



SECTION PHYSIQUE

TRAVAUX PRATIQUES 3

Détecteur d'objets orbitaux par caméra optique

Prototype de démonstration au sein du projet SSA

Manon BÉCHAZ
Baptiste CLAUDON
Jules ESCHBACH
Salomon GUINCHARD

2021

Résumé

Avec l'augmentation exponentielle des objets en orbite, la mise en place de méthodes de recensement et de catalogues est aujourd'hui essentielle. Ce travail s'attelle à démontrer la possibilité d'utiliser du matériel optique accessible au grand public pour la détection, depuis le sol, de ces objets en orbite. Trois montages avec différents matériels ont été testés, et se sont avérés pour deux d'entre eux à mêmes de détecter efficacement des traces de satellites. En parallèle, des algorithmes ont été développés pour traiter les images obtenues et en détecter automatiquement les traces, permettant le calcul automatique d'orbites et donc la mise en place d'un catalogue. Le traitement total des images s'est avéré à même de détecter correctement 22% des centres des traces observables, permettant d'obtenir 11 orbites sur les 117 images considérées pour l'étude statistique.

Table des matières

Introduction	3
1 Prototypes et obtention des images	4
1.1 Caméra	4
1.1.1 Caméras testées	4
1.1.2 Exemples d'images obtenues	5
1.1.3 Discussion et comparaison des caméras	5
1.2 Capteurs	6
1.3 Prototype complet	7
2 Traitement des images et obtention des orbites	7
2.1 Méthode	7
2.1.1 Traitement initial des images	7
2.1.2 Détection des lignes	8
2.1.3 Détection du centre des traces et de la position du satellite	9
2.1.4 Traces correspondant au même satellite	10
2.1.5 Détermination des orbites	10
2.2 Résultats et Discussion	10
2.2.1 Exemples de détection des traces et de leur centre	10
2.2.2 Statistiques	15
2.2.3 Importance du seuil de Hough et des paramètres empiriques τ et a	17
Conclusion	18
A Spécifications des caméras et objectifs utilisés	21
B Prototype final : Montage	22
B.1 Montage	22
B.2 Schéma de l'ensemble du processus de commande de l'appareil photo	23
C Schéma de l'ensemble du traitement des images	25

Introduction

Si la conquête spatiale a réussi un exploit considérable en permettant l'apparition entre autres de nouveaux moyens de communication, cette dernière s'est accompagnée d'une pollution progressive et invisible des alentours terrestres. Depuis le début de la conquête spatiale en 1957, le nombre d'objets en orbite autour de la Terre ne fait ainsi qu'augmenter et ce sont aujourd'hui des milliers de débris spatiaux de quelques centimètres de diamètre à plusieurs mètres de large qui gravitent autour de notre planète [1]. Du fait de leur densité toujours plus importante, ces déchets constituent une menace grandissante aussi bien pour les satellites encore en fonction que pour les vols habités [2][3].

Si de nombreuses solutions sont en cours d'étude, force est de constater qu'aujourd'hui une des seules réponses à cette menace reste la prévision de la position des déchets et leur intégration aux calculs d'orbites des satellites, d'où la nécessité d'une classification complète et précise de ces derniers [4][5].

Par ailleurs, beaucoup de ces déchets (et notamment les plus problématiques de par leur taille) étant suffisamment importants pour être détectés depuis la Terre, la création de télescopes et d'algorithmes adaptés permettrait une première détection et classification des débris, depuis le sol [6]. Cependant, face à l'importance de la tâche de détection et classification (Environ 20'000 objets sont actuellement en orbite autour du globe [7]) associé au nombre limité d'observatoires pouvant être mis à contribution, la mise en place d'un catalogue complet s'avère de première approche complexe dans sa réalisation.

Pourtant, si de nombreux débris sont observables depuis le sol, parfois même à l'oeil nu, ne serait-il pas possible d'adapter les techniques astronomiques à du matériel grand public, accessible au plus grand nombre ? Si l'astrophotographie a su démontrer la possibilité d'observer de nombreux corps à l'aide d'un simple appareil photo, il semble dès lors envisageable d'utiliser des méthodes similaires dans le but de détecter les débris. L'idée serait alors de permettre par exemple la mise en place d'une base de données participative.

L'objectif de ce projet est dès lors double : à la fois démontrer la possibilité de détecter avec du matériel accessible débris et satellites et en parallèle déterminer les minimums matériels nécessaires à une détection et classification fiable. Le projet s'articule ainsi autour de deux parties.

Dans un premier temps, il s'agit d'obtenir des images exploitables où sont observées sous la forme de traces (lignes sur des images longue exposition) des satellites et/ou débris. Une méthode ascendante dans le choix des composants a été utilisée dans le but de définir les minimums nécessaires et de comparer les résultats de montages basiques avec ceux de montages plus complexes.

Ensuite il est aussi nécessaire d'être en mesure de fournir une méthode automatique de

traitement des images obtenues ne nécessitant aucune connaissance astronomique avancée. La seconde partie de ce projet s'attelle donc à la construction d'un algorithme de traitement efficace des données, en mesure de détecter sur les photos les traces, et d'en calculer les orbites des objets détectés, orbites pouvant par la suite être intégrées à une base de données globale.

1 Prototypes et obtention des images

L'objectif de la première partie de ce travail est l'obtention d'images exploitables, de manière la plus automatique possible. Il est par ailleurs nécessaire de pouvoir mesurer un certain nombre de paramètres tels que la position GPS ou encore l'orientation de la caméra, nécessaires au calcul d'orbite qui suivra. Le choix a été fait de travailler avec un Raspberry Pi comme ordinateur contrôlant caméra et capteurs.

1.1 Caméra

1.1.1 Caméras testées

Les spécifications de la caméra doivent permettre la prise de photo de nuit. Les caractéristiques sont ainsi similaires aux caractéristiques utilisées en astrophotographie. Sont donc nécessaires :

- un capteur le plus grand possible, avec des photosites de bonne taille,
- un objectif le plus ouvert possible,
- la possibilité de régler les ISO qui mesurent la sensibilité à la lumière,
- la possibilité de réaliser des images longue exposition (une dizaine de secondes environ).

La présence d'une mise au point manuelle est un plus. Du fait de l'objectif d'automatisation du système il est exigible de pouvoir commander la caméra directement depuis un ordinateur.

Le choix de l'objectif est tout aussi important. Ce dernier doit permettre la transmission du plus de lumière possible. Un grand angle est intéressant dans le but de pouvoir couvrir le plus de ciel possible.

Deux caméras et trois objectifs ont été testés, respectivement :

- La Raspberry Pi HQ Camera associée d'un objectif Waveshare 6mm Wide Angle Lens,
- Le Sony α 7 avec les objectifs Sony FE 28-70 mm f/3.5-5.6 OSS et Sony FE 1.4/24 GM

Les principales spécifications de ces caméras sont données dans l'annexe A.

1.1.2 Exemples d'images obtenues

La figure 2 présente des exemples de photos longue exposition (6 secondes) obtenues avec la Raspberry Pi - HQ Camera et l'objectif Waveshare 6mm Wide Angle Lens ainsi que le Sony $\alpha 7$ monté de l'objectif Sony FE 28-70 mm f/3.5-5.6 OSS et l'objectif Sony FE 1.4/24 GM. Noter que la caméra Raspberry Pi - HQ Camera n'est pas dotée d'un mode longue exposition, les photos sont donc obtenues par superposition des images d'une vidéo de 20 secondes.

