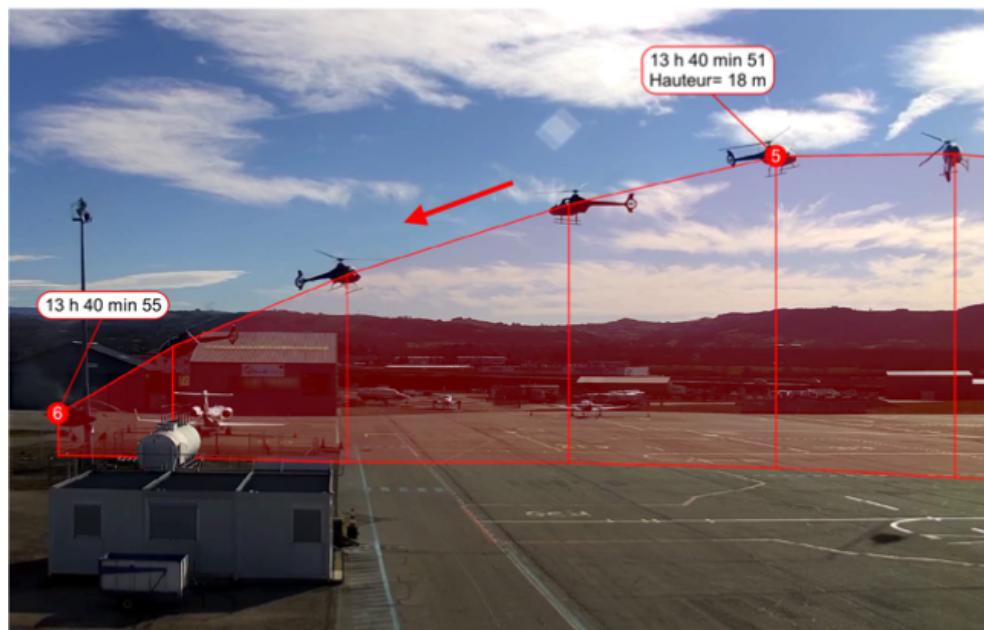


Projet Informatique

# Création d'une photoséquence à partir d'une trajectoire d'avion et d'une mise en place absolue de caméra



Jules Rosier  
Commanditaire : Jean-François Villeforceix

# Table des matières

<b>1</b>	<b>Contexte du projet</b>	<b>1</b>
<b>2</b>	<b>Objectifs de ce projet</b>	<b>2</b>
2.1	Objectifs . . . . .	2
2.2	Contraintes . . . . .	4
2.3	Recueil des besoins - Les acteurs . . . . .	4
<b>3</b>	<b>Analyse fonctionnelle - Solutions proposées</b>	<b>4</b>
<b>4</b>	<b>Réalisation technique</b>	<b>7</b>
4.1	Transformation des coordonnées . . . . .	8
4.1.1	Implémentation de la formule image . . . . .	9
4.2	Création de la photoséquence . . . . .	9
4.3	Ajout de l'avion sur la photoséquence . . . . .	10
<b>5</b>	<b>Réalisation et suivi de projet</b>	<b>12</b>
<b>6</b>	<b>Conclusion et perspectives</b>	<b>13</b>
6.1	Conclusion . . . . .	13
6.2	Perspectives . . . . .	14

## Tableau des figures

Numéro	Description	Page
1	Exemple de photoséquence d'un F22.	1
2	Trajectoire d'un aéronef dont on veut générer la photoséquence.	3
3	Photoséquence générée à partir de la trajectoire.	3
4	Diagramme des cas d'utilisation du projet.	4
5	Diagramme d'activité.	5
6	Diagramme de séquence du plugin.	6
7	Diagramme de classe du plugin.	7
9	Exemple de fichier de calibration au format XML MicMac.	10
14	Diagramme de Gantt du projet.	13

TABLE 1 – Tableau des figures du rapport

# 1 Contexte du projet

Le Bureau d'enquêtes et d'analyses pour la sécurité de l'aviation civile (BEA) est responsable des enquêtes de sécurité concernant les accidents et incidents graves impliquant des aéronefs civils français.

