Université de Montpellier
Faculté des sciences

Master 1 Informatique

Programmation d'un drone pour la photogrammétrie

Groupe
TER_QBEP

Présenté par

Quentin Andre
Baptiste Darnala
Eliott Duverger
Pierre Van-Iseghem

Encadré par

Hinde Bouziane

Année universitaire : 2018/2019

Table des matières

1 Introduction

De nos jours, l'utilisation des nouvelles technologies telles que les drones est de plus en plus fréquente dans le domaine du civil. Cela fait une dizaine d'année que nous assistons à un essor technologique dans ce domaine. Après avoir été utilisés en premier lieu dans le domaine militaire, les drones équipés de caméra embarquée se sont ouverts au grand public. Efficaces et peu onéreux, ils représentent une alternative sérieuse à une pléthore de problèmes divers et variés.

Dans le cadre de notre projet semestriel de master informatique, nous avons eu pour but de réaliser une application permettant à un drone d'effectuer un vol automatique afin de photographier une scène donnée. Les photos prises sont ensuite utilisées dans un autre projet afin de reconstituer cette même scène en 3 dimensions.

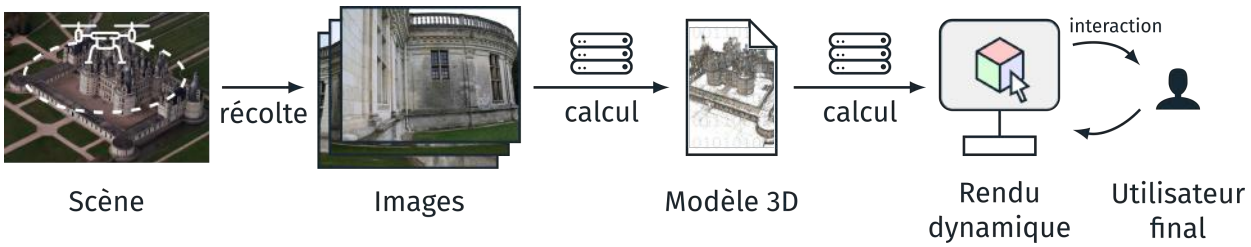


FIGURE 1 – Workflow des groupes TER

Cette modélisation pourrait permettre d'apporter un point de vu neutre dans diverses situations. On peut par exemple imaginer la reconstitution d'une scène d'accident routier ou d'une scène de crime afin de conserver l'état initial des lieux. On peut aussi imaginer une utilisation pour la modélisation de bâtiments célèbres pour la sauvegarde du patrimoine (Notre Dame de Paris).

Durant ce projet, nous devions choisir entre deux approches : Programmer une trajectoire dynamique ou produire un plan de vol. Pour effectuer une trajectoire dynamique, le drone utilise toutes les données en sa possession pour calculer sa trajectoire jusqu'à son prochain point de passage. Chaque trajectoire est calculée d'un point à un autre jusqu'à avoir terminé sa mission. D'un autre côté, la production d'un plan de vol nécessite certaines données sur la scène à photographier en amont. A partir de ces données, nous créons une liste de points de passage fixes que le drone va suivre à la lettre.

Nous avons choisi de travailler sur la deuxième approche, les objectifs étant donc de se familiariser avec le drone à notre disposition (DJI Phantom 3 SE 4K) puis de prendre en main les bibliothèques fournies par le constructeur afin d'implémenter des algorithmes de génération de plan de vol. Il s'agira de développer un ou plusieurs algorithmes qui permettront au drone de voler autour d'une scène de manière automatique tout en prenant des photographies utilisables dans le projet de modélisation 3D.

2 Le drone

2.1 Présentation

Nous avons réalisé ce projet avec le drone DJI Phantom 3 SE 4K [1] mis à notre disposition par la faculté des sciences. Ce drone, produit par l'entreprise chinoise DJI, a d'abord était une exclusivité chinoise avant d'être commercialisé de manière internationale en 2017. Ses caractéristiques principales sont les suivantes :

— Poids : 1235g

— Précision verticale : ± 0,5 mètre (en utilisant les données GPS)

— Précision horizontale : ± 1,5 mètre (en utilisant les données GPS)

— Autonomie de vol : environ 20 minutes

— Support possible sur smartphone et tablette

— Cartes SD prises en charge

Ses principaux outils d'orientation sont une boussole, les données GPS et les données visuelles. Il est déconseillé de voler trop près d'une surface réfléchissante en utilisant les données visuelles qui pourraient se trouver faussées. Afin de s'assurer la meilleure précision possible, il est fortement recommandé d'étalonner la boussole à chaque nouveau vol. En effet, cette dernière est grandement sensible aux interférences électro-magnétiques et cela peut causer une instabilité et une grande perte de précision durant le vol.

2.2 Utilisation

L'utilisateur peut évidemment contrôler le drone et sa caméra avec la manette comme n'importe quel objet volant télécommandé.

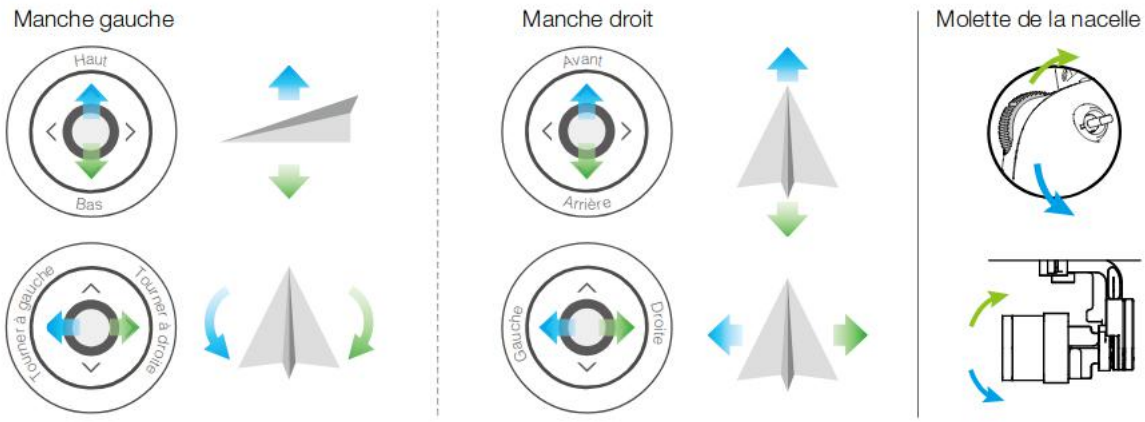
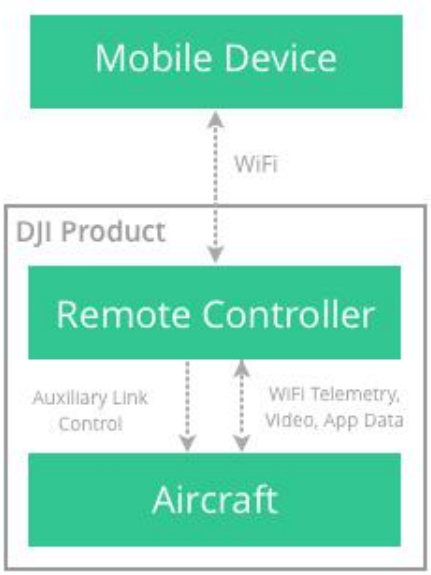


