Élaboration d'un prototype bas coût d'instrumentation mobile pour l'acquisition de nuages de points géoréférencés par photogrammétrie

Flavian Couty

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

ECOLE SUPERIEURE DES GEOMETRES ET TOPOGRAPHES

MEMOIRE

présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : Géomètre et Topographe

par

Flavian COUTY

Élaboration d'un prototype bas coût d'instrumentation mobile pour l'acquisition de nuages de points géoréférencés par photogrammétrie

Soutenu le 9 septembre 2022

JURY

Madame Joëlle NICOLAS-DUROY	Présidente du jury
Madame Elisabeth SIMONETTO	Maître de stage
Monsieur José CALI	Maître de stage
Monsieur Jérôme VERDUN	Enseignant référent

Remerciements

Lors de mon TFE au sein du laboratoire GeF de l'ESGT, j'ai pu travailler et discuter avec de nombreuses personnes que je souhaite remercier ici.

Tout d'abord je souhaite remercier Elisabeth SIMONETTO et José CALI qui ont été mes maîtres de stage durant ces 5 mois de TFE. Leurs compétences dans leurs domaines m'ont aidé à la réalisation de ce TFE. Ils m'ont également fait progresser dans ces domaines que ce soit en photogrammétrie, en programmation ou encore en positionnement dynamique. Je les remercie d'avoir été le plus disponible possible et surtout d'avoir répondu à mes questions tout au long de ce TFE. Je les remercie également pour m'avoir accordé leur confiance dès le début du projet ce qui m'a permis de travailler le plus efficacement possible.

Je souhaite aussi remercier LABERGERIE Éric et CHARLET Christophe, pour leur accompagnement dans ce projet avec leurs conseils dans leurs domaines qui m'ont apporté de nombreuses solutions.

Je tiens à remercier également VERDUN Jérôme en tant que directeur du laboratoire pour l'accueil qu'il m'a été réservé mais également en tant que professeur référent pour toutes les réponses auxquelles il a répondu avec la plus grande clarté. Plus généralement je veux remercier toutes les personnes du laboratoire GeF pour leur accueil, leur sympathie et leur disponibilité tout au long de ce TFE.

Enfin je veux remercier toute ma famille qui m'a encouragé pendant toute ma scolarité à l'ESGT mais également avant. Ces simples mots ne peuvent pas témoigner de toute la reconnaissance que j'ai pour eux.

Liste des abréviations

API : Application Programming Interface

CRS : Coordinate Reference System

EPSG : European Petroleum Survey Group

GeF : Géomatique et Foncier

GNSS : Global Navigation Satellite System

HDOP : Horizontal Dilution Of Precision

IPS : Image Par Seconde

LiDAR : Light Detection And Ranging

NMEA : National Marine Electronics Association

NRTK : Network Real Time Kinematic

RGF : Réseau Géodésique Français

RTK : Real Time Kinematic

USB : Universal Serial B

Table des matières

Introduction

La photogrammétrie est une technique permettant de déterminer les dimensions et les volumes d'un objet à partir de photos nous le montrant sous plusieurs angles [CNRTL, 2022]. Il est alors possible de représenter un objet en 3D pouvant être texturisé et ainsi pouvoir le visualiser. On peut également exploiter ce nuage de points puisqu'il est possible de mesurer des surfaces, des hauteurs et même des volumes. Le domaine d'application est assez étendu puisque que la photogrammétrie peut être utilisée à la fois dans l'aménagement d'un terrain quand il s'agit de déplacer un volume assez important de matériaux. Il peut être important de connaître la texture et les dimensions d'un objet. Il peut aussi être utilisé dans l'animation avec la modélisation 3D ou encore dans la métrologie industrielle quand il s'agit de mesurer les pièces devant respecter certaines dimensions [EXID, 2022]. En ajoutant des points d'appui sur l'objet, il est possible de géoréférencer le nuage de points et ainsi permettre de joindre plusieurs projets entre eux ou encore transmettre le nuage à d'autres personnes qui peuvent à leur tour l'utiliser pour compléter leur projet. Cette opération implique cependant de placer des points d'appui, de les mesurer avec la station totale pour ensuite pouvoir commencer le lever photogrammétrique. C'est un processus long, surtout si la surface de l'objet à relever fait plusieurs centaines de mètres carrés comme par exemple un corps de rue. Il faut alors trouver un moyen de se passer de ces points d'appui tout en gardant une précision satisfaisante.

Un système de photogrammétrie mobile répond à cette problématique puisqu'il permet de s'affranchir de points d'appui mais également de nous déplacer pendant l'acquisition des mesures. Cela est possible grâce à l'utilisation du géoréférencement direct permettant le calcul des positions et des orientations (donc les poses) de la caméra à chaque instant. Pour un géoréférencement direct, trois composantes sont importantes. La première est la composante de position obtenue le plus souvent grâce à un récepteur GNSS. La seconde est celle de l'orientation qu'on obtient à l'aide d'une centrale d'attitude. Enfin la dernière composante est celle du lever, c'est-à-dire l'instrument qui sera utilisé pour avoir des données sur la scène. Cet instrument peut être un système LiDAR qui nous permet d'avoir un nuage de points 3D via un relevé d'un scanner par exemple. Dans notre cas, l'instrument sera une caméra capable de capturer des images qui seront, par la suite, utilisées pour la formation d'un nuage de points.

Les systèmes de photogrammétrie mobiles sont de plus en plus utilisés puisqu'ils permettent d'obtenir plus rapidement les mêmes résultats que les procédures classiques de photogrammétrie sans avoir besoin de relever des cibles pour géoréférencer le nuage de points. Ainsi des constructeurs comme Leica ont conçu un backpack capable de synchroniser les données du scanner laser avec celles des images prises sur le terrain et donc de combiner les données visuelles avec la précision du nuage de points. Cet instrument est bien adapté au relevé d'un corps de rue puisqu'en se promenant, avec ce sac à dos, il est possible d'obtenir un nuage de points avec une précision relative de 2 à 3 centimètres, mais surtout une précision de 5 centimètres en absolue en un minimum de temps [Sitalia, 2022] [Leica Pegasus, 2022]. On peut également équiper une voiture de cette technologie pour couvrir une plus grande zone avec une influence non négligeable sur la précision du lever. Cependant, cette technologie, bien que très pratique, se révèle très coûteuse pour de petites et moyennes entreprises qui viennent de naître. En effet, Leica, avec son backpack Pegasus, propose à ses utilisateurs de se procurer ce dernier moyennant la somme de 200 000 €. Cette technologie est alors limitée aux grandes entreprises dans le sens où elles seules peuvent s'acheter ce genre d'appareil.

L'objectif de ce TFE est alors de développer un système de photogrammétrie mobile low-cost, c'est-à-dire de créer un prototype capable de capturer des images et de géoréférencer ces dernières avec des appareils les moins coûteux tout en gardant la plus grande précision possible. Cela implique de faire un compromis entre prix et précision des appareils. Ce travail s'inscrit dans le cadre des travaux du laboratoire, GeF, du Cnam associé à l'ESGT dans le domaine instrumental et en particulier les systèmes multi-capteurs bas coût. Cette instrumentation pourra également être utilisée dans le cadre des travaux pratiques en photogrammétrie. Ainsi ce travail a pour but de monter en compétence le laboratoire en ce qui concerne l'utilisation de systèmes multicapteurs en photogrammétrie, mais aussi dans le domaine du positionnement et plus particulièrement dans le positionnement dynamique.

L'objectif de ce TFE est donc la réalisation d'un prototype instrumental bas coût et le développement de la chaîne de traitement permettant le géoréférencement du nuage de points. Dans un premier temps, nous présenterons les différents instruments utilisés puis, dans un second temps, leur intégration dans le nouveau prototype instrumental. Ensuite, nous présenterons la chaîne de traitement mise en place pour le géoréférencement direct du nuage de points. Enfin, nous validerons cette chaîne de traitement en analysant la qualité des nuages de points obtenus.

I Les différentes composantes du système

Pour faire de la photogrammétrie mobile, plusieurs capteurs sont nécessaires. En effet, un lever photogrammétrique nécessite un appareil photographique ou une caméra afin de visualiser la scène en 3D. Cet appareil possède son propre repère (nous verrons ce dernier ultérieurement). De plus pour géoréférencer la caméra, c'est-à-dire connaître ses positions et ses orientations à chaque instant, deux capteurs sont nécessaires. Le premier est un récepteur GNSS qui permet, sous certaines conditions, de déterminer la position de son antenne et donc d'en déduire celle du centre de prise de vue de la caméra. Le second appareil est une centrale d'attitude qui permet de déterminer son orientation dans l'espace par rapport à son repère de référence. A partir des angles d'attitude estimés par la centrale, on va pouvoir en déduire l'orientation de la caméra par rapport à ce repère terrain. La Figure 1 montre ces différents repères qui viennent d'être énoncés.

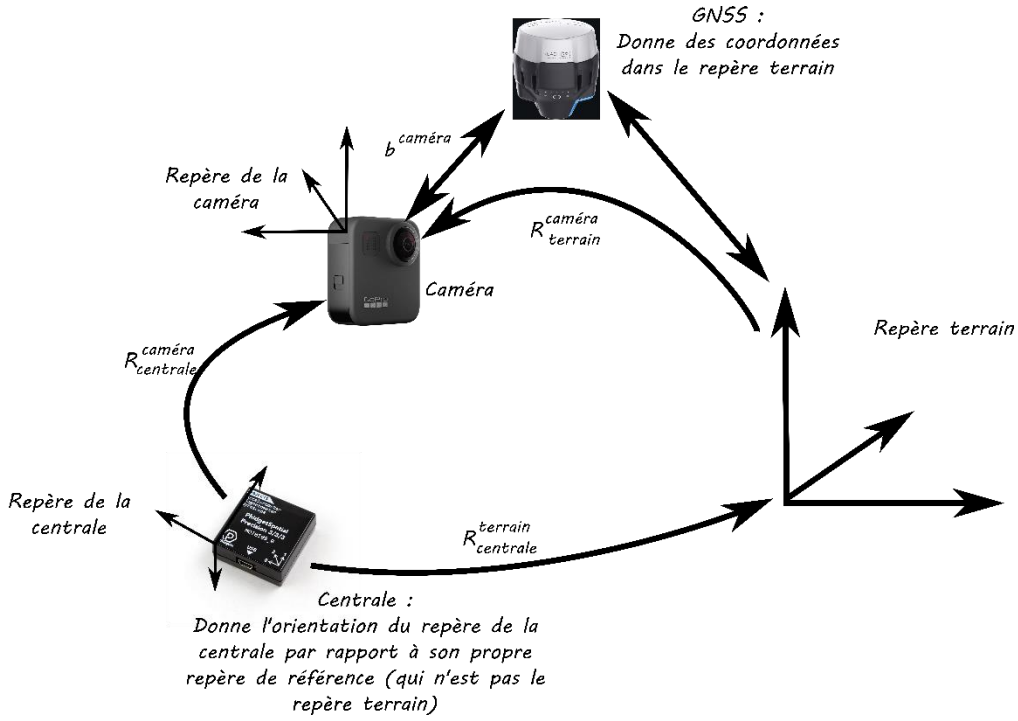


Figure 1 : Schéma des différents repères utilisés

Pour être exploitable, les trois données (position, orientation et photographie) doivent être réalisées au même instant. Il est donc nécessaire de les synchroniser ou de les dater avec le même repère temporel. Un ordinateur est alors une pièce centrale du prototype puisqu'il est indispensable pour piloter les instruments et traiter les différentes données. (cf. Figure

2). A noter ici que les instruments utilisés étaient ceux déjà disponibles au laboratoire GeF. Le choix des instruments n'est donc pas abordé dans ce travail.

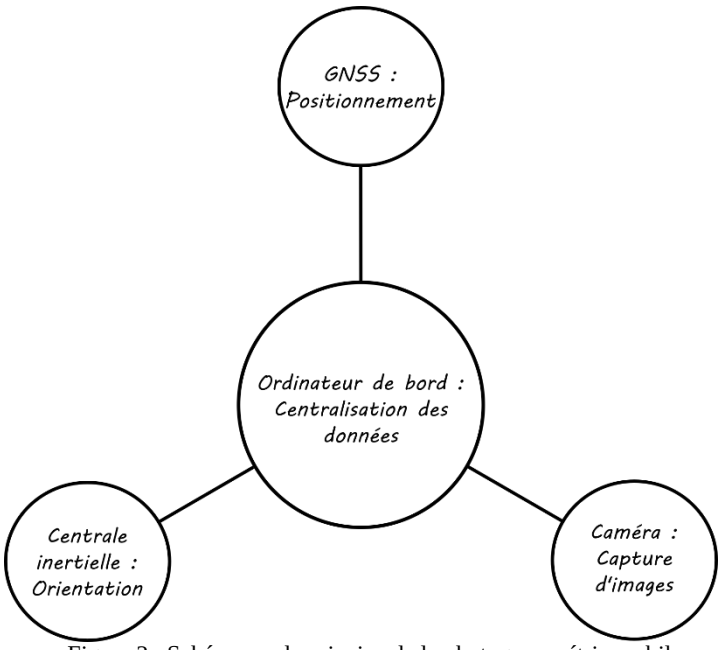


Figure 2 : Schéma sur le principe de la photogrammétrie mobile

I.1 Le récepteur GNSS : Reach RS2

Figure 3 : Reach RS2
Source : emlid.fr

Comme il a été dit plus tôt dans cette partie, un des appareils essentiels pour faire de la photogrammétrie mobile est le GNSS puisqu'il permet d'obtenir les coordonnées de l'appareil à tout instant à condition d'avoir une couverture satellitaire suffisante. Pour cette tâche, nous avons à notre disposition le récepteur-antenne Reach RS2, un GNSS développé par la société Emlid (voir Figure 3). Cette société est spécialisée dans les récepteurs GNSS dit « bas-coût » pouvant convenir à toute une gamme de professions allant de la topographie, en passant par l'agriculture ou encore l'archéologie [Emlid, 2022]. Cela est possible puisque ce GNSS peut fonctionner en mode RTK ou du NRTK à condition d'avoir un abonnement chez un fournisseur de données en temps réel tel que Teria, Orphéon ou S@tinfo. Dans le cadre de ce TFE, nous utiliserons le NRTK puisqu'il permet de s'affranchir d'une base et d'avoir les coordonnées du GNSS en temps réel ce qui nous évite de faire du post-traitement une fois le lever terminé.

Cependant, l'élément à vérifier impérativement avant de l'utiliser dans notre prototype de photogrammétrie mobile est la précision de ce GNSS. En effet, une grande partie de la précision du prototype dépend de la précision du Reach. Il est alors important de

comparer ce dernier avec la précision des GNSS dit « plus classiques ». L'information que nous donne le constructeur sur la précision en mode RTK/NRTK [Emlid, 2022] est donnée dans le Tableau 1.

Précision du GNSS en planimétrie	Précision du GNSS en altimétrie
7 mm ± 1ppm	14 mm ± 1ppm

Tableau 1: Précision du Reach RS2 donnée par le constructeur

Cette étude et ce comparatif ont déjà été effectué lors du rapport préprofessionnel mené en 2022 par des étudiants de l'ESGT [Bastard-Rosset et al, 2022]. Ce rapport donne avant tout les écarts sur les coordonnées obtenues avec Reach RS2 et avec un GNSS dit « plus classique » (Trimble R6) sous plusieurs conditions et en particulier lorsque le dispositif est en mouvement. En effet, cette étude a évalué l'influence des conditions de lever induites par les différents types d'environnement que nous pouvons rencontrer couramment. Ainsi il a été testé dans des zones dites « rurales », « périurbaines » et « urbaines ». A noter que les zones sont déterminées en fonction des indices de masquage qui sont obtenus en regardant le pourcentage de masque à l'endroit où la mesure est faite. Par exemple une zone dite « rurale » à un indice de masquage faible tandis qu'une zone dite « urbaine » à un indice de masquage de plus de 50 %. De plus, comme nous faisons de la photogrammétrie mobile, le récepteur GNSS sera en mouvement lors des mesures et il convient d'évaluer la précision de cet appareil dans les conditions d'utilisation. Les résultats obtenus en mode NRTK sont présentés dans le Tableau 2 suivant :

Milieux	Moyenne des écarts entre le Reach et le Trimble R6 (en mètres)	Écart-types (en mètre)
Rural	0.037	0.015
Périurbains	0.064	0.020
Urbains	0.815	0.040

Tableau 2 : Moyennes et écart type des écarts en fonction des zones issue du travail de [Bastard-Rosset et al]

Le Tableau 2 présente les résultats de cette étude, ils montrent que les écarts de positions entres les deux solutions restent inférieure à 7 centimètres en zones rurales et périurbaines avec un écart-type inférieur à 2 centimètres. Ces valeurs se dégradent en milieu urbain puisque la moyenne des écarts peut atteindre 80 centimètres. On peut donc en conclure, grâce à ce tableau, que le Reach RS2 reste un GNSS tout à fait convenable pour des zones où les masques sont peu nombreux, mais devient nettement moins fiable quand il s'agit de mesures effectuées dans des zones beaucoup plus couvertes. Ainsi il faut utiliser ce GNSS dans des zones où l'indice de masquage est inférieur à 50 %. Dans cette étude, notre zone correspond au jardin de l'ESGT ce qui correspond à une zone dite « périurbaine ».

I.2 La centrale d'attitude : PhidgetSpatial

Figure 4 : Centrale d'attitude Phidgets

Source : phidgets.com

La centrale attitude est un élément indispensable dans notre prototype de photogrammétrie mobile puisqu'elle permet de l'orienter et de le positionner. Nous avions à notre disposition une centrale développée par la société Phidget, une société canadienne, spécialisée dans la création de capteur, de moteur et autres appareils électroniques (voir Figure 4). L'avantage est que ces appareils peuvent être commandés par ordinateur, téléphone via un code ou via une interface graphique développée par cette même entreprise [Phidget, 2022]

Cette centrale est composée d'un gyromètre capable de mesurer le vecteur vitesse angulaire, c'est-à-dire la vitesse de variation des angles de roulis/tangage/lacet. D'un accéléromètre capable de mesurer l'accélération de l'appareil. Cette accélération, une fois diminuée de l'accélération gravitationnelle, permet d'obtenir la vitesse et la position de l'utilisateur par intégration. Enfin, la centrale est aussi composée d'un magnétomètre capable de mesurer la direction et l'intensité du champ magnétique [Cadden, 2022]. Les mesures du gyromètre, de l'accéléromètre et du magnétomètre pouvant se faire que sur un axe, il est nécessaire d'en avoir un sur chaque axe. Ainsi la centrale est composée de trois gyromètres, trois accéléromètres et trois magnétomètres.

