaliser un test en plein air, un lieu a été choisi en dehors de Montpellier, et cette vidéo sera tournée avec comme drone de test le Phantom 3 SE du département informatique, qui est l'appareil cible de notre application. L'ensemble des tests sur simulateur ayant été concluants, ce dernier vol ne devrait pas poser de problèmes.

7 Conclusion

Notre objectif était de produire une solution permettant d'automatiser un vol de drone autour d'un sujet donné, de prendre des photos avec suffisamment d'informations pour recréer un modèle de ce dernier en 3D.

Au terme de ce projet, l'application réalisée interagit avec le drone, permet d'entrer les caractéristiques du sujet, génère un plan de vol, et le fait exécuter par le drone. Nous obtenons ainsi une quantité conséquente de photos accompagnées de métadonnées qui correspondent au résultat attendu pour une procédure de photogrammétrie.

L'application a cependant quelques limites, la plan de vol est généré sans prendre en compte la limite de batterie de l'appareil et l'exécution d'un plan de vol comportant plus de 99 points de passages nécessite que l'opérateur entre de nouveau les paramètres et relance manuellement la mission.

Bien qu'ayant rempli le cahier des charges établi, nous aurions cependant aimé rajouter quelques fonc-

tionnalités en plus, par exemple, un moyen de contrôle de type "joystick" directement dans l'application, afin de pouvoir contrôler manuellement le drone, par ailleurs la carte, même si parfaitement fonctionnelle actuellement aurait profité de quelques heures de travail en plus pour être plus agréable à utiliser, notamment au niveau de sa fréquence de rafraîchissement.

Ce projet, un peu ambitieux du fait de nos faibles connaissances préalables en matière de programmation android, nous a permis de découvrir la photogrammétrie, un domaine qui nous était inconnu auparavant, et par la même occasion, les drones. Ce fut par ailleurs une expérience enrichissante que de travailler sur un SDK assez peu populaire, cette expérience semblant assez proche de ce qui se fait dans l'industrie du logiciel. Ce projet fut, d'autre part, une occasion de mettre à l'épreuve nos capacités de gestion, qui semblent avoir été à la hauteur, l'organisation de notre groupe ayant permis d'outrepasser les difficultés liées au manque de connaissance vis à vis, notamment d'android.

8 Bibliographie

Références

- [1] "Wikipedia Drone", <https://fr.wikipedia.org/wiki/Drone>
- [2] "Comment vole un drone?", <http://drone-zone.e-monsite.com/pages/comment-vole-un-drone.html>
- [3] "Phantom 3SE", <https://www.dji.com/fr/phantom-3-se>
- [4] "DJI Mobile-SDK Doc", <https://developer.dji.com/mobile-sdk/documentation/introduction/index.html>
- [5] "Android Doc", <https://developer.android.com/guide>
- [6] "OSMDroid", <https://github.com/osmdroid/osmdroid>

A Manuel d'utilisation

Installez l'application sur un smartphone android de version 4.0 ou ultérieure, et lancez la.

A.1 Paramétrage

Vous pouvez d'ores et déjà appuyer sur le bouton "Load" afin d'entrer les caractéristiques du bâtiment sujet, il vous est demandé d'entrer la largeur du bâtiment, sa longueur, sa hauteur , la position de son centre ; pour cela, trois options sont possibles, vous pouvez :

- Entrer les coordonnées à la main dans les champs associés.
- Utiliser le bouton "Current loc" pour entrer la position actuelle de votre téléphone (matérialisé par le symbole "onde" bleu).
- Appuyer quelques secondes sur la carte à l'endroit voulu, afin de placer un marqueur, si ce dernier ne se place pas au bon endroit, vous pouvez le supprimer en appuyant sur "remove" et le replacer. Enfin, entrez l'angle formé entre le nord et l'axe de la largeur du bâtiment (le côté dont vous avez entré la dimension dans le premier champ).

Les mesures de distance sont à mentionner en mètre et les autres informations en degré. Pour valider les paramètres, appuyez sur "create path", cela vous renverra à l'écran précédent.

A.2 Jonction avec le drone

Démarrez l'appareil DJI, et connectez y votre téléphone en utilisant la jonction Wifi. Une fois cette étape effectuée, un bouton "Start" devrait être présent sur l'accueil de l'application. Appuyez sur ce bouton, vous arrivez alors sur un autre écran où la caméra du drone est affichée à gauche et présentant une carte à droite, où sont représentés les points de passage du drone. Positionnez le drone au sol à proximité directe du point visualisé en rouge, il s'agit du premier point de passage.

A.3 Démarrage de la mission

Une fois le drone positionné, vous pouvez lancer la mission en appuyant sur "Start" en bas à droite de l'écran, le drone réalisera alors la mission et restera, à son terme au dernier point de passage. Vous pouvez mettre en pause le vol durant son exécution via le bouton pause, ou l'interrompre via le bouton stop.