ENSG Géomatique

ÉCOLE NATIONALE
DES SCIENCES
GÉOGRAPHIQUES

BEA

Bureau d'Enquêtes et d'Analyses
pour la sécurité de l'aviation civile

Développement d'un outil de tracking d'objets dans des vidéos

Rapport d'analyse

Commanditaires : Jean-François VILLEFORCEIX et Nicolas HOLVOET
Bureau d'Enquêtes et d'Analyses

Hugo Tardy

Décembre 2020

Table des matières

1 Introduction

1.1 Contexte et acteurs du projet

Le Bureau d'Enquêtes et d'Analyse pour la sécurité civile est un organisme chargé des enquêtes des accidents de l'aviation civile. Pour déterminer les circonstances précises de certains accidents impliquant des prises de vue vidéo, le BEA utilise Micmac un logiciel de photogrammétrie open source développé par l'IGN pour reconstruire la trajectoire et certains paramètres de vol de l'avion.

Enquêteurs de sécurité au Bureau d'Enquêtes et d'Analyse, Jean-François Villeforceix et Nicolas Holvoet sont les commanditaires du projet et représentent aussi l'utilisateur final de l'application.

1.2 Problématique

Les vidéos utilisées par le BEA dans le cadre de ses enquêtes présentent souvent des élèments ininteressants voire gênants appelés masques qu'il convient d'exclure dans les calculs de MicMac. Ces masques sont mobiles et doivent aujourd'hui être saisis à la main ce qui est très chronophage.

Jean-François Villeforceix et Nicolas Holvoet souhaitent à travers ce projet trouver une solution permettant d'automatiser la création de masques dans des vidéos d'accidents. Ceci permettrait d'accélérer la phase de pré-traitement et donc l'enquête tout en libérant des ressources humaines pour d'autres tâches.

2 Objectifs de l'étude

Le projet vise à développer un outil de tracking vidéo permettant de construire automatiquement sur une vidéo ou une pile d'images renseignée par l'utilisateur des masques. Ces masques sont mobiles

2.1 Besoin du commanditaire

Le besoin du commanditaire peut se schématiser de la manière suivante :