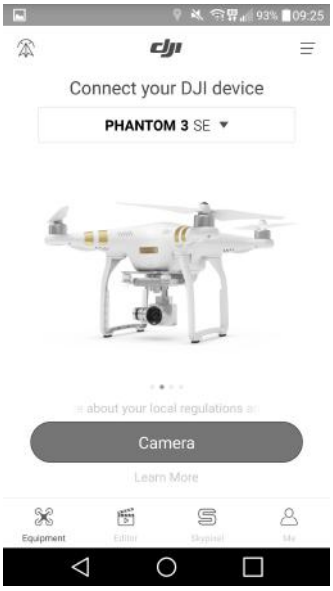
FIGURE 2 – Contrôles de la manette

1. https ://www.dji.com/fr/phantom-3-se

Cependant, il est aussi possible de connecter son smartphone à la manette en Wifi grâce à une application développée par le constructeur. Le smartphone donnera alors des ordres à la manette retransmis au drone. C'est avec cette méthode de communication que les plus néophytes du pilotage pourront contrôler le drone.



(a) Contrôle du drone



(b) Application DJI GO

Avec cette application, il est possible de se connecter au produit DJI. Une fois le produit identifié, l'application dispose d'une grande quantité de fonctionnalités :
Il peut contrôler le drone avec des modes de vol intelligents ou des missions par point de passage. Il a aussi un retour direct de la caméra qui lui permet de prendre des photos et vidéos.
En outre il y a énormément d'options et paramètres qui concernent le drone ainsi que sa caméra pour un usage professionnel.

FIGURE 4 – Interface DJI GO

3 Production d'un plan de vol

Dans cette partie, nous présenterons les études préalables que nous avons mené sur la génération d'un plan de vol.

3.1 Étude de la trajectoire du drone

Dans un premier temps, nous avons étudié les différentes solutions proposées concernant la représentation d'une trajectoire de vol.

Après étude, nous avons trouvé que le trajet du drone pouvait être considéré de 2 manières différentes : la première consiste à considérer le trajet du drone comme totalement continu, sans aucun arrêt ni décélération. La seconde, quant-à-elle, consiste à considérer le trajet comme une série de points de passage sur lesquels le drone s'arrête avant de poursuivre.

Dans notre cas, la seconde solution été plus facilement exploitable car elle comporte plusieurs avantages. En effet, considérer le trajet comme continu peut poser problème. En sachant qu'il est nécessaire de prendre des photos à certains endroits autour de l'objet, ne réaliser aucun arrêt rendrait difficile l'obtention de prises de qualités. De plus, réaliser des arrêts à chaque points de passage permet de réorienter le drone facilement vers la structure.

Au final, nous avons trouvé que la méthode de génération de mouvements point à point à profil d'accélération bang-bang était très proche de celle proposée par le constructeur sur son drone. Elle consiste à voir le trajet du drone comme une série de phases d'accélérations constantes suivie d'une phase symétrique de freinage entre des points de passage fixés que nous appellerons waypoints. Ces waypoints ont pour caractéristiques des coordonnées GPS précises (latitude et longitude) ainsi qu'une hauteur. Ils ont pour rôle de construire le trajet du drone ainsi que de repérer les points de prises photo.

La trajectoire en elle même se base sur une loi de mouvement composée de 2 splines du second degré les plus simples avec T la durée du mouvement :

$$\begin{aligned} x(t) &= at^2 + bt + c \quad 0 \leq t \leq \tfrac{T}{2} \\ x(t) &= d(t-T)^2 + e(t-T) + f \quad \tfrac{T}{2} \leq t \leq T \end{aligned}$$

Les coefficients a, b, c, d, e, f sont déterminés à partir des contraintes initiales, finales et de continuité en :

$$x(t=0) = x^d \quad x(t=0) = 0$$
$$x(t=T) = x^a \quad x(t=T) = 0$$
$$x(t=\tfrac{T^-}{2}) = x(t=\tfrac{T^+}{2})$$
$$\dot{x}(t=\tfrac{T^-}{2}) = \dot{x}(t=\tfrac{T^+}{2})$$

$$a = -2(x^a - x^d)/T^2$$
$$b = 0 \quad c = x^d$$
$$d = -2(x^a - x^d)/T^2$$
$$e = 0$$
$$f = x^a$$

3.2 Problématiques préalables pour la production d'un plan de vol

L'objectif principal étant de générer un plan de vol, nous avons commencer par trouver les éventuels problèmes que nous pourrions rencontrer. En y réfléchissant, nous avons trouvé plusieurs problématiques préalables :

3.2.1 Paramètres de production d'un plan de vol

Pour commencer, nous nous sommes concentrés sur les aspects et paramétrages possibles pour nos algorithmes.
Nus avions imaginé faire des mesures manuelles sur place, lancer notre programme, puis stocker les résultats dans un fichier. Malheureusement, cette méthode peut s'avérer être imprécise et difficile à mettre en oeuvre sur certains bâtiments (des bâtiments ayant des restrictions d'accès par exemple). De plus, les plans de vols générés seront exécutés sur une application mobile, il est donc difficile d'imaginer un utilisateur rentrer une série de waypoints sur son smartphone, surtout si ce nombre est conséquent. De ce fait, nous avons choisi de séparer les plans de vol en 2 catégories.

La première sera basée uniquement sur une série de waypoint définis à l'avance qui seront ensuite utilisés pour générer le plan de vol lors du lancement du drone. La deuxième quant à elle, prendra en paramètre la position initiale du drone puis les waypoints seront générés à partir cette dernière. En soit, nous avons donc choisi de définir un algorithme manuel et un algorithme automatique.

3.2.2 Précision et représentation des coordonnées

Il est légitime de se questionner sur la précision des waypoints utilisés et de la manière de les représenter. Dans notre cas, nous nous sommes basés sur ce que proposait Google en terme de maping, notamment sur leur manière de représenter leur coordonnés GPS.

En effet, l'application Google Maps utilise des coordonnées en degrés décimaux (41.40338, 2.17403), bien plus pratiques à utiliser car n'utilisant que 2 mesures (latitude et longitude). De plus, afin de rester cohérents, nous avons choisi une précision allant jusqu'à 10^{-6}, précision utilisée par leur application et nous permettant donc de nous appuyer sur celle-ci pour récupérer des coordonnées.

3.2.3 Prendre des photographies exploitables

En outre, nous nous sommes questionnés sur le nombre et les caractéristiques des photographies à prendre pour la modélisation 3D. Il est certain qu'instaurer un nombre de photo minimum est nécessaire, mais en prendre trop serait contre-productif. Effectivement, cela demanderait de faire un tri potentiellement non-nécessaire vis-à-vis des photos prises et augmenterait donc la complexité de la reconstitution. Au final, nous avons choisi de laisser cette variable paramétrable et de la modifier au fil des divers tests si les résultats ne correspondaient pas aux attentes données. A noter que notre choix a aussi été influencé par le groupe de modélisation 3D. En effet, il faut un nombre relativement élevé de photos pour qu'une reconstitution soit possible, surtout si la structure est complexe, il est donc normal de laisser cette entrée paramétrable.