Cette centrale permet à la fois de s'orienter dans l'espace grâce aux magnétomètres et aux gyromètres mais aussi d'avoir une position relative par rapport à la position initiale grâce aux accéléromètres. Ainsi, de premier abord, elle pourrait se suffire à elle-même étant donné qu'elle regroupe à elle seule toutes les informations nécessaires pour pouvoir calculer les poses des caméras à chaque instant. Cependant, comme l'a montré [Gomez, 2018] avec l'expérience de Woodman, la position de la centrale dérive avec le temps de manière exponentielle ce qui rend cette dernière inutilisable pour notre cas après quelques secondes. Il convient alors d'utiliser un récepteur GNSS qui fournit des positions stables dans le temps.

Pour les besoins de notre prototype, nous aurons besoin des angles d'Euler mais les seules précisions données sont celles sur l'accéléromètre, le gyromètre et le magnétomètre. Ainsi pour estimer la précision des angles nous avons procédé comme suit : la centrale d'attitude est posée à un endroit stable et on va collecter les mesures pendant une longue période (ici 15 minutes). A l'issue de cette dernière, on va calculer la moyenne et l'écart-type afin d'avoir la précision de l'appareil. Nous n'aurons malheureusement pas le biais

puisqu'on ne possède pas les vraies valeurs des angles à cet endroit. Les valeurs sont résumées dans le Tableau 3 :

	Roulis (°)	Tangage (°)	Lacet (°)
Moyenne	-36.8	-21.1	105.1
Écart-type	0.04	0.04	0.13

Tableau 3 : Écart type sur les mesures de la centrale d'attitude

La précision sur les mesures de la centrale est identique pour le roulis et le tangage mais est trois fois moins grande pour le lacet.

I.3 La caméra : GoPro

Le dernier élément essentiel du prototype est la caméra. Cette dernière va nous permettre de capturer les images indispensables pour le calcul du nuage de points. Il vient alors à nous deux choix possibles : le premier est de prendre des photos avec une certaine fréquence d'acquisition tout au long du lever. L'autre choix est de faire une vidéo qui commence juste avant le début du lever et qui se termine une fois fini. Chacun des deux cas comportent des avantages et des inconvénients. En effet, avoir directement les images de la scène permet de les traiter sans avoir recours à d'autres programmes ou logiciels. Cependant avec une fréquence d'acquisition de photo élevée, la GoPro risque de ne pas suivre la cadence imposée. Quant au choix de la vidéo, même s'il faut découper au préalable cette dernière pour extraire les images, c'est beaucoup plus simple à utiliser lors du lever puisqu'il suffit de la lancer en début et de l'arrêter en fin de lever. De plus une vidéo est une série d'images par seconde (appelé IPS). On peut alors paramétrer une vidéo pouvant aller de 30 IPS à même 60 IPS. Cela répond donc à notre problématique qui est d'avoir une fréquence d'acquisition permettant d'avoir au minimum une image par seconde voire même plus. Cependant le format des images issues d'une vidéo est plus petit provoquant une moins bonne résolution de ces dernières par rapport à des photos prisent directement. Nous avons tout de même décidé d'utiliser la vidéo afin de capturer les images qui permettront de former le nuage de points.

Nous avons à notre disposition deux types de GoPro : la HERO 5 et la Max (voir Figure 5). Ces deux caméras ont des caractéristiques communes mais avec quelques particularités. En effet, la GoPro Max possède non seulement une caméra frontale mais également une caméra qui pointe vers l'arrière. Cela permet de faire des photos ou vidéos à 360° et ainsi offrir d'autres possibilités lors du lever de la scène. Cependant, dans la suite de ce TFE, les photos et vidéos en 360° n'ont pas été étudiées. De plus, cette GoPro Max possède un mode SuperView qui permet d'avoir un champ de vision encore jamais atteint par les autres GoPro. Mais nous verrons plus tard que cette caméra, bien qu'elle offre de nouvelles possibilités, est assez difficile à piloter lors du lever notamment avec un programme.

Figure 5 : Go Pro Max
Source : gopro.com

Tous ces appareils reprennent les différentes composantes nécessaires pour un géoréférencement direct mais la nécessité de coordonner ces derniers est essentielle. Ainsi une quatrième composante apparaît avec la centralisation des données via un ordinateur de bord pour que le géoréférencement soit complet.

I.4 L'ordinateur de bord : Tablette (Getac)

L'ordinateur de bord est essentiel dans la conception d'un prototype de photogrammétrie mobile. En effet, c'est la pièce centrale permettant de lancer les mesures des appareils utilisés, de centraliser les données reçues et de les traiter. Il faut également que cet ordinateur soit léger et portatif afin qu'il soit facilement transportable. Ainsi, notre premier choix s'est tourné vers un Raspberry (voir Figure 6) après avoir lu un TFE sur le même thème [NIEDERBERGER G., 2021]. Ce dernier est un micro-ordinateur de la taille d'une carte de crédit et est composé de tous les éléments nécessaires afin de faire tourner un système d'exploitation Linux. Cela a été développé par la société Eben Upton en 2006 [KUBII, 2022]. Bien que ce micro-ordinateur corresponde parfaitement aux critères énoncés précédemment, des difficultés ont été rencontrées très tôt dans la prise en main du Raspberry. En effet avec le Raspberry, il n'était pas possible de se connecter en wifi avec la caméra. De plus, la bibliothèque python utilisée

Figure 6 : Raspberry Pi 4
Source : KIBII.fr

pour le contrôle de la centrale ne s'installait pas correctement nous empêchant d'avoir des données issues de cet instrument. Nous avons alors voulu trouver une autre solution concernant l'ordinateur de bord. Nous avons donc étudié la possibilité d'utiliser la tablette tactile Getac disponible à l'école.

La tablette tactile est développée par Getac et est sous système d'exploitation Windows 10 (voir Figure 7). Bien que cette tablette soit nettement plus lourde que le Raspberry, elle reste tout à fait portable pour de la photogrammétrie mobile. Elle comporte également un écran avec lequel on peut interagir pendant le lever ouvrant un champ des possibles plus grand qu'avec le Raspberry. Cela ne serait pas possible avec ce dernier à moins d'y ajouter un écran servant de moyen de communication avec l'utilisateur mais cela encombrerait le prototype.

Figure 7 : Tablette tactile GETAC
Source : GETAC.com

En comparant les deux solutions, on comptait des avantages et des inconvénients de chaque côté. Pour le Raspberry, bien qu'il soit la solution la plus adéquate en ce qui concerne le poids, il l'est beaucoup moins pour ce qui est de la praticité lors du lever. En effet, il faudrait lancer le programme permettant de récupérer les données dès que le Raspberry est alimenté ou alors après quelques secondes mais nous n'aurions aucune mainmise sur le déroulement du programme. C'est-à-dire que nous ne saurions pas si les appareils sont bien connectés au Raspberry, si le programme se déroule sans encombre. De l'autre côté, avec la tablette, nous avons un ordinateur de bord plus lourd que le Raspberry mais avec un avantage non négligeable puisqu'il possède un écran tactile. Ce dernier permettant de voir le déroulé du programme, voir si tous les appareils sont bien connectés, mais aussi et surtout permettant d'interagir avec l'utilisateur pour s'adapter un maximum à l'objet levé. C'est cet élément qui a fait pencher la balance du côté de la tablette tactile Getac. Ainsi pour toutes les raisons évoquées, nous avons fait le choix d'utiliser la tablette tactile comme ordinateur de bord aux dépens du Raspberry.

II Intégration des différentes composantes du prototype

Pour commencer à utiliser les appareils, il faut d'abord les agencer entre eux pour former notre prototype d'instrumentation. Il faut ensuite pouvoir les contrôler et les faire fonctionner afin de lancer nos lever. La tablette servant d'ordinateur de bord est la pièce centrale de ce prototype puisqu'elle contrôlera les instruments à distance. Cela sera possible depuis un script python qui se décompose en trois phases. La première phase concerne celle de la connexion, c'est-à-dire que l'on va s'assurer que tous les appareils soient connectés à la tablette. Nous allons ensuite initialiser les appareils et enfin commencer le lever pour l'acquisition des observations nécessaires.

II.1 Un chariot multicapteur

Dans un premier temps, nous avions à notre disposition un chariot (voir Figure 8) pouvant être déplacé à notre guise. Le récepteur GNSS devait être positionné en hauteur de manière à ce qu'aucun objet ou personne puisse gêner la réception des signaux et ainsi ne pas avoir de données à cet instant t. Cet appareil doit donc être le plus haut possible. De plus, la caméra devait être au plus près de l'objet à lever pour qu'aucun élément du chariot ne puisse créer un cache ou même assombrir la vidéo. Enfin, la centrale étant reliée par câble USB à la tablette, il faut qu'elle ne soit pas trop loin de cette dernière pour qu'elle puisse être branchée. La Figure 8 montre la configuration que nous avons choisie en fonction des paramètres énoncés précédemment. A noter que d'autres dispositions sont possibles tant qu'elles respectent ces mêmes contraintes.

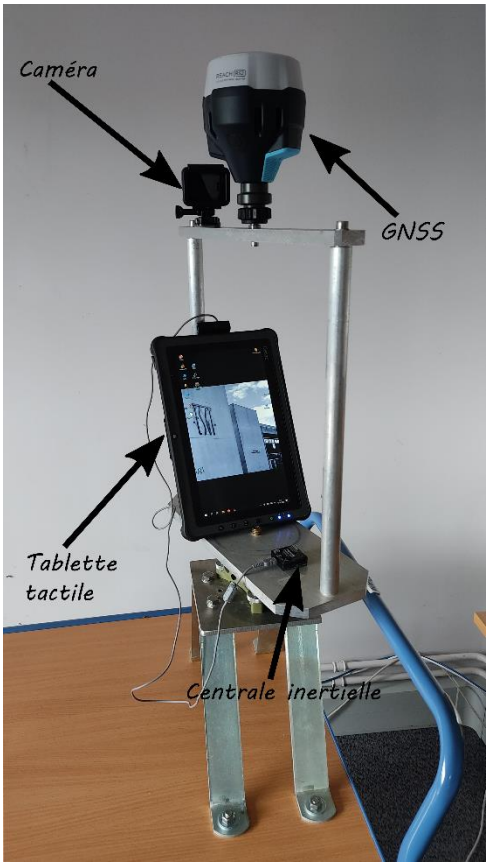


Figure 8 : Montage de tous les appareils sur un chariot

II.2 Connexion et utilisation des appareils

Cette phase va nous permettre de connecter chaque instrument à la tablette afin qu'ils puissent, *in fine*, envoyer les données à l'ordinateur de bord.

II.2.1 Pour le récepteur GNSS

La connexion au récepteur GNSS peut se faire de deux façons différentes : soit par câble USB ou soit par bluetooth. Dans le premier cas, cela impliquerait d'ajouter des ports USB à la tablette puisque celle-ci n'en est équipée que d'un seul (ce port USB étant utilisé par la centrale). Il faudrait alors avoir un HUB USB pouvant accepter au moins deux ports USB (un pour le GNSS et un pour la centrale). Cependant, ce prototype a pour vocation d'être le plus portatif possible et donc, par définition, à être le plus léger possible afin de permettre une utilisation plus facile de ce dernier. Ainsi, pour éviter d'ajouter un appareil non essentiel au prototype, il est plus judicieux de se connecter au GNSS par bluetooth.

La connexion bluetooth de la tablette vers le GNSS va se faire avec un code python (la Figure 9 illustre ce code) et plus particulièrement grâce à la bibliothèque *pybluez* [Albert Haung & contributors, 2022]. Cette dernière permet de détecter les appareils bluetooth à proximité, de s'y connecter afin de pouvoir recevoir des informations. Dans un premier temps, il est nécessaire de connaître les appareils bluetooth présents dans l'environnement proche de la tablette afin d'y trouver notre récepteur GNSS. Pour cela, nous utiliserons la fonction *bluetooth.discover_devices()* qui permet d'avoir le nom ainsi que l'adresse de connexion de tous les appareils détectés par la tablette. Après avoir choisi notre GNSS parmi ceux trouvés, nous devons apparier notre tablette au GNSS. C'est ce que la fonction *bluetooth.find_service()* nous permet de faire en lui indiquant une adresse de connexion. Il faut à présent créer un socket, c'est-à-dire une « maison » qui permet de stocker les données envoyées par le GNSS avant qu'elles soient enregistrées dans la tablette. Ainsi, nous utiliserons *bluetooth.BluetoothSocket()* afin de créer ce socket puis *.connect()* pour permettre au serveur (ici le Reach) d'envoyer ses données au client (ici la tablette) par l'intermédiaire de la « maison » (le socket). Le GNSS est connecté à la tablette et il est maintenant possible de prendre les données présentes dans ce socket

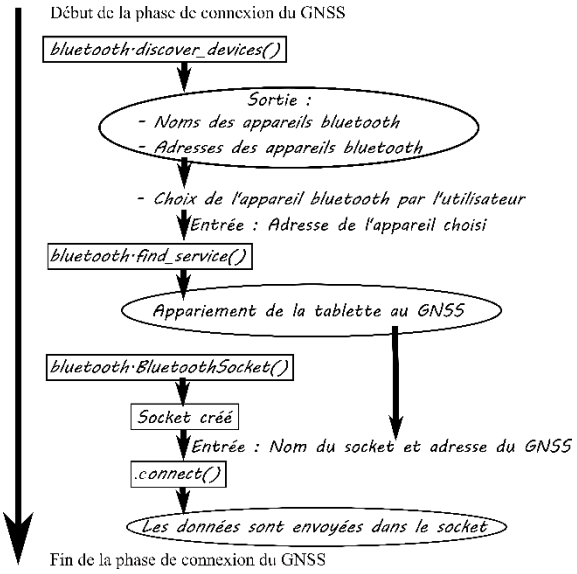


Figure 9 : Diagramme du déroulé de la phase de connexion du GNSS

grâce à *.recv()* et de les enregistrer dans un fichier texte. On peut alors résumer les trois dernières fonctions avec le schéma suivant :

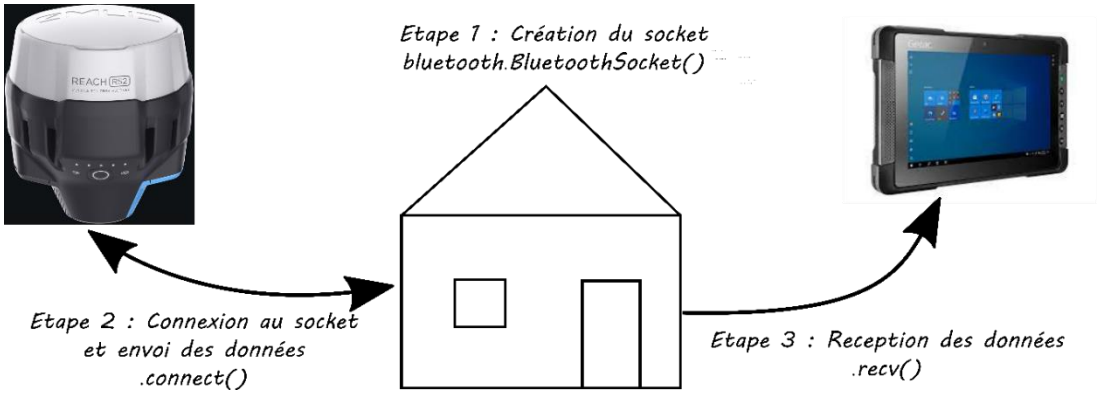


Figure 10 : Schéma pour la création et la connexion d'un socket

Une fois la tablette connectée au GNSS il faut pouvoir récupérer et traiter les données envoyées par le Reach toutes les secondes. Pour ce faire, deux solutions ont été envisagées. La première consiste à utiliser une boucle while infinie avec à l'intérieur la fonction *.recv()* qui permet de prendre les données présentes dans le socket dès qu'une donnée arrive donc toutes les secondes. L'avantage de cette méthode : peu importe le nombre d'octets entrés dans la fonction *.recv()*, toutes les données seront récoltées. Cependant, il faut utiliser une boucle infinie qui peut être contraignante pour la suite du programme. La deuxième consiste à faire appel à la fonction *.recv()* à la fin du programme, dès que toutes les données ont été enregistrées dans le socket. Cette manière est la plus pratique puisqu'on n'utilise pas de boucle infinie cependant le choix du nombre d'octets entrés dans la fonction est indispensable si l'on veut récolter l'entièreté des données du GNSS. On verra par la suite que les deux manières peuvent être utilisées avec plus ou moins de succès.

II.2.2 Pour la centrale

La connexion de la tablette avec la centrale est beaucoup plus simple puisque sur leur site internet, il existe une API (Application Programme Interface) permettant de piloter la centrale en développant un script en Python. Il existe également une application dédiée pour contrôler le fonctionnement de la centrale mais elle ne permet pas l'enregistrement des données [Phidgets Inc (1), 2022]. Dans la bibliothèque python de phidget que nous allons voir ultérieurement, il existe plusieurs classes mais une seule nous sera utile puisque nous ne voulons que les angles d'Euler. Ainsi grâce à la classe *Spatial* nous pouvons directement accéder à ces informations. Après avoir branché la centrale, ouvert un canal avec *openWaitForAttachment()* et attendu le nombre de secondes défini, la centrale est connectée à la tablette et il est alors possible d'enregistrer les données venant de cette dernière.

L'utilisation de cette centrale est alors très simple puisque, dès que le canal est ouvert et jusqu'à sa fermeture, le programme va enregistrer les données selon une fréquence d'acquisition définie au préalable.

II.2.3 Pour la caméra

La caméra va permettre d'acquérir les données visuelles nécessaires pour la formation de notre nuage de points. Pour rappel, la vidéo sera utilisée pour lever notre objet et sera ensuite découpée en images comme il a été expliqué dans la partie I.3.