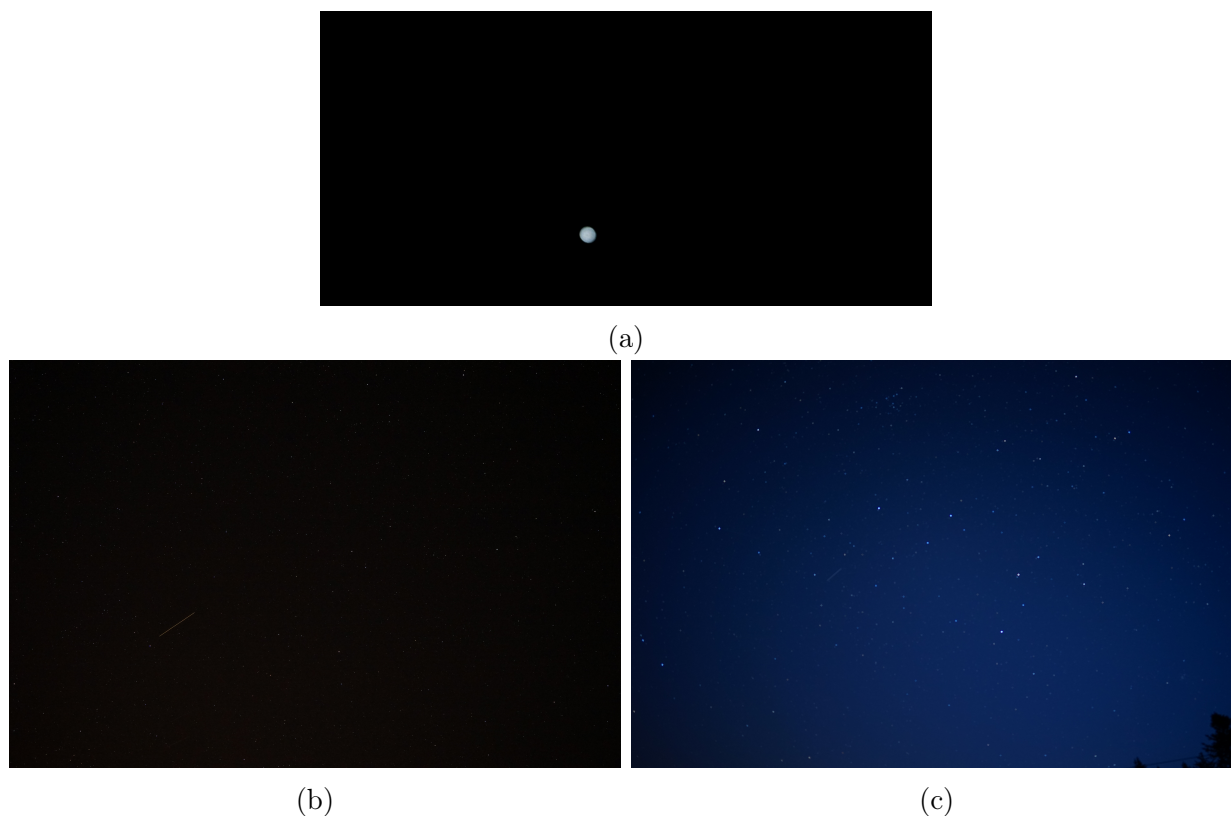


FIGURE 2: Exemple d'images réalisées par (a) la Raspberry Pi - HQ Camera et l'objectif Waveshare 6mm Wide Angle Lens ainsi que le Sony $\alpha 7$ monté de (b) l'objectif Sony FE 28-70 mm f/3.5-5.6 OSS et (c) l'objectif Sony FE 1.4/24 GM.

1.1.3 Discussion et comparaison des caméras

Pour ce qui est de la première caméra, il est clair sur la photo 2a que les images obtenues rendent difficile toute détection. La luminosité est trop faible, aucune étoile n'est visible, même la tâche blanche correspondant à la lune est floue. Il est donc impossible de pouvoir espérer repérer une quelconque trace de satellite ou de débris. Si cette dernière offre l'avantage de pouvoir être facilement programmable et compatible avec le Raspberry Pi utilisé, la taille du capteur et le nombre de pixels sont trop faibles pour offrir une sensibilité suffisante.

Par ailleurs, l'absence d'écran ne permet pas de faire la mise au point de manière correcte, diminuant encore la qualité des images.

La seconde caméra permet de résoudre ces problèmes en offrant un capteur beaucoup plus grand et avec plus de pixels. Il est indéniable sur les photos 2b et 2c que la qualité est bien meilleure. L'utilisation d'appareils photos réflex ou hybrides en astrophotographie et dans le cadre de détections de satellites amène toute fois quelques problèmes qu'il est nécessaire de souligner. Tout d'abord, la prise d'images longue exposition et avec des ISO élevés comme il est ici nécessaire donne lieu à du bruit non négligeable [8][9]. Ce bruit est à prendre en considération dans la mesure où, comme détaillé dans la suite de ce rapport, sa présence perturbe la détection des traces, et sa réduction supprime au contraire les traces les moins lumineuses. Enfin, un problème spécifique au **Sony α 7** est à remarquer. Ce dernier possède en effet un filtre spatial interne visant à réduire le bruit mais conduisant dans le cadre d'astrophotographie à une réduction du nombre d'étoiles et de traces détectées, lui donnant le nom de *Star Eater* [10][11]. Il est cependant à noter que ceci est propre au **Sony α 7** et donc non présent dans d'autres appareils photos similaires.

Pour ce qui est enfin du choix de l'objectif pour le second appareil, deux objectifs ont été testés. Le premier, objectif polyvalent non tourné vers la photo de nuit, s'est avéré déjà suffisant pour obtenir des photos exploitables. Le second, plus spécifique à la photographie nocturne possède une ouverture plus importante permettant à une plus grande quantité de lumière de passer et améliorant donc le nombre d'étoiles détectées. La photo 2c paraît en effet plus lumineuse que la photo 2b. Ce dernier objectif possède aussi une focale plus petite permettant un plus grand angle de vue et offrant donc la possibilité de couvrir une surface plus importante du ciel.

Si la diminution de la focale semble à première vue positive dans la mesure où cette dernière permet un angle de vue plus important, il est tout de même nécessaire de faire attention aux défauts optiques engendrés par l'utilisation d'un objectif grand angle [12]. En particulier, la déformation des images peut interférer avec la suite du processus et notamment la détection des lignes droites et de leur centre. Ici, les objectifs utilisés ont une focale suffisamment importante pour que ces effets aient été considérés comme négligeables, mais l'utilisation de lentilles de focale plus faible nécessiterait la prise en compte de ces facteurs dans les algorithmes suivants.

1.2 Capteurs

Deux modules contenant différents capteurs ont été utilisés :

- Le module **GPS Waveshare GSM/GPRS/GNSS HAT**, permettant d'obtenir la position GPS mais aussi de téléphoner, de se connecter à internet, de transférer des données par Bluetooth etc.

- Le module BNO055 permettant d’obtenir l’orientation absolue du capteur, et donc de la caméra rattachée à l’aide d’un accéléromètre, d’un gyroscope et d’un magnétomètre.

L’ensemble des spécifications de ces modules peuvent être trouvées dans les fiches descriptives des capteurs [13][14].

1.3 Prototype complet

Le prototype final retenu pour la suite du projet et l’analyse des images, est constitué de la caméra Sony Alpha 7, des deux objectifs associés et des capteurs ci dessus. L’ensemble des spécificités de montage est présenté dans l’annexe B.

2 Traitement des images et obtention des orbites

2.1 Méthode

Une fois les images obtenues, il est nécessaire de pouvoir détecter de manière automatique les traces, d’autant plus que certaines ne sont pas ou peu visibles à l’oeil nu. L’ensemble du processus, décrit schématiquement sur la figure 15 de l’annexe C se divise en trois parties distinctes :

- Le traitement initial des images, permettant de filtrer les images et de faciliter les étapes ultérieures.
- La détection des traces sur l’image et la détermination de leur position absolue dans le système de coordonnées local ou système de coordonnées altitude/azimut (ALT-AZ) [15].
- La détermination des orbites.

Une description plus approfondie de ces étapes est fournie dans la suite de ce rapport. L’ensemble des codes permettant leur réalisation est donné dans le GitHub associé [16].

2.1.1 Traitement initial des images

La détection automatique des traces sur les images directement récupérées des caméras est complexe de part la présence importante de bruit. De plus, nombreuses traces sont très peu lumineuses nécessitant un traitement initial des images avant de les soumettre aux algorithmes de détection.

Ce premier traitement a ainsi pour but de rendre plus visibles ces traces tout en simplifiant suffisamment les images (limiter notamment le nombre d’étoiles visibles, pouvant pour les

plus grosses êtres considérées comme des traces par les algorithmes de détection), afin de rendre la détection plus rapide et efficace.

Pour se faire, sont appliqués successivement :

- un filtre augmentant le contraste
- un filtre augmentant la luminosité
- un filtre permettant de passer en noir et blanc.

Les seuils de ces trois filtres ont été choisis de manière empirique à la suite de divers tests, afin d'obtenir des images suffisamment filtrées sans pour autant perdre trop d'information sur les traces. Divers autres filtres, tels que des filtres gaussiens par exemple ont été testés sans que ces derniers n'apportent de changements significatifs. Ils n'ont donc pas été retenus.

Un exemple d'image avant et après avoir été filtrée est présentée sur la figure 3. Les couleurs ont été inversées pour plus de clarté. Noter entre autres que certaines traces non visibles à l'oeil nu ont déjà été révélées par cette première étape.

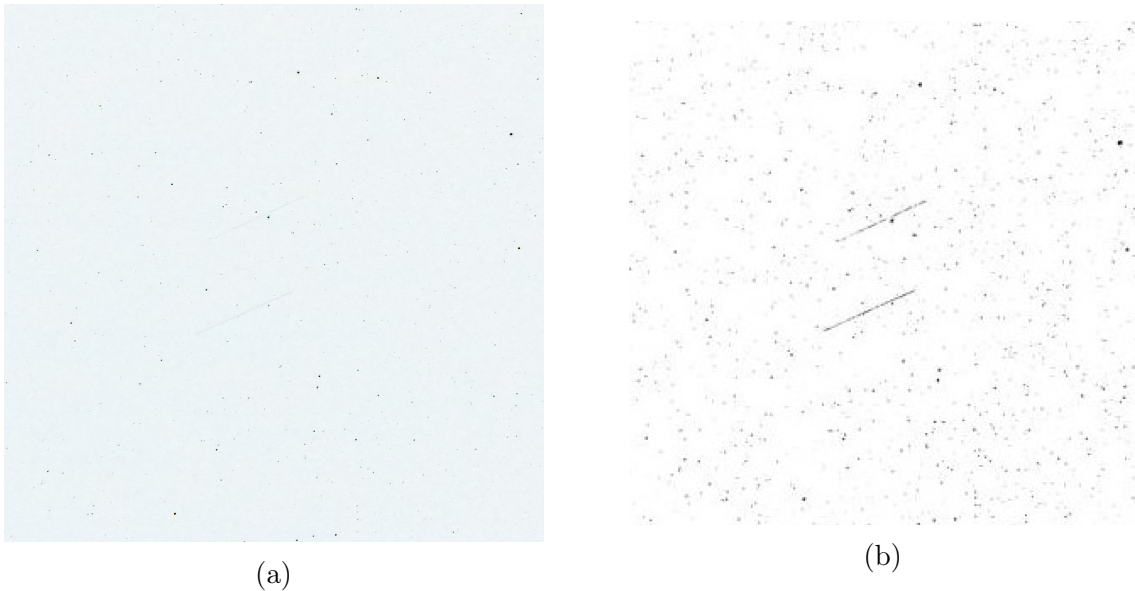


FIGURE 3: Exemple d'image obtenue (a) avant et (b) après traitement initial. Les couleurs ont été inversées.