3.3 Approche manuelle

Le premier algorithme se base sur les études précédemment réalisées sur le projet et prend en entrée la hauteur de la scène ainsi qu'un ensemble de waypoints indiqués au préalable par l'utilisateur. Ces points correspondront, après diverses modifications, aux positions auxquelles le drone effectuera ses prises, il est donc important de bien les positionner. Ce choix a pour but de laisser plus de libertés à l'utilisateur vis à vis des photos qu'il veut obtenir mais il demande aussi plus de travail.

En effet, la reconstitution 3D nécessite qu'il y ait au moins un point caractéristique commun à 2 images adjacentes, il est donc indispensable de s'assurer que la distance entre 2 points ne soit jamais trop grande. Dans le cas contraire, il sera préférable d'augmenter la distance entre le drone et l'objet afin d'avoir une vue plus globale.

Lorsque tous les points sont établis, l'algorithme cherchera la distance maximale possible entre 2 points. Cette étape a pour but de créer un premier cercle autour de la structure, cercle qui sera parcouru par le drone lors du vol et dont le centre correspondra à la position vers laquelle le drone s'orientera pour ses prises. À cela est ajouté une distance de sécurité (+-5 mètres) autour de la structure en augmentant le rayon du cercle. Enfin, les points au préalables placés sur la structure sont rabattus sur ce nouveau cercle par projection puis les points de passage sont dupliqués en fonction de la hauteur définie par l'utilisateur. Cette étape permettra de fonctionner par étage et donc de prendre en photo l'entièreté de la structure.

Différentes étapes de calcul des waypoints :

1. Calcul de la distance maximale possible entre les points.
2. Création d'un cercle avec distance de sécurité.
3. Repositionnement des points existants sur le cercle.
4. Duplication du cercle en fonction de la hauteur donnée.

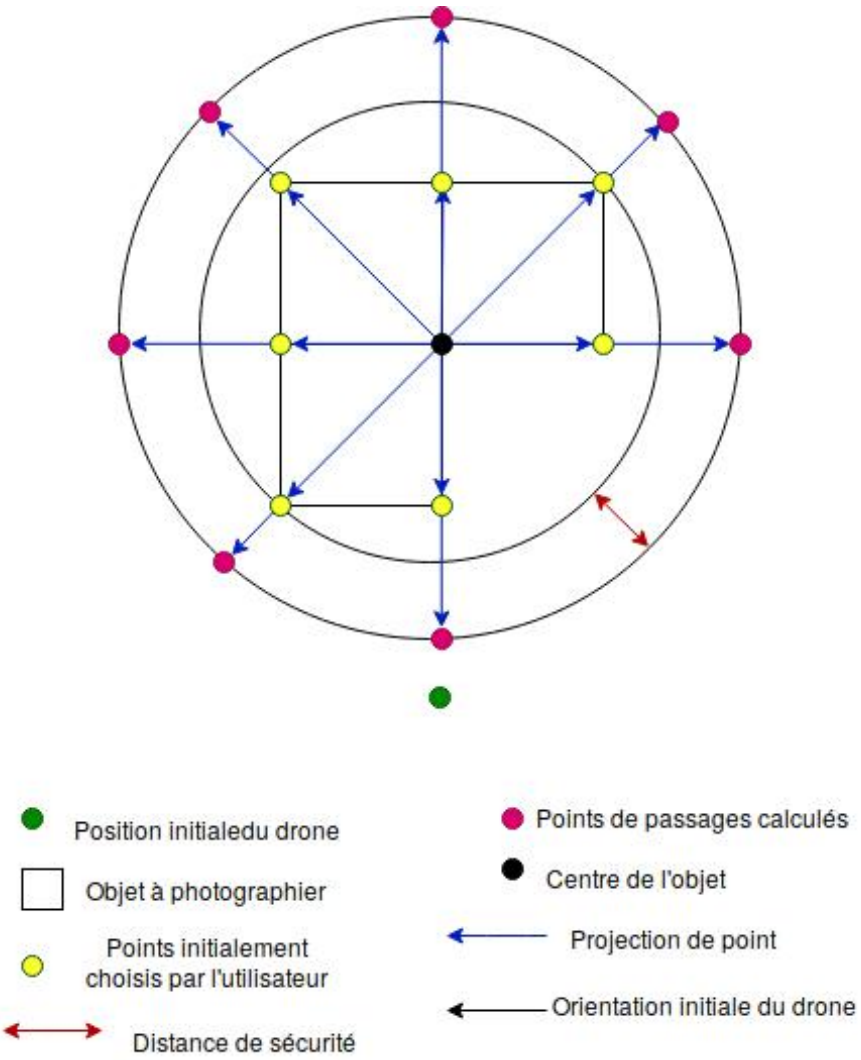


FIGURE 5 – Application de l'algorithme 1 sur une structure quelconque

‘

Algorithm 1 Renvoi un nouvel ensemble adapté de waypoints

Require: E un ensemble de waypoint, h la hauteur de la scène
  $EFinal \leftarrow null$
  for i $<$ taille(E) **do**
    for j $<$ taille(E) **do**
      $d \leftarrow distanceMax(E(i), E(j))$
      $Amax \leftarrow E(i)$
      $Bmax \leftarrow E(j)$
    end for
  end for
  $dSecu \leftarrow d + distanceSecurit$
  $g \leftarrow getMiddle(Amax, Bmax)$
  for i $<$ h **do**
    for j $<$ taille(E) **do**
      $az \leftarrow getazimut(g, j)$
      $azInverse \leftarrow getazimut(j, g)$
      $F \leftarrow projectionPoint(g, dSecu, az)$
      $ajout\ \ F\ de\ l'orientation\ du\ drone$
      $EFinal \leftarrow add(F)$
    end for
  end for
  $ReturnEFinal$

Dans cet algorithme, on parcourt tout d'abord l'ensemble des waypoints mis à notre disposition afin de trouver la distance maximale possible entre eux. Lorsque celle-ci est trouvée, on y ajoute une distance de sécurité puis on crée le nouveau waypoint correspondant au centre du segment qui sera le centre du cercle. Enfin, on réalise deux boucles : la première se basant sur la hauteur et qualifiant le nombre de tours autour de l'objet (1 tour pour 1 mètre de hauteur) ; la deuxième parcourant les waypoints existants pour les transposer sur le nouveau cercle que l'on vient de créer. Finalement, on obtient des waypoints positionnés autour de l'objet et qui sont répartis tous les 1 mètre de hauteur.

Dans le cas de l'algorithme manuel, le calcul de l'orientation du drone pour chaque waypoint est différent. En effet, il est possible de positionner le drone n'importe où avant d'appliquer le plan de vol, il est donc impossible de récupérer des informations préalables. De ce fait, afin d'obtenir l'orientation du drone pour chaque waypoints, ceux-ci sont précalculés à partir du centre du cercle. Cette orientation correspondra à un azimut qui sera détaillé plus loin.

