

Rapport de projet de photogrammétrie : Solutions photogrammétriques légères

Gabin Jubault, Hoang-Viet Nguyen, Edson Olivares Medina, Victor Oxombre, Félix Quinton





Sommaire

I.	Introduction	3
н.	Prise en main	4
	1) Fonctionnalités des appareils	4
	2) Exportation des données	5
	3) Premiers tests	7
III.	Calibration	
	1) Chantiers mis en place	
	2) Traitements de données via <i>MicMac</i>	
	3) Résultats obtenus	13
	4) Solutions envisagées	
IV.	Dérive	
	1) Chantiers mis en place	
	2) Traitement des données	
	3) Résultats obtenus	20
	4) Comparaison des données	24
V.	Conclusion	27
VI.	Annexe	
VII.	Bibliographie	

I. Introduction

Ce projet intitulé *Solutions photogrammétriques légères* se déroule durant trois semaines. Celui-ci rentre dans le cadre plus global d'un projet de fin de stage. En effet, l'ensemble des élèves de première année à l'ENSG se retrouve au centre IGN de Forcalquier pour un stage de neuf semaines. Ce lieu a été choisi pour sa richesse de paysages propices aux techniques liées à la géomatique.

Ce stage a pour objectif de mettre en application l'ensemble des enseignements appris tout au long de cette première année. Cela permet notamment de mieux comprendre certaines notions et de mettre en pratique les techniques liées au domaine de la géomatique tels que la géodésie, la topométrie, la photogrammétrie et la télédétection.

Ainsi, lors de ces six premières semaines de stage, nous avons appris à utiliser les outils permettant de réaliser ces projets de fin d'année.

Notre projet a pour thématique la photogrammétrie. La photogrammétrie est l'acquisition et l'exploitation de photos pour mesurer d'une part la forme des objets ainsi que leurs couleurs et d'autre part leur restitution en 3D. Des outils informatiques adaptés sont utilisés pour traiter ces données.

Le matériel fournit pour ce stage était constitué de deux appareils :

- le *BLK3D* prêté par *Leica*, qui s'intéresse notamment aux tests que nous allons réaliser sur leur appareil
- la Mi Sphere Camera achetée par l'ENSG et mis à notre disposition

Notre cahier des charges nous demandait de tester ces produits afin de vérifier leur compatibilité avec le domaine de la photogrammétrie.

Il nous a été demandé dans un premier temps de prendre en main le matériel et les logiciels associés. Puis dans un second temps de tester leurs aptitudes à la photogrammétrie.

Pendant les 3 semaines, nous nous sommes organisés de façon à pouvoir gérer les deux appareils en même temps. Ainsi deux équipes se sont formées, un duo pour le *BLK3D* et un trio pour la *Mi Sphere Camera*. Les premiers jours ont servi à découvrir les deux caméras dont la manipulation était assez aisée, ce qui a permis de passer rapidement aux étapes suivantes. Des premières images ont été prises au centre IGN pour se familiariser avec le traitement des images fournies par les deux appareils. Des déplacements sur le terrain ont été faits pour pouvoir tester les deux caméras et ainsi analyser leurs performances.

Premièrement, nous avons voulu déterminer la calibration des deux appareils à l'aide du polygone d'étalonnage du centre IGN. Puis, nous avons effectué une batterie de tests spécifiques aux appareils. Enfin, le test principal lors de ces trois semaines de stage fut le calcul de la dérive de chacun des appareils.

II. Prise en main





Le *BLK3D* est un appareil permettant de faire des mesures rapidement en faisant de la photogrammétrie en temps réel. Ce système permet d'obtenir des mesures de distance en 3D avec une précision millimétrique. Ces acquisitions sont faites grâce aux deux capteurs photos calibrés (gauche – droite) qu'il possède dont la base fixe est connue. Cet appareil est fourni avec un flash adaptable sur ce dernier et avec une lanière afin de limiter les risques de chute. Le prix de cet appareil est d'environ 5000€. De plus, l'appareil est associé à une licence afin de traiter les données du *BLK3D* depuis un logiciel propre à *Leica*. Néanmoins, l'installation de ce logiciel n'ayant pas été possible pour les ordinateurs mis à notre disposition, il nous a été impossible de l'utiliser.