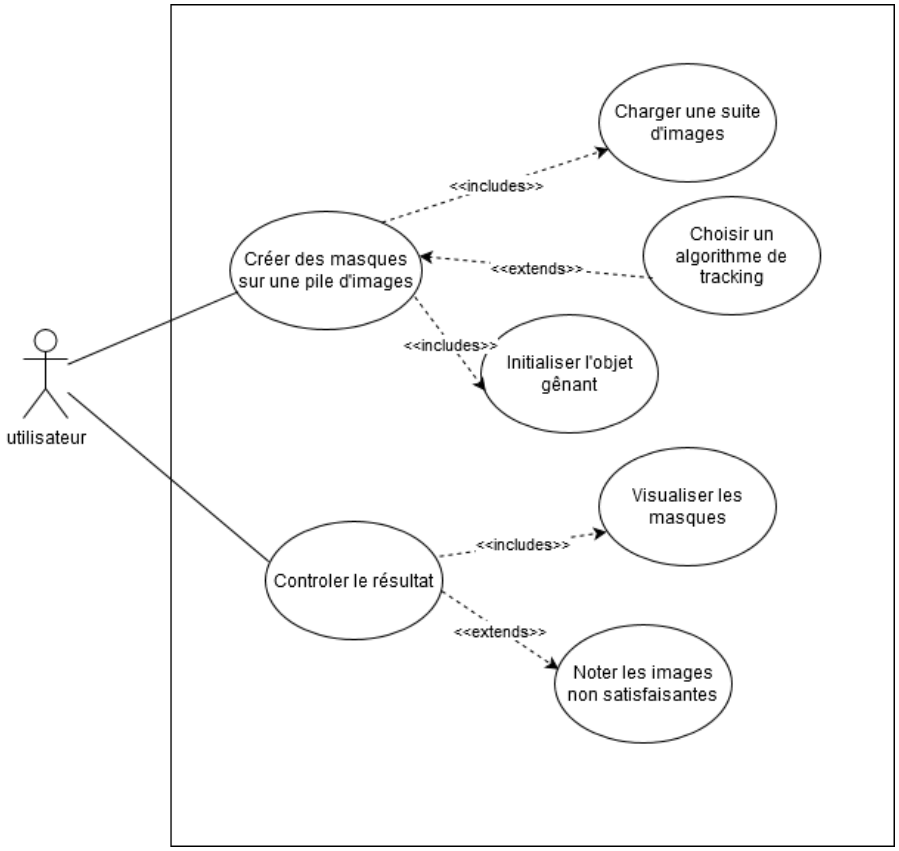

FIGURE 1: Diagramme d'utilisation

L'utilisateur doit avoir à disposition une pile d'images ou une vidéo de la séquence qui l'intéresse. Si plusieurs méthodes sont implémentées, celui ci peut alors choisir une méthode de calcul selon son besoin et les contraintes de la vidéo (occlusion, mouvements rapides, changements de luminosité...). Avant de lancer le calcul, l'utilisateur doit initialiser le calcul en indiquant la position ou le masque correspondant à l'objet gênant sur la ou les premières images de la séquence.

Il est à prévoir que l'application ne soit pas parfaite dans ses prédictions, un outil de controle permettant de visualiser facilement le résultat des masques est donc à prévoir dans l'application. Le système de contrôle le plus intuitif est de pouvoir visualiser la série d'images avec les masques en transparence par dessus.

2.2 Données

Les données de travail sont des vidéos ou séquences d'images. Ces données sont confidentielles et seront fournies par le BEA via LinShare, une plateforme de partage. La problématique est centrée sur les masques mobiles, il s'agit donc majoritairement de prises de vue par des caméras de type action cam comme des GoPro, ou potentiellement de prises de vue fixe présentant un sujet gênant très mobile.

Les objets à détecter sont de nature variées, il peut s'agir de la tête, de la main ou du pied de la personne en vol, d'objets volants (autres avions, parachutes,..), d'une partie du cockpit et des ailes. Les masques peuvent donc concerner des objets qui se déplacent dans le champ de vision mais ausi des objets qui ne sont présents que sur les bords des images, ponctuellement et non visibles en entier, ce qui peut compliquer détection et création de masques pertinents.

2.3 Micmac

Micmac est un logiciel de photogrammétrie développé par l'IGN et l'ENSG. Les masques utilisés par Micmac suivent une nomenclature précise. La création d'un masque sur un fichier "photo_id.jpg" aboutit à la création de deux fichiers :

— photo_id_Masq.tif : une image binaire représentant visuellement le masque

— photo_id_Masq.xml : un fichier XML de métadonnées

3 Analyse fonctionnelle

L'aspect technique majeur du projet réside dans l'algorithme utilisé pour réaliser les masques. Cet algorithme doit potentiellement aller plus loin que la détection de l'objet et pouvoir déterminer sa forme sur une suite d'images.

Parmi les outils "clés en main" répondant à cette problématique, on peut citer les réseaux de neurones réalisant une segmentation et une classification automatique des objets selon des classes précises.

FIGURE 2: Exemple de segmentation sémantique par deeplearning
Source : ICNet for Real-Time Semantic Segmentation on High-Resolution Images - Youtube

Cette solution nécessite cependant d'entrainer le réseau. Dans notre cas, les objets étant très variés, il n'est pas concevable de réaliser un réseau polyvalent qui nécessiterait un très grand nombre de données ni d'espérer utiliser un extrait de la vidéo que l'on souhaite traiter comme entrainement, qui demanderait trop de travail humain.

3.1 Tracking

Dans le cas d'objets de petite taille, un masque grossier entraine peu de pertes d'informations sur l'image. La forme du masque n'étant pas déterminante dans ce cas de figure, il suffit de connaitre la position de l'objet, sa taille approximative et de suivre sa position.