Dans le cadre de ces enquêtes, le BEA utilise la photogrammétrie pour reconstituer la trajectoire des aéronefs à partir d'enregistrements vidéo, tels que ceux provenant de caméras GoPro, de dispositifs de vidéosurveillance ou de smartphones. Ces trajectoires sont ensuite analysées en 2D dans le logiciel QGIS et en 3D sous forme d'animations. Cependant, la visualisation actuelle manque d'immersion réaliste, car elle ne reproduit pas précisément les conditions visuelles de l'événement. Ce projet vise à générer des photoséquences en réalité augmentée en intégrant la trajectoire aux images ayant servi à sa reconstitution. Une photoséquence est la décomposition d'un sujet en mouvement sur la même image, voir figure 1 :



FIGURE 1 – Exemple de photoséquence d'un F22

## 2 Objectifs de ce projet

### 2.1 Objectifs

Jusqu'à présent, les photoséquences étaient créées manuellement, image par image, à l'aide de logiciels de retouche comme Photoshop ou GIMP. L'objectif de ce projet est d'automatiser la création de ces photoséquences grâce à un outil capable de les générer directement à partir de données, notamment la trajectoire de vol d'un aéronef.

L'utilisateur devra fournir en entrée les données suivantes :

- Les images d'une vidéo d'accident ou des photographies aériennes du site. Si plusieurs photos sont fournies, chacune avec des positions différentes de l'avion, il faudra également inclure les masques binaires créés avec MicMac et détournant l'avion.
- Des fichiers XML issus de MicMac contenant les calibrations et orientations absolues.
- La trajectoire de vol et sa trace au sol, dont certains points clés sont numérotés et légendés spécifiquement.

L'outil devra ensuite générer en sortie :

- Une couche raster correspondant à l'image du site de l'accident.
- Plusieurs couches raster, représentant l'aéronef à différents instants de sa trajectoire.
- Une couche de points, avec les coordonnées calculées en pixels (à partir de l'orientation absolue de la caméra et de la trajectoire 3D), stylisée conformément à la charte graphique du BEA.
- Une couche vecteur, représentant la trajectoire de l'aéronef sous forme de polygones.

En concertation avec le commanditaire, il a été décidé que cet outil serait développé sous la forme d'un plugin QGIS avec une interface graphique. Ce plugin devra extraire les coordonnées des points des données en entrée, les transformer en coordonnées image (en utilisant les fichiers de calibration et d'orientation de la caméra), puis générer une couche de polygones reliant les points pour permettre une visualisation 3D de la trajectoire.

Par exemple, si l'utilisateur souhaite obtenir une photoséquence pour la trajectoire représentée à la Figure 2, l'outil générera la photoséquence illustrée à la Figure 3.