2.1.2 Détection des lignes

L'algorithme de détection des lignes a été mis au point par Yann Bouquet [17]. La méthode est basée sur une transformée de Hough qui analyse l'alignement de points dans un espace à 2 dimensions. Un filtre de Canny est d'abord appliqué sur l'image dans le but d'éliminer des points lumineux isolés ayant peu de chance d'appartenir à une trace de satellite. La transformée de Hough est ensuite effectuée, détectant généralement un nombre de traces

nettement supérieur au nombre d’objets sur l’image. En effet, la méthode de détection est très sensible et trouve plusieurs lignes pour une même trace réelle. Ainsi, la procédure se termine par le regroupement des lignes correspondant à la même trace, c’est-à-dire en renvoyant la moyenne des lignes correspondant à des orientations et positions similaires. L’analyse des voisinages de ces groupes permet alors une bonne estimation des orientations et positions réelles des traces. Une description plus complète est fournie dans le rapport de Yann Bouquet [17].

2.1.3 Détection du centre des traces et de la position du satellite

L’approche la plus simple pour connaître la position du satellite est d’estimer son orbite par une ellipse. Trois positions à trois instants différents suffisent pour interpoler une ellipse autour de la terre. À partir d’une image longue exposition et des traces qu’elle contient, ainsi que le temps à laquelle l’image a été prise, on doit pouvoir déterminer la position du satellite observé. Dans un premier temps, il s’agit donc de trouver le centre des traces précédemment déterminées. Cette position est exprimée en pixel sur l’image et est déterminée par l’approche suivante.

Notons S l’ensemble des points de l’image appartenant à la ligne détectée à l’étape précédente, et pour un point $p \in S$, notons le carré de côté $a \in \mathbb{N}$ pixels centré en p , $C_{p,a}$. Définir ensuite une fonction poids par $w : S \rightarrow \mathbb{R}$:

$$p \mapsto \sum_{p' \in C_{p,a} \cap S} I(p') \quad (1)$$

où I renvoie l’intensité du pixel. Lorsque le poids $w(p)$ d’un point dépasse un seuil empirique τ , celui-ci est compris comme appartenant à la trace T du satellite.

Une fois tous les points appartenant à la trace déterminés, le point milieu est alors calculé comme point du milieu de la trace. Cette approche s’est révélée efficace, et plus rapide qu’un calcul de barycentre, donc conservée.

Un commentaire peut sans doute être fait concernant le choix de ce poids. Celui-ci est fait pour favoriser les points appartenant à la trace générée par le satellite par rapport aux traces d’autres satellites (ayant des traces d’orientation et de position différentes) et aux étoiles en arrière-plan de l’image, en favorisant les points dans l’alignement de la trace.

Une fois les coordonnées du centre des traces obtenues, ces dernières sont reliées à l’heure de passage du satellite en ce point, t_{mid} , défini comme :

$$t_{\text{mid}} = t_{\text{photo}} + \frac{t_{\text{exposition}}}{2}, \quad (2)$$

avec t_{photo} l’heure à laquelle la photo a été prise et $t_{\text{exposition}}$ le temps d’exposition.

Les coordonnées sont ensuite converties dans un système de coordonnées altazimutales (ou horizontales) ou dans un système de coordonnées équatoriales à partir des paramètres (position GPS, inclinaison de la caméra, etc) obtenus des capteurs.

2.1.4 Traces correspondant au même satellite

Une fois que les coordonnées des satellites dans l'image sont connues, il reste à déterminer quelles traces, capturées sur des images différentes, correspondent au même satellite (Pour calculer une orbite il faut en effet deux à trois positions et temps associés au même satellite, sur des images différentes). Pour que des traces de deux images (ρ, θ, t) et (ρ', θ', t') (avec ρ, θ la paramétrisation de la ligne et t le temps de la photo) soient considérées comme étant causées par le même satellite, le critère simple suivant est établi :

$$\chi_{t-t' < t^*} \left(\left(\frac{\rho - \rho'}{\rho^*} \right)^2 + \left(\frac{\theta - \theta'}{\theta^*} \right)^2 \right) + \chi_{t-t' \geq t^*} \eta^2 < \eta^2 \quad (3)$$

pour $\eta > 0$ une tolérance fixée de manière empirique et ρ^*, θ^*, t^* des grandeurs caractéristiques.

2.1.5 Détermination des orbites

Une fois les traces regroupées par satellite, les données des positions obtenues peuvent enfin être fournies à un programme permettant le calcul d'orbites à partir de deux ou trois observations, tel que ELFIND [18].

2.2 Résultats et Discussion

2.2.1 Exemples de détection des traces et de leur centre

Résultats L'ensemble des images obtenues a été traité par les algorithmes décrits précédemment dans le but de détecter les traces. Les figures 4, 5 et 6 présentent les résultats obtenus pour trois images types. Les différentes étapes du processus sont illustrées, respectivement :

- l'image initiale obtenue directement de la caméra,
- l'image filtrée par le traitement initial décrit dans la section 2.1.1,
- le résultat de la détection des lignes par l'algorithme de Hough (section 2.1.2),
- les traces (en violet) et leurs centres (en jaune) obtenus de l'algorithme de la partie 2.1.3.

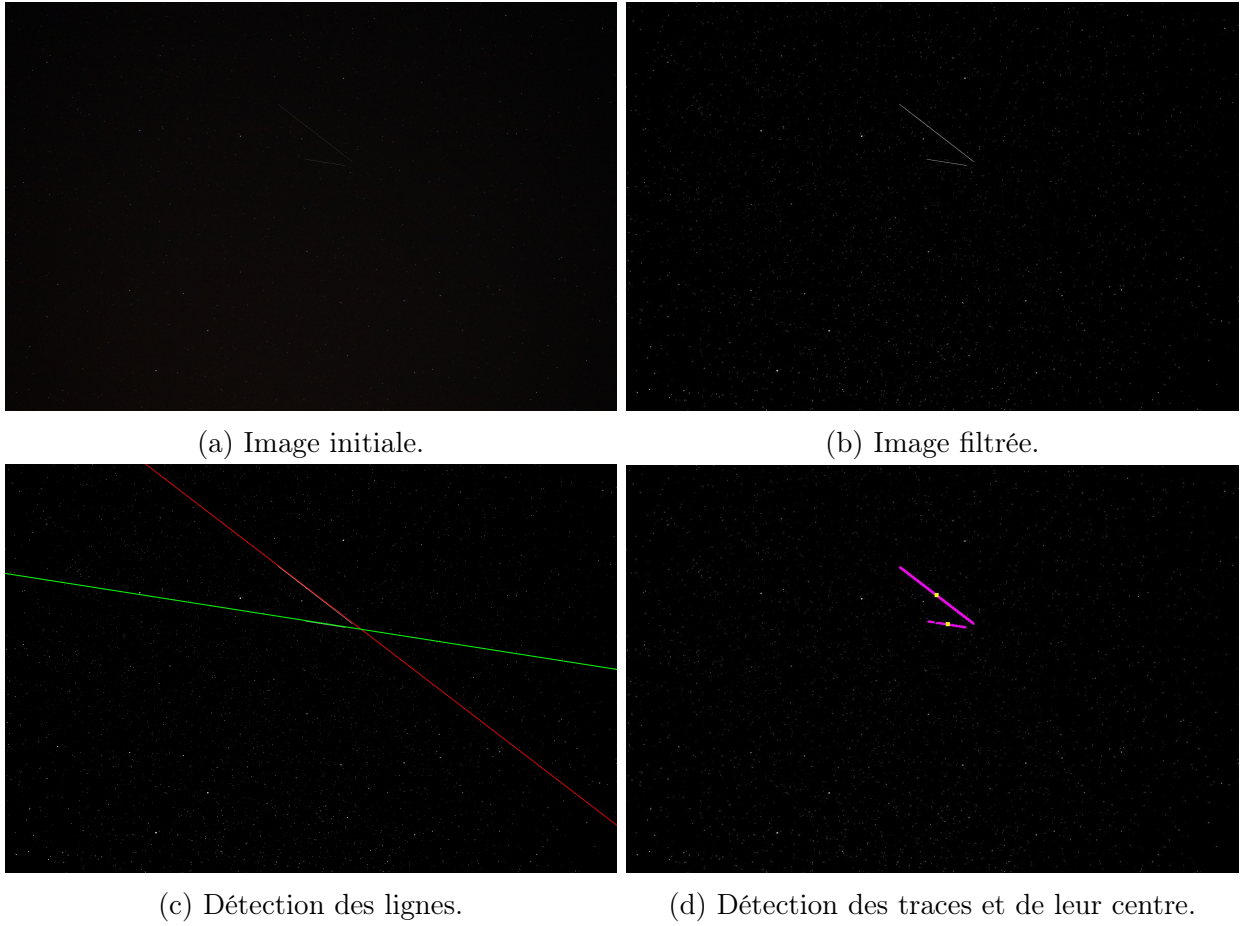


FIGURE 4: Exemple de détection des traces et illustration de l'ensemble du processus, cas d'une détection correcte.