Algorithm 2 Calcul de l'azimut/orientation entre deux waypoints

Require: a et b deux waypoints avec des coordonnées en radian
  $x \leftarrow \cos(a.lat) \times \sin(b.lat) - \sin(a.lat) \times \cos(b.lat) \times \cos(b.longi - a.longi)$
  $y \leftarrow \sin(b.longi - a.longi) \times \cos(b.lat)$
  $Renvoyer 2 \times (y/sqrt(x^2 + y^2) + x)$

Enfin, lorsque l'orientation est obtenue, elle est associée au waypoint pour la photo.

3.4 Approche automatique

L'objectif du second algorithme est avant tout de pallier aux défauts du premier et notamment au choix des points de passage qui peut s'avérer être particulièrement chronophage. Dans cet algorithme, on se basera donc seulement sur la position de départ du drone ainsi que les paramètres clés du parcours voulu :

1. le nombre de photos désirées
2. la distance entre le drone et l'objet
3. la hauteur de l'objet
4. le nombre de tours effectués par le drone

À partir de ces informations, on calculera un cercle faisant le tour de la scène que l'on transformera par la suite en hélicoïdale où les points de passage seront répartis de manière équitable. Pour cela, on prendra la position de départ du drone, son orientation, et grâce à la distance drone-objet passée en paramètre, on en déduira les coordonnées GPS de l'objet. À partir de ces coordonnées et de cette distance, on pourra ensuite projeter les points de passage avec le pas d'angle déduit du nombre total de photos désirées, ainsi que du nombre de tours que devra effectuer le drone. On obtient alors les points de passage qui correspondent à l'emplacement depuis lequel les photos seront prises.

Ces points de passage sont ensuite répartis sur toute la hauteur de l'objet ce qui nous donne un plan de vol à la forme hélicoïdale. Enfin, lorsque le drone commence sa mission et qu'il arrive à un point de passage, il suffit donc de corriger l'angle grâce à l'orientation de départ afin de diriger le drone vers l'objet et de prendre les photos.

Algorithm 3 Plan de vol hélicoïdal

Require: N le nombre de photos voulues, H la hauteur de l'objet, DO la distance drone-objet, T le nombre de tour de l'hélice.

$A \leftarrow Orientation\ du\ drone$
$DRONE \leftarrow Coordones\ GPS\ du\ drone$
$OBJET \leftarrow projectionPoint(DRONE, A, DO, 0)$
$Pas\_angle \leftarrow \frac{T \times 2 \times \pi}{N}$
$A \leftarrow A + \pi$
$i \leftarrow 0$
$WaypointList \leftarrow \{\}$
while $i < N$ **do**
  $A \leftarrow A + Pas\_angle$
  $WaypointList[i] \leftarrow projectionPoint(OBJET, A, DO, \frac{i \times H}{N})$
  $WaypointList[i] \longrightarrow Corriger\ l'orientation\ du\ drone$
  $WaypointList[i] \longrightarrow Prendre\ la\ photo$
  $i++$
end while

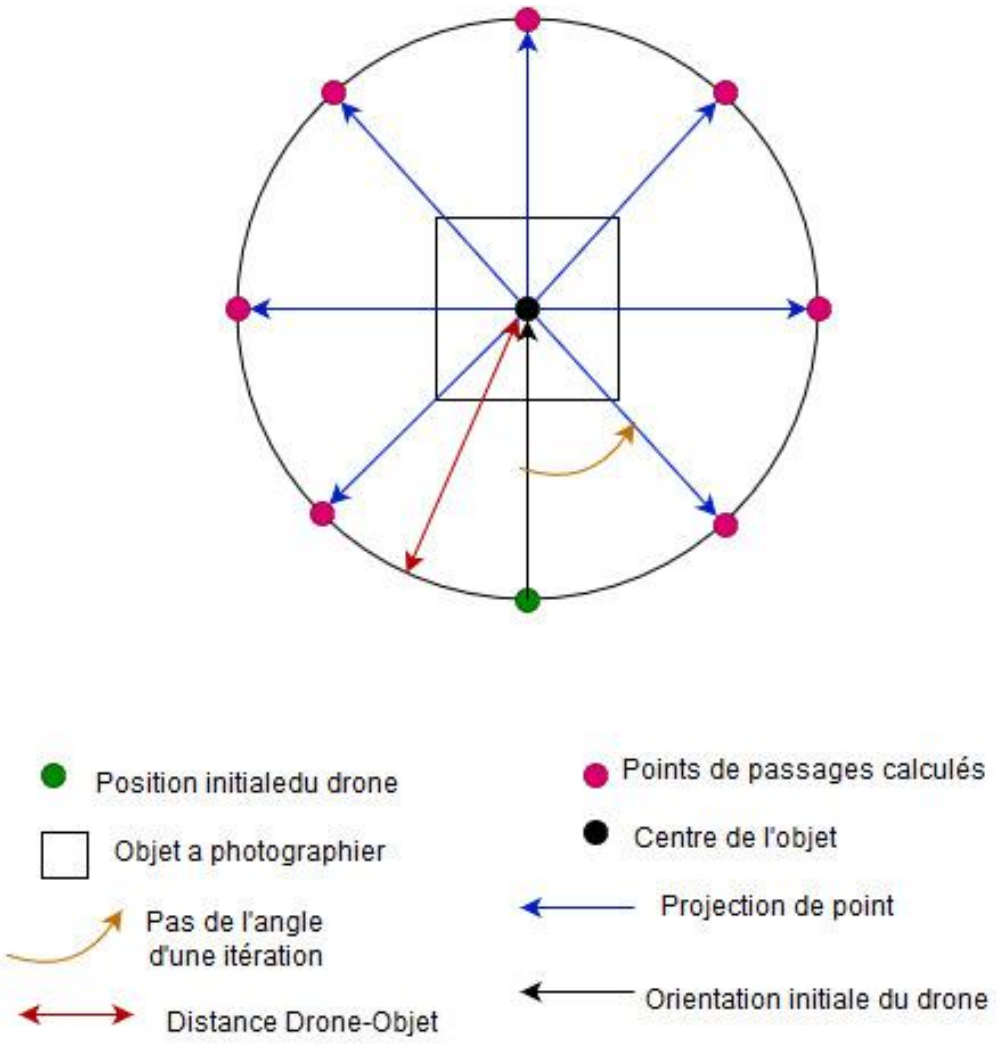


FIGURE 6 – Application de l'algorithme 2 dans un cas général avec 8 Waypoints répartis sur une hélice à un tour

La fonction projectionPoint(d, a, do, alt) permet de créer un nouveau Waypoint en projetant le point d dans la direction a, à une distance donnée do et en ajoutant l'altitude alt au nouveau point.
Pour calculer les points de passage automatiquement, il nous faut diviser l'hélice en sections équivalentes. Ainsi, le drone parcourra la même distance entre chaque points et l'angle de prise de vues entre chaque photographies sera le même. Pour calculer ces points, il nous faut donc récupérer l'orientation initiale du drone avec la distance entrée en paramètre ainsi que les coordonnées GPS du drone. On utilise la fonction de projection décrite un peu plus bas pour calculer les coordonnées GPS de l'objet.
Grâce à la distance et à ces coordonnées, il ne manque plus qu'une direction pour calculer les nouvelles coordonnées de chacun des points de passage. Enfin, nous diviserons notre hélice en **N** points de passage. Ainsi à chaque itérations, l'azimut utilisée pour générer le nouveau waypoint augmente de : $\frac{T \times 2 \times \pi}{N}$, avec **N** le nombre total de point de passage et T le nombre de tours de l'hélice.