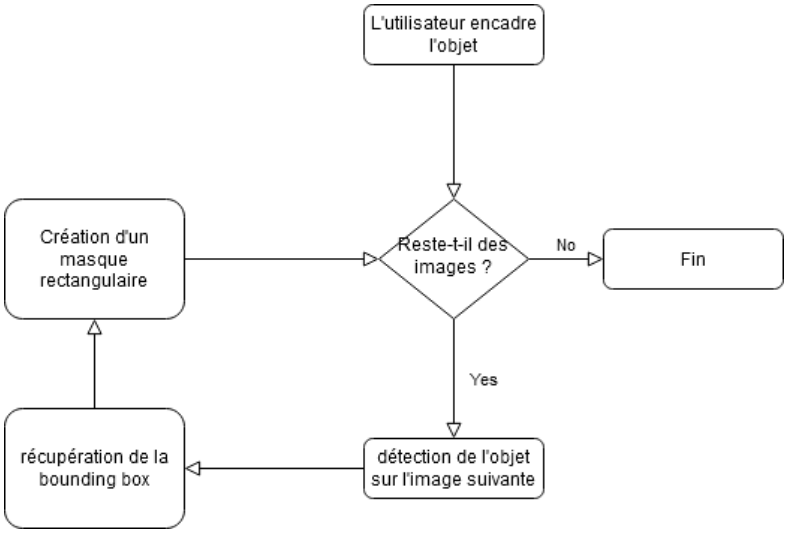

FIGURE 3: Schéma de traitement des petits objets

Pour suivre un objet, plusieurs algorithmes dits de tracking existent, la bibliothèque OpenCV en intègre 8 à ce jour. Ces algorithmes se différencient par des critères tels que la vitesse d'éxécution, la précision, et leur robustesse :

- Boosting Tracker
- MIL Tracker
- KCF Tracker
- CSRT Tracker
- MedianFlow Tracke
- TLD Tracker
- MOSSE Tracker
- GOTURN Tracker

Les objets suivis présentent des caractéristiques importantes pour le choix d'un outil de tracking. En effet les conditions d'acquisition peuvent varier et ces objets sont susceptibles d'être cachés ou de sortir de l'emprise de la caméra (occlusion), de bouger plus ou moins rapidement (mouvements brusques) ou de subir un changement de luminosité important.

La vitesse d'éxécution est intéressante dans le cas d'applications de détection d'objet en temps réel mais dans le cadre de notre projet elle est secondaire et la priorité est d'assurer la précision et la robustesse du suivi.

Table 4. Strengths and weaknesses of all eight trackers on 360-degree videos.

Features / Trackers	Principle	Strength	Weakness	Improvement Suggestion
GOTURN	Pretrained CNN model	Recovery from failure and occlusion	Target not in training data	Include specific targets for training in advance
MedianFlow	Min forward–backward error	Reliable on slow changing target	Fast-moving target	Support motion detection
TLD	Track, learn, and detect	Recovery from failure and occlusion	High false alarm	Combine with reliable filter
KCF	Kernelized correlation filter	Report tracking failure	Fixed target size	Adaptable target size
BOOST	AdaBoost	Decent accuracy	Seldom report tracking failure	Adaptable tolerance
CSRT	Discriminative correlation filter	Robust and high accuracy	Long-term occlusion	Failure recovery with spatial relationship
MIL	Multi-instance learning	High accuracy	Long-term occlusion	Failure recovery with spatial relationship
MOSSE	Min square error	High tracking speed	Fixed target size	Adaptable target size

FIGURE 4: Tableau de comparaison des algorithmes de tracking pour des vidéos à 360°
Source : Comparison of Tracking Techniques on 360-Degree Videos By Tzu-Wei Mi and Mau-Tsuen Yang, 14 August 2019

CSRT se montre le plus prometteurd'après l'étude de la 4 car plus précis et robuste, en plus de permettre de changer la taille de la fenête de suivi dynamiquement, bien qu'il soit parfois mis en échec par les occlusions de longue durée.