Discussion La figure 4 donne un exemple d'un fonctionnement correct de l'algorithme et d'une détection correcte des traces. Le traitement initial permet comme attendu de rendre plus lumineuses les traces, bien que déjà visibles à l'oeil nu (figure 4b); les lignes sont correctement détectées par la transformée de Hough (figure 4c) et les traces et leurs centres semblent eux aussi correctement évalués (figure 4d).

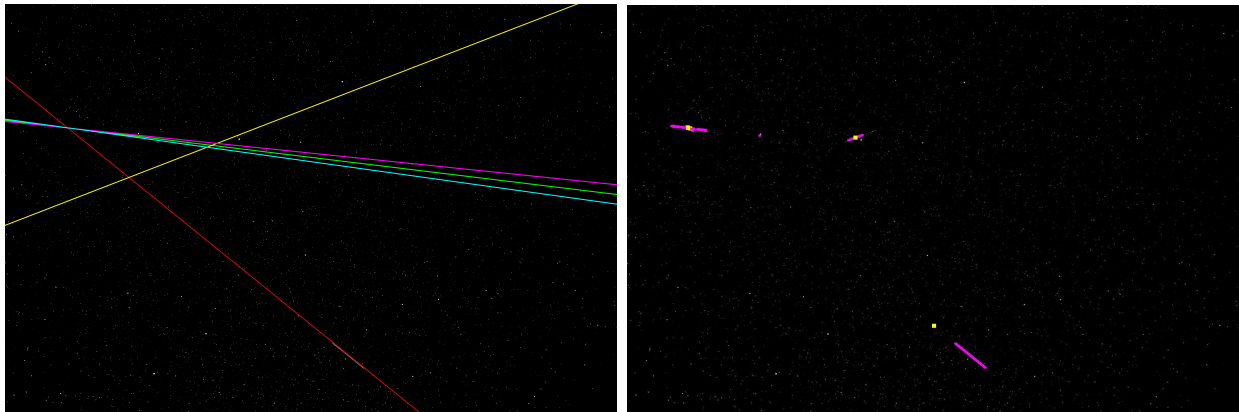
Néanmoins, de nombreuses images n'ont pu être correctement analysées. La figure 5 est typique d'une analyse incorrecte et présente simultanément plusieurs des principales erreurs observées.

Dans un premier temps, il est à noter sur la figure 5c que l'une des traces a été détectée en triple par l'algorithme de Hough. Ce problème, déjà soulevé dans des précédents rapports [17] a des conséquences limitées dans la suite du processus dans la mesure où les satellites détectés dans des images successives sont ensuite regroupés, permettant in fine de ne garder qu'une seule trace. Noter que plus est sur la figure 5d qu'un seul point jaune est observé sur la trace plusieurs fois détectée, indiquant que les trois centres calculés sont identiques, et donc que leur estimation n'a pas été altérée par la détection multiple.



(a) Image initiale.

(b) Image filtrée.



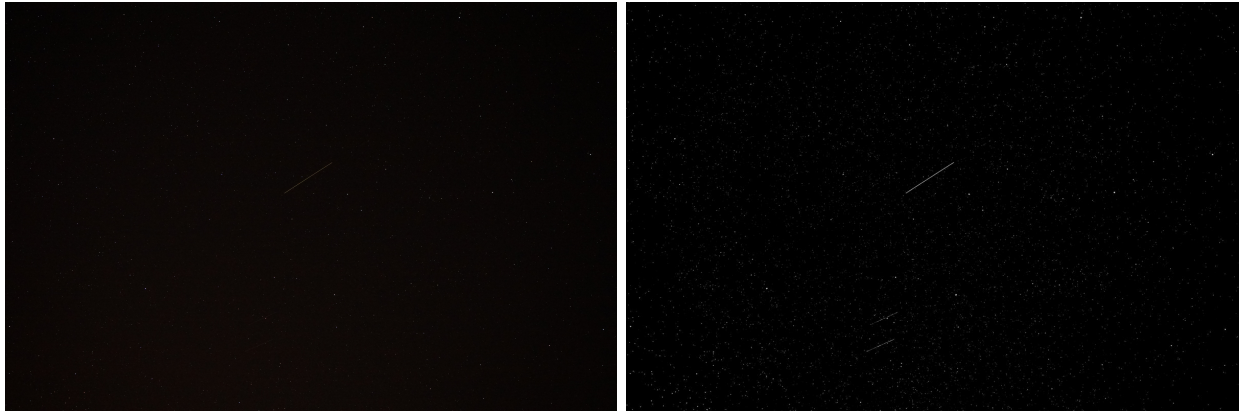
(c) Détection des lignes.

(d) Détection des traces et de leur centre.

FIGURE 5: Exemple de détection des traces et illustration de l'ensemble du processus, cas d'erreurs dans l'algorithme affectant la détection.

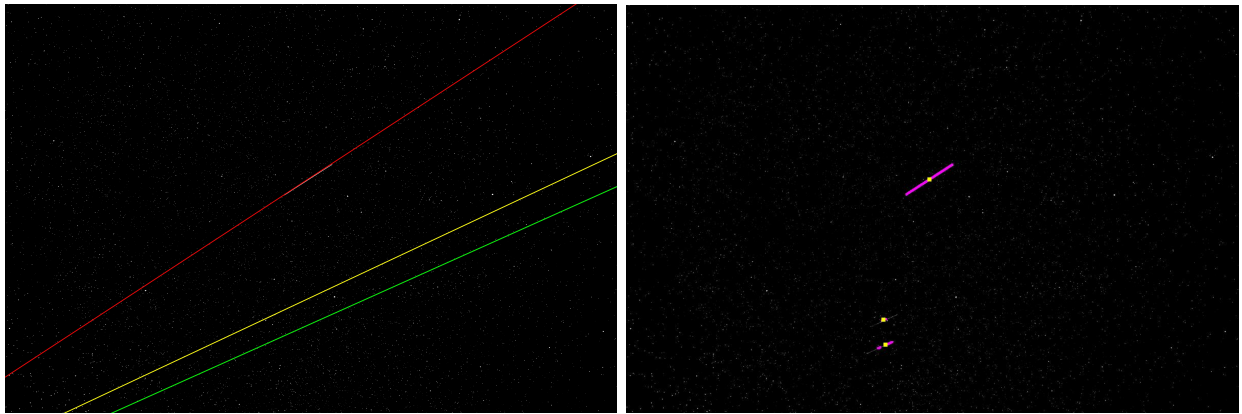
Une solution pour éviter ce problème pourrait cependant être de prendre en input plusieurs images successives, afin que l'algorithme puisse comparer les traces obtenues sur chacune d'elles. Ne seraient alors retenues que les lignes présentant le même angle sur des images successives. De la sorte, pourraient être éliminées les traces correspondant à un centre commun, mais différentes de par l'angle qu'elles forment entre elles. Sur l'exemple de la figure 7, un tel processus permettrait notamment de conserver sur l'image 7a uniquement la trace avec la même orientation que celle de la figure 7b.

La seconde erreur, plus problématique, concerne la détection du centre des traces. Il est en effet clair sur la trace la plus basse de la figure 5d que le centre est mal positionné. Ceci est dû à la présence d'étoiles suffisamment lumineuses le long de l'orientation de la ligne, détectées par l'algorithme comme faisant partie de la trace (ici il s'agit du point violet observable entre les deux autres traces) et biaisant la moyenne utilisée pour calculer le centre de la trace. Ce problème peut être résolu en diminuant le seuil d'intensité ou le côté du carré utilisé dans la définition du poids des pixels (équation (1)), permettant de ne détecter que les zones les plus lumineuses correspondant aux traces. Cependant, certaines traces étant plus faibles et



(a) Image initiale.

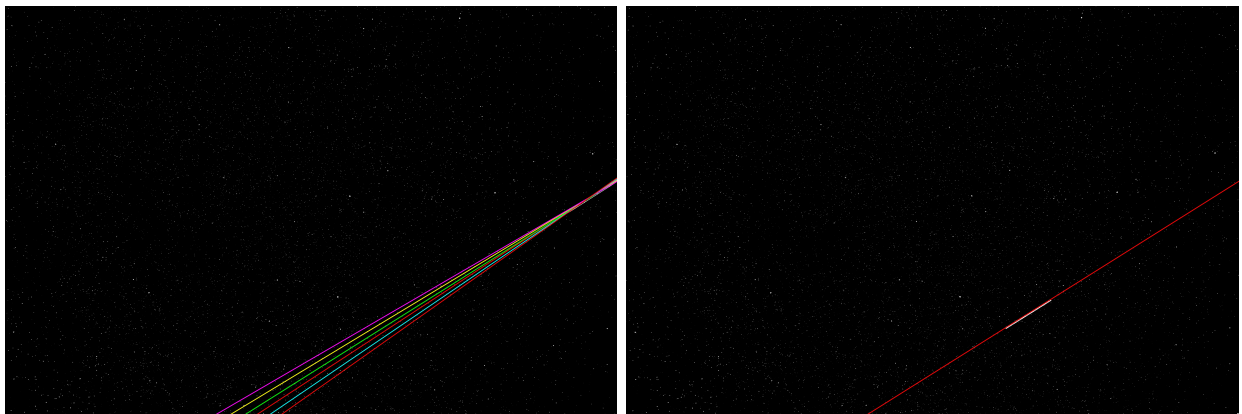
(b) Image filtrée.