La connexion entre la tablette et la caméra va se faire par wifi, plus particulièrement avec la bibliothèque *pywifi* [Awk, 2022]. La Figure 11 illustre le déroulé de ce programme. Cette bibliothèque va nous permettre de connecter la tablette à des appareils wifi disponibles aux alentours à condition d'avoir leur nom et leur mot de passe si nécessaire. Nous allons tout d'abord lancer une recherche des réseaux wifi pour détecter les instruments à proximité puis observer les appareils détectés (respectivement .scan() et *.scan_results()*). Ainsi nous pouvons choisir la caméra avec laquelle on veut établir la connexion si elle a été détectée. On va pour cela créer un profil, c'est-à-dire qu'on va renseigner, pour l'appareil précédemment sélectionné, des éléments tels que son nom et son mot de passe qui vont nous permettre de nous connecter à la caméra depuis la tablette. Une petite différence entre les deux caméras est à signaler ici et concerne la détection de ces dernières lors de la recherche des réseaux wifi par la tablette. En effet, pour la GoPro HERO 5, le simple fait de l'allumer la rend détectable par la tablette. Il faut quelque fois aller dans le menu de connexion de la caméra pour la rendre visible. Cependant, pour la GoPro Max cela est plus compliqué puisque pour pouvoir être détectée par la tablette, il faut avoir l'application GoPro Quik sur notre téléphone, connecter ce dernier à la caméra via cette application et c'est à ce moment-là que la tablette peut détecter la GoPro Max. A la fin de ce programme, la caméra est connectée à la tablette et il est alors possible de la piloter depuis la tablette.

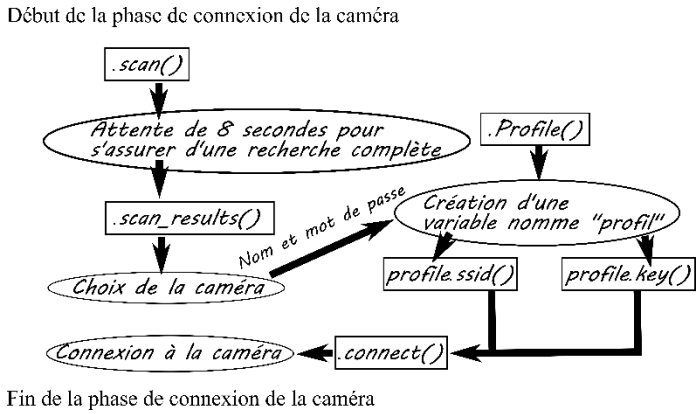


Figure 11 : Diagramme du déroulé de la phase de connexion à la caméra

II.3 Phase d'initialisation

La deuxième phase est celle de l'initialisation, c'est-à-dire qu'on va s'assurer que les instruments soient prêts à être utilisés et à envoyer des données pertinentes.

II.3.1 L'antenne/récepteur GNSS

Commençons par le récepteur GNSS qui renvoie les coordonnées de la position de l'antenne avec la qualité du signal et le temps utilisé pour obtenir cette initialisation. Pour faire cela, il faut d'abord mettre sous format Ascii les données brutes du GNSS. Ces données brutes sont transmises sous forme de trames conformes à la norme NMEA qui définit le « protocole de transmission des données entre les instruments et équipements électroniques liés au GPS. » [BAC PRO SEN EIE, 15/03/2022] (voir Figure 12).

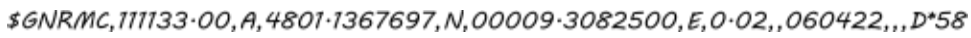


Figure 12 : Exemple de trame NMEA GNRMC après le décodage en Ascii

Une fois ces données transmises au format Ascii, nous pouvons utiliser la bibliothèque *pynmea* et notamment la fonction *.parse()* qui permet de lire ces trames et en sortir des valeurs disponibles (par exemple pour la

Figure 12 on peut voir que l'on peut ressortir les coordonnées géographiques du GNSS à l'instant t). Pour notre projet de photogrammétrie mobile, les données essentielles se situent principalement dans la trame GPGGA puisque dans cette dernière, on retrouve les coordonnées géographiques du GNSS, la qualité du signal, le nombre de satellites visibles, l'affaiblissement de la précision horizontale (HDOP : Horizontale Dilution Of Precision) ainsi que l'altitude orthométrique accompagnée de l'ondulation du géoïde. Une autre trame peut nous être utile et il s'agit de la trame GPVTG car on peut avoir, grâce à elle, la vitesse au sol de l'instrument. Cela peut être utilisé pour avertir l'utilisateur qu'il va trop vite ou tout simplement pour montrer la vitesse et ainsi garder une vitesse constante. A ce niveau-là nous avons les coordonnées géographiques de l'appareil mais pour que ces coordonnées soient plus parlantes, il est nécessaire de les transformer en coordonnées planes et plus précisément dans le système RGF93 avec la projection Lambert93. On utilise pour cela la bibliothèque *pyproj* ainsi que ces fonctions [Pyproj, 2022]. Tout d'abord il faut choisir le système de coordonnées de référence (CRS) du Lambert93 avec *.CRS.from_esgp()* en indiquant l'EPSG du Lambert93 (2154). De plus, comme on souhaite transformer les coordonnées géographiques en coordonnées planes, il faut le CRS géographique du Lambert93. Pour cela on utilise la fonction *crs_lambert93.geodetic_crs* qui nous permet

d'avoir le bon CRS sans donner un EPSG. Ainsi on peut déterminer la transformation pour effectuer le changement de système de coordonnées de l'un à l'autre avec la fonction *Transformer.from_crs()*. On peut ensuite l'appliquer sur l'ensemble des coordonnées en question. On obtient alors les coordonnées Est et Nord de l'antenne ainsi que sa hauteur ellipsoïdale. Toutes ces étapes sont essentielles dans l'étape d'initialisation et on les retrouvera dans le traitement des données du lever.

II.3.2 La caméra

Concernant la caméra, une vidéo sera prise de quelques secondes avec les paramètres que l'utilisateur aura choisi directement sur la caméra. Il aura alors le choix de continuer avec cette configuration ou de recommencer cette phase jusqu'à trouver la bonne configuration. C'est ici que la GoPro Max est plus difficile à utiliser puisque pour pouvoir initialiser la caméra, il faut au préalable pouvoir la contrôler à distance avec un programme. Pour cela, on utilise la bibliothèque *goprocam* et surtout la fonction *GoProCamera().GoPro()* qui, une fois attribuée à une variable, permet d'utiliser la caméra [Konrad, 2022]. Cependant, avec la GoPro Max, cette fonction ne fonctionne pas à tous les coups empêchant la suite du programme de se dérouler sans qu'on ait réussi à en identifier les raisons.

II.3.3 La centrale d'attitude

Enfin, l'initialisation de la centrale va consister à préchauffer la centrale à l'aide d'un circuit électronique jusqu'à la température à laquelle elle a été étalonnée (45°C) pour pouvoir corriger les mesures des différentes erreurs systématiques [Phidgets Inc. (2), 2022]. Cette étape est la plus compliquée à programmer puisque dans la bibliothèque *Phidget*, la fonction permettant de mettre la centrale à la bonne température se situe dans la classe *Spatial()* avec la fonction *setHeatingEnabled(True)* alors que la fonction permettant de connaitre la température (*getTemperature()*) de la centrale se situe dans la classe *TemperatureSensor()*. De plus, pour se connecter à la centrale, il faut ouvrir un canal qui va nous permettre de communiquer avec elle. Or, pour ouvrir un canal, il faut le faire soit avec la bibliothèque

Spatial() ou dans celle de *TemperatureSensor()* et il n'est pas possible d'ouvrir deux canaux pour un seul phidget. Une solution trouvée (et est illustrée dans la Figure 13) consiste à ouvrir d'abord le canal avec la classe *Spatial()* pour utiliser la fonction nécessaire, puis basculer rapidement sur l'autre classe afin de connaître la température et recommencer ainsi de suite.. La centrale monte alors à la bonne température pour être utilisée (45°C) et la phase d'initialisation est terminée.

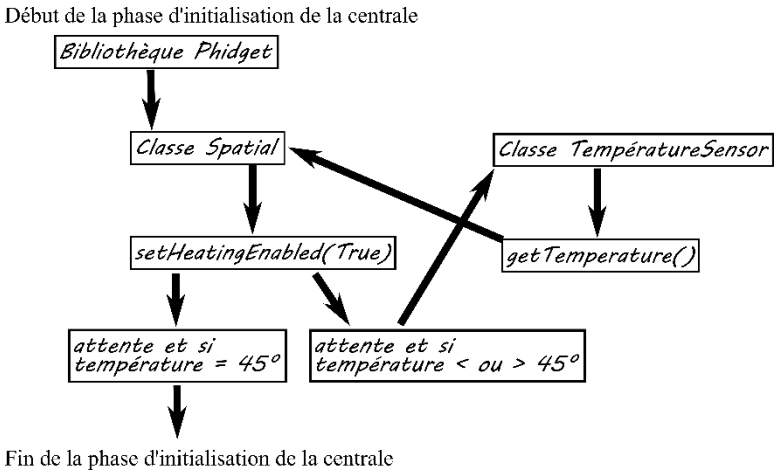


Figure 13 : Illustration de la phase d'initialisation de la centrale

II.4 Phase de lever

La troisième phase est celle qui nous intéresse le plus puisqu'il s'agit de celle du lever. La plus grande difficulté de cette phase est d'avoir au même instant t les données de chaque appareil. C'est ce qu'on appelle la synchronisation temporelle. Cependant chaque appareil a une fréquence d'acquisition différente ce qui rend la synchronisation compliquée. Pour cela deux manières sont envisageables : la première consiste à débuter les observations de chaque instrument au même instant t et ensuite dater toutes les observations sur la même échelle temporelle. La deuxième méthode consiste à débuter les observations de chaque instrument à des instants différents et ensuite dater toutes les observations sur la même échelle temporelle. A noter ici que c'est les instants d'exécution des commandes des différents instruments qui sont utilisés comme début de lever de chaque instrument. D'autres méthodes peuvent être envisagées mais par manque de temps, cette recherche n'a pas été faite ici.

II.4.1.1 Déclenchements simultanés des instruments de mesures

Dans un premier temps nous allons voir comment déclencher les observations des trois instruments à partir d'un seul programme principal python. Avec la bibliothèque *asyncio*, il est possible de lancer plusieurs programmes de manière quasi simultané [Python, 2022]. Pour cela, il faut que ces derniers ne comportent pas de boucle infinie qui empêcherait de faire tourner le programme principal. Cela est possible puisque dans le programme de

lever de la centrale, on stocke les données dès qu'elles arrivent sans faire de boucle infinie. Pour celui de la caméra, on lance simplement la vidéo et enfin pour celui du GNSS, on va stocker les données dès que le lever est fini. Dans aucun des cas nous utilisons donc de boucle infinie. Notre instant 0 est donc le même pour chaque appareil mais il faut régler le problème dû au fait que les instruments n'ont pas la même fréquence d'échantillonnage. Pour cela comme l'instant 0 est le même pour chaque appareil, il suffit de trouver une fréquence commune à chaque instrument. La vidéo à une fréquence variable entre 30 et 60 images par secondes donc potentiellement 30 à 60 données par seconde. La centrale à une fréquence d'acquisition pouvant aller de 1 à 250 données par seconde. Enfin les données du GNSS sont obtenues toutes les secondes et c'est donc lui qui a la fréquence la plus basse. Ainsi pour une synchronisation des données avec ce programme, il faut prendre les données toutes les secondes à partir de l'instant 0. Ce choix nous affranchit dans un premier temps d'interpoler les données du GNSS mais nous nous assurerons, plus loin, d'un recouvrement d'au moins 80 % entre images pour la photogrammétrie. Comme le montre la Figure 14, les données qui seront utilisées pour l'ajustement de notre prototype sont celles obtenues à l'instant t où les trois instants d'acquisition se chevauchent.

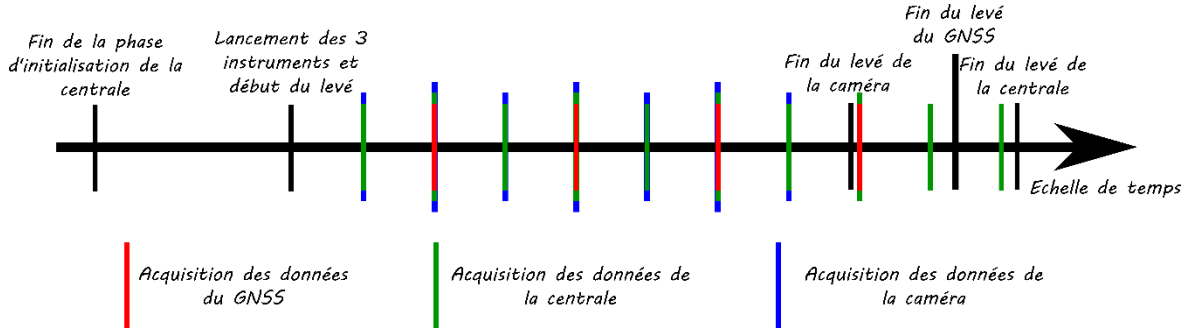


Figure 14 : Schéma illustrant la synchronisation temporelle

Un premier problème est apparu dans la synchronisation au moment des lancements des appareils et plus précisément lors le lancement de la caméra. En effet, de premier abord, on lance les mesures de la centrale et du GNSS en même temps qu'on lance la vidéo. Cependant, la fonction permettant de lancer la vidéo est trop longue à s'exécuter (en moyenne 0.7s) ce qui décale le début de la vidéo par rapport à celui de la centrale et du GNSS. Ce résultat a été obtenu en mettant la fonction *time.time* avant et après avoir lancé la commande permettant de lancer la vidéo. Pour résoudre ce problème, la solution trouvée est de lancer la vidéo puis de placer un *hilight* dans la vidéo qui correspondra au moment du lancement des données du GNSS et de la centrale. Un *hilight* est un point de repère que l'on place n'importe quand sur la vidéo et qui permet de retrouver plus facilement cet instant choisi. Le vrai début de la vidéo correspondra alors, à l'instant du point de repère et il suffira

de couper la vidéo jusqu'à ce moment-là. Le temps de réponse de la fonction permettant de placer un point de repère étant nettement moins long (0.04s au lieu de 0.7s), la synchronisation temporelle ne sera que meilleure. Cependant, avec l'ajout de cette fonction, il est nécessaire d'utiliser la fonction *getClip()* de la bibliothèque *goprocam* permettant de ne sélectionner qu'une partie de la vidéo. Néanmoins cette fonction ne fonctionne pas sur les vidéos prises avec la GoPro Max rendant l'utilisation de cette caméra, dans ce programme, plus difficile. Pour synchroniser les données il ne faut donc plus prendre le début de la vidéo comme instant 0 mais bien l'instant où le point de repère est placé. La Figure 15 illustre la solution énoncée.

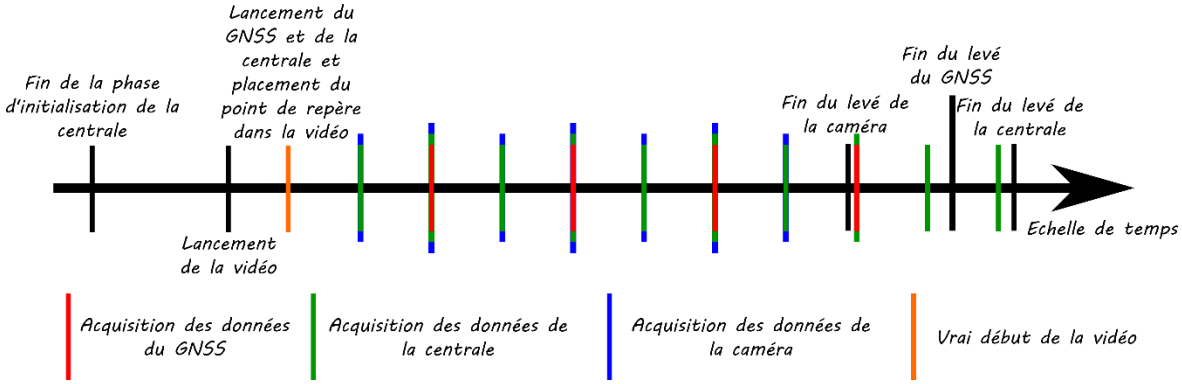


Figure 15 : Schéma illustrant le vrai début de la vidéo

Un autre problème, et pas des moindres, apparait lorsque le lever dure plus d'une minute trente. En effet, quand cette durée est dépassée, le GNSS ne stocke pas toutes les données et on se retrouve avec un fichier comportant la première minute de donnée et les 2-3 dernières mesures. Cela est sûrement dû au fait que le socket servant de stockage des données du GNSS n'est pas assez grand pour recevoir autant de données et qu'elle ne garde que les premières reçues ainsi que 2-3 mesures finales. Ce problème n'a pas été résolu et nous force donc à utiliser la deuxième méthode énoncée dans la partie II.2.1. Elle consiste à utiliser une boucle infinie pour stocker les données du GNSS toutes les secondes. Il n'est alors plus possible d'utiliser la bibliothèque *asyncio* avec cette boucle while et il faut alors trouver un autre moyen de lancer simultanément les programmes. Il est tout de même possible d'utiliser ce programme mais pour un lever de moins d'une minute trente.

L'autre manière pour pouvoir lancer en simultané les différents programmes de chaque appareil tout en utilisant une boucle infinie est d'avoir un programme pour le fonctionnement du GNSS et un autre pour celui de la centrale et de la caméra. Mais pour ne pas risquer d'inverser des parties de codes de la caméra et de la centrale, nous avons décidé d'avoir un code distinct pour chaque appareil. Ainsi avec ces trois scripts, nous allons lancer de manière simultanée les différents instruments mais cette fois depuis trois programmes

distincts. Maintenant que chaque programme se lance indépendamment les uns des autres, il est possible de faire une boucle infinie sans pour autant gêner le déroulement des autres programmes. Ensuite, l'idée pour un lancement simultané malgré trois programmes distincts est de dire à ces derniers de commencer le stockage des données à partir d'une date déterminée. En effet à la fin de la phase d'initialisation, on va dire au programme pilotant le récepteur GNSS et de la centrale « d'attendre » une date et plus précisément une seconde précise. Cette seconde nous sera donnée par le code pilotant la caméra une fois la phase d'initialisation terminée. En effet, à la fin de la phase d'initialisation de la caméra, on va demander la date actuelle, extraire la seconde correspondante et y ajouter un certain nombre de secondes (par exemple 5s). On va ensuite enregistrer cette donnée dans un fichier texte. Ce fichier texte sera lu par les trois programmes et une fois cette seconde atteinte ils se lanceront quasiment en même temps. L'instant 0 est alors le même pour chaque appareil et résout ainsi notre problème de synchronisation temporelle *via* un lancement simultané. En ce qui concerne les différentes fréquences d'acquisition, on fait le même procédé que celui énoncé dans le paragraphe précédent. La Figure 16 illustre le principe venant d'être énoncé. A noter que l'échelle de temps des trois instruments doit être la même pour tous les appareils.