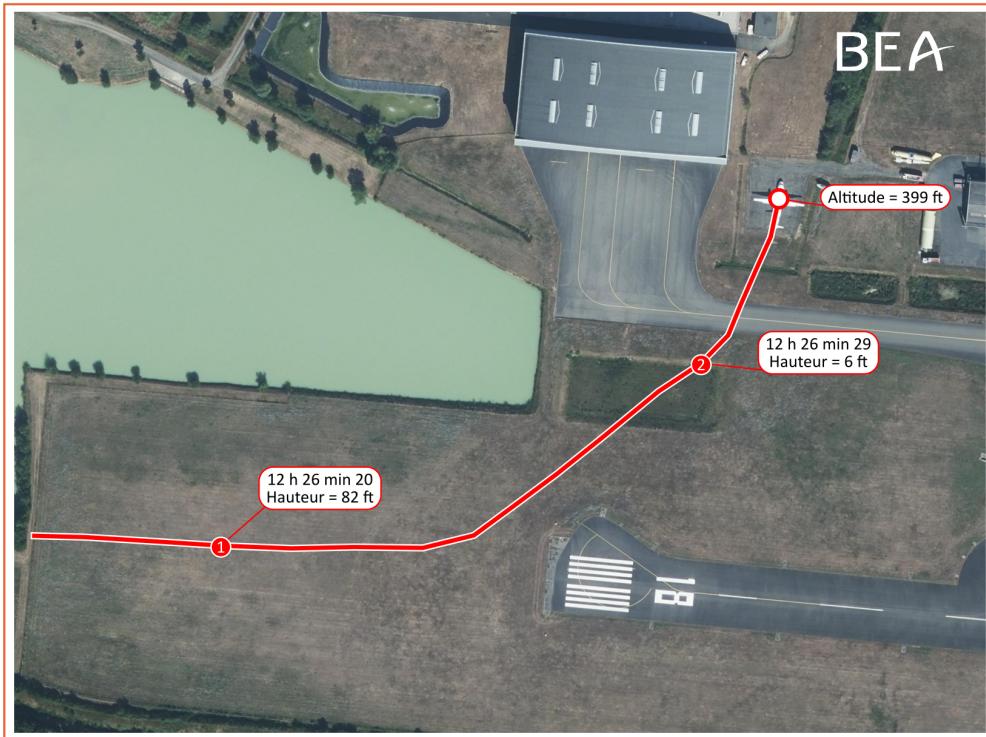


FIGURE 2 – Trajectoire d'un aéronef dont on veut générer la photoséquence.

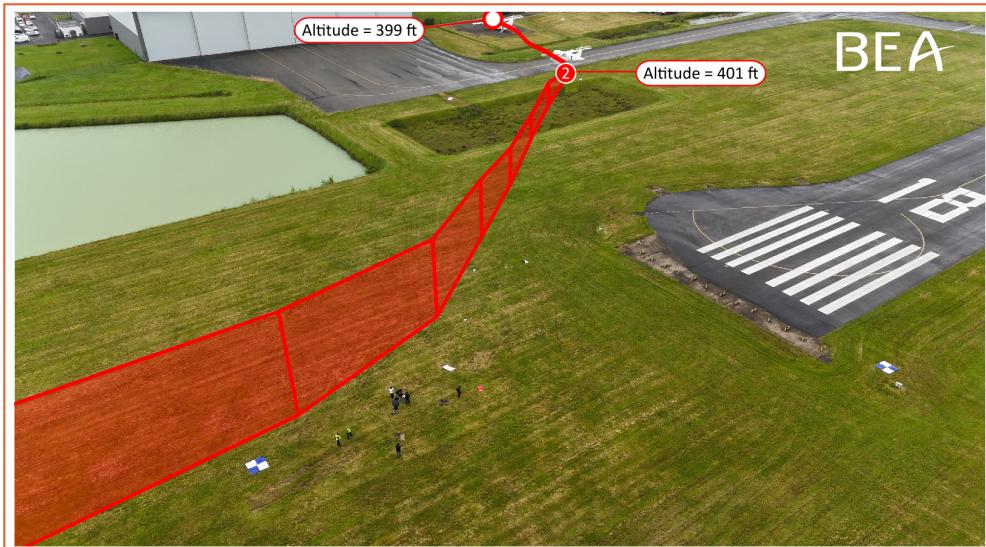


FIGURE 3 – Photoséquence générée à partir de la trajectoire.

## 2.2 Contraintes

Aucune contrainte spécifique n'a été imposée quant à la méthode de réalisation de l'outil. Toutefois, il est impératif qu'il dispose d'une interface graphique pour en faciliter la prise en main par un large éventail d'utilisateurs. Le langage Python ainsi que l'API pyQGIS devront être utilisés pour implémenter les fonctionnalités, en particulier les opérations liées à la manipulation des données géospatiales. De plus, l'utilisation de MicMac est conseillée pour les transformations de coordonnées.

## 2.3 Recueil des besoins - Les acteurs

Cet outil s'adresse principalement aux enquêteurs du BEA, qui réalisent actuellement leurs photoséquences à l'aide de logiciels de retouche comme Photoshop ou GIMP. Une interface simple et intuitive est donc essentielle pour permettre une prise en main rapide et efficace par ces utilisateurs.

La figure 4 illustre le diagramme des cas d'utilisation de ce projet. On y observe que l'utilisateur n'a qu'à fournir les données en entrée et lancer l'exécution du plugin. Toutes les étapes, telles que les transformations des coordonnées et la création des couches de points et de polygones, seront automatisées par l'outil.