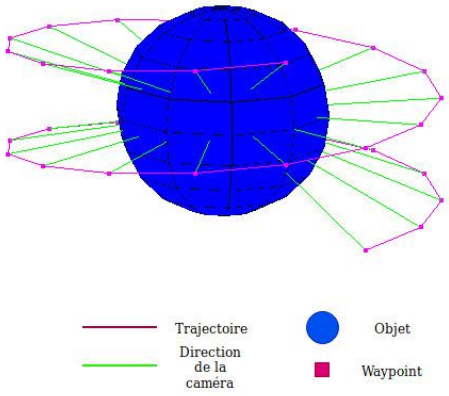


FIGURE 7 – Application de l'algorithme 2 sur un objet sphérique en 3D

À chaque point de passage généré, il faut ajouter une altitude. Afin de la calculer, nous récupérons l'altitude initiale **AI** du drone lors de la génération des points et l'altitude **Alt** passée en paramètre par l'utilisateur dans l'application. Et pour chaque point de passage N_i créé, l'altitude $AI + \frac{i*Alt}{N}$ y est associée.

Lorsque le drone parcourt les points de passage de sa mission, il doit s'orienter vers l'objet afin de prendre une photo à chaque point de passage. Pour cela, nous pouvons nous aider de l'action attribuable aux waypoints ROTATE_AIRCRAFT qui demande en paramètre l'azimut vers lequel doit s'orienter le drone.
L'azimut que demande le drone est un entier allant de -180°à 180°. Il faut également ajouter a cela le mode de rotation du drone qui peut tourner dans le sens horaire ou anti-horaire selon son mode.
Ainsi, s'il est en mode horaire, l'azimut 0°représente le nord, 90°l'est, -90°l'ouest et 180°le sud. Sinon, l'est sera à l'azimut -90°, l'ouest 90°et le sud -180°.
Ainsi, pour calculer l'orientation du drone, on récupérera l'orientation initiale que l'on réduira à chaque itération lors de la création de points de passage. Cette réduction s'effectue avec le même pas que celui utilisé pour la générations de waypoints : $\frac{T \times 2 \times \pi}{N}$. Enfin, on veillera a garder l'azimut dans l'intervalle [-180° ;180°].

Projection d'un point en coordonnées GPS

Lorsque l'azimut est calculé à chaque orientation, il est nécessaire d'appliquer une projection à partir du centre de l'objet afin de répartir les waypoints. Pour cela, on utilisera une distance **d** que l'on projettera à partir du centre sur une orientation donnée. Dans notre algorithme de projection de point, notre azimut de référence est le Nord et il est orienté dans le sens horaire, c'est-à-dire que l'est vaudra 90°, le sud 180°, l'ouest 270° :

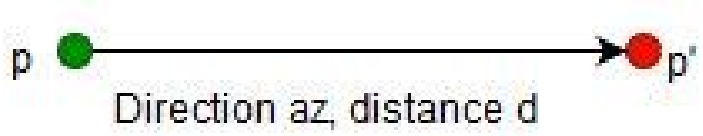


FIGURE 8 – Projection de p sur une distance d dans une direction az

La distance **d** variera selon l'algorithme utilisé. Dans le cas du premier algorithme, cette distance correspondra au rayon de la distance maximale possible entre les points à laquelle on ajoutera une distance de sécurité. Dans le cas du second algorithme, elle est définie par l'utilisateur au lancement de la mission. Cette distance doit être exprimée en radian. Pour convertir une distance de mètre à radian, on se base sur l'expression d'un Mille marin, c'est-à-dire 1852 mètres, ce qui est égal à une minute de latitude de la Terre.

Algorithm 4 Projection d'un Point en coordonnées GPS

Require: **p** un point avec des coordonnées GPS convertis en radian, **d** distance en radian, **az** la direction en radian

$p'.latitude \leftarrow \arcsin\left(\sin\left(p.latitude\right) \times \cos az + \cos\left(p.latitude\right) \times \sin\left(d\right) \times \cos\left(az\right)\right)$

$p'.longitude \leftarrow IEEERemainder^{2}(p.longitude - (arctan2(\sin\left(az\right) \times \sin\left(d\right) \times \cos\left(p.latitude\right), \cos\left(d\right) - \sin\left(p.latitude\right) \times \sin(p'.latitude)) + \pi), \pi/2)$

Renvoyer p'

4 Application mobile pour l'exécution d'un plan de vol

4.1 Objectifs

Afin d'implémenter les différents algorithmes proposés, nous nous sommes fixé pour objectif de développé une application mobile capable de se connecter au drone et de lancer des missions de prise de vue. Notre but est de proposer une application simple permettant à n'importe qui de l'utiliser, quelque soit sa connaissance sur les drones. Avec une série de paramètres modifiables dans l'application, un utilisateur pourrait donc commander son drone pour tourner autour d'une scène donnée mais aussi interrompre à tout moment le vol en cas de problème. Enfin, il lui suffira ensuite de récupérer les photographies enregistrées sur la carte micro SD du drone.

4.2 Spécifications Techniques

Pour la réalisation de cette application, les kits proposés par DJI permettent un développement sur Android Studio (Android) ou Xcode(IOS). Nous avons décidé de privilégier le système d'exploitation Android car chacun des membres du groupe possède un smartphone Android.
Le langage utilisé pour l'application et l'implémentation des algorithmes est Java. Java étant natif à Android Studio, cela ne nous posait aucun problème car l'intégralité de notre groupe a suivi le cursus scolaire de la licence informatique de Montpellier.

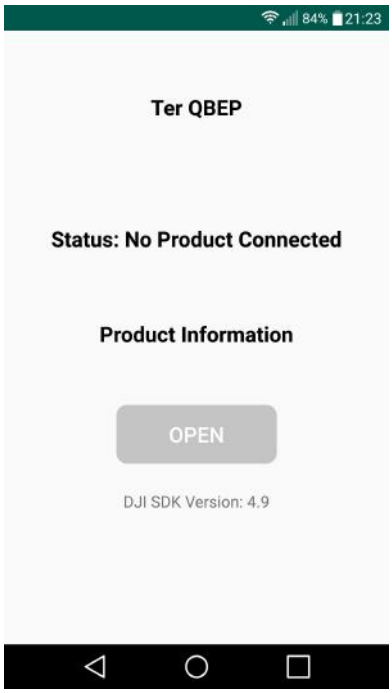
Comme aucun des membres du groupe n'était familier ni à Android Studio, ni aux bibliothèques misent à disposition par DJI pour contrôler le drone, nous avons appris à utiliser les deux outils en suivant les différents tutoriels sur le site du constructeur et sur les différents réseaux communautaires. Ces tutoriels nous ont permis d'apprendre à nous connecter au drone, lui donner des instructions (décollage, atterrissage, etc...), générer des missions avec des waypoint et utiliser l'API de Google Maps pour afficher les waypoints calculés. Ces tutoriels nous ont aussi permis de travailler sur le type d'architecture derrière une application conçue avec Android Studio : une architecture MVC (Model Vue Controller). Pour certains des membres du groupe, nous avions déjà rencontré ce type d'architecture en cours de Données du Web avec Angular.