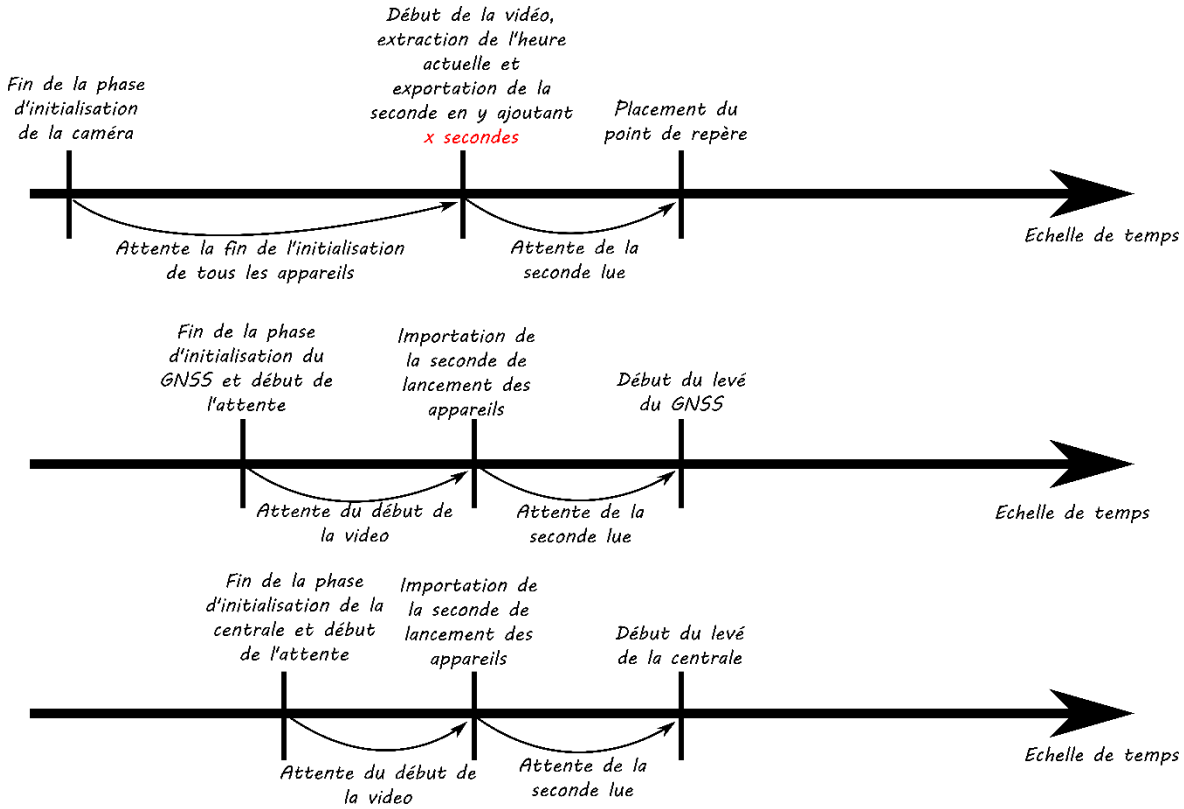


Figure 16 : Principe du lancement simultané des instruments avec 3 programmes

L’avantage de cette méthode est qu’il n’y a pas de temps limite pour faire le lever, toutes les données seront acquises. Cependant comme il y a trois programmes pour faire fonctionner le prototype, il est plus difficile à prendre en main.

II.4.1.2 Déclenchements différés des instruments de mesures

Les problèmes énoncés précédemment concernant la caméra et le GNSS restent toujours présents avec un lancement différé. Cela nous oblige alors à ne pas utiliser le lancement de la vidéo comme instant 0 de la caméra mais bien un point de repère comme « vrai » début de vidéo. Il faut également utiliser trois programmes distincts pour les trois appareils. Le principe de cette méthode est simple puisqu’en théorie il faut dater le commencement de la phase de lever pour chaque appareil afin de, *in fine*, retrancher les informations où les trois appareils ne sont pas déclenchés en même temps. Pour cela, on peut utiliser la fonction *time.time()* de la bibliothèque *time* afin d’avoir le moment du déclenchement des différents appareils lors de la phase de lever. Ainsi à la fin de cette phase, en plus des données brutes des appareils nous avons également les instants où les appareils ont été déclenchés. Comme nous voulons que la caméra soit le dernier appareil à être démarré, nous allons faire la différence entre le temps de début du GNSS avec celui de la caméra ainsi que celui entre la centrale et la caméra. Nous pouvons alors retirer autant de données du GNSS/centrale qu’il y a de secondes d’écart.

Un problème apparaît quand nous nous retrouvons avec un écart entre le temps du GNSS/centrale et celui de la caméra inférieure à une seconde. En effet il reste quelques microsecondes d’écart qui peuvent faire la différence lorsqu’on va vouloir calculer le bras de levier et la matrice de passage. Il faut minimiser au maximum l’écart de temps entre le déclenchement des différents appareils pour avoir une meilleure précision. En ce qui concerne la centrale, rien de plus simple puisqu’il suffit de réduire l’intervalle de temps entre les mesures lors du lever pour pouvoir réduire au maximum l’écart de temps entre les lancements. Sachant que l’intervalle minimal entre deux mesures est de 4 millisecondes, il est possible de jouer sur ce curseur. Pour ce qui concerne l’écart de temps entre le GNSS et la caméra, cela se complique. En effet, les données du GNSS sont obtenues toutes les secondes et donc après avoir retiré autant de données que de secondes d’écart, il n’est pas possible de diminuer cet écart au niveau des données GNSS. La solution trouvée est alors de ne pas couper la vidéo depuis le point de repère mais plutôt de couper la vidéo depuis le point de repère diminué de l’écart restant entre le GNSS et la caméra. En faisant cela, l’écart entre les données GNSS et celles de la caméra, juste après la fin du lever, est de x secondes

pile. Cela permettra de synchroniser les données du GNSS et de la caméra dès que les x premières données du GNSS seront retirées. La figure suivante explique le procédé décrit :

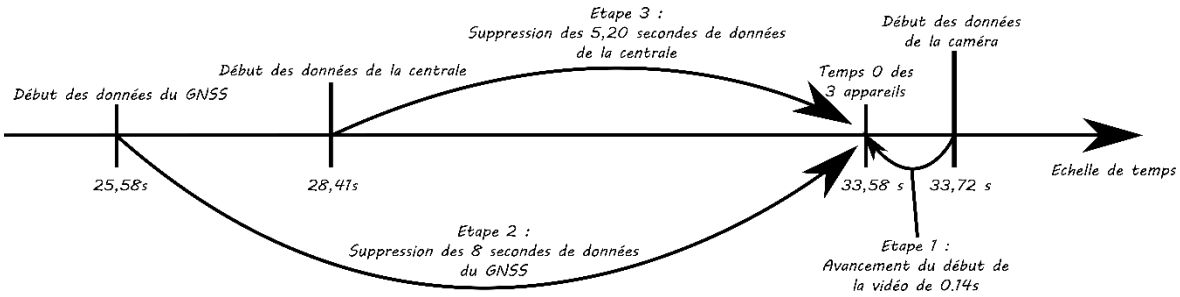


Figure 17 : Principe de la synchronisation en temps différé

Il est important de noter que le début du lever de la caméra qu'on retrouve dans cette Figure 17 (noté « Début des données de la caméra ») ne correspond pas au vrai début de la vidéo mais bien à l'instant où le point de repère est placé. Le début de la vidéo a proprement parlé se situe quelques secondes avant. Les données du GNSS ont commencé à être acquises au bout de 25,58 secondes et la vidéo a commencé au bout de 33.72 secondes. Nous observons qu'il y a 8,14 secondes entre le début des données GNSS et des données de la caméra et qu'une fois les 8 premières données du GNSS retirées, il ne reste plus que 0.14 seconde. La première étape est alors de mettre le début de la vidéo non pas au moment du point de repère (ici ce dernier est à 33,72 secondes) mais au moment du point de repère auquel on aura retranché les 0.14 seconde. Ainsi le début de la vidéo sera à 33.72-0.14=33.58 secondes. Après nous pouvons donc enlever les 8 premières données du GNSS correspondant aux 8 secondes écoulées entre les débuts d'acquisition des deux instruments. En ce qui concerne la synchronisation de l'instant 0 de la centrale et de la caméra (en prenant comme hypothèse que la fréquence d'acquisition choisie pour la centrale est de 10 Hz) on peut retirer les 50 premières mesures correspondant au 5 secondes séparant les deux déclenchements. Le début des mesures de la centrale est alors à 33.41 secondes. Comme la fréquence d'acquisition de la centrale correspond à une mesure toutes les 10 millisecondes, on peut retirer 2 mesures de la centrale ce qui amène le début des mesures de la centrale à 33.61 et il y aura donc un délai de 0.03 milliseconde entre les deux lancements. On aurait pu interpoler les données de la centrale pour combler ce délai mais pour nos premiers tests et pour ne pas ajouter des sources d'imprécision, nous avons décidé de ne pas interpoler les mesures. On aura donc retiré les 52 premières mesures de la centrale afin de synchroniser au mieux les instants 0 des instruments. Mais un résidu de 0.03 seconde persiste et ne peut être réduit que si la fréquence d'acquisition de la centrale est augmentée. En ce qui concerne la synchronisation des mesures des différents instruments, il suffit de prendre les instants t où les trois instruments fournissent une donnée (toutes les secondes ici).

Bien que ces programmes permettent une durée de lever quelconque, la synchronisation de l'instant 0 n'est pas parfaite ce qui peut entrainer une perte de précision lors du calcul du bras de levier et de la matrice de passage. Il faut également trois programmes pour faire fonctionner le prototype ce qui est plus compliqué à prendre en main contrairement à la première méthode énoncée dans la partie II.4.1.1 qui nécessite un seul programme.

III Acquisition et traitement des données

Le géoréférencement direct nécessite de déterminer les poses (positions et orientations) de la caméra. Cependant notre instrumentation ne permet d'obtenir que la position de l'antenne GNSS et l'orientation de la centrale. Une étape d'ajustage est donc nécessaire pour déterminer les poses de la caméra. L'ajustage du prototype va permettre de déterminer le bras de levier entre le centre de phase de l'antenne GNSS et l'origine du repère caméra ainsi que la matrice de passage du repère de la centrale au repère de la caméra appelé matrice de boresight. Ces paramètres ne changeront pas tant que les instruments resteront rigidement fixés au chariot sinon il faudra les ajuster à nouveau. Une fois ces deux éléments déterminés, on peut passer au lever à proprement parlé qui va permettre de créer un nuage de points de la scène. Ces deux phases vont être énoncés dans les parties suivantes.

III.1 Phase d'ajustage du prototype

Dans cette partie, nous verrons d'abord les calculs permettant d'estimer le bras de levier et la matrice de passage

III.1.1 Théorie

Pour comprendre ce qu'est le bras de levier et la matrice de passage il faut d'abord énoncer l'équation de géoréférencement. Comme il a été énoncé plus tôt, le géoréférencement repose sur trois composantes : celles de position, d'orientation et de lever. Avec ces dernières, il est possible de construire l'équation de géoréférencement qui permet de connaître la position et l'orientation (c'est-à-dire la pose) de la caméra à chaque image dans le repère de référence. Dans le cas présent, le repère de référence est le repère dans lequel sont exprimées les coordonnées de l'antenne GNSS et qui sera nommé repère terrain.

La Figure 18 nous montre le chemin à suivre afin de connaître les poses pour chaque images prises par la caméra à partir des donnés du GNSS et de la centrale.

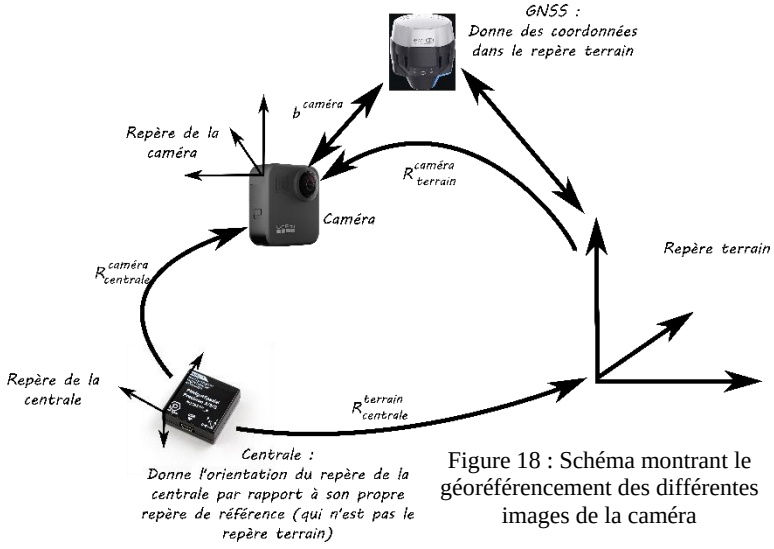


Figure 18 : Schéma montrant le géoréférencement des différentes images de la caméra

Cela va alors nous permettre de former l'équation de géoréférencement. Notre but est alors d'avoir les positions des images de la caméra dans le repère terrain : $r_{caméra}^{terrain}$. Pour cela nous avons les coordonnées du centre de phase de l'antenne GNSS toujours dans le repère terrain : $r_{GNSS}^{terrain}$. Nous pouvons alors écrire que :

$$r_{caméra}^{terrain} = r_{GNSS}^{terrain} + b^{terrain} \quad (1)$$

Avec $b^{terrain}$ un vecteur allant du centre de phase de l'antenne GNSS au centre de prise de vue de la caméra. Ce vecteur définit le bras de levier et permet d'obtenir la position de la caméra à partir de celle de l'antenne GNSS. Cependant ce bras de levier va changer à chaque instant dans le repère terrain et il faut donc le transformer dans le repère de la caméra pour qu'il soit constant quel que soit l'instant t (2).

$$b^{terrain} = R_{cam}^{terrain} * b^{cam} \quad (2)$$

Comme on peut le voir avec la Figure 18, il faut utiliser la matrice de passage du repère caméra vers le repère terrain et ainsi obtenir un bras de levier dont les coordonnées exprimées dans le repère caméra sont invariantes au cours du temps. A noter que le repère de la caméra se définit comme suit : l'axe de la coordonnée x est dirigé vers la droite de la caméra, celui de y est dirigé vers le haut et enfin l'axe de la coordonnée z vers la direction opposée à la direction de visualisation de la caméra. Ainsi l'utilisation de l'équation (2) avec l'équation (1) donne :

$$r_{caméra}^{terrain} = r_{GNSS}^{terrain} + R_{cam}^{terrain} * b^{cam} \quad (3)$$

Or dans cette équation (3), la matrice de passage de la caméra vers le terrain nous est inconnue lors du lever. Il faut trouver un moyen de la déterminer pour chaque image. Pour cela, on peut dire que cette matrice peut être décomposée en deux autres matrices. La première c'est la matrice de passage entre la centrale et le terrain puis la seconde est celle entre caméra et la centrale (4).

$$R_{centrale}^{terrain} * R_{cam}^{centrale} = R_{centrale}^{terrain} * R^{boresight} = R_{cam}^{terrain} \quad (4)$$

La première matrice permet de passer du repère de la centrale vers celle du terrain et nous sera donnée grâce à la centrale d'attitude. Le repère de la centrale est défini comme indiqué sur la Figure 19 :

Figure 19 : Repère de la centrale d'attitude

L'axe des x est dirigé vers la gauche de l'instrument, l'axe des y vers le devant de la centrale et enfin l'axe des z vers le bas de l'instrument. Le repère de référence de cette centrale étant le repère défini comme suit : l'axe des x pointe vers le Nord magnétique, l'axe des y vers l'Est magnétique et l'axe des z pointe vers le bas. Les angles de roulis, tangage, lacet fournis par la centrale permettent de construire la matrice de passage entre le repère de la centrale et ce repère précédemment défini. Lorsque les repères sont alignés, les angles de rotations sont nuls. Ainsi on verra ultérieurement qu'il faut ajouter d'autres rotations afin d'avoir la matrice de rotation de la centrale vers le terrain.

On appelle alors la deuxième matrice énoncée, la matrice de boresight puisqu'elle ne change pas quel que soit l'instant t du lever [NIEDERBERGER G., 2021]. On peut alors écrire l'équation (3) comme suit :

$$r_{caméra}^{terrain} = r_{GNSS}^{terrain} + R_{centrale}^{terrain} * R^{boresight} * b^{cam} \quad (5)$$

On obtient alors l'équation (5) qui est l'équation de géoréférencement contenant le bras de levier (b^{cam}) ainsi que la matrice de passage cam/centrale ($R^{boresight}$). Dans cette équation, le prototype instrumental nous donne les coordonnées de l'antenne GNSS dans le repère terrain ($r_{terrain}^{GNSS}$) ainsi que la matrice de passage centrale/terrain ($R_{centrale}^{terrain}$). Il nous faut alors déterminer les deux autres inconnues puisque peu importe l'instant t, ces deux éléments seront constants.

C'est alors là qu'intervient le lever d'ajustage puisqu'il va permettre de déterminer à la fois le bras de levier et la matrice de boresight. Pour cela on utilise l'équation (2) pour le bras de levier et l'équation (4) pour le calcul de la matrice de boresight :

$$R_{terrain}^{cam} * b^{terrain} = b^{cam} \quad (2)$$

$$R^{boresight} = R_{terrain}^{centrale} * R_{cam}^{terrain} \quad (4)$$

Dans ces deux équations, il y a une matrice qui revient et c'est la matrice de passage du repère terrain vers le repère caméra (ou l'inverse, il suffit de prendre sa transposée puisque comme les deux repères étant orthonormés, les matrices de passage vérifient l'équation suivante : inverse(R)=transposé(R)). Il faut donc l'orientation du repère caméra par rapport au repère terrain afin de déterminer cette matrice. De plus, d'après l'équation (1) il nous faut connaître la position de la caméra pour déterminer le bras de levier $b^{terrain}$dans le repère terrain. Cette matrice de passage terrain/caméra et la position de la caméra sont déterminées à l'aide du logiciel Metashape et de points d'appui.