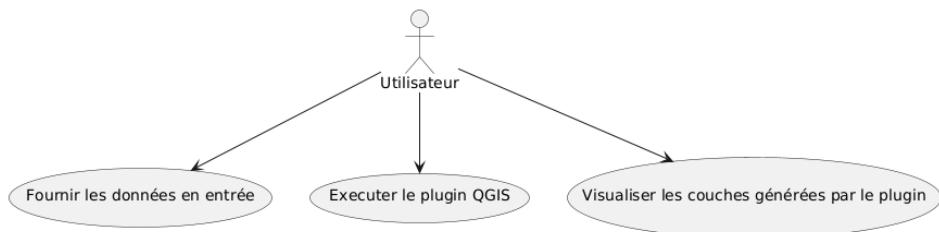


FIGURE 4 – Diagramme des cas d'utilisation du projet.

## 3 Analyse fonctionnelle - Solutions proposées

Le principal enjeu de ce projet réside dans la fusion de plusieurs technologies : l'API pyQGIS, l'API MicMac, la manipulation d'arborescences XML et l'interfaçage via PyQt. La figure 5 présente le diagramme d'activité du projet, illustrant les différentes étapes du fonctionnement du plugin.

Le plugin prend en entrée les données détaillées dans la section 2.1. Une fois exécuté, les étapes suivantes sont réalisées :

1. Extraction des données fournies en entrée.
2. Analyse des fichiers XML à l'aide de scripts de parsing, fournis par le commanditaire.
3. Transformation des coordonnées du repère terrain vers le repère caméra, en utilisant l'API MicMac.
4. Génération des couches de données grâce à l'API PyQGIS.
5. Affichage des couches de données dans l'interface QGIS.

Ce processus permet d'automatiser et d'optimiser l'intégration et la visualisation des données, répondant ainsi aux besoins définis par le commanditaire et les utilisateurs finaux.

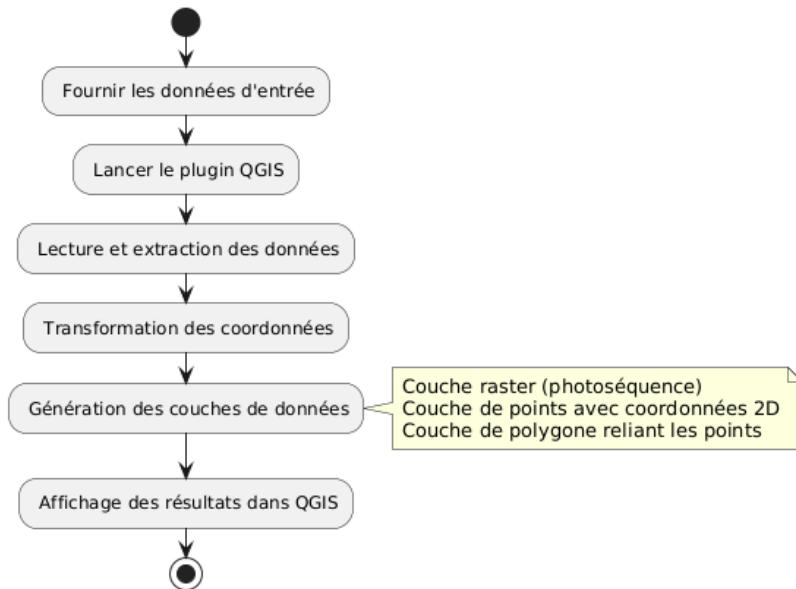


FIGURE 5 – Diagramme d'activité

Ce diagramme d'activité est complété par le diagramme de séquence (figure 6). Ce dernier illustre l'intégration des différentes API dans le projet. L'objectif principal est d'intégrer une méthode de transformations des coordonnées à un plugin QGIS.