(c) Détection des lignes.

(d) Détection des traces et de leur centre.

FIGURE 6: Exemple de détection des traces et illustration de l'ensemble du processus, cas d'erreurs dans l'algorithme n'affectant pas ou peu la détection.



(a)

(b)

FIGURE 7: Exemple d'image présentant des traces multiples correspondant au même centre (a) et une trace unique pour le même objet (b) lors de la prise suivante.

moins lumineuses, une telle opération les rendrait indétectables.

Le troisième exemple est typique de tels phénomènes. Les lignes sont correctement détectées par Hough, et bien que les seuils soient identiques à ceux utilisés pour la figure 5d, il est observé sur la figure 6d que les traces ne sont pas entièrement détectées comme telles, parce que leur intensité est suffisamment faible pour ne pas être prises en compte (Les zones violettes correspondant aux zones détectées comme appartenant à la trace par l'algorithme ne couvrent pas l'ensemble de la trace). Dans cet exemple, cela n'affecte pas la détection du centre puisque celui-ci est comme souvent associé aux zones les plus lumineuses, et ce sont donc seulement les extrémités qui sont coupées, de manière symétrique, n'affectant pas la position détectée du centre (qui est placé par calcul du point milieu de la trace comme expliqué dans la section 2.1.4). Dans l'hypothèse où une ou plusieurs des extrémités de la trace ne sont pas détectées du fait du poids $w(p)$ utilisé, la position du centre serait cependant faussée. Afin de résoudre ce problème, il pourrait être pris en compte le gradient d'intensité le long de la trace afin de symétriser celle-ci et d'éviter un placement latéral du centre. Une diminution du seuil n'est ici pas envisageable, puisque cela rendrait les traces complètement indétectables (la seconde notamment).

Ainsi, il est difficile de trouver une solution au problème de la figure 6d sans compromettre d'autres résultats. Les seuils ont été réglés de sorte à pouvoir détecter de manière correcte le plus grand nombre de traces (cf section 2.2.3), mais une détection parfaite de toutes les traces semble difficile, voir impossible.

Une autre solution consisterait à filtrer de manière plus efficace les étoiles lors de l'étape de traitement initiale. Néanmoins, une nouvelle fois, cela serait au dépens de la qualité des traces et donnerait lieu à une perte d'information. Aucun traitement ou filtre plus efficace que celui présenté n'a été trouvé pour limiter l'impact des étoiles sur les traces. La définition du point centre en tant que point médian de la trace et non point moyen pourrait aussi donner lieu à de meilleurs résultats.

Ainsi, il a donc été observé que la majorité des traces excédentaires provenait d'une multiplication des traces correspondant à un même objet, et des possibilités ont été discutées afin de pallier ce problème. Néanmoins, l'effet inverse est tout aussi fréquent, et si l'algorithme semble donc efficace pour détecter les traces les plus lumineuses, ce dernier peine à rendre compte des traces qui le sont moins. Les figures 8a et 8b donnent un exemple flagrant de cette lacune. Pour une même image initiale, la figure 8a présente les résultats fournis par l'algorithme et sur la figure 8b sont encadrées les traces observables à l'oeil nu. Quand deux traces sont observables à l'oeil nu, seulement une est détectée par l'algorithme.

Afin de réduire le nombre de traces non détectées, il pourrait être envisagé d'appliquer un traitement algorithmique par $w(p)$ décroissant dans un certain intervalle, en combinant les idées introduites précédemment. De la sorte, en plaçant un seuil d'intensité $w(p)$ élevé, seules les traces les plus lumineuses seraient détectées. Ces traces seraient ensuite gardées en mémoire et éliminées de l'image (en 'verrouillant' ces pixels par exemple). Le processus serait ensuite répété avec un $w(p)$ plus faible, les traces déjà trouvées n'interféreraient alors

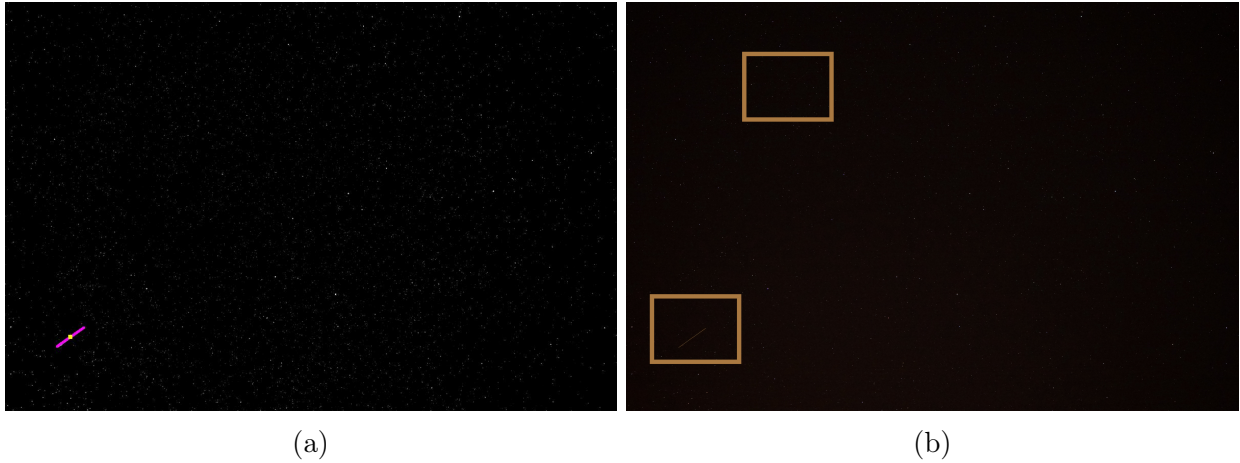


FIGURE 8: Exemple de détection de traces (a) par l'algorithme et (b) à l'oeil nu.

pas avec la détection à un seuil d'intensité inférieur.

2.2.2 Statistiques

Détection des traces : Un recensement manuel des traces observables à l'oeil nu sur l'ensemble des photos tel que réalisé sur la figure 8b a permis une comparaison systématique avec les résultats fournis par l'algorithme, permettant à la fois d'attester sur un nombre important de traces la performance de l'algorithme et de quantifier l'impact des erreurs précédemment citées. La figure 9 et la table 1 présentent les résultats obtenus par l'algorithme et à l'oeil nu sur un set de 117 images capturées par le Sony $\alpha 7$ et l'objectif Sony FE 28-70 mm f/3.5-5.6 OSS.

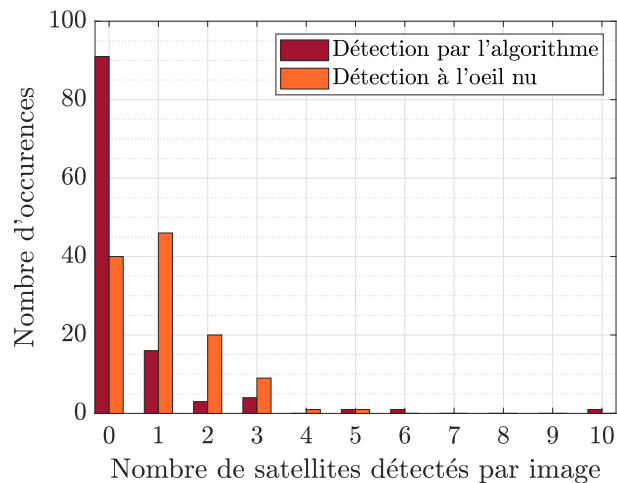


FIGURE 9: Histogramme du nombre de satellites détectés par image, par l'algorithme et à l'oeil nu.

	Algorithme	Oeil nu
Nombre de photos présentant des traces	26	77
Nombre de traces détectées	55	122
Nombre moyen de traces détectées par image	0.47	1.04

TABLE 1: Statistiques sur le nombre de satellites détectés par l’algorithme et à l’oeil nu, et comparaison.

Détection des centres : En ce qui concerne la détection des centres des traces par l’algorithme, on note que sur les 26 images comportant des traces détectées, seulement 24 ont pu être exploitées pour placer les centres. Après traitement, 14 images comportaient les centres placés sur 20 traces, alors que 34 traces avaient été détectées. Sur ces 20 centres, 12 sont correctement placés, ce qui correspond à une erreur de placement du centre sur la trace de 59%, et une erreur de 40% sur la position des centres.

La détection des traces et le placement des centres sur celles-ci se faisant successivement, les résultats obtenus correspondent à une erreur totale de

$$e_{rr} = 100 \cdot \left(1 - \frac{N_{eff}^t}{N_{tot}^t} \frac{N^c}{N_{eff}^c} \frac{N_{eff}^c}{N^c} \right) = 100 \cdot \left(1 - \frac{34}{55} \frac{20}{34} \frac{12}{20} \right) \simeq 78\%$$

par rapport à un détection-placement parfait, et où l’on a introduit N_{eff}^t et N_{tot}^t le nombre effectif de traces et le nombre total de traces. N^c correspond au nombre de centres et N_{eff}^c le nombre de centre correctement placés.