III.1.2 Utilisation de Metashape pour le calcul des poses

Metashape est un logiciel permettant notamment de calculer les poses des images importées grâce à des points d'appui. La première étape est d'importer la vidéo dont on veut extraire les images. Il suffit alors de choisir la fréquence à laquelle on veut les photos ainsi que le début et la fin de la série. A savoir que pour nos tests, nous avons décidé de n'avoir qu'une photo par seconde afin d'avoir la même fréquence d'acquisition que celle du GNSS. Une fois cela fait, on peut détecter les cibles Metashape utilisées lors du lever et importer leurs coordonnées mesurées dans le repère terrain. On peut maintenant lancer un « alignement des images » de Metashape qui permet d'estimer les poses des images. Cette étape d'alignement des photos est cruciale puisqu'elle permet d'estimer au mieux les paramètres externes de la vidéo (donc les poses) ainsi que les paramètres internes de la caméra y compris les distorsions par une approche Structure From Motion [Agisoft, 2022]. Une fois cette étape terminée, nous avons pour chaque image, sa position dans le repère dans lequel sont exprimées les coordonnées des points d'appui (pour notre cas, il s'agit du repère terrain) mais également l'orientation des images par rapport ce même repère. L'orientation des images est exprimée selon trois angles de rotation que sont oméga, phi et kappa ou encore roulis, tangage et lacet. Ces angles permettent de passer du repère terrain au repère de la caméra. [Argisoft, 2022].

On peut utiliser les positions et les orientations pour nos calculs de bras de levier et de matrice de boresight. Maintenant que la théorie est en place, on peut l'appliquer à notre prototype instrumental en suivant le protocole suivant.

III.1.3 Protocole

Nous allons d'abord voir les différentes recommandations afin de réaliser les meilleurs lever possibles.

III.1.3.1 Condition de lever

Pour un lever réussi, il est nécessaire de respecter plusieurs règles afin d'avoir en sortie les données les plus exploitables possible. Tout d'abord, il est important de vérifier que le trajet suivi lors du lever ait un accès au ciel dégagé, qu'aucun arbre ne se trouve sur la route et que le chemin emprunté ne traverse pas de chemin couvert. En effet, la position de la caméra étant déterminée par des données GNSS, il est important d'avoir un accès au ciel assez important tout au long du lever pour avoir une détermination de la position la plus

fiable possible. Un saut dans les données du GNSS nous empêcherait de déterminer les coordonnées de la caméra, autrement que par interpolation des mesures GNSS, ce qui nous empêcherait de géoréférencer le nuage de points de manière précise sur la zone concernée.

Ensuite, pour obtenir les différentes images de la caméra, il a été dit dans la partie I.3 que nous utiliserons une vidéo qui sera découpée avec un pas défini par l'utilisateur. Ainsi lors du lever et plus précisément lors du déplacement des appareils il est important de ne pas aller trop vite afin d'éviter l'effet du filé sur les images. Il faut alors adapter sa vitesse de croisière pour éviter cet effet et donc ne pas aller trop vite. Cet effet dépendant de la focale, de la distance qui sépare l'appareil photo de l'objet à lever, la dimension du photosite et la vitesse d'obturation. La vitesse de croisière va alors dépendre de ces éléments. Mais il faut également avoir une vitesse suffisante pour ne pas avoir trop d'images de la même scène (avoir un recouvrement entre deux images de plus de 85 %) ce qui augmenterait le temps de traitement inutilement sans pour autant apporter d'informations supplémentaires. Une vitesse constante est un élément également à prendre en compte.

Ces conseils doivent d'autant être suivis s'il s'agit du lever servant à la détermination du bras de levier et de la matrice de passage. En effet, de ce lever va dépendre la qualité et la précision des deux éléments précédemment cités et donc de la qualité du nuage de points de l'objet levé. On peut alors ajouter d'autres recommandations en ce qui concerne ce genre de lever. Tout d'abord concernant les positions des cibles servant de points d'appui ou de contrôle qui vont nous permettre de déterminer les poses des caméras avec Metashape, il faut qu'elles soient :

- réparties de manière uniforme sur l'ensemble du mur servant de mur d'ajustage en couvrant aussi les bords du lever ;

- en nombre suffisant pour pouvoir estimer au mieux les poses des images.

Enfin, il est conseillé d'avancer de manière parallèle au mur. Avec toutes ces recommandations, il est alors possible de commencer son lever aussi bien pour ajuster le prototype ou simplement pour l'utiliser et produire un nuage de points 3D.

Ainsi dans notre cas, nous avons fait notre lever d'ajustage sur le mur de la salle de photogrammétrie en positionnant nos points d'appui et des points de contrôle comme le montre la Figure 20.

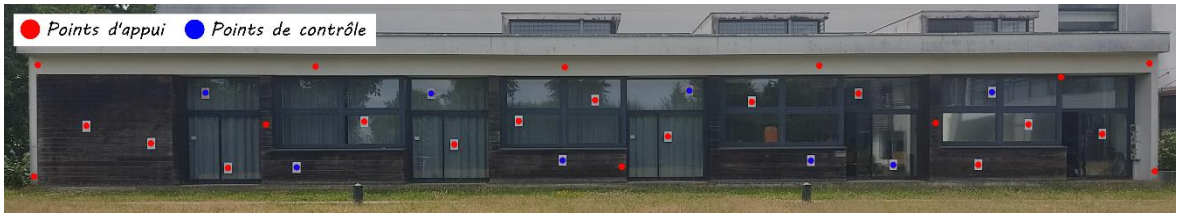


Figure 20 : Positionnement de points d'appui et de contrôle sur le mur de photogrammétrie

III.1.3.2 Exploitation sur Metashape

Une fois le lever terminé et les données enregistrées dans la tablette, il nous faut calculer le bras de levier et la matrice de passage. Comme nous l'avons vu dans la partie III.1.1, nous avons besoin des coordonnées de l'antenne GNSS obtenues à la fin du lever, des angles d'attitude déterminés par la centrale Phidget ainsi que des poses de la caméra. Ces poses ne sont pas encore connues et pour pouvoir les déterminer, nous allons utiliser Metashape comme il a été annoncé dans la partie III.1.2.

Ce logiciel permet d'importer une vidéo pour qu'elle soit découpée en une séquence d'images. Pour nos premiers tests, nous avons paramétré le logiciel afin que la séquence d'images ait la même fréquence que les données du GNSS (en l'occurrence la fréquence est de 1 Hz). Pour le calcul des poses de la caméra pour chaque image, il a fallu poser des cibles sur le mur qui seront soit des points d'appui ou soit des points de contrôle. Deux types de cible sont utilisés pour géoréférencer le nuage de points :

- la première sont des cibles rouges et blanches fixées sur le bâtiment dont les coordonnées sont connues suite à un lever topographique de façon centimétrique (voir Figure 21).

- l'autre type de cible sont des cibles Metashape que l'on peut imprimer à l'aide du logiciel et qui dont les coordonnées ont été déterminé par rayonnement à partir des cibles précédentes (voir Figure 22).

Ces deux types de cible pouvant à la fois être utilisés comme point d'appui mais également en point de contrôle.

Figure 21 : Cible rouge et blanche

Figure 22 : Cible Metashape

Source : Photoscan_ModeEmploi_2016.pdf, 12/06/2022

La première chose à faire sur Metashape est de détecter les cibles propres au logiciel qu'il est capable de reconnaître. Une fois fait, il faut importer les coordonnées de ces cibles précédemment citées et également les coordonnées des cibles rouges et blanches. On peut alors lancer un premier « alignement des photos » permettant à Metashape de positionner lui-même les cibles rouges à quelques pixels. Il suffit alors à l'utilisateur de vérifier si les cibles sont bien positionnées et, le cas échéant, de les déplacer. C'est après ce moment-là qu'on choisit les cibles servant de points de contrôle ou pas. Enfin un autre « alignement des photos » est nécessaire ce qui nous permet d'avoir des erreurs en mètre et en pixel sur chaque cible. Pour savoir quand l'erreur en mètre est tolérable, il faut faire la somme entre l'erreur en mètre sur les points d'appui et une proportion (ici on va prendre ¼) de la taille équivalente en pixel au sol. La précision des points d'appui est centimétrique et nous verrons plus tard que la taille pixel sol sera de 4 millimètres pour tous nos lever, l'erreur en mètre ne doit pas excéder 1.5 centimètre. Ainsi, si les erreurs en mètres de toutes les cibles sont en dessous du centimètre et que les erreurs en pixels sont en dessous de 1, on peut exporter les poses des images de la caméra en format texte. On a alors toutes les données nécessaires pour le calcul du bras de levier et de la matrice de passage.

III.1.4 Calcul du bras de levier et de la matrice de de passage

Reprenons les équations (1), (2) et (4) énoncées dans la partie III.1.1 qui permettent de calculer nos deux inconnues. Nous allons voir comment les calculer plus précisément dans les parties qui vont suivre.

III.1.4.1 Bras de levier

Ce bras de levier correspond au vecteur qui sépare le centre de prise de vue de la caméra au centre de réception du GNSS. Pour obtenir ce vecteur, il suffit de faire la différence entre les coordonnées du GNSS à l'instant t avec les coordonnées de l'image au

même instant (équation (1)). Le problème est que ce vecteur change à chaque instant dans le repère du GNSS appelé repère terrain. Il faut alors transformer ce vecteur dans le repère de la caméra pour qu'il soit constant peu importe l'instant t choisi. Pour cela, on utilise la matrice de rotation entre le repère terrain vers le repère caméra [Agisoft, 2022] que l'on obtient grâce à l'expression (7) suivante. A noter qu'Oméga (ω), Phi (φ) et Kappa (κ) sont obtenus grâce aux calculs fait sur Métashape et traités dans la partie III.1.2. A savoir que le sens positif de rotation est le sens horaire.

$$R_{terrain}^{cam} = \begin{pmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\ \omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{pmatrix} \quad (7)$$

On obtient toutes les inconnues de l'équation (2) ce qui nous permet d'obtenir notre bras de levier. La première étape est alors réalisée et on peut faire la moyenne entre tous les bras obtenus avec toutes les images aux instants t différents pour obtenir un unique bras de levier.

III.1.4.2 Matrice de passage entre le repère de la caméra et le repère de la centrale

La matrice de boresight est donc une matrice de rotation permettant de passer directement du repère de la caméra au repère de la centrale. Cette matrice s'obtient à partir de l'équation (4) connaissant la matrice de passage caméra/terrain. Plus précisément, nous avons la matrice de passage terrain/caméra mais il suffit de prendre sa transposée ou son inverse pour obtenir la matrice voulue. Il nous faut alors déterminer la matrice de passage du repère terrain vers le repère de la centrale. Pour cela, on utilise l'équation (8) suivante :

$$R_{centrale}^{terrain} = R_z^{90} * R_x^{180} * R_{centrale}^{référence} \quad (8)$$

Avec R_z^{90} une matrice de rotation selon l'axe z de 90°, R_x^{180} une matrice de rotation selon x de 180° et $R_{centrale}^{référence}$ la matrice de rotation formée par les angles (roulis, tangage, lacet) issues des données de la centrale allant du repère de la centrale à son repère de référence. A savoir que le sens positif de rotation est dans le sens anti-horaire.

$$R_{centrale}^{référence} = \begin{pmatrix} \cos L & -\sin L & 0 \\ \sin L & \cos L & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos T & 0 & \sin T \\ 0 & 1 & 0 \\ -\sin T & 0 & \cos T \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\ \mathrm{R} & -\sin \mathrm{R} \\ 0 & \sin \mathrm{R} & \cos \mathrm{R} \end{pmatrix} \quad (9)$$

Un fois les deux matrices calculées, on utilise l'équation (4) pour obtenir la matrice de passage caméra/centrale. Il ne reste plus qu'à calculer les angles d'Euler [Fabio Morbidi, 2022] issue de la matrice précédemment calculée comme suit :

$$Roulis = \varphi = actan(r_{32}, r_{33}) \text{ (10)}$$

$$Tangange = \theta = actan\left(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\right) \text{ (11)}$$

$$Lacet = \psi = actan(r_{21}, r_{11}) \text{ (12)}$$

avec r_{ij} les coefficients de la matrice de boresight $R^{boresight} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$

Il ne nous reste plus qu'à exporter le bras de levier et les coefficients de la matrice de boresight. Ces derniers sont exportés puisqu'ils permettent de restituer plus simplement la matrice. Cependant, il sera vu par la suite que Metashape a besoin de ces angles pour calculer l'orientation du nuage de points.

III.2 Phase de lever

Il faut tout d'abord lever la scène en question en appliquant toutes les recommandations énoncées dans la partie III.1.3.1. Une fois le lever terminé, nous pouvons passer au calcul des poses en utilisant les données fournies par le prototype instrumental ainsi que les ajustages vus précédemment.

III.2.1 Calcul des poses des caméras

L'étape suivante consiste à déterminer les poses des caméras pour les lever effectués. Il est important de préciser que la position des appareils entre eux ainsi que les paramètres de la caméra ne doivent pas changer entre le lever servant d'ajustage et les autres lever puisqu'on va utiliser le bras de levier et la matrice de boresight qui ont été déterminés avec une configuration précise. Si pour n'importe quelle raison, un des paramètres venait à changer, il faudrait recommencer le lever dit d'ajustage. De ce fait, il arrive souvent que le lever d'ajustage soit fait juste avant les autres lever pour s'assurer que les paramètres soient les mêmes.

Revenons à présent sur le calcul des nouvelles poses des images obtenues avec un nouveau lever. Les données en notre possession sont celles du GNSS, de la centrale, les différentes images de la scène sans oublier le bras de levier et la matrice de boresight que nous venons de calculer. La première étape consiste à trouver la matrice de passage du repère terrain au repère caméra afin de transformer les coordonnées du GNSS du repère terrain vers le repère caméra et d'y appliquer le bras de levier. Pour cela on connait déjà la matrice de passage entre le repère de la caméra à celui de la centrale (matrice de boresight) et grâce à

la centrale on peut calculer la matrice de passage centrale/terrain. On obtient ainsi la matrice de passage caméra/terrain que l'on va inverser puis utiliser pour transformer les coordonnées de l'antenne GNSS qui sont dans le repère terrain dans le repère caméra. On peut ensuite appliquer le bras de levier puis utiliser la matrice caméra/terrain afin d'avoir les coordonnées de la caméra dans le repère terrain. On obtient ainsi les positions des caméras. Tous les calculs sont résumés dans l'équation (12) :

$$r_{caméra}^{terrain} = R_{cam}^{terrain} * r_{caméra}^{cam}$$

$$= R_{cam}^{terrain} * \left(b^{cam} + r_{gnss}^{cam}\right)$$

$$= R_{cam}^{terrain} * (b^{cam} + R_{terrain}^{cam} * r_{gnss}^{terrain}) \quad (13)$$

En ce qui concerne l'orientation de la caméra pour chaque image, il suffit simplement de calculer les angles de rotation de la matrice de passage terrain/caméra avec les équations (9), (10), (11). Comme nous l'avons vu dans la partie III.1.2 , Metashape utilise ces angles-là pour construire son nuage de points de la scène. A la fin de ce programme nous avons pour chaque image, sa position dans le repère terrain ainsi que son orientation dans ce même repère.

III.2.2 Calcul du nuage de points sur Métashape

Le calcul du nuage de points de la scène sur Metashape peut paraître simple *a priori* car on se dit qu'il faut simplement importer les images et les poses des images de la caméra puis lancer un « alignement des photos » Cependant cela s'est révélé plus difficile que prévu puisque cela ne fonctionne pas avec des fichiers texte comportant les positions ainsi que les orientations des caméras. Il a fallu développer un script python permettant d'importer les poses pour chaque images correspondantes grâce à la bibliothèque *Metashape* seulement disponible sur leur logiciel [Agisoft, 2022]. Dans un premier temps, on importe les données de position et d'orientation de chacune des images dans des variables python. On définit ensuite le système de coordonnées dans lequel on va travailler *Metashape.CoordinateSysteme()* ainsi que le système d'angle utilisé *Metashape.EulerAnglesOPK* (pour notre cas il s'agit du RGF93 en Lambert93 (EPSG 2154) et on travaille avec les angles oméga, phi et kappa). On importe ainsi les photos dans Métashape (*.addPhotos())* puis pour chacune d'entre elles, on leur attribue leur position (*.reference.location)* et orientation (*.reference.location).* Par défaut, Metashape ne prend pas en considération l'orientation des images que l'on importe. Pour y remédier, on utilise une

fonction qui permet de le faire (*reference.rotation_enabled=True*). On pourrait alors lancer « l'alignement des images » pour obtenir notre nuage de points grâce à l'utilisation des poses importées. Cependant Metashape va ajuster les poses en fonction de celles rentrées en paramètre. On peut alors dire au logiciel de prendre plus ou moins en considération les poses calculées par notre instrumentation. Cela se règle dans la colonne « Précision » comme le montre la Figure 23.

Cameras	Easting (m)	Northing (m)	Altitude (m)	Accuracy (m)	Error (m)	Omega (°)	Phi (°)	Kappa (°)	Accuracy (°)
image0	487953.437642	6772548.676465	157.151382	10.000000		-100.360	45.234	-169.951	10.000
image1	487953.810519	6772548.374128	157.148880	10.000000		-101.124	47.297	-170.981	10.000
image2	487954.202207	6772548.012549	157.131852	10.000000		-100.820	47.439	-171.179	10.000
image3	487954.561924	6772547.645382	157.157734	10.000000		-100.701	46.754	-170.911	10.000
image4	487954.927350	6772547.285202	157.154333	10.000000		-100.344	47.684	-170.266	10.000

Figure 23 : Illustration de la colonne "Précision" dans Metashape

Plus ce nombre est élevé, moins Metashape va le prend en considération et donc plus la donnée associée à cette précision sera ajustée par le logiciel. On utilise une fonction qui permet d'attribuer la précision en mètre pour les positions et en degré pour l'orientation (*.reference.location_accuracy* et *.reference.rotation_accuracy)*. On peut réaliser l'étape d'alignement des images effectué par Metashape *via* un script python. Une fois ce script lancé et réalisé, nous avons les images avec leurs poses qui sont importées ainsi qu'un nuage de points. On peut alors créer un maillage pour ensuite le texturer et obtenir une reproduction en 3D de la scène.