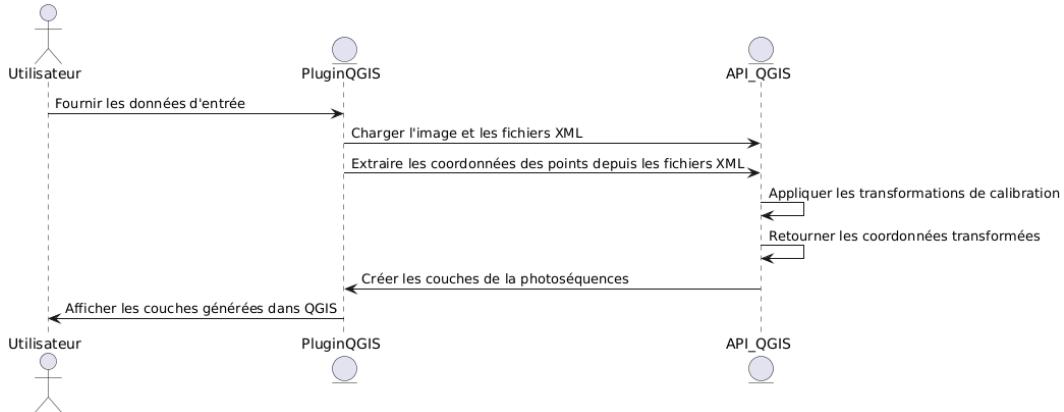


FIGURE 6 – Diagramme de séquence du plugin.

Enfin, la figure 7 présente le diagramme de classe du plugin. Les classes **ClassFactory** et **PluginDialog** font partie de la structure standard d'un plugin QGIS. La classe **Plugin** constitue le cœur du projet, où toutes les méthodes principales seront implémentées.

Les méthodes présentées dans le diagramme sont indicatives et pourraient être modifiées ou réparties en plusieurs fonctions selon les besoins. Elles décrivent cependant l'ensemble des fonctionnalités que le plugin devra prendre en charge.

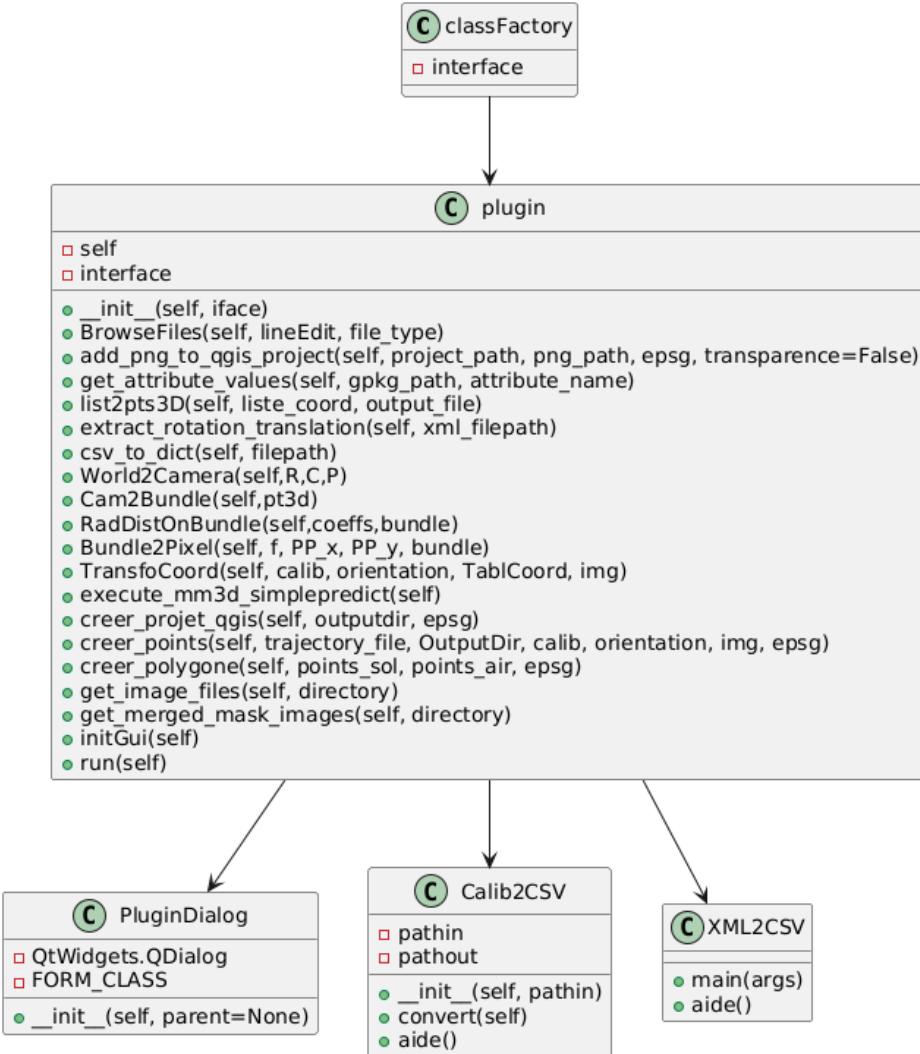


FIGURE 7 – Diagramme de classe du plugin.