3.2 Segmentation

Dans le cas d'objets plus gros pour lesquels on ne peut pas se permettre de réaliser un masque rectangulaire sans perdre trop d'informations, on utilise une segmentation pour construire le masque. Pour déterminer les régions appartenant au masque, on se base sur le masque de l'image précédente récursivement basée sur la précédente, la première utilisant le masque défini par l'utilisateur

En utilisant un algorithme de détection et de description des points clés du masque, on peut établir la position de l'objet dans l'image suivante. Une segmentation adaptée réalisée en parallèle sur l'image suivante nous permet de dire si chaque région appartient au masque ou non en utilisant les descripteurs.

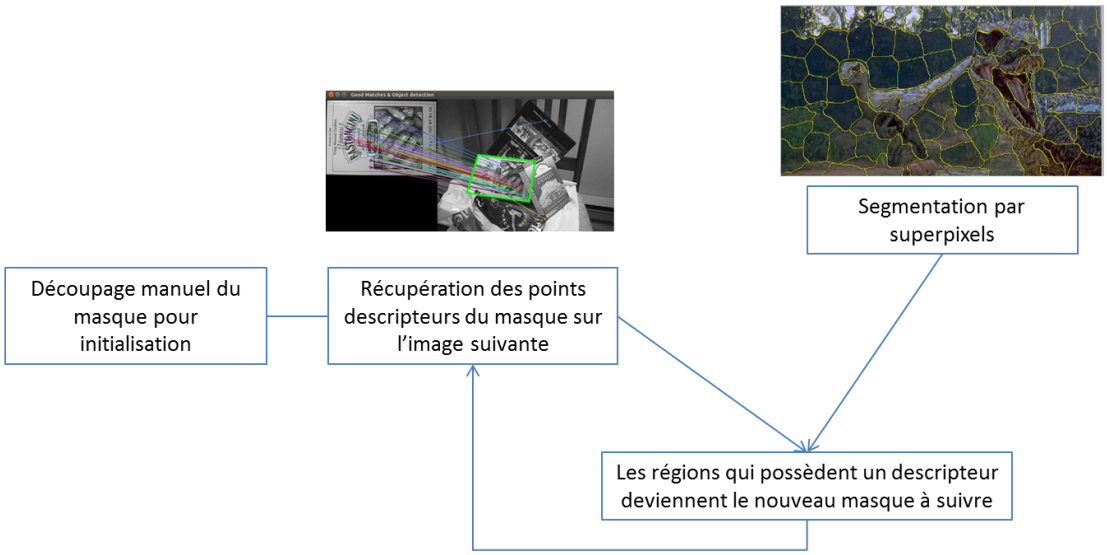

FIGURE 5: Caption

Le choix de la segmentation s'est porté sur la méthodes des superpixels qui a l'avantage de sursegmenter. Ceci permet de minimiser les régions ambigues qui comporteraient des élèments d'un objet à masquer et du reste de l'image non masquée. Ces objets gênants massifs seront probablement la principale difficulté du projet.

3.3 Suivi multiple d'objets

Plusieurs objets gênants peuvent apparaitre en même temps sur les vidéos. Selon les objets, l'algorithme peut fonctionner pour un élèment de la vidéo et échouer sur un autre élèment. L'application doit donc permettre de pouvoir "reprendre" un objet spécifique sans pour autant refaire les autres suivis considérés comme satisfaisant. Pour cela, on assigne à chaque objet un nom unique et une série de masques sera créé pour chaque objet. Une fois satisfaisant, les masques seront fusionnés en un seul masque.