De plus, si sur cet objet il existe des points de contrôle, il est possible de vérifier la précision du modèle ce qui nous permettra d'évaluer notre prototype instrumental de photogrammétrie mobile.

IV Validation de prototype

Pour pouvoir valider notre prototype, plusieurs points sont à prendre en considération. Tout d'abord, il faut valider la précision du géoréférencement que nous produisons avec notre instrumentation. En effet, pour que l'objectif du TFE soit réalisé, il faut surtout que la précision obtenue sur notre modèle soit bonne vis-à-vis des précisions des appareils. Ensuite, le second point important de ce TFE est l'aspect low-cost du prototype. Effectivement, quand bien même la précision du nuage est satisfaisante, il faut regarder l'aspect financier du prototype et le comparer à ce qui se fait actuellement dans le domaine de la photogrammétrie mobile. Enfin, il convient d'analyser la praticité du prototype comparé à d'autre méthode comme la photogrammétrie ou encore le lever scanner.

IV.1 Étude sur la précision du prototype

Pour évaluer la précision du prototype, on peut faire plusieurs comparaisons. La première est de faire la différence entre les poses calculées avec Métashape avec des points d'appui et les poses déterminées par notre instrumentation. En effet, comme à chaque lever on utilise un mur avec des cibles, il est possible de calculer les poses des caméras sur Métashape pour tous les lever. La seconde façon de procéder est de comparer les coordonnées des points de contrôle avec celles obtenues par géoréférencement direct. Comme nous l'avons vu précédemment, trois façons de synchroniser les données peuvent être utilisées pour déterminer les poses des images de la caméra. Nous allons d'abord déterminer quelle synchronisation donne de meilleurs résultats. Pour cela on va étudier pour chaque type de lancement, la différence entre les poses calculées avec notre instrumentation et celles calculées avec Metashape.

Tout d'abord il est important de noter que la précision du prototype sera impactée par la précision de la synchronisation des données. En effet, bien qu'en vérifiant que les trois appareils aient été lancés en même temps en utilisant un *time.time* au moment du lancement des instruments, il existe un délai entre le moment où la fonction est lancée et le moment où l'instrument a réellement démarré les mesures. Ce délai créé un décalage des données puisque si on suppose un déplacement de trois kilomètres par heure avec un décalage de 0.05 seconde, les données ne seront pas récoltées au même endroit mais à une distance de 4 centimètres ce qui peut également influencer sur la détermination du bras de levier et la matrice de passage et/ou sur les poses calculées par notre instrumentation. Un moyen de réduire cette influence est de diminuer la vitesse du prototype pour les lever et plus

particulièrement pour les lever d'ajustage. Par exemple, en diminuant par deux la vitesse, l'écart ne serait plus que de 2 centimètres. Nous pouvons maintenant passer à la comparaison des points connus en coordonnées.

A noter que h signifie hauteur ellipsoïdale et que Metashape sera utilisé avec le même paramétrage pour tous les lever.

IV.1.1 Choix de la méthode de synchronisation

La première méthode de synchronisation que nous allons étudier est celle avec un lancement simultané avec un seul programme. A noter que pour ces lever, la caméra utilisée est la GoPro HERO 5. Il faut noter également que plusieurs lever ont été réalisés avec ce type de lancement et nous avons moyenné les erreurs sur l'ensemble des lever. Les tailles pixel au sol et le recouvrement valant respectivement 4 millimètres et 91 % pour les deux lever. Nous faisons la moyenne des écarts sur les 100 poses calculées. Les graphiques montrant les erreurs pour chaque lever sont donnés en annexe 1.

	Erreur sur la coordonnée E (en cm)	Erreur sur la coordonnée N (en cm)	Erreur sur la coordonnée h (en cm)
Moyenne	39,0	-35,5	9,6
Écart type	4.2	3.2	5.5
	Erreur sur le roulis (en °)	Erreur sur le tangage (en °)	Erreur sur le lacet (en °)
Moyenne	-0.18	0.14	-0.03
Écart type	0.99	1.11	0.43

Tableau 4 : Écart entre les poses calculées sur Metashape et les poses calculées avec notre instrumentation avec un lancement simultané et un seul programme

Le Tableau 4 nous montre que l'erreur moyenne en position en planimétrie se situe entre 35 et 40 centimètres avec un écart type de 3 à 4 centimètres. En ce qui concerne l'altimétrie, l'erreur moyenne est proche du décimètre avec un écart-type de 5 centimètres. En ce qui concerne les angles, l'erreur moyenne est très proche de 0°. L'écart type, quant à lui, est autour de 1° pour les trois angles. Ces erreurs sont bien trop importantes pour espérer utiliser notre prototype comme substitue à la photogrammétrie traditionnelle. Passons maintenant au deuxième type de lancement, celui en différé avec un programme pour chaque appareil. Notons que la caméra utilisée est aussi la Go Pro HERO pour les mêmes raisons

évoquées précédemment. Plusieurs lever ont aussi été réalisés avec ce type de lancement et les graphiques montrant les erreurs moyennes par lever sont en annexe 2. La moyenne des écarts a été faite sur 153 poses calculées. Notons ici que la taille pixel sol ainsi que le recouvrement valent respectivement 4 millimètres et 92 %

	Erreur sur la coordonnée E (en cm)	Erreur sur la coordonnée N (en cm)	Erreur sur la coordonnée h (en cm)
Moyenne	9.5	-9.5	0.1
Écart type	5.1	5.1	2.5
	Erreur sur le roulis (en °)	Erreur sur le tangage (en °)	Erreur sur le lacet (en °)
Moyenne	-0.15	-0.09	0.30
Écart type	1.09	1.40	0.73

Tableau 5 : Écart entre les poses calculées sur Metashape et les poses calculées avec notre instrumentation avec un lancement différé

Avec le Tableau 5 nous observons un écart moyen sur les coordonnées planimétriques de l'ordre du décimètre et du millimètre pour l'altimétrie. L'écart type en planimétrie est, quant à lui, de l'ordre de 5 centimètres et de 2.5 centimètres en altimétrie. Pour les écarts moyens sur les angles, il est très proche de 0°. L'écart type sur les angles reste proche de 1°. On peut alors étudier le dernier type de lancement. Il est fait de manière simultanée pour les trois appareils mais avec un programme pour chaque appareil. La caméra est la même que celle utilisée pour les deux autres types de lancement.

	Erreur sur la coordonnée E (en cm)	Erreur sur la coordonnée N (en cm)	Erreur sur la coordonnée h (en cm)
Moyenne	-7.5	7.5	0.6
Écart type	5.5	4.1	3.1
	Erreur sur le roulis (en °)	Erreur sur le tangage (en °)	Erreur sur le lacet (en °)
Moyenne	-0.30	0.73	-0.04
Écart type	1.07	1.74	0.97

Tableau 6 : Écart entre les poses calculées sur Metashape et les poses calculées avec notre instrumentation avec un lancement simultané et 1 programme par instrument

A noter également ici que plusieurs lever ont été fait et les graphiques des erreurs sont en annexe 3. La moyenne des écarts est faite sur 196 poses calculées. De plus la taille pixel sol ainsi que le recouvrement valant respectivement 4 millimètres et de 93 %. Le Tableau 6 montre un écart moyen en planimétrie de 7 centimètres tandis que cet écart est en dessous du centimètre pour l'altimétrie. L'écart type est ici entre 3 et 5 centimètres pour les 3 coordonnées. En ce qui concerne les angles, les écarts restent entre de 0 et 1°. L'écart type, quant à lui, est autour du degré pour le roulis et le lacet mais proche de 2° pour le tangage.

En comparant les poses calculées avec Métashape et les poses obtenues *via* notre instrumentation, nous observons que la première est à écarter puisqu'elle nous donne des écarts nettement supérieurs au décimètre. En ce qui concerne les deux hypothèses, leurs précisions sont plus satisfaisantes pour notre travail puisqu'elles sont inférieures au décimètre. Cependant, la troisième hypothèse donne une meilleure précision ce qui nous permet de départager ces dernières entre elles. De plus, le choix du type de lancement peut se faire selon la praticité de ce dernier. Pour le lancement de manière différé, il a également été vu dans la partie II.4.1.2] que la synchronisation n'est pas parfaite notamment entre les données de la centrale et celles de la caméra. De plus l'extraction des données de tous les appareils sur un intervalle commun est assez compliquée à mettre en place. Pour toutes ces raisons, il a été décidé de choisir le lancement simultané avec un programme par instrument. Nous pouvons à présent passer à la comparaison des points de contrôle sur le nuage de points obtenus avec les poses calculées par notre prototype instrumental.

IV.1.2 Comparaison des points de contrôle sur les nuages de points obtenus avec les poses calculées par notre instrumentation

Pour pouvoir analyser correctement les résultats que l'on va obtenir, il est important d'analyser la précision des appareils vue jusqu'à présent. Tout d'abord, comme il a été vu dans la partie I.1, la précision du GNSS va jouer un rôle important dans la précision du nuage de points. En effet, en mouvement et en présence de masques, la précision du Reach est proche de 7 centimètres. Un autre point pouvant affecter nos résultats est la précision de la centrale. Comme nous l'avons vu dans la partie I.2, bien que nous n'ayons pas le biais sur les mesures d'angles, l'imprécision sur les mesures de la centrale peut jouer un rôle dans la précision des poses calculées pas l'instrumentation. Enfin, le dernier point qui exerce une influence sur la précision de nos données est au moment du pointage des cibles sur Métashape. Tout cela va avoir un premier effet sur le calcul du bras de levier et de la matrice

de boresight mais également au moment du calcul des poses par notre instrumentation. Ainsi on s'aperçoit que la précision des appareils utilisés va avoir un impact sur la précision des poses obtenues par notre instrumentation et donc sur la précision de notre géoréférencement.

Avant de s'intéresser à l'évaluation de la précision du nuage de point, il est important de s'assurer que les lever suivent bien les recommandations énoncées dans la partie III.1.3.1. Pour cela, pour chaque lever, nous regarderons si les bases sont relativement constantes, si le recouvrement est au-dessus de 80 % et si les images sont nettes. Sur Metashape il est possible de calculer un indice de qualité des images. Pour que l'image soit validée, il faut que l'indice soit supérieur à 0.5 [Agisoft, 2022].

Lever	Base (en m)		Recouvrement	Qualité de l'image
	Moyenne	Écart-type		
1	0.500	0.023	93 %	0.96
2	0.479	0.023	93 %	0.96
3	0.514	0.033	93 %	0.95

Tableau 7 : Validation des lever

Le Tableau 7 nous montre une homogénéité sur la distance entre deux poses successives, avec un recouvrement suffisamment élevé et avec une qualité d'image très satisfaisante pour les trois lever. Ces derniers sont alors validés et nous pouvons alors étudier leur précision.

Comme il a été présenté dans la partie III.2.2, on peut importer des poses sur Metashape tout en ayant la possibilité de les prendre plus ou moins en considération lors de l'alignement des images. On peut alors émettre quelques hypothèses sur l'utilisation des poses importées et leur prise en compte dans le traitement Metashape. Tout d'abord nous pouvons émettre l'hypothèse que les poses sont loin d'être parfaites et donnent au logiciel un ordre d'idée sur les positions et les orientations des images de la caméra. Des points d'appui étant placés sur l'objet levé afin qu'ils soient utilisés dans le calcul. Nous pouvons également décider que ces poses sont parfaites et qu'elles n'ont pas besoin d'être ajustées lors de l'alignement des images. Les points seront ici mis en point de contrôle pour évaluer la précision du modèle. Ces deux hypothèses sont diamétralement opposées et il convient d'en émettre une intermédiaire. Pour cela, on supposera que nos poses ne sont pas parfaites et ont besoin d'être ajustées par Metashape. Dans ce cas les points vont être utilisés en points

de contrôle ce qui permettra d'observer si, avec les poses calculées par notre instrument, Metashape est capable de nous donner un alignement précis.

IV.1.2.1 Première hypothèse : Ajustement des poses avec les points d'appui

Pour cette hypothèse et les deux autres, nous allons utiliser les données issues du lever fait avec un lancement simultané avec un programme par instrument comme il a été spécifié dans la partie IV.1.1. Nous pourrons regarder les écarts moyens entre les poses importées et celles ajustées par Metashape mais également les erreurs moyennes obtenues sur les points d'appui. Cependant, Metashape utilisera ces points lors de l'alignement. Il n'est alors pas pertinent de regarder les écarts sur ces points d'appui puisqu'ils seront de l'ordre du millimètre. Nous allons donc nous concentrer sur les écarts moyens entre les poses ajustées et importées sur les trois lever effectués. A noter que dans les colonnes « Précision » de Metashape, nous mettons 1 mètre pour les positions et 10° pour les orientations. Cela aura notamment comme effet la non prise en compte des poses importées comme il a été énoncé pour cette hypothèse.

Lever	Erreur en E (en cm)	Erreur en N (en cm)	Erreur en h (en cm)	Erreur en roulis (en °)	Erreur en tangage (en °)	Erreur en lacet (en °)
1	9.4	10.5	3.3	1.08	1.29	0.85
2	1.1	1.8	3.6	1.17	1.39	1.14
3	11.6	11.2	3.4	1.10	1.66	0.89

Tableau 8 : Écarts moyens entre les poses importées et ajustées par Metashape avec points d'appui

Le Tableau 8 affiche des écarts autour de 10 centimètres en planimétrie (à part pour le lever 2) et de 2 à 3 centimètres en altimétrie. Comme Metashape était libre d'ajuster les poses en prenant en compte les points d'appui et sans prendre en compte les poses importées, on obtient les mêmes écarts moyens entre les poses que dans la partie IV.1.1. Cependant en condition d'utilisation, aucun point d'appui n'est utilisé. Il faut donc placer ces points en points de contrôle afin d'évaluer la précision de l'alignement.

IV.1.2.2 Deuxième hypothèse : Poses importées parfaites

Dans cette hypothèse on suppose que le prototype instrumental donne des poses parfaites et que le logiciel va ajuster les paramètres internes lors de l'alignement. On va également utiliser des points de contrôle afin d'évaluer la précision de l'alignement des

images. Pour cela on va définir les colonnes « Précision » avec 0.1 millimètre pour les positions et 1 milli- degré pour les orientations. Cela imposera à Metashape qu'il aligne avec les poses importées sans les ajuster au préalable. Ainsi la différence entre les poses ajustées et importées sera inférieure au millimètre pour les positions et du milli-degré pour les angles. Nous ne nous intéresserons alors qu'aux erreurs sur les points de contrôle.

Lever	Erreur en E (en cm)	Erreur en N (en cm)	Erreur en h (en m)
1	115.0	42.9	28.3
2	106.4	95.6	10.1
3	174.5	55.3	47.4

Tableau 9 : Erreurs moyennes sur les points de contrôle avec des poses « parfaites »

Le Tableau 9 montre des erreurs moyennes entre 50 centimètres et presque 2 mètres. Ce résultat bien que désastreux était prévisible puisqu'en début de partie nous avons analysé les précisions des différents appareils et de la synchronisation et donc toutes les sources d'erreurs qui rentrent en compte lors du calcul des poses. De plus, le logiciel n'avait pas d'autres informations tels que des points d'appui lors de l'alignement ce qui amène à ces résultats. Il convient alors de laisser à Metashape une certaine liberté quant au fait d'ajuster les poses importées et les paramètres internes pour produire le meilleur nuage de points possible.

IV.1.2.3 Troisième hypothèse : Ajustage des poses sans points de contrôle

Dans cette hypothèse, nous allons mettre les valeurs des colonnes « Précision » à 10 centimètres pour les positions et 1° pour les orientations. Les cibles sont dans ce cas considérés comme des points de contrôle et ne sont donc pas prises en compte pour le calcul les poses. Ces points doivent être répartis de façon homogène sur l'ensemble de l'objet levé afin d'avoir une vue de la précision sur l'ensemble du nuage de points.

Figure 24 : Répartition des points de contrôle sur l'objet levé

La Figure 24 montre une répartition homogène des points de contrôle sur l'ensemble du mur ce qui permettra de visualiser la précision du nuage à travers ces points sur la totalité

de l'objet. On peut alors comparer l'écart moyen entre les poses ajustées par Metashape et celles importées mais également les écarts moyens sur les points de contrôle

Lever	Erreur en E (en cm)	Erreur en N (en cm)	Erreur en h (en cm)	Erreur en roulis (en °)	Erreur en tangage (en °)	Erreur en lacet (en °)
1	2.4	1.7	1.7	1.02	1.24	0.84
2	1.6	1.7	1.9	1.04	1.35	1.17
3	5.4	4.1	2.0	1.00	1.28	0.85

Tableau 10 : Écarts moyens entres les poses importées et ajustées par Metashape sans points d'appui

On observe grâce au Tableau 10 que les écarts moyens entre les poses ajustées et importées ont diminué par rapport à la première hypothèse. Ceci est logique puisque Metashape n'a plus de points d'appui pour son calcul de pose et va donc très peu ajuster les poses importées. Les écarts sont entre 2 et 5 centimètres ce qui est moins important que dans la première hypothèse où les écarts étaient autour de 10 centimètres à part pour le lever 2. On peut en conclure ceci : bien que Metashape n'avait que des poses approchées et aucun point d'appui, il lui a été possible d'ajuster des poses. Ce résultat doit s'accompagner d'écarts pas trop importants sur les points de contrôle pour que le nuage de points soit bien géoréférencé.

Lever	Erreur en E (en cm)	Erreur en N (en cm)	Erreur en h (en cm)
1	3.2	20.0	3.9
2	9.2	9.6	5.6
3	9.4	15.7	2.8

Tableau 11 : Erreurs moyennes sur les points de contrôle

Le Tableau 11 nous montre que les écarts sur les points de contrôle sont de l'ordre de la dizaine de centimètres pour la planimétrie (à part pour la coordonnée N du lever 1 et 3) et entre 3 et 6 centimètres pour l'altimétrie. Cette différence peut s'expliquer par le fait que quand bien même la précision du GNSS est moins importante en altimétrie, la hauteur ellipsoïdale reste la même sur l'ensemble du lever et si dans la synchronisation il existe quelques microsecondes d'écart, l'influence de cette dernière n'aura pas une aussi grande importance que pour la planimétrie. En effet, à l'instant t ou t+1 la coordonnée h est sensiblement la même contrairement aux coordonnées E et N. Les erreurs moyennes sur les points de contrôle sont cohérentes avec les imprécisions énoncées en début de partie puisque

le GNSS a une précision proche de 7 centimètres à laquelle il faut ajouter la précision de la centrale. Il ne faut surtout pas oublier l'imprécision dû au décalage de la synchronisation qui peut elle aussi augmenter les erreurs sur les points de contrôle.