## 4 Réalisation technique

L'outil développé prend la forme d'un plugin QGIS, ce qui justifie le choix du langage de programmation Python. Ce dernier permet un accès pratique à l'API QGIS via la bibliothèque Python PyQGIS. Les principales bibliothèques utilisées seront donc :

- PyQGIS, l'API Python de QGIS, qui sera exploitée pour réaliser divers traitements, notamment la création de polygones représentant la trajectoire.
- PyQt, qui sera utilisée pour concevoir l'interface graphique du plugin.

## 4.1 Transformation des coordonnées

Lors de la phase d'analyse du projet, il avait été convenu d'utiliser MicMac pour effectuer la transformation de coordonnées. En effet, une commande de MicMac permet de réaliser cette transformation : la commande `mm3d SimplePredict`. Cette commande permet de convertir des coordonnées d'un repère terrain à un repère image. Pour cela, elle nécessite une image, son fichier d'orientation, ainsi qu'un fichier `xml` au format MicMac contenant les points dont les coordonnées doivent être transformées. En sortie, la commande génère un nouveau fichier `xml` contenant les coordonnées images des points terrain fournis.

En théorie, cette fonction semble idéale. Cependant, elle ne retourne que les coordonnées images des points présents dans le champ de l'image. Or, pour créer les polygones représentant la trajectoire de l'aéronef, il est crucial d'obtenir également les coordonnées du point voisin du dernier point de l'image. Ce point ainsi que sa projection au sol, bien qu'en dehors de l'image, sont indispensables pour construire le dernier polygone de la trajectoire. Sur la Figure 8) on peut remarquer à gauche de l'image que le dernier polygone dépasse des dimensions de l'image.



FIGURE 8 – Couche de polygone générée par le plugin.

Par ailleurs, une modification de la fonction `SimplePredict` de MicMac est envisagée afin d'y inclure un argument de type booléen permettant d'enregistrer toutes les coordonnées, y compris celles situées en dehors des limites de l'image. Pour anticiper l'intégration de cette nouvelle fonctionnalité, un bouton a été implémenté dans le plugin, offrant à l'utilisateur la possibilité de choisir la méthode de transformation des coordonnées : soit la méthode classique basée sur les calculs matriciels, soit celle utilisant MicMac. Il sera cependant nécessaire que l'utilisateur dispose d'une version de MicMac intégrant cette modification pour pouvoir utiliser cette dernière option.

Il a été décidé, en accord avec le commanditaire, de se concentrer en priorité sur

l'implémentation complète de toutes les fonctionnalités, afin de disposer d'un plugin entièrement fonctionnel avant d'intégrer la méthode de calcul basée sur MicMac. La plupart des fonctions nécessaires à cet objectif ont déjà été programmées et sont présentes dans le script du plugin. Le développement d'une nouvelle fonctionnalité dans MicMac requiert des tests plus approfondis car les modifications peuvent affecter d'autres utilisateurs extérieurs au projet.

Pour l'instant, le bouton permettant de sélectionner l'algorithme MicMac a simplement été désactivé.

Ainsi, il a été décidé d'utiliser une autre méthode pour transformer les coordonnées, reposant sur la formule image avec l'application des distorsions. Cette méthode s'appuie sur des calculs matriciels utilisant les données issues des fichiers de calibration et d'orientation.

#### 4.1.1 Implémentation de la formule image

Afin d'implémenter la formule image, il est nécessaire de récupérer les différents coefficients utiles pour les calculs, tels que les coefficients de distorsion radiale, la focale de la caméra, les coordonnées du point principal, etc.