Discussion : Premièrement, il est à noter sur la table 1 qu’un nombre important d’images présentait des traces (77 sur 117), confirmant l’intérêt de développer un tel processus de détection des satellites depuis le sol, puisque de nombreux objets sont en effet détectables. Pour ce qui est de l’algorithme, ce dernier ne détecte qu’environ la moitié des traces observables, les plus lumineuses, les moins lumineuses étant en effet supprimées par le traitement initial des images. La proportion de traces détectées par image, définie comme le rapport entre le nombre de traces détectées par l’algorithme et le nombre de traces observées à l’oeil nu, vaut en moyenne pour ce set 0.504.

Pour ce qui est du nombre total de traces détectées, celui ci s’élève comme indiqué sur la table 1 à 55. Cependant, similairement au cas de la figure 5, certaines traces sont détectées plusieurs fois (entre 2 et 10 fois). En supprimant ces traces détectées plusieurs fois, ce sont in fine seulement 34 satellites qui ont été détectés. De plus, il est à noter que pour pouvoir calculer les orbites, il faut avoir 2 à 3 traces correspondant au même satellite, détectées sur des images successives, ce qui limite encore le nombre de données obtenues pouvant être

exploitées pour le calcul d'orbites. Au total, sur les 117 images, seulement 11 orbites ont pu être obtenues. En prenant finalement en compte l'erreur sur la détection des centres, l'erreur totale de l'ensemble du processus s'élève à 78%.

Il est clair que le rendement est faible, et de nombreuses améliorations (dont un certain nombre ont déjà été discutées dans la partie 2.2.1) sont donc à prévoir afin de détecter plus efficacement les traces et réduire les erreurs précédemment citées.

2.2.3 Importance du seuil de Hough et des paramètres empiriques τ et a

Il a été discuté dans la partie 2.2.1 l'importance du réglage des seuils de détection de l'algorithme. Une étude de l'impact de ces seuils a donc été réalisée.

Les figures 10 et 11 montrent l'importance du seuil de Hough et des paramètres empiriques a et τ utilisés dans la fonction poids w (Équation (1)) qui permet d'associer à chaque satellite une position et un temps. L'analyse a été effectuée sur l'ensemble d'images `set_final_analysis` (à retrouver dans le GitHub associé [16]), typiques des images obtenues. Elle démontre qu'un seuil de Hough suffisamment élevé permet de réduire de manière satisfaisante le nombre de faux positifs dans la détection des traces. De plus, un bon choix de paramètres a et τ permet d'obtenir un taux de réussite (nombre de positions trouvées sur nombre de traces correctement détectées) de $2/3$ dans le calcul des coordonnées position-temps du satellite. À noter qu'il n'est pas nécessaire de prendre a supérieur à 16, ce qui est intéressant puisque le temps de calcul grandit selon $\mathcal{O}(a^2)$. Les meilleurs paramètres pour ce type de données sont donc $a = 16$, $\tau = 0.013$.

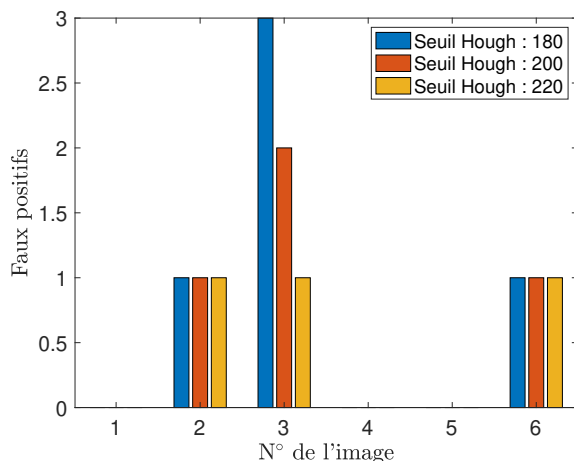


FIGURE 10: Impact du seuil de Hough sur le nombre de faux positifs.

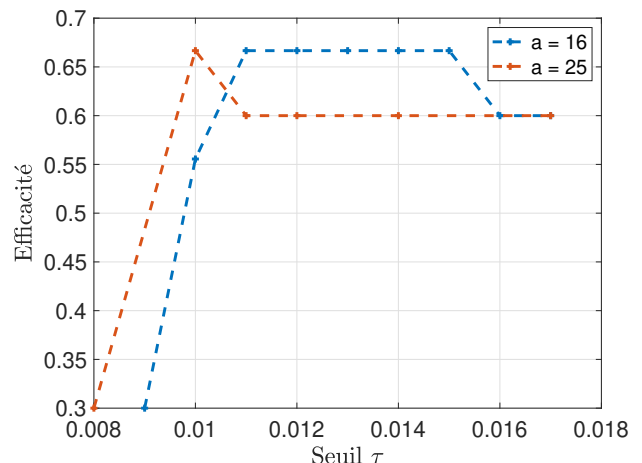


FIGURE 11: Impact des paramètres a et τ de la fonction poids w sur l'efficacité de l'algorithme.

Conclusion

En conclusion, le montage effectué à partir du Sony Alpha 7 a su démontrer la possibilité d’obtenir, à partir d’un appareil photo classique et d’objectifs adaptés, des images exploitables présentant des traces d’objets en orbite.

L’approche optique s’est pourtant avérée limitée par le traitement des images obtenues, nécessaires à l’obtention des orbites. Si de nombreuses traces ont été observées à l’oeil nu, l’algorithme a montré des difficultés à les détecter et seules les plus lumineuses ont pu offrir des résultats concluants, le rendement total de l’algorithme restant faible. En particulier, le bruit créé par les étoiles environnantes a interféré de manière non négligeable avec le processus de détection, entraînant multiples erreurs non négligeables.

Ainsi, au vu des algorithmes actuels, il est donc à penser que l’approche optique, bien que prometteuse de par la capacité des appareils optiques actuels à détecter un nombre importants d’objets parfois même invisibles à l’oeil nu, est limitée par le traitement nécessaire à la détection des objets les moins lumineux. Si cette dernière peut permettre la mise en place d’un premier catalogue, par le grand public, des objets les plus lumineux, il est aussi nécessaire de considérer en parallèle d’autres méthodes de détection, telles que les capteurs radars [19][20], les capteurs RF (Radio-frequency sensors) [19][21] ou encore les techniques de télémétrie laser [19][22], en vue de fournir un recensement complet des objets orbitaux.

Références

- [1] NASA. Technical report on space debris. https://www.orbitaldebris.jsc.nasa.gov/library/un_report_on_space_debris99.pdf, 1999.
- [2] ESA. Space debris : assessing the risk. http://www.esa.int/About_Us/ESOC/Space_debris_assessing_the_risk, 2005.
- [3] Rong yu Sun, Jin wei Zhan, Chang yin Zhao, and Xiao xiang Zhang. Algorithms and applications for detecting faint space debris in geo. *Acta Astronautica*, 110 :9–17, 2015. Dynamics and Control of Space Systems.
- [4] Walter D. Bennette, Kayla Zelif, and Joseph Raquepas. Classification of objects in geosynchronous earth orbit via light curve analysis. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6, 2017.
- [5] Sunita Jahirabadkar, Prajakta Narsay, Shivani Pharande, Gargi Deshpande, and Anusha Kitture. Space objects classification techniques : A survey. In *2020 International Conference on Computational Performance Evaluation (ComPE)*, pages 786–791, 2020.
- [6] Norman C. Anheier and Cliff S. Chen. A new approach to space situational awareness using small ground-based telescopes. 12 2014.

- [7] NASA. Orbital debris, quarterly news. <https://orbitaldebris.jsc.nasa.gov/quarterly-news/pdfs/odqnv23i4.pdf>, 2019.
- [8] How to avoid and reduce noise in your photos. <https://digital-photography-school.com/how-to-avoid-and-reduce-noise-in-your-images/>. Accessed 04.2021.
- [9] Noise in digital cameras. <https://www.johnsankey.ca/noise.html>.
- [10] Ian Norman. The star eater issue : Why i no longer recommend sony cameras for astrophotography. <https://petapixel.com/2017/05/04/star-eater-issue-no-longer-recommend-sony-cameras-astrophotography/>, 2017. Accessed 04.2021.
- [11] Star eater : Documentation of an issue with the sony cameras for astrophotography. <https://www.lonelyspeck.com/sony-star-eater-and-how-to-fix-it/>, author=Ian Norman, year=2017.
- [12] Miguel Alemán-Flores, Luis Alvarez, Luis Deniz, and D. Santana-Cedrés. Wide-angle lens distortion correction using division models. volume 8258, pages 415–422, 11 2013.
- [13] Gsm/gprs/gnss hatuser manual. https://www.waveshare.com/w/upload/4/4a/GSM_GPRS_GNSS_HAT_User_Manual_EN.pdf.
- [14] Bno055 intelligent 9-axis absolute orientation sensor, datasheet. https://cdn-shop.adafruit.com/datasheets/BST_BNO055_DS000_12.pdf. Accessed 04.06.2021.
- [15] Altitude/azimuth. <https://astronomy.swin.edu.au/cosmos/H/Horizontal+Coordinate+System>. Accessed 04.06.2021.
- [16] <https://github.com/kiteros/DetecteurObjetsOrbitaux>.
- [17] Yann BOUQUET. Delineating satellite tracks in astronomical images. <https://github.com/YBouquet/detectsat/blob/master/report.pdf>, 2020.
- [18] Elfind. <http://sat.belastro.net/satelliteorbitdetermination.com/ELFIND.htm>. Accessed 04.06.2021.
- [19] Becaja M. Caldwell Reina S. Buenconsejo Sara A. Carioscia Bhavya Lal, Asha Balakrishnan. Global trends in space situational awareness (ssa) and space traffic management (stm), 2018.
- [20] Jan Eilers, Simon Anger, and Thomas Neff. Radar based system for space situational awareness. *The Journal of Space Operations Communicator*, 13, 10 2016.
- [21] Matthew Prechtel. Event detection from rf sensing, 2018.
- [22] Jizhang Sang and Craig Smith. Performance assessment of the eos space debris tracking system. 08 2012.
- [23] Raspberry pi camera modules documentation. <https://www.raspberrypi.org/documentation/hardware/camera/>. Accessed 04.2021.
- [24] Raspberry pi hq camera. https://www.waveshare.com/wiki/Raspberry_Pi_HQ_Camera. Accessed 04.2021.