4.3 Développement de l'application

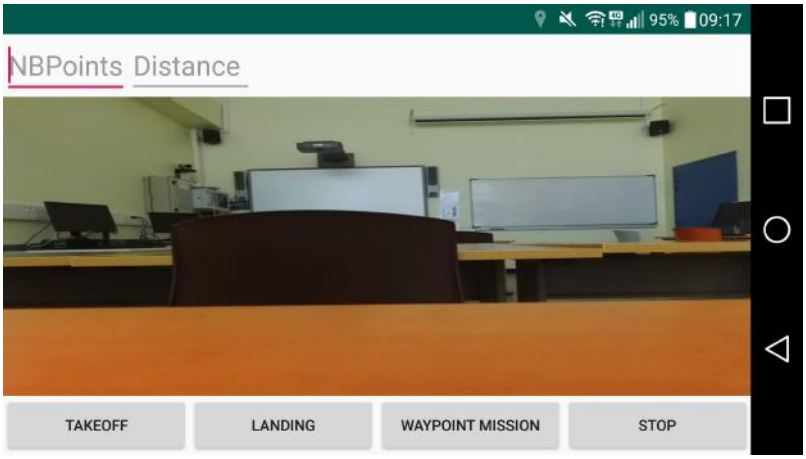
Dès le début du développement, nous voulions avoir un prototype qui nous permettrait d'effectuer des essais réels avec le drone dans le but d'affiner nos algorithmes. Nous avons donc rapidement développé une application test.
Connexion aux serveurs de DJI Pour programmer une application qui contrôle un produit DJI, il faut s'enregistrer en tant que développeur. Chaque compte développeur peut alors demander une clé d'identification pour chaque projet qu'il développe.

Pour poursuivre la vérification, l'application va alors se connecter aux serveurs de DJI avec la clé d'identification, c'est pourquoi il est nécessaire d'être connecté à internet pour valider cette dernière étape.

Enfin, une fois toutes ces étapes vérifiées, l'activité de contrôle du drone va s'ouvrir. En revanche, si un des pré-requis précédents n'a pas été rempli, l'application bouclera en essayant de se connecter.



(a) Écran d'accueil du prototype



(b) Interface graphique du prototype

Pour ce prototype nous avions donc un retour caméra, un bouton pour décoller et un bouton pour se poser (les autres boutons n'étant pas encore implémentés ou utiles). Nous utilisions ces simples commandes afin de vérifier que la connexion au drone était bien établie.

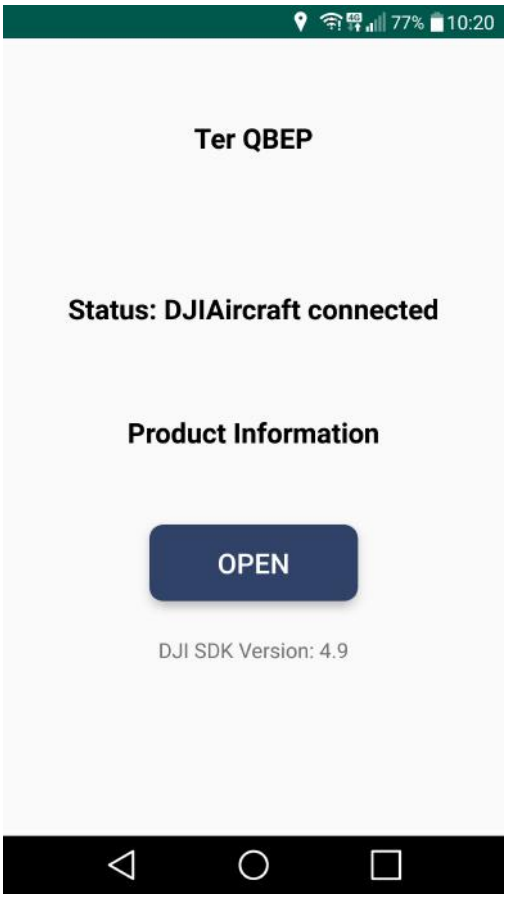
Quelques fois, durant nos tests, le drone ne répondait à aucun ordre. La connexion s'était déroulée sans erreur, nous avions un retour caméra direct, mais il était impossible de le faire décoller. Après plusieurs jours de recherche, le drone fonctionnait correctement sans aucune modification du code. Nous supposons que ce dysfonctionnement provenait d'une faible liaison GPS , d'interférences électro-magnétiques, ou d'un problème interne au drone (boussole mal initialisée). Ces problèmes étant des problèmes que nous avions rencontrés également dans l'application DJI GO, nous avons pu les corriger rapidement.

4.4 Produit final

Après plusieurs semaines de développement en parallèle à la conception des algorithmes, nous avons fini par avoir un aperçu de notre application finale. Celle-ci est divisée en deux activités.

Il y a tout d'abord un menu de connexion afin de pouvoir passer à l'activité de contrôle du drone. Celle-ci vérifiant certains pré-requis :

Connexion à la manette
L'application va vérifier si le smartphone est bel et bien connecté au réseau Wifi de la manette ainsi que la manette est également connectée au drone.



(a) Menu de connexion

(b) Interface graphique

Interface graphique
La deuxième activité sert à contrôler le drone et mettre en place des missions de waypoints. C'est à partir de là que l'utilisateur va saisir ses paramètres, exécuter un algorithme de production de plan de vol, vérifier si les Waypoints calculés lui conviennent, puis lancer la mission.

4.4.1 Paramètres d'application

Les algorithmes que nous avons implémentés ont besoin de différents paramètres pour générer leur liste de waypoints. Pour que l'utilisateur puisse rentrer ses paramètres, nous avons mis en place un formulaire. Il se divise en 4 parties :

NBPoints
Le nombre de waypoint total que devra suivre le drone et ainsi le nombre de photos prise.
Alt
L'altitude maximale du drone en mètre atteinte lors du vol.
Dist
La distance en mètre qu'il y a entre le drone et le centre de l'objet à photographier.
NBTours
Le nombre de révolution autour de la scène que le drone effectuera lors de son vol.

Remarque : Pour l'algorithme manuel nécessite seulement l'altitude maximum tandis que l'algorithme automatique utilise la totalité de ces paramètres pour pouvoir générer une mission de waypoints.

4.4.2 Instructions basiques de sécurité

Takeoff
Nos algorithmes ne prenant pas en compte le décollage du drone depuis le sol, nous avons décidé d'implémenter le bouton **Takeoff**. Faire les choses de cette manière permet d'ajouter une étape de sécurité durant la mission : il faut tout d'abord faire décoller le drone en vol stationnaire à 1 mètre du sol. En effet, cela permet de vérifier que le drone n'a pas de problème technique quelconque tel qu'un signal GPS faible ou une boussole mal initialisée. Ces problèmes se détectent facilement en vol stationnaire car ils font dériver le drone ou le rendent instable. Si le drone ne présente aucun problème apparent lors du décollage, la mission peut alors être initialisée.
Stop et **Landing**
Par ailleurs, si pendant le vol, pour quelque raison que ce soit, le drone agit de manière dangereuse, il est possible d'appuyer sur le bouton **Stop**. Ce bouton équivoque va rendre le drone stationnaire le plus rapidement possible. Il faut alors vérifier qu'il puisse atterrir sans complication au niveau du sol. Si tel est le cas, le bouton **Landing** va simplement le faire se poser. Si, en revanche, la mission été interrompue au dessus d'un fossé par exemple, il faudra le déplacer manuellement avec la manette vers un endroit sécurisé puis appuyer sur le bouton **Landing**.