Pour extraire ces informations, j'ai développé une classe Python qui convertit un fichier XML de calibration interne de caméra en un fichier CSV, en extrayant les paramètres de calibration mentionnés précédemment.

L'utilisateur devra toujours fournir les fichiers de calibration MicMac suivants pour permettre ces transformations :

- Un fichier au format XML pour la calibration de la caméra.
- Un fichier au format XML contenant l'orientation absolue de ou des image(s).

La figure 9 est un exemple de fichier de calibration au format XML MicMac, dans lequel on peut retrouver, entre autres, les coefficients de distorsion ainsi que les coefficients de distorsion inverse.

Cette méthode sera appliquée aux coordonnées de l'avion dans les airs ainsi qu'aux coordonnées de la trace au sol de l'avion.

## 4.2 Création de la photoséquence

L'API Python de QGIS a permis de gérer la création des couches vectorielles de la photoséquence.

```

<?xml version="1.0" ?>
<ExportAPERO>
    <CalibrationInternConique>
        <KnownConv>eConvApero_DistM2C</KnownConv>
        <PP>2028.50466514128448 1140.74474362282126</PP>
        <F>2933.88223404226301</F>
        <SzIm>4032 2268</SzIm>
        <CalibDistortion>
            <ModRad>
                <CDist>2014.3147171287419 1102.87590834279854</CDist>
                <CoeffDist>1.35929433093155689e-08</CoeffDist>
                <CoeffDist>-3.09597171001779215e-15</CoeffDist>
                <CoeffDist>2.97222521246652611e-22</CoeffDist>
                <CoeffDistInv>-1.34192770799197935e-08</CoeffDistInv>
                <CoeffDistInv>3.28270356748486257e-15</CoeffDistInv>
                <CoeffDistInv>-3.83099238257061114e-22</CoeffDistInv>
                <CoeffDistInv>1.12820532037850351e-29</CoeffDistInv>
            </ModRad>
        </CalibDistortion>
    </CalibrationInternConique>
</ExportAPERO>

```

FIGURE 9 – Exemple de fichier de calibration au format XML MicMac

Les polygones représentant la trajectoire finale de la photoséquence (illustrée par le polygone rouge dans la figure 3) seront générés à partir des coordonnées de l'aéronef en vol et de leur projection au sol. Les points utilisés pour la création des polygones sont uniquement ceux situés dans l'image, ainsi que le premier point précédant le dernier point visible de l'image, afin d'éviter la création de polygones inutiles en dehors du champ de l'image.

De plus, une couche de points, possédant le style du BEA, est créée, permettant de visualiser certaines informations. Ainsi, la seule chose qui manque à la photoséquence est d'afficher l'avion à plusieurs moments de sa trajectoire.

### 4.3 Ajout de l'avion sur la photoséquence

Lorsque l'utilisateur fournit les données, il doit fournir un dossier contenant des images d'une photoséquence. Ce dossier contient, en plus des images tirées d'une vidéo de la photoséquence, des masques MicMac associés à chaque image. Chacun de ces masques permet de "détourer" l'aéronef pour l'ajouter à la photoséquence.

Parmi les données fournies, on retrouvera l'image du site de l'accident sans l'aéronef, qui servira de couche raster de base pour notre photoséquence, comme illustré dans la Figure 10. De plus, d'autres photos seront disponibles à différents instants. Sur ces

autres photos, l'aéronef sera positionné différemment et sera accompagné de masques MicMac indiquant la position de l'avion, permettant ainsi de l'isoler, comme montré dans la Figure 11. À l'aide de la bibliothèque Python OpenCV, une méthode a été développée pour permettre au plugin de fusionner le masque MicMac avec son image associée, produisant ainsi une image ne contenant que l'avion, comme illustré dans la Figure 12.