- [25] Sony alpha7 e-mount camera with full frame sensor full specifications and features. <https://www.sony.com/electronics/interchangeable-lens-cameras/ilce-7-body-kit/specifications>. Accessed 05.2021.
- [26] Stabilisateur optique steadyshot fe 28-70 mm f3.5-5.6 spécifications et fonctionnalités complètes. <https://www.sony.ch/fr/electronics/objectifs-appareil-photo/sel2870/specifications>. Accessed 05.2021.
- [27] Fe 35 mm f1.4 gm spécifications et fonctionnalités complètes. <https://www.sony.ch/fr/electronics/objectifs-appareil-photo/sel35f14gm/specifications>. Accessed 05.2021.

A Spécifications des caméras et objectifs utilisés

Les principales caractéristiques des caméras et objectifs testés sont présentés dans les tables 2 à 6. Une description plus détaillée est à retrouver dans la documentation des caméras [23][24][25][26][27].

Raspberry Pi - HQ Camera	
Taille du capteur	6.287x4.712 mm
Nombre de pixels	12.3 Megapixels
Taille des pixels	1.55x1.55 μm

TABLE 2: Principales caractéristiques de la caméra Raspberry Pi - HQ [23].

Waveshare 6mm Wide Angle Lens	
Focale	6 mm
Ouverture	f/1.2
Angle de vue	f/36

TABLE 3: Principales caractéristiques du Waveshare 6mm Wide Angle Lens [24].

Sony Alpha 7	
Taille du capteur	Full Size 24x36mm
Nombre de pixels	24.3 Megapixels
Gamme de sensibilité ISO	100-51200 ISO
Temps de pose min/max [s]	1/8000 - ∞

TABLE 4: Principales caractéristiques du Sony α 7 [25].

Sony FE 28-70 mm f/3.5-5.6 OSS	
Focale	28-70 mm
Ouverture maximale	f/3.5-5.6
Ouverture minimale	f/36

TABLE 5: Principales caractéristiques du Sony FE 28-70 mm f/3.5-5.6 OSS [26].

Sony FE 1.4/24 GM	
Focale	24 mm
Ouverture maximale	f/1.4
Ouverture minimale	f/16

TABLE 6: Principales caractéristiques du Sony FE 1.4/24 GM [27].

B Prototype final : Montage

B.1 Montage

Le prototype retenu est montré sur la photo 12. Ce dernier est composé de plusieurs parties :

- la caméra permettant d'obtenir les images,
- les capteurs renvoyant les données essentielles au calcul d'orbites,
- le Raspberry pi permettant de contrôler caméras et capteurs,
- la batterie externe alimentant le Raspberry,
- le disque dur permettant de stocker les données récoltées.

Une vue schématique du montage est donnée sur la figure 13.

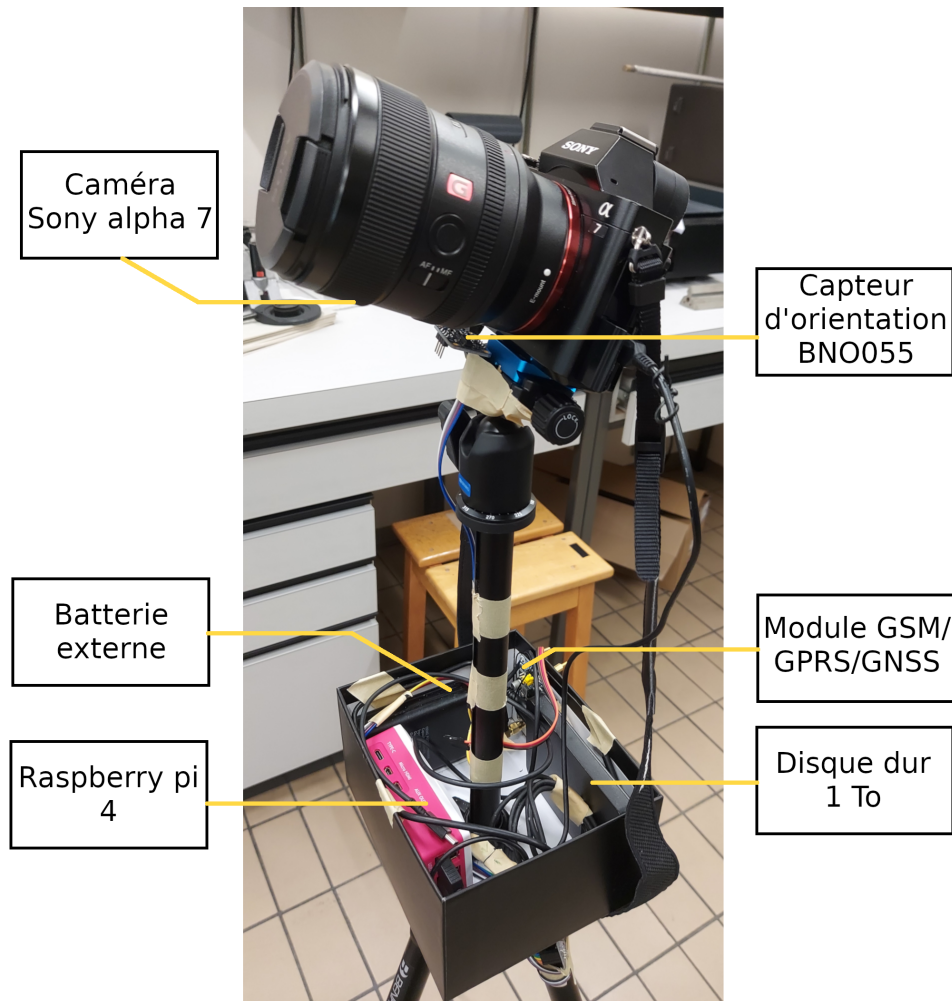


FIGURE 12: Photo annotée du prototype retenu.

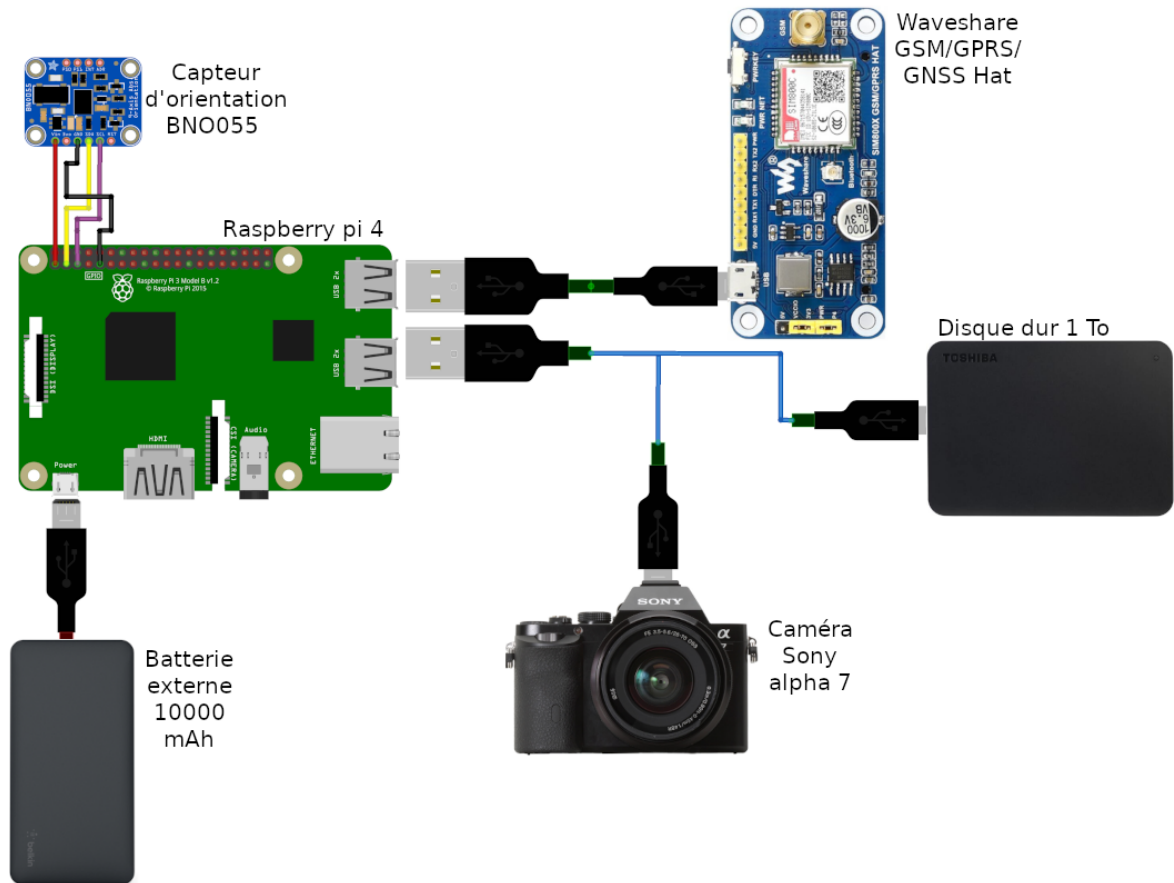


FIGURE 13: Schéma du montage du prototype.