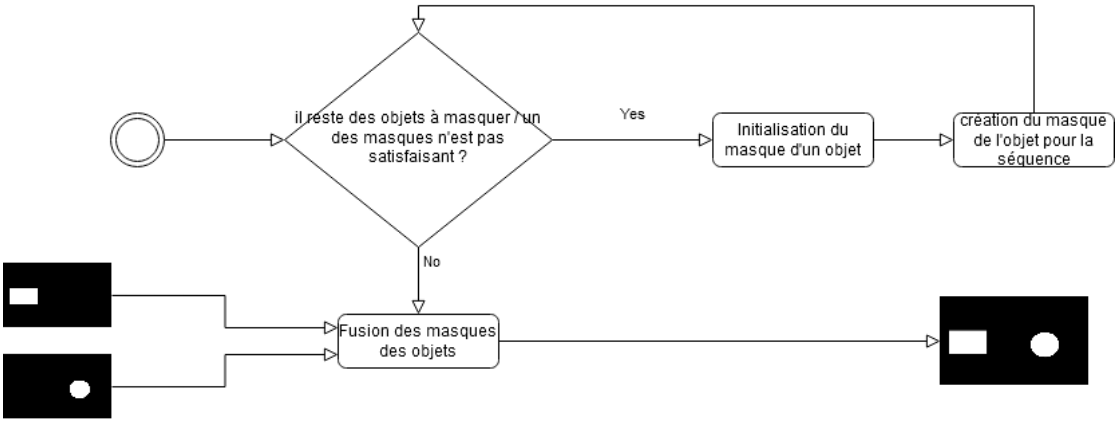

FIGURE 6: Caption

3.4 Visualisation du résultat

Afin de visualiser les masques et de s'assurer de leur pertinence, un outil simple doit être mis en place pour l'utilisateur. Pour cela, les frames seront enregistrées dans un dossier de vérification avec le masque en transparence et un identifiant affiché sur chaque masque. Une reconstruction de la vidéo à partir des frames masquées sera créée.

3.5 Architecture

Chaque vidéo possède un nombre potentiellement important de frames, qui peuvent elles mêmes contenir différents objets. Afin de gérer cette multiplicité, les objets (associés à une vidéo) constituent une classe à part entière dans laquelle est contenue la liste des masques de l'objet en question sur les frames de la vidéo. La fonction *Mask* de la classe *Video* prend en entrée la frame où l'utilisateur saisira manuellement le masque, les frames qu'il souhaite traiter, et enfin l'objet s'il s'agit de corriger un masque existant qui n'est pas satisfaisant.

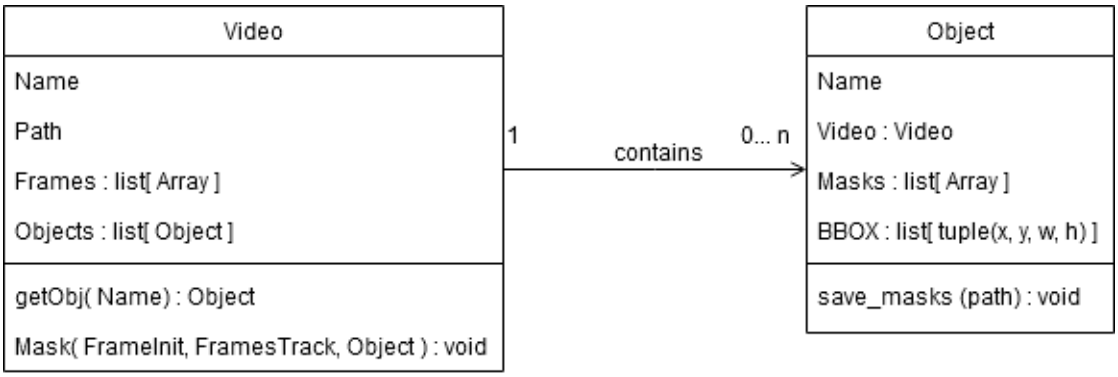

FIGURE 7: Diagramme de classe

4 Etude technique

4.1 Python

Le projet sera réalisé avec le language Python en version 3 avec l'environnement de développement Pycharm.

4.2 OpenCV

OpenCV est une librairie python spécialisée dans la vision par ordinateur. Certains algorithmes tels que SURF utilisés pour la récupération de points descripteurs sont sous licence dans les versions au delà de la 3.4.2. La bibliothèque OpenCV sera donc utilisée en version 3.4.2.