4.4.3 Missions de Waypoints

La partie graphique de l'application de compose de plusieurs boutons :

Manual Mission / Automission
Ce bouton permet de passer d'un algorithme à l'autre. L'algorithme affiché est celui qui sera utilisé lors de la mission.
Load et **Waypoint Mission**
Le bouton **Load** permet de valider les différents paramètres rentrés par l'utilisateur et de générer la mission en fonction de l'algorithme courant. Des messages visuels indiqueront à l'utilisateur si la mission s'est générée correctement ou si, au contraire, une erreur s'est

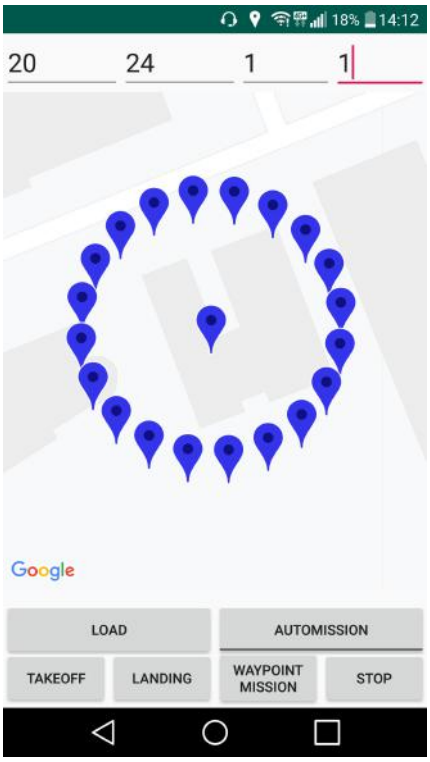


FIGURE 11 – Affichage des waypoints d'une mission chargée

produite.
Si la mission est prête, en appuyant sur le bouton **Waypoint Mission**, le drone se mettra en mouvement en suivant les Waypoints fraîchement calculés.

Dans la bibliothèque DJISDKManager, un waypoint est représenté par trois valeurs : des coordonnées polaires (latitude et longitude) ainsi qu'une altitude. Contrairement aux coordonnées, la valeur de l'altitude dépend de la hauteur du drone lors de l'initialisation de la mission.

Pour mener à bien une mission, l'utilisateur va en premier lieu appuyer sur **Takeoff**. Le drone qui aura alors décollé en vol stationnaire servira d'étalon pour l'altitude. L'utilisateur remplira ensuite les champs de paramètres en fonction de l'algorithme choisi. Il lui suffira ensuite d'appuyer sur le bouton **Load** pour charger la mission puis **Waypoint-Missions** pour exécuter la mission.

Google Maps
Durant ce projet, nous avons également implémenté la carte de l'API Google Maps dans notre application. Une fois le plan de vol calculé, les waypoints s'afficheront sur la carte, ce qui permettra à l'utilisateur de visualiser le trajet que le drone va effectuer et donc d'effectuer une pré-vérification du plan de vol calculé. Nous pouvons noter que le waypoint central représente le centre de la scène, il ne sera donc pas survolé lors de la mission.

5 Expérimentations

5.1 La législation des drones

Une fois l'intégration faite, il a fallu procéder à la phase d'expérimentation. Cependant, faire voler un drone n'est pas si simple. L'ouverture au grand public de cette technologie a mené à un renforcement des législations dans ce domaine. En résumé, il est interdit de faire voler un drone en ville ou au dessus d'une zone interdite (à moins d'obtenir une dérogation spécifique) et il est interdit de dépasser le plafond des 150 mètres. Nous avons donc utilisé la carte gouvernementale de restriction de vol pour savoir à quel endroit nous pouvions tester nos algorithmes.

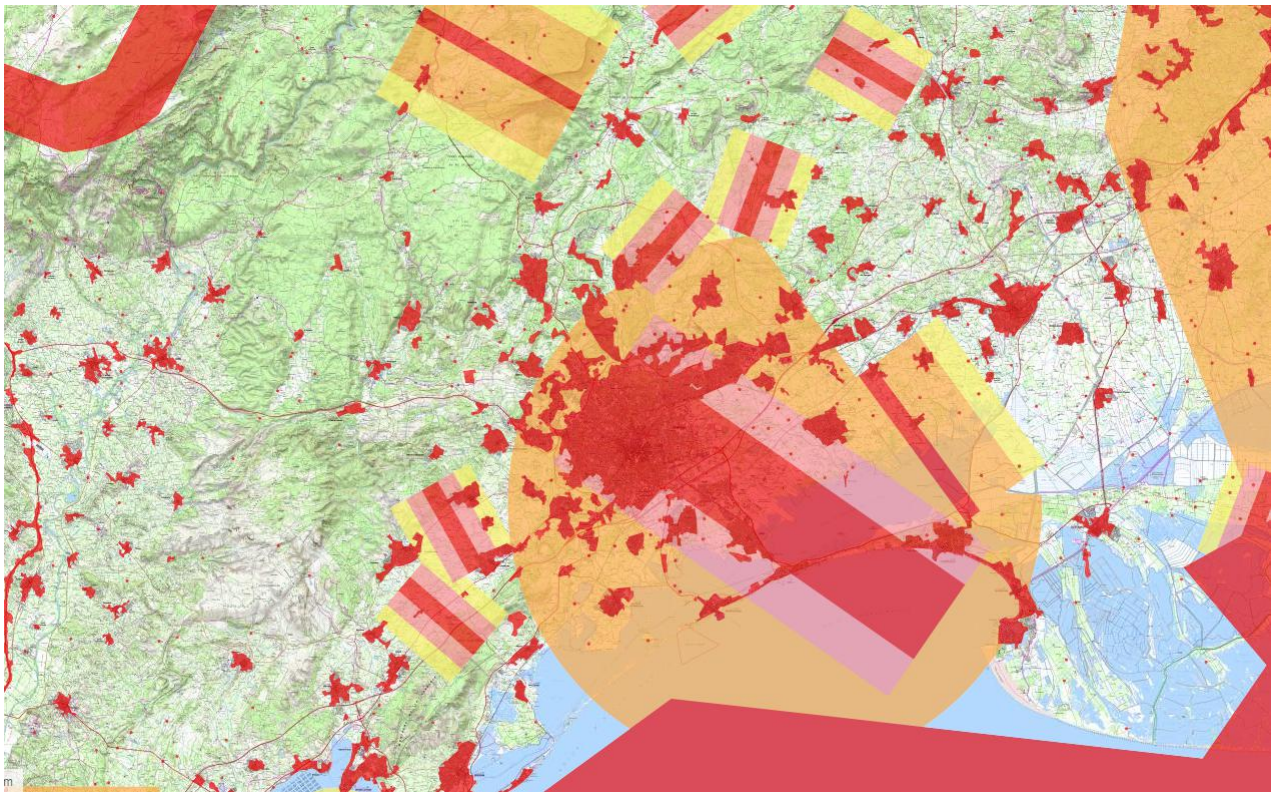


FIGURE 12 – Carte de restriction de vol pour drone de loisir (zone de Montpellier)

Fort heureusement, un membre de notre groupe avait accès à un terrain privé dans une zone autorisée grâce auquel nous avons pu mener à bien nos expérimentations. Nous avons également trouvé une structure qui nous permettrait d'analyser la qualité des clichés photographiques pris. Cette structure est un arbre de taille suffisante pour de premières expérimentations.