B.2 Schéma de l'ensemble du processus de commande de l'appareil photo

La figure 14 présente l'ensemble du processus de commande de l'appareil photo et des capteurs par le Raspberry Pi. À l'allumage du Raspberry, sont ainsi réalisés successivement les mesures des paramètres nécessaires aux calculs d'orbite (orientation, position GPS), leur enregistrement sous des formats adaptés (param.txt, STATION.IN) et la commande, via USB de l'appareil photo. Les photos sont prises à intervalle de temps régulier jusqu'à l'arrêt du programme. L'ensemble des codes permettant ce processus ainsi que leur mode d'emploi sont donnés dans le GitHub associé [16].

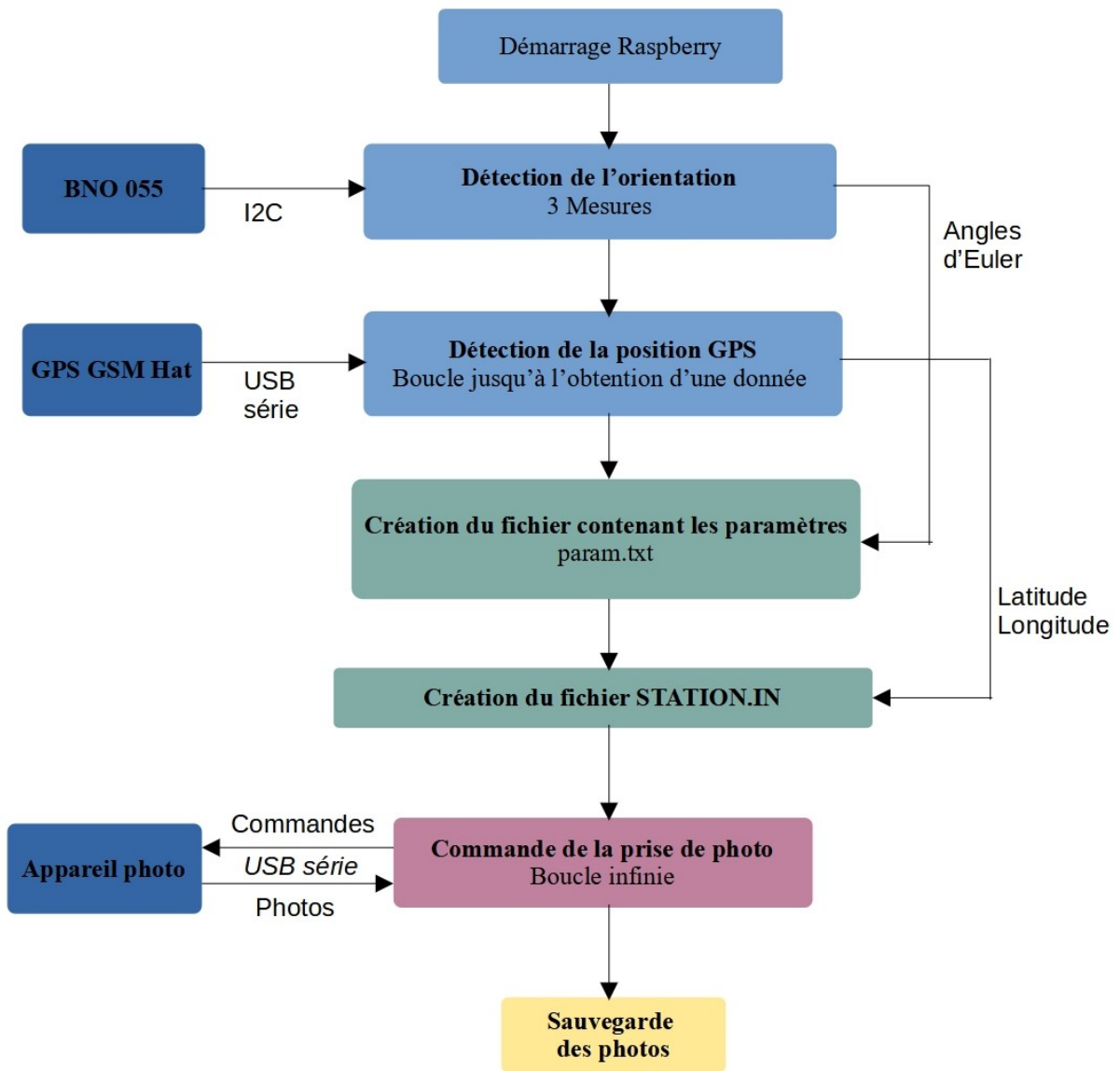


FIGURE 14: Vue schématique de l'ensemble du processus de commande de l'appareil photo.

C Schéma de l'ensemble du traitement des images

La figure 15 présente de manière schématique l'ensemble des étapes nécessaires à l'obtention des orbites à partir des images brutes, comme décrit dans la section 2. L'ensemble des codes permettant le traitement des images est donné dans le GitHub [16].

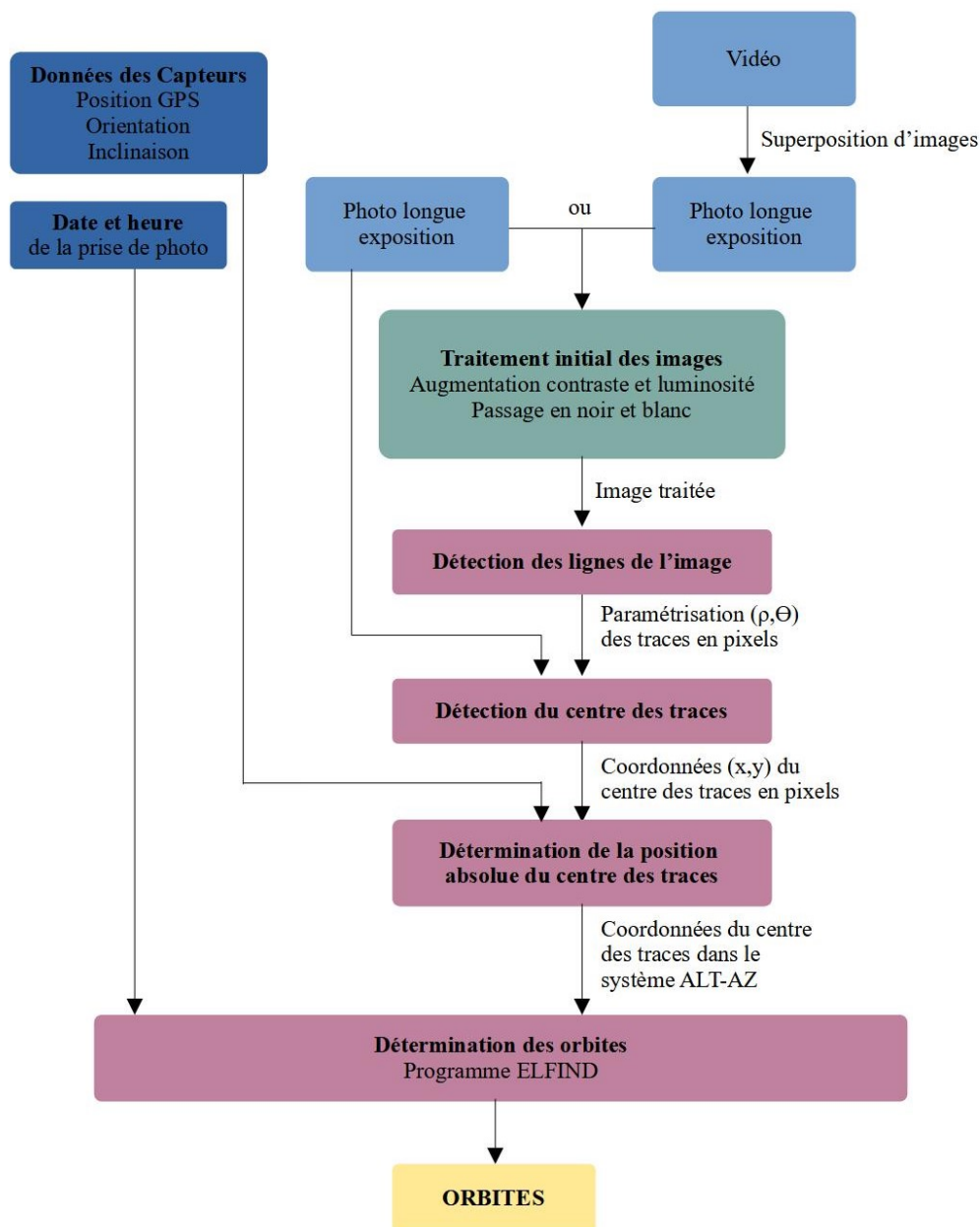


FIGURE 15: Vue schématique de l'ensemble du traitement des images.