4.3 Gitlab

Le suivi du projet se fera sur Gitlab, et se fera sur un Git partagé par les commanditaires dans le but d'assurer un suivi du projet.

4.4 LaTeX

Les rapports et comptes rendus du projet sont rédigés avec LaTeX qui est un langage et un système de composition de documents. L'éditeur utilisé est Overleaf.

5 Organisation du projet

5.1 livrables attendus

Les livrables attendus par le commanditaire sont :

— Un code fonctionnel et commenté

— Une documentation utilisateur

— Une documentation développeur pour faciliter la prise en main du code

5.2 GANTT

Le planning prévisionnel est réprésenté dans le GANTT. La mise en place de l'application et du tracking ainsi que la visualisation des résultats sont réalisés en premier. Le traitement des gros objets sur les séquences vidéo se fera dans un deuxième temps car les résultats temps d'implémentation sont plus difficiles à prévoir. Un temps est réservé à l'écriture de commentaires dans le code ainsi qu'à la rédaction des documentations et rapports en fin de projet.

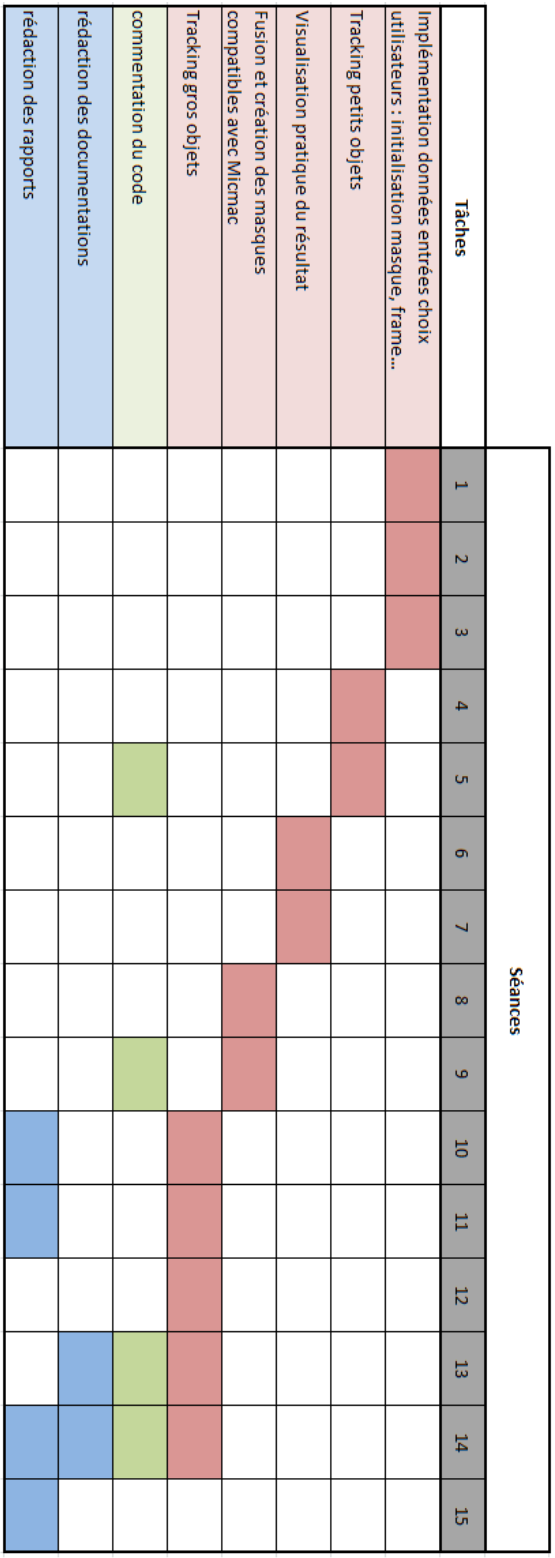

FIGURE 8: GANTT