Photo LEICA BLK3D

La *Mi Sphere Camera* est, contrairement au *BLK3D*, une caméra destinée au grand public (environ 200€) qui possède aussi deux capteurs *fisheye* ayant un angle de visée de 190°, visant l'opposé l'un de l'autre afin d'obtenir une photo 360 degrés avec le moins d'angles morts possible. Cet appareil est fourni avec un trépied adapté à la taille de l'appareil.

Les mesures sur les photos prises par le BLK3D peuvent



Photo Xiaomi Mi Sphere Camera

être faites directement sur l'appareil ou sur le logiciel fournis avec (à condition qu'il soit fonctionnel). A chaque acquisition, le *BLK3D* prendra deux photos grâce aux appareils situés de part et d'autre de la machine puis affichera uniquement la photo prise avec la caméra gauche corrigée de la distosion sur l'écran de l'appareil.

Il y a deux modes d'acquisition différents disponibles sur la machine, soit prendre un unique jeu de deux photos (*single-shot*), soit prendre deux jeux de photos (*multi-shot*) en translatant l'appareil entre les deux prises de vue. Cette méthode permet une meilleure stéréoscopie et donc à l'appareil d'effectuer de meilleures mesures.



Single-shot



Multi-shot

Il est également possible de tracer des plans directement sur le *BLK3D* ou sur le logiciel auxquels l'on peut ajouter les photos prises. Cependant, cette fonctionnalité reste très restrictive et ne s'intégrait pas dans le cadre de notre projet. De plus, le logiciel permet de tracer les contours de différents objets et d'en obtenir une « modélisation 3D ». Le *BLK3D* possède également un laser servant de distance-mètre que nous n'étudierons pas ici car nous voulions nous concentrer principalement sur l'aspect photogrammétrique que peut apporter cet appareil.

L'interface de contrôle de la *Mi Sphere Camera* est très sommaire, en effet il y a uniquement trois boutons sur cette dernière permettant de l'allumer, de prendre des photos/vidéos et d'activer la Wi-Fi. Néanmoins, nous pouvons le manipuler directement avec un smartphone à partir de l'application *Mi Sphere Camera*, ce qui rend son utilisation plus pratique et complète. Grâce à cette application, nous pouvons régler les paramètres internes des deux caméras simultanément sauf la focale et l'ouverture qui sont fixes. Comme indiqué précédemment, Xiaomi a créé cet appareil avec deux capteurs, les deux optiques montées en symétrie de part et d'autre nous fournissent une image 360°, sachant que chaque optique a un champ de vision de 190°, cela permet d'avoir en théorie une zone de sécurité. Grâce à ce recouvrement, il est possible d'avoir une vue en 360° en direct depuis l'application sur smartphone.

2. Exportation des données

Dans le cas de la *Mi Sphere Camera*, lorsque nous essayons de lire les images ou vidéos depuis la carte SD nous avons deux images sous forme de cercle dans un seul fichier, ce qui correspond en effet aux images prises par chaque capteur de l'appareil. Il est donc difficile de pouvoir traiter ces images avec un logiciel comme *Micmac*.



Image Xiaomi brute



Photo Xiaomi après script Python

Nous avons mis au point un petit programme *Python* (voir annexe 9) afin de couper ces images en deux, les deux cercles correspondant à chaque capteur, afin de les utiliser séparément avec le logiciel adapté. En effet, les fichiers en sortie de l'appareil sont aux formats JPG et DNG. L'avantage du format JPG par rapport au format DNG, est le fait que nous obtenons une image avec un poids plus faible et une colorimétrie exploitable par le logiciel *MicMac.* Ce pourquoi nous avons opté, dans un premier temps, pour ce format d'image de par sa simplicité d'utilisation.

Néanmoins, utiliser les images DNG serait préférable : elles ne présentent pas de compression et ne sont pas affectées par les algorithmes internes de l'appareil photo. Cependant, un script *Python* permettant d'obtenir une bonne colorimétrie et un étalement de l'histogramme aurait été nécessaire, ce qui aurait pu être fait une fois que les traitements avec les images compressées soient aboutis. De plus, par soucis de poids, nous rajoutons une compression aux fichiers après séparations en les enregistrant au format JPG, les enregistrer dans un format *lossless* est conseillé.