Pour résumer, les poses obtenues ne peuvent pas être considérées comme parfaites quand elles sont importées dans Metashape pour la construction du nuage de points. Au contraire il faut permettre à Metashape de pouvoir ajuster les poses importées. On peut alors regarder l'erreur sur les points de contrôle pour valider la précision de l'alignement obtenu. On remarque ainsi que les erreurs sur ces points sont proches de la dizaine de centimètres. Cela peut paraître important de premier abord mais en regardant les précisions des différents instruments, on s'aperçoit qu'elles peuvent expliquer ces erreurs. Pour améliorer la précision du prototype, il est, dans un premier temps, possible de placer un ou deux points d'appui sur l'objet. En effet, mettant un point d'appui sur le lever 1, l'erreur sur les points de contrôle a nettement diminué passant à 7.0 centimètres sur la coordonnée E, 7.3 centimètres sur la coordonnées N et a 2.4 centimètres sur la coordonnée h. On peut ensuite s'inntéresser à un autre moyen de synchroniser les données instrumentales. Maintenant que la précision du prototype instrumental est vérifiée, on peut étudier le coût de ce prototype.

IV.2 Étude sur le coût du prototype

La précision est un élément indispensable pour juger de la qualité d'un prototype de photogrammétrie mobile mais pour notre cas, il faut également que ce prototype soit à bas-coût. Ce terme ne signifie pas forcément que le prix global du prototype soit à la portée de tout le monde mais plutôt qu'il se différencie de ses homologues par un prix nettement moins élevé. L'objectif de cette partie est alors d'estimer le prix global de ce prototype. Ce dernier est équipé d'un ordinateur de bord et plus précisément d'une tablette tactile GETAC. Après quelques recherches internet, il s'avère que le prix de cette tablette soit de l'ordre de 2500 €. Ensuite il y a comme récepteur-antenne GNSS un Reach RS2 qui est au prix de 2110 € sur le site du constructeur Emlid. Comme caméra on utilise la GoPro HERO 5 qui a un prix moyen de 250 €. Enfin pour notre centrale inertielle, on utilise la PhidgetSpatial développée par Phidget au prix de 95 € sur leur site internet. On pourrait compter dans le prix du prototype le coût du chariot qui nous permet de le déplacer à notre guise. Cependant on peut également faire fonctionner l'ensemble du système sur un chariot plus petit ou sur une canne. On aurait alors des cas différents selon la méthode pour déplacer le prototype. Il ne faut pas oublier le prix pour avoir accès à Metashape puisque ce logiciel est essentiel pour la création du nuage de points. Cependant, un autre logiciel pourrait être utilisé pour la réalisation de

cette tâche. Ainsi le prix du système de photogrammétrie mobile s'élève à près de 5000 € pour le prototype. A cela va s'ajouter le prix du logiciel de photogrammétrie utilisé (par exemple pour Metashape, il faut ajouter 180 € pour une licence standard et 3335 € pour une licence professionnelle). En regardant ce qui se fait de similaire dans le même domaine, on s'aperçoit que ce prototype est huit fois moins cher que les premiers appareils vendus. On peut donc en conclure que ce prototype entre bien dans la catégorie low-cost des instruments de photogrammétrie mobile. Il faut cependant nuancer cette conclusion puisque, comme nous l'avions vu dans l'introduction, le système de Leica permet à ses utilisateurs d'avoir une précision sur son nuage de point de l'ordre de 5 centimètres sans point d'appui. La précision finale n'est alors pas la même.

IV.3 Étude sur la praticité du prototype

Pour étudier la praticité d'un prototype, il faut regarder s'il apporte des avantages sur certains aspects ou du moins qu'il ne soit pas trop contraignant vis-à-vis d'autres méthodes. On va alors comparer notre prototype utilisant la photogrammétrie mobile avec la photogrammétrie sans géoréférencement direct mais également avec les relevés scanner qui aboutissent à des résultats quasi-identiques. Les aspects qui vont être comparés sont le matériel utilisé, le temps passé sur le terrain, le personnel mobilisé et enfin la précision des données.

En ce qui concerne le matériel à prendre lors des lever sur le terrain, pour la photogrammétrie classique il faut un appareil photo, des cibles Metashape et quelques fois des perches permettant de prendre des photos à une hauteur plus haute. Ces éléments sont assez légers sans prendre excessivement de place. Cependant, pour pouvoir géoréférencer notre nuage, il est nécessaire d'apporter une station totale avec son trépied et une antenne GNSS s'il n'existe aucun point connu à proximité. De la même manière, le matériel à prendre pour faire un lever LIDAR est aussi imposant. En effet il faut d'abord le scanner qui peut être plus ou moins imposant, un trépied ainsi que des cibles permettant d'assembler et géoréférencer les nuages de points. A cela peut s'ajouter d'autres trépieds s'il est nécessaire d'avoir des cibles à des endroits où elles ne peuvent être accrochées. Quant au matériel nécessaire pour la photogrammétrie mobile, il nous faut seulement le prototype instrumental. On peut considérer que les instruments sont solidement attachés à leur socle et qu'ils ne bougent pas et donc réaliser le lever d'ajustage au préalable.

Passons maintenant au temps passé sur le terrain. En ce qui concerne la photogrammétrie, en plus de prendre des photos, il faut lever les cibles avec un tachéomètre

depuis une référence connue. Le temps passé sur le terrain peut vite devenir important s'il faut, en plus, placer un point GNSS. Ainsi, sur des objets pouvant être long ou avec une forme peu commune, le temps sur le terrain peut être très long. Pour un lever LiDAR, il faut attendre que le balayage s'effectue ce qui est d'autant plus long que la zone à lever est grande. On peut également imaginer que pendant que le scanner enregistre ses données, le technicien puisse lever les cibles servant pour le géoréférencement et l'assemblage du nuage de points. Enfin, pour la photogrammétrie mobile, le temps passé sur le terrain dépend de la longueur ou la complexité de l'objet à lever. Il reste tout de même inférieur aux deux autres méthodes surtout si l'objet regorge de recoins.

Pour le personnel utilisé, chacune des trois méthodes peut être effectuées seules, sans l'aide d'une autre personne. Cependant pour la photogrammétrie, il est plus rapide de venir à deux sur le terrain afin que l'un s'occupe du lever photogrammétrique pendant que l'autre s'occupe du lever tachéométrique.

Enfin, en ce qui concerne la précision du lever, on a vu dans la partie IV.1.2 que la précision du prototype utilisant la photogrammétrie mobile est proche de 10 centimètres. Pour la méthode du lever LIDAR ou encore pour la photogrammétrie classique, la précision du géoréférencement dépend de la précision du GNSS utilisé afin d'obtenir un point géoréférencé. On peut diminuer cette imprécision en visant non pas un point connu mais plusieurs points. La précision d'un GNSS classique tourne autour de 2 centimètres en planimétrie et 5 en altimétrie en utilisant le NRTK. La précision de ces deux méthodes est deux à quatre fois plus grande que pour le prototype instrumental utilisant la photogrammétrie mobile.

La Figure 25 reprend les critères énoncés précédemment en attribuant à chaque méthode une note entre 0 et 5 en fonction des avantages et inconvénients qu'elles ont par rapport aux autres. Ainsi une note proche de 5 montre que la méthode est la moins contraignante dans ce critère par rapport

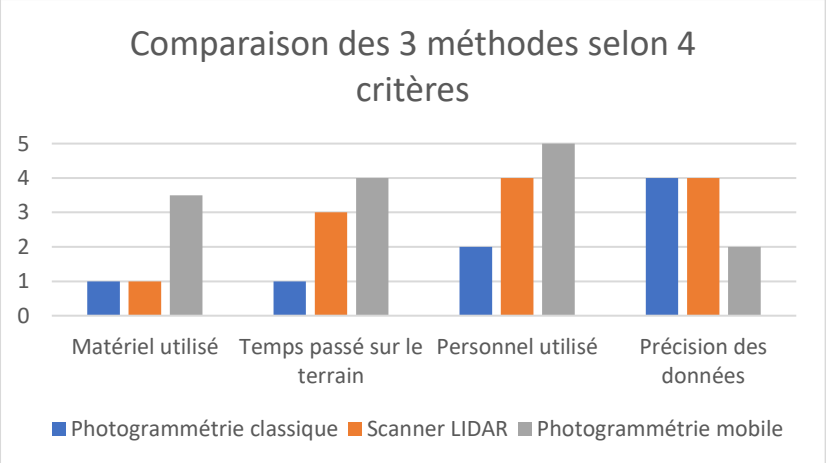


Figure 25 : Comparaison des méthodes selon plusieurs critères

aux autres. Par exemple, on a vu que le matériel utilisé sur le terrain pour le prototype utilisant la photogrammétrie est assez léger vis-à-vis des deux autres méthodes qui ont besoin de beaucoup d'éléments. Elle aura donc une note plus proche de 5 que le lever LiDAR ou la photogrammétrie.

Globalement, on observe que même si la photogrammétrie mobile n'est pas première dans tous les critères, elle a de bons résultats sur l'ensemble de ces derniers. La précision de son géoréférencement étant son plus gros point faible actuellement. On peut donc en conclure que la photogrammétrie mobile reste une méthode très pratique si l'on veut avoir un nuage de points géoréférencé.

IV.4 Exemples d'utilisation du prototype

Afin de donner un exemple d'utilisation du prototype, nous avons décidé de l'utiliser pour lever des logements se situant à l'entrée du parking de l'école. Nous devons d'abord ajuster le prototype puis nous pouvons ensuite lever les logements. L'objectif premier lors de ce test était de savoir si on pouvait utiliser le mode SuperView de la GoPro HERO afin d'avoir un champ de vision le plus grand possible et donc de visualiser un plus grand espace. Cependant avec la taille des cibles utilisées pour le calcul des poses avec Metashape, le logiciel n'arrivait pas à pointer ses propres cibles. Il fallait alors se rapprocher du mur afin qu'il puisse les détecter. Cela impliquait donc de faire le lever d'ajustage sur l'herbe avec de légère bosse et non plus sur du bitume ce qui rend le déplacement du charriot plus compliqué. De plus les photos seront impactées puisqu'il y a un risque de perte de netteté. Nous sommes donc restés avec le champ de vision utilisé lors de nos tests précédents pour lever le logement. Un fois les données obtenues et les poses des images de la caméra calculées avec notre prototype, nous pouvons lancer le calcul du nuage de points sur Metashape. La colonne précision est réglée sur 1° pour ce qui concerne l'orientation et 0.20 mètre pour les positions. Pour permettre une meilleure analyse de la précision du géoréférencement, nous avons levé une quinzaine de points sur une façade des logements afin que l'on puisse utiliser ces points comme points de contrôle. Après les avoir pointés sur les images et après avoir lancer l'alignement des images, il est possible d'évaluer sa précision. On peut maintenant analyser les résultats obtenus et notamment l'erreur sur les points de contrôle mais également l'écart entre les poses importées et celles ajustées par Metashape.

	Erreur en E (en cm)	Erreur en N (en cm)	Erreur en h (en cm)
Moyenne des erreurs	25.5	31.6	62.6

Tableau 12 : Erreur sur les points de contrôle des logements

On observe dans le Tableau 12 une erreur sur les points de contrôle de 20 à 60 centimètres. Cela est très étonnant puisqu'on s'attendait à une erreur de l'ordre de la dizaine de centimètre. Pour tenter de comprendre cette erreur, on regarde les écarts entre les poses importées et celles ajustées par Metashape.

	Erreur sur la coordonnée Est (en cm)	Erreur sur la coordonnée Est (en cm)	Erreur sur la coordonnée Est (en cm)
Moyenne	3.6	7.5	54.8
	Erreur sur le roulis (en °)	Erreur sur le tangage (en °)	Erreur sur le lacet (en °)
Moyenne	0.813	1.864	8.630

Tableau 13 : Écart entre les poses importées et celles ajustées par Metashape sur le lever des logements

Le Tableau 13 nous montre un écart assez important au niveau de la coordonnée h, près de 50 centimètres mais également un écart sur le lacet de plus de 8 degrés. Deux hypothèses peuvent expliquer ces écarts : la première est que le GNSS nous a donné de mauvaises coordonnées ce qui provoque ces écarts constatés. Cependant cela n'explique par l'écart au niveau des lacets que nous observons dans le Tableau 13. La deuxième hypothèse est que la centrale ne nous donne pas les bons angles ou bien que la matrice centrale/terrain n'est pas bonne. C'est cette hypothèse que nous allons étudier. Le site phidget nous indique que le Nord est indiqué par l'axe 0 de la centrale c'est-à-dire l'axe des X. Nous allons alors vérifier cela grâce à l'interface graphique développée par phidget qui nous permet de visualiser l'orientation de la centrale par rapport au Nord.

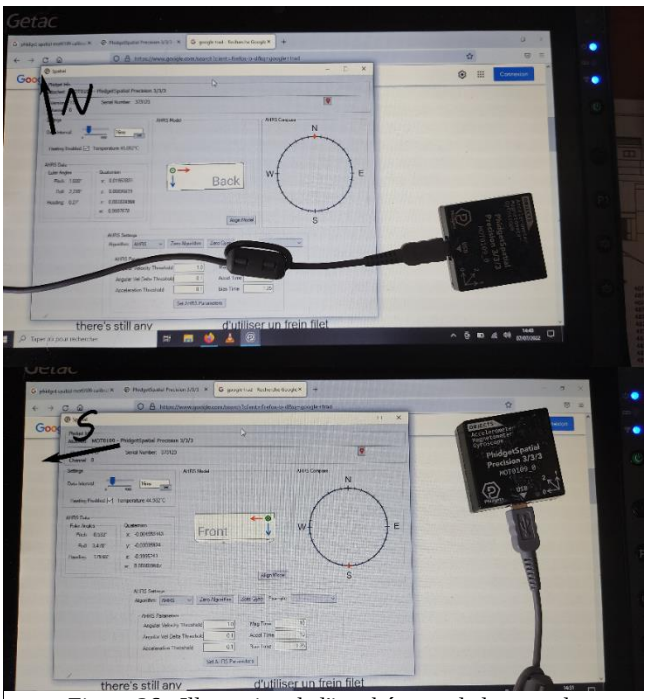


Figure 26 : Illustration de l'incohérence de la centrale

Comme le montre la Figure 26, sur la première image le Nord pointe vers le haut de l'image. En toute logique, le Sud devrait alors être à l'opposé du Nord et donc pointé vers le bas de l'image. En sachant que la centrale a juste subit une rotation autour de l'axe des z sur la seconde image par rapport à la première, on constate que le

Sud pointe non pas vers le bas de l'image mais vers la gauche de l'image. La centrale nous donne alors de mauvais angles ce qui va avoir une influence sur la création de la matrice centrale/terrain et donc sur la détermination des poses des images de la caméra. Cette erreur de fonctionnement ne pouvait pas être détectée plus tôt puisque pour nos différents lever, le mur utilisé était le même. L'orientation de la centrale et de la caméra par rapport au repère terrain étaient donc identiques pour nos lever d'ajustage et pour les autres lever. Ainsi nous retrouvions la bonne matrice terrain/caméra. Dans le cas présent, le lever d'ajustage et le lever des logements n'ont pas la même orientation par rapport au repère terrain. Cela explique pourquoi l'erreur de fonctionnement de la centrale n'a pas pu être détectée dans nos différents tests. Il faut alors régler ce problème pour pouvoir exploiter le prototype et qu'il nous retourne de meilleurs résultats. En état actuel, le prototype ne peut produire de bon résultat que si l'orientation du lever servant d'ajustage et le lever de l'objet ont la même orientation par rapport au terrain.

Conclusion

Ce TFE a été réalisé au sein du laboratoire GeF de l'ESGT afin de les aider à acquérir des compétences en ce qui concerne les systèmes multicapteurs et plus précisément dans le domaine de la photogrammétrie mobile. Il avait comme principal enjeu de permettre à ce laboratoire de se munir d'un prototype instrumental d'acquisition de nuages de points géoréférencés. L'objectif est qu'il puisse être utilisé dans différentes opérations menées par ce dernier ou plus simplement utilisé par le corps professoral dans les différents TP mis en place dans le cycle ingénieur de l'ESGT. Pour cela, il a fallu d'abord comprendre comment chaque appareil fonctionnaient et pouvaient être utilisés afin de répondre à notre problématique. Il a fallu ensuite déterminer les données qui allaient servir pour l'acquisition du nuage de points sur les différents appareils. Un fois que les données ont été choisies, il fallait savoir comment faire fonctionner les instruments les uns par rapport aux autres et surtout comment les synchroniser de façon quasiment parfaite. On a pu ensuite tester notre prototype avec différents tests qui nous ont indiqué les points à améliorer et à modifier. Enfin le prototype abouti peut être utilisé afin de démontrer les domaines dans lesquels sa précision lui permet d'être utilisé. Une des premières avancées a été de déterminer quel ordinateur de bord utiliser. Il fallait faire un choix entre légèreté ou praticité de l'ordinateur lors du lever. On en a conclu que pour notre cas la praticité l'emportait sur la légèreté puisque cela va nous permettre de nous adapter aux spécificités de notre objet à lever. Il permet d'avoir également un œil tout au long du lever en s'assurant notamment que tous les instruments fonctionnent et envoient des données voulues. La deuxième avancée a été de déterminer la meilleure synchronisation temporelle entre les instruments qui aura une influence non négligeable dans la précision du prototype. En effet une meilleure synchronisation entraînera une meilleure précision du nuage. Pour cela nous devions choisir entre un lancement simultané ou différé des instruments. Après de nombreux tests nous avons déterminé que le lancement simultané des appareils donnait un meilleur résultat et était moins contraignant à utiliser. Avec toutes ces avancées, nous obtenons un nuage de points avec une erreur moyenne sur des points de contrôle de 10 centimètres sans utiliser de points d'appui. Cela permet d'utiliser ce prototype dans le lever de corps de rue ou encore pour le lever de bâtiment servant de modélisation 3D. Avec un simple passage, on est capable de modéliser une rue entière sans points d'appui avec une précision décimétrique. Cependant, en l'état, il serait préférable de mettre quelques points d'appui sur l'objet à lever pour être certain d'avoir une précision meilleure que 10

centimètres. On pourrait alors voir tous les axes d'améliorations possibles afin d'améliorer cette précision ou pour augmenter la praticité du prototype.