FIGURE 10 – Image du site sans aéronef

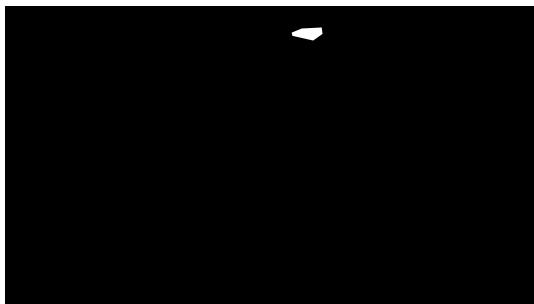


FIGURE 11 – Exemple de masque Mic-Mac



FIGURE 12 – Fusion entre le masque et l'image

Les images issues de la fusion seront ajoutées à la photoséquence. Il suffit de rendre transparents les pixels noirs à l'aide de PyQGIS pour que l'avion apparaisse à différentes positions sur la photoséquence. Les deux couches contenant uniquement les avions seront superposées à l'image du site sans aéronef. La Figure 13 illustre l'avion à deux positions différentes superposé à l'image sans aéronef.



FIGURE 13 – Superposition des couches d'avions à l'image de base

## 5 Réalisation et suivi de projet

La figure 14 illustre l'organisation et la répartition des tâches de ce projet. Quatre étapes principales peuvent être identifiées :

1. **Prise en main des outils** : Compréhension et familiarisation avec les différentes API et bibliothèques utilisées.
2. **Création de l'interface utilisateur** : Conception de l'interface du plugin à l'aide de PyQt.
3. **Transformation des coordonnées** : Implémentation de la méthode de transformation de coordonnées
4. **Création de la photoséquence** : Génération et visualisation de la photoséquence dans QGIS à l'aide de PyQGIS.

Si ces étapes principales sont achevées rapidement, le plugin pourra être enrichi avec des fonctionnalités supplémentaires suggérées par le commanditaire.

Le projet sera jalonné par des réunions régulières avec le commanditaire pour valider les orientations prises et s'assurer qu'elles correspondent à ses attentes. Une visite au BEA est également prévue courant janvier pour rencontrer le commanditaire et mieux comprendre les besoins de l'organisme.

Séance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Prise en main de PyQGIS et de Qt															
Prise en main de MMVII															
Création de l'interface du plugin															
Programmation des transformations de coordonnées															
Programmation de la création de la photoséquence															
Ajout de nouvelles fonctionnalités selon l'avancée du projet															
retour commanditaire															
Visite du BEA															
Rédaction du rapport															
Préparation soutenance															

FIGURE 14 – Diagramme de Gantt du projet

## 6 Conclusion et perspectives

### 6.1 Conclusion

L'objectif initial du projet était de créer un outil robuste et cohérent permettant d'automatiser la génération de photoséquences. Dans un premier temps, il avait été décidé d'utiliser MicMac avec la commande `mm3d SimplePredict` pour transformer les coordonnées terrain en coordonnées image. Cependant, cette commande ne permettant pas d'exporter les coordonnées en dehors des limites de l'image, il a été décidé d'utiliser la formule image avec application des distorsions pour effectuer la transformation des coordonnées.

L'API PyQGIS permet, quant à elle, de générer toutes les couches nécessaires à la création de la photoséquence. Cet outil prend la forme d'un plugin QGIS, permettant ainsi à l'utilisateur de visualiser la photoséquence directement dans QGIS. Actuellement, le plugin est entièrement fonctionnel et permet de générer une photoséquence correctement. La Figure 3 illustre la mise en page des photoséquences générées par

l'outil pour les enquêteurs du BEA.

Enfin, il est important de noter que cet outil ne vise pas à créer la mise en page des photoséquences, un autre outil interne au BEA étant déjà prévu à cet effet.

## 6.2 Perspectives

Bien que l'outil actuel soit entièrement fonctionnel, plusieurs pistes d'améliorations peuvent encore être explorées. La première concerne l'ajout d'un nouvel algorithme de transformation de coordonnées : celui de MicMac avec la commande `SimplePredict`. La plupart des méthodes étant déjà implémentées, il suffirait de modifier la fonction dans le code source de MicMac. Cette modification permettrait d'ajouter un argument facultatif indiquant si la fonction doit également exporter les coordonnées en dehors des limites de l'image.

De plus, la création de la couche de points présente un temps d'exécution qui semble assez long (environ 5 secondes) par rapport à celui de la couche de polygones (moins d'une seconde). Il pourrait donc être utile de réduire la complexité algorithmique de la méthode utilisée pour créer les points.