5.2 Résultats de l'algorithme à points de passage manuel

Nous avons donc expérimenté l'algorithme avec 8 points de passage pris manuellement. Le premier constat est que les photos obtenues ne sont pas optimales. Cela est explicable parle fait que la scène photographiée soit trop petite. En effet, cet algorithme utilise des

waypoints précis obtenus à partir de Google Maps, ce qui n'est pas adapté à une scène aussi petite car les points de passage sont censés être un minimum éloignés les uns des autres. Nous supposons que cet algorithme fonctionne bien mieux sur des structures plus volumineuses telles qu'une maison par exemple.

De plus, trouver le centre de la scène grâce aux waypoints peut être problématique. Dans les photos, on s'aperçoit que l'arbre n'est pas tout à fait le centre de la scène. Or, sur des scène de petites tailles, il est important que ce centre soit bien positionné afin d'avoir des photos bien orientées.

5.3 Résultats de l'algorithme de génération automatique des points de passage

Voici le résultats de l'exécution de l'algorithme de génération automatique des points de passage. Les paramètres entrés via l'application sont les suivant : 24 Waypoints, 5 mètres de distance, 1 mètre d'altitude, 1 tour. Ce qui nous donne comme résultats les 24 photographies suivantes :
Les 24 prises de vue sont équitablement réparties autour de l'objet. Cependant, on observe un léger décalage sur certaines images : le drone a photographié la zone derrière l'arbre. Cela est sûrement dû à une imprécision de la mesure de la distance entre le drone et l'arbre à l'initialisation ainsi qu'au rafales de vents pouvant perturber la prise des photographies.

Dans leur globalité, les photos sont de bonne qualité et représente bien la scène. On peut déduire que calculer les waypoints de manière automatique est bien adapté à de petites scènes, surtout si celle-ci est dégagée ainsi que circulaire.

6 Perspectives d'évolution

Ce projet touchant à sa fin, nous aurions aimé effectuer plusieurs améliorations.

Au niveau de l'application mobile, malgré le fait qu'elle soit complètement fonctionnelle, nous aurions aimé concevoir une meilleure ergonomie. Un utilisateur extérieur au projet aura peut-être du mal pour à la prendre en main. En effet, certains paramètres ne sont pas utilisés par le premier algorithme. Nous pourrions alors concevoir un véritable bouton "switch" qui adapterai l'affichage à l'algorithme courant.

Aussi, il serait pratique pour l'utilisateur de saisir directement des points de passage sur Google Maps (pour l'algorithme manuel).

En outre, nous avons finalement supprimé le retour caméra qui n'était plus réellement utile. Cependant, il aurait été appréciable d'avoir un bouton pour afficher ou cacher l'image générée par le drone. À cela s'ajoute un problème d'affichage sur l'écran de connexion qui ne s'actualise pas automatiquement quand le drone est connecté.

Au niveau algorithmique, il aurait été préférable de prendre en compte plusieurs points de passages et non seulement 2 pour générer le centre de la scène (essentiellement sur l'algorithme manuel). Ce changement aurait pour effet d'avoir une plus grande précision et donc de meilleures photos.

Enfin, l'algorithme de génération automatique du plan de vol ne peut que produire des trajectoires hélicoïdales, donc circulaires. Il pourrait être intéressant d'avoir une approche elliptique. De cette façon, le drone aurait une trajectoire plus optimisée par rapport à certains types de scènes .

7 Bilan et conclusion

Lors de ce semestre, notre projet ne s'est pas déroulé exactement comme nous l'avions prévu. Malgré tout, nous avons réussi a atteindre les objectifs que nous nous étions fixés.

Aucun d'entre nous n'étant familier avec l'outil Android Studio, nous avons du tout apprendre en parallèle de la recherche d'information et de la mise au point des méthodes de production de plan de vol. Nous avons également était retardé par certaines particularité du kit de développement d'application du drone fourni par DJI. Cependant, nous avons réussi à terminer le projet dans les temps grâce une bonne organisation, ainsi qu'une bonne répartition du travail entre les membres du groupe.

Dans un même temps, ce projet à été une aventure enrichissante sur le plan technique, et nous a permis d'acquérir de nombreuses connaissances. C'est notre premier projet "de terrain" qui nécessitait bien plus d'organisation que les autres projets que nous avions réalisé. Nous avons été heureux de participer a ce projet.

Enfin, nous avons pour regrets de n'avoir que quelques résultats représentés par les photographies de l'arbre. Nous aurions aimé avoir une meilleure cible à photographier, et également pouvoir donner ces résultats au groupes chargés de la reconstitutions de l'objet en 3D pour avoir un rendu complet de l'ensemble du projet. Nous tenterons d'atteindre cet objectif pour la soutenance.

8 Glossaire

Coordonnées GPS : Coordonnées géographique caractérisée par un système trois coordonnées : **Latitude, Longitude, Altitude**.

Latitude : Valeur angulaire, expression du positionnement Nord ou Sud d'un point sur Terre.

Longitude : Valeur angulaire, expression du positionnement Est ou d'un point sur Terre.

Altitude : Grandeur qui exprime un écart entre un point donnée et un niveau de référence.

Waypoints : Points de passage du drone, caractérisés par des coordonnées GPS ainsi qu'une suite d'action réalisé par le drone.

Azimut : L'azimut (parfois orthographié azimut) est l'angle dans le plan horizontal entre la direction d'un objet et une direction de référence. La direction de référence est en général le Nord.

Radian : Unité de mesure des angles plan (symbole : rad).

Mille marin : Unité de mesure de longueur utilisé en navigation marine et aérienne, il représente la mesure d'une minute de latitude de la Terre soit 1852 mètres $= \frac{1}{60} \times \frac{\pi}{180}$ rads.

Spline : Une fonction définie par morceaux par des polynômes.

Activité (Android Studio) : L'architecture générale des applications Android est multi-fenêtrée. Une activité désgine une fenêtre individuelle à l'intérieur d'une application.

9 Bibliographie

— https ://gitlab.com/ftornil/lucilia/blob/master/docs/2017-2018/rapports/final/1-recolte/rapport-final-1.pdf

— https ://fr.wikipedia.org/wiki/Coordonnées_géographiques

— http ://www.aip-drones.fr/restrictions-specifiques/more-31

— https ://www.geoportail.gouv.fr/donnees/restrictions-pour-drones-de-loisir

— https ://fr.wikipedia.org/wiki/Mille_marin

— https ://developer.dji.com/api-reference/android-api/Components/SDKManager/DJISDKManager.html

— https ://developer.dji.com/mobile-sdk/documentation/android-tutorials/index.html

— https ://fr.wikipedia.org/wiki/Azimut