Dans un premier temps il serait intéressant d'utiliser une Go Pro plus récente puisque cela permettrait d'utiliser une bibliothèque python développée par Go Pro directement et donc avoir des fonctions créées expressément pour la caméra en question. On pourrait également réfléchir à un autre moyen de synchroniser les instruments entre eux pour tenter d'améliorer la précision du nuage de points. Ensuite dans l'optique d'avoir un prototype le plus portatif possible, il serait intéressant de se passer du chariot et ainsi mettre tous les instruments sur une canne ou créer un dispositif backpack. Cela permettrait d'utiliser le prototype peu importe le type de sol et d'être également plus maniable sur des terrains ou le sol n'est pas lisse. Cela induit donc de changer l'ordinateur de bord et de passer de la tablette actuellement utilisée par une autre tablette moins lourde et moins imposante. Il faut néanmoins qu'elle soit suffisamment puissante pour permettre le déroulement complet des programmes sans pour autant avoir un impact sur le temps de déclenchement des fonctions permettant le déclenchement des instruments. Il faut donc développer un prototype capable d'être tenu avec une main ou alors sur le dos afin que notre problématique de la mobilité puisse être résolue à 100 %. On pourrait également se pencher sur la problématique permettant à l'utilisateur de voir ce qu'il lève en lui proposant soit un nuage de points qui se crée en direct ou plus simplement de voir ce que la vidéo filme et donc de repasser par des endroits qu'il aurait manqués. Il faudrait alors que la vidéo soit retransmise en direct sur l'ordinateur de bord ce qui permettrait à l'utilisateur de visualiser l'objet filmé. Comme vu dans la dernière partie, il faut également régler le problème de la centrale puisqu'elle ne nous donne pas les bons angles nous empêchant ainsi de déterminer les bonnes matrices de rotation. Pour finir, afin d'améliorer l'aspect low-cost du prototype, on peut tenter d'utiliser un logiciel gratuit permettant de calculer les poses des caméras ou permettant de former un nuage de points géoréférencé à partir de ses poses.

Bibliographie

Travaux universitaires

BASTARD-ROSSET C., BATTISTELLA C., DOUYER A., EL HADDAD M., 2022, Évaluation des solutions GNSS temps réel récentes et à bas coût, Rapport de projet pré-professionnel, ESGT, 31p

GOMEZ C., 2018, Estimation d'une trajectoire par couplage inertie-GNSS, Mémoire de TFE, ESGT, 68 p.

Sites web

Agisoft, Manuel de l'utilisateur Agisoft Métashape, [en ligne], Disponible sur https://www.agisoft.com/pdf/manuals_other/Métashape_pro_fr_1_5.pdf (consulté le 8/04/2022)

Agisoft, Métashape python reference, [en ligne], Disponible sur https://www.agisoft.com/pdf/Métashape_python_api_1_6_0.pdf (consulté le 05/05/2022)

Albert Haung & contributors, Pybluez, [en ligne], Disponible sur https://pybluez.readthedocs.io/en/latest/index.html (consulté le 11/03/2022)

Awk, piwifi, [en ligne], Disponible sur https://github.com/awkman/pywifi (consulté le 24/03/2022)

BAC PRO SEN EIE, Les Trames NMEA, [en ligne], Disponible sur http://www.cedricaoun.net/eie/trames%20NMEA183.pdf (consulté le 14/03/2022)

Tom Flanagan, pynmea2, Disponible sur https://github.com/Knio/pynmea2 (consulté le 15/03/2022)

Cadden, La centrale inertielle, [en ligne], Disponible sur https://www.cadden.fr/centrale-inertielle-fonctionnement/, (consulté le 14/05/2022)

Centre National de Ressources Textuelles et Lexicales (CNRTL), Ortolang, [en ligne], Disponible sur https : //www.cnrtl.fr/definition/photogramm%C3%A9trie (consulté le 14/05/2022)

Emlid, Emlid Reach RS2 datasheet, [en ligne], Disponible sur https://files.emlid.com/docs/Datasheet%20RS2%20ENG%20web.pdf (consulté le 09/05/2022)

EXID, Le domaine d'application de la photogrammétrie, [en ligne], Disponible sur https : //www.exid-diagnostic.fr/les-domaines-dapplication-de-la-photogrammetrie/ (consulté le 14/05/2022)

Fabio Morbidi, Robotique industrielle, Disponible sur https://home.mis.u-picardie.fr/~fabio/Eng/documenti/Teaching/RI18-19/RobInd_Ch2p1.pdf (consulté le 8/04/2022)

Konrad Iturbe, gorpo-py-api, [en ligne], Disponible sur https://github.com/KonradIT/gopro-py-api (consulté le 17/03/2022)

KUBII, Qu'est-ce qu'un Raspberry, [en ligne], Disponible sur https://www.kubii.fr/content/72-quest-ce-que-le-raspberry-pi#:~:text=Le%20Raspberry%20Pi%2C%20con%C3%A7u%20en,de%20l'universit%C3%A9%20de%20Cambridge (consulté le 25/05/2022)

Leica Pegasus, Solution mobile pour capturer la réalité, Disponible sur file:///C:/Users/couty/Downloads/Leica_PegasusBackpack_DS.pdf, (consulté le 18/08/2022)

Phidget, What is the phidget ?, [en ligne], Disponible sur : https://www.phidgets.com/docs/What_is_a_Phidget%3F (consulté le 10/06/2022)

Phidgets Inc. **(1)**, Phidgets, [en ligne], Diposnible sur https://www.phidgets.com/docs/OS_-_Windows (consulté le 7/03/2022)

Phidgets Inc. **(2)**, Phidgets, [en ligne], Diposnible sur https://www.phidgets.com/?tier=3&catid=10&pcid=8&prodid=1204 (consulté le 8/03/2022)

Pix4D, Yaw, Pitch, Roll and Omega, Phi, Kappa angles and conversion, Disponible sur https://s3.amazonaws.com/mics.pix4d.com/KB/documents/Pix4D_Yaw_Pitch_Roll_Omega_to_Phi_Kappa_angles_and_conversion.pdf (consulté le 07/04/2022)

Pyproj, Installation, [en ligne], Diposnible sur https://pyproj4.github.io/pyproj/stable/examples.html (consulté le 15/04/2022)

Python, Asyncio, Disponible sur https://docs.python.org/fr/3/library/asyncio.html (consulté le 01/04/2022)

Sitalia, LEICA Pegasus – BackPack, [en ligne], Disponible sur : https://www.sitalia.it/fr/c/fq9bzv/leica_pegasus_-_backpack.html (consulté le 25/05/2022)

Document personnel

NIEDERBERGER G., Prototypage, analyse et qualification d'une solution de photogrammétrie mobile, 2021, Travail de fin d'étude, 76p

Liste des figures

Liste des tableaux

Élaboration d'un prototype bas coût d'instrumentation mobile pour l'acquisition de nuages de points géoréférencés par photogrammétrie mobile

Mémoire d'Ingénieur C.N.A.M., Paris 2022

RESUME

La photogrammétrie est une technique permettant de modéliser des objets ou même des bâtiments en 3D. Cela est permis grâce à l'utilisation de photos prises sous différents points de vue permettant ainsi à l'objet d'être visualiser sous plusieurs angles. La photogrammétrie est utilisée par beaucoup de professions puisqu'elle permet également de faire des mesures dans la modélisation pouvant permettre la création de projet sans pour autant être sur le terrain.

Cependant, cette méthode implique l'utilisation de points d'appui permettant au nuage de points d'être géoréférencé. La photogrammétrie mobile permet de s'affranchir de ces points puisque les images sont géoréférencées de manière direct. Bien qu'innovatrice, cette technologie n'en reste pas moins onéreuse pour la plupart des professionnels. Le développement d'un prototype bas coût permet alors à un plus grand nombre de se doter de cette technologie avec un prix nettement moins élevé.

Mots clés : photogrammétrie mobile, géoréférencement direct, poses des images, ajustage, nuage de points

---

SUMMARY

Photogrammetry is a technique to model objects or even buildings in 3D. This is possible tanks to the use of photos taken from different viewpoint allowing the object to be visualized from several angles. Photogrammetry is used by many professions as it also allows measurements to be made in modelling that allow project creation without being in the site.

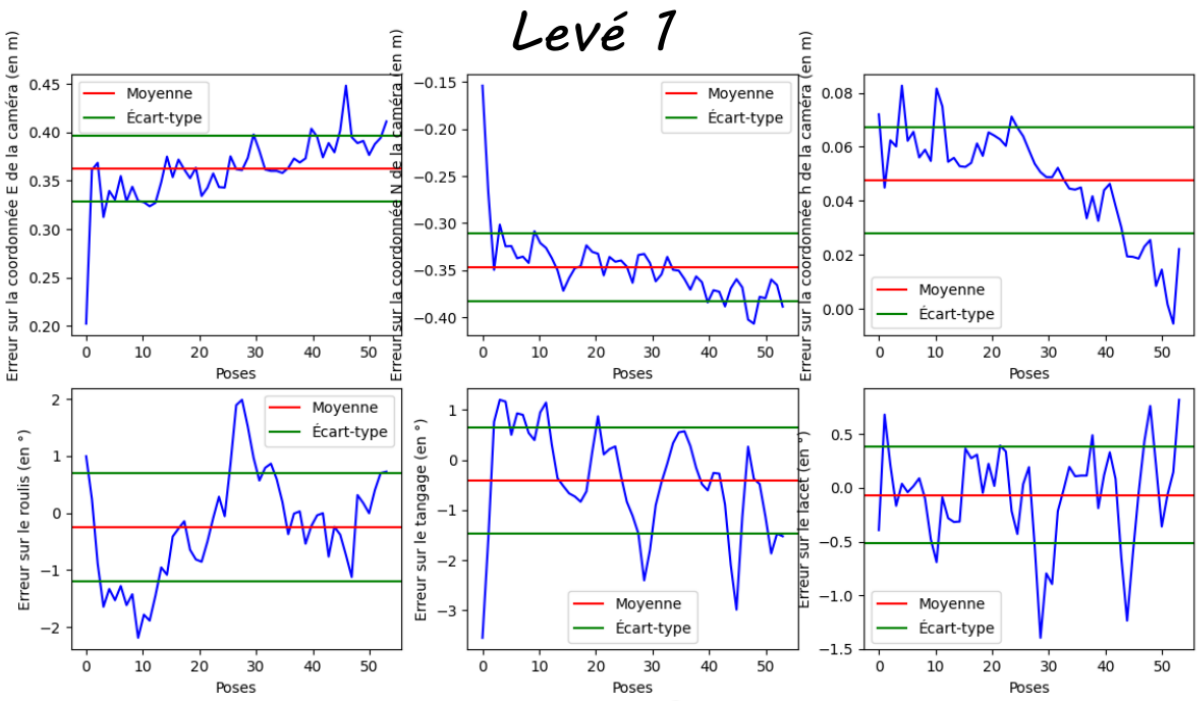
However, this method involves the use check points allowing the scatter plot to be georeferenced. Mobile photogrammetry makes it possible to dispense with these points because the images are georeferenced directly. Although innovative, this technology is still expensive for most professionals. The development of a low-cost prototype allows more people to acquire this technology at a significantly lower price.

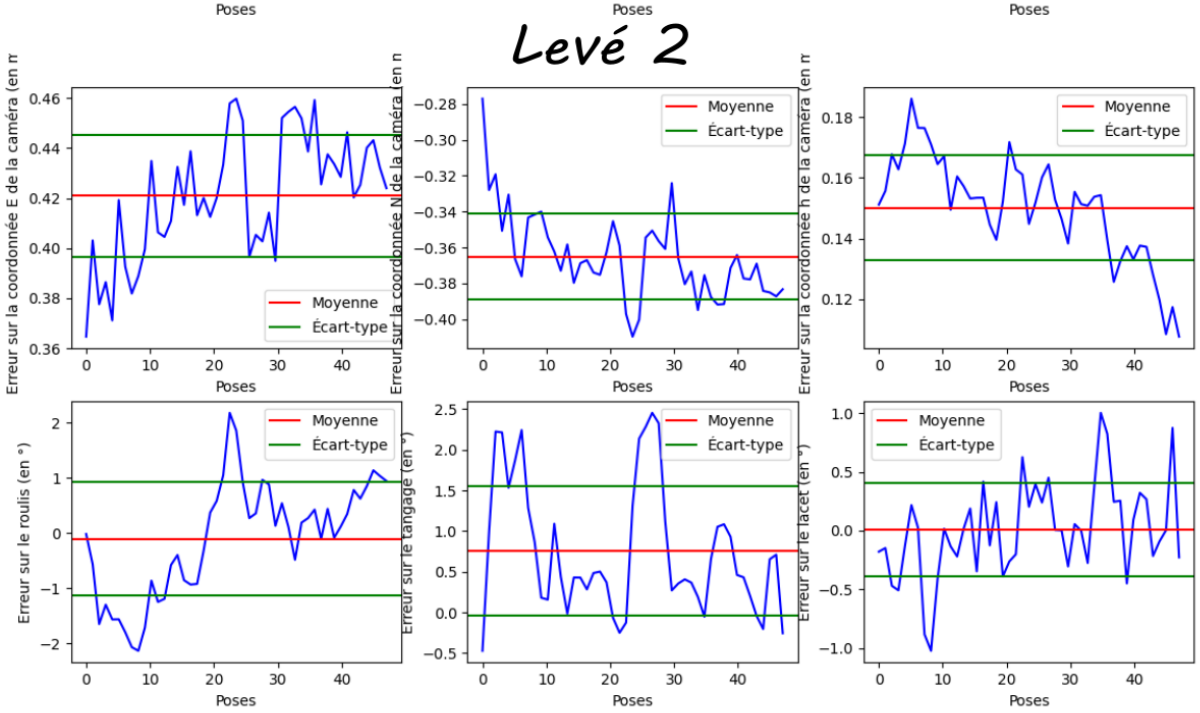
.

Key words : mobile photogrammetry, direct georeferencing, posing of images, fitting, scatter plot

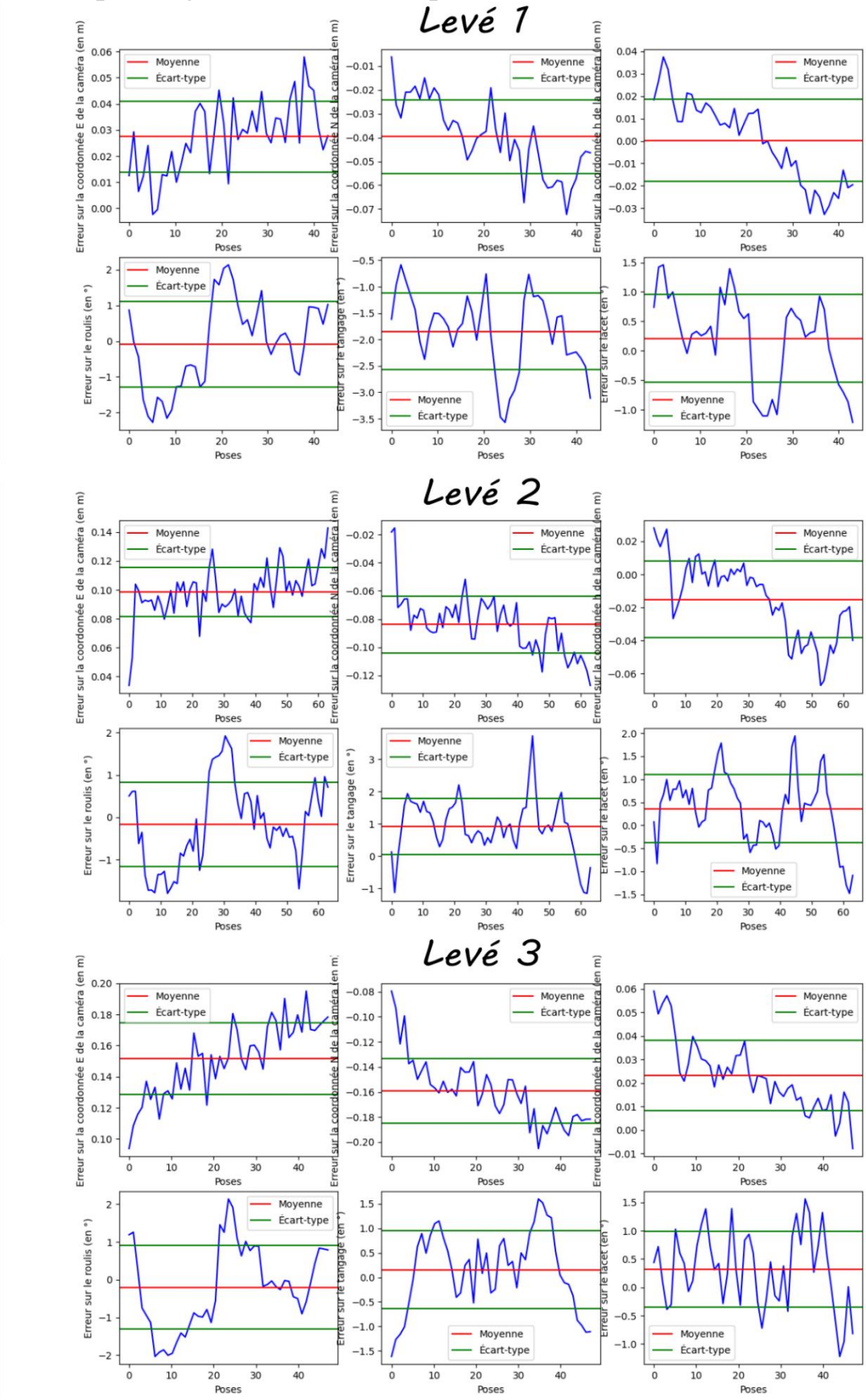
Table des annexes

Annexe 1
Graphiques des écarts entre les poses calculées par notre prototype et les poses ajustées de Metashape avec un déclenchement simultané depuis 1 programme





Annexe 2
Graphiques des écarts entre les poses calculées par notre prototype et les poses ajustées de Metashape avec un déclenchement différé



Annexe 3
Graphiques des écarts entre les poses calculées par notre prototype et les poses ajustées de Metashape avec un déchenchement simultané et 1 programme par instrument