Du côté du *BLK3D*, nous avons également créé un script *Python* pour différentes raisons. Il s'agit seulement d'aller chercher les différentes images contenues dans les fichiers « .cmi » de l'appareil. Le script (voir annexe 8) va alors les renommer afin de pouvoir les différencier tout en prenant en compte l'appareil d'origine des photos et si elles sont issues d'un premier jeu de photos ou d'un deuxième. Finalement les images apparaîtront sous le format suivant : « Img_k_L_1.jpg » et seront copiées dans un dossier image prévu à cet effet, k étant le numéro de l'image, « L » signifiant *left* et donc que l'image est issue de l'appareil gauche et « R » pour l'autre, enfin le « 1 » correspondant au jeu de photos *single-shot* et « 2 »pour les *multi-shot*.

Pour l'utilisation des photos par les différents logiciels, il est nécessaire d'avoir des métadonnées complètes. Or, ces informations n'étaient pas toutes contenues dans les photos fournies par le *BLK3D*. Concernant la *Mi Sphere Camera*, toutes ces informations étaient perdues lors de l'utilisation du script de séparation des images. Ainsi, il a fallu créer un fichier du type « MicMac – LocalChantierDescripteur.xml » comprenant le nom des caméras, leurs focales associées et la taille des capteurs.

3. Premiers tests

Dans un premier temps nous avons cherché à évaluer la fiabilité des mesures effectuées par le BLK3D nous avons donc réalisé quatre séries de tests. Pour cela, nous avons fixé trois ciblettes *black & white* sur trois coins d'une planche. Les distances entre ces ciblettes étant connues, nous avons pu les comparer avec les mesures effectuées par le *BLK3D*.

Test 1 : Le premier test consistait à s'éloigner de plus en plus de la planche (de 1 à 9 mètres) et à relever les distances entres les ciblettes. Voici les écarts moyens obtenus entres les différentes mesures (tableau complet en annexe 1 et 2). Les tests ont été réalisés en prenant un jeu et deux jeux de photos.



Photo test 1 du BLK3D

Ce test a, dans un premier temps, permis de mettre en évidence la précision des mesures du *BLK3D*. Il nous a aussi permis d'observer l'utilité de prendre deux jeux de photos. En effet, à partir de 7 mètres, il arrive que l'erreur de mesures avec un seul jeu de photos devienne trop importante contrairement à celle avec 2 jeux.

Leica nous fournit ces courbes où nous remarquons une erreur qui croit rapidement :



Graphique des erreurs de mesures en fonction de la distance de prise de vue fourni par *Leica*

Ici nous pouvons comparer les valeurs de précision des mesures du *BLK3D* en fonction de la distance fournie par *Leica* avec les valeurs obtenues lors de nos tests. Nos tests montrent que le BLK3D est en réalité plus précis que les données fournies par Leica. Par contre, nous n'avons pas réussi à tracer de courbes équivalentes puisque nous obtenons des résultats très bruités.

En guise de comparaison, nous pouvons tout de même préciser certaines valeurs remarquables :

- En single-shot, l'erreur n'excède pas 2cm pour des mesures entre 1m et 7,5m.
- En *single-shot*, l'erreur se situe entre 2cm et 3cm pour des mesures entre 8m et 9m.
- En multi-shot, l'erreur n'excède pas 1,5cm pour des mesures entre 1m et 9m.





Test 2 : Le second test résidait cette fois-ci non pas dans un changement de distance mais dans un changement d'angle horizontal, nous avons donc fait tourner l'appareil de 10 à 170 degrés autour de la planche (avec un rayon de 2 mètres) et avons encore une fois pris des mesures régulières.



Les résultats de ce test (fournis en annexe 3) ont également été concluants. Les mesures de distances sont certes moins précises lorsque l'angle augmente mais les variations restent très faibles.



Photo test 3 du BLK3D

Test 3: Le troisième test similaire au second consistait en une variation d'angle vertical de 0 à 70 degrés. De même, nous avons effectuées des mesures régulièrement.

Encore une fois les mesures se sont avérées précises. L'appareil semble donc peu sensible aux variations d'angles et de distances (résultats en annexe 4).

	а	b	С	d
Angles horizontaux à 2m	0.001	0.005	0.003	-
Angles verticaux à 2m	0.002	0.002	0.002	-
Distance (1-6,5m) / Single-				0.017
shot	0.002	0.006	0.005	
Distance (6,5-9m) / Single-				
shot	0.008	0.011	0.009	0.026
Distance (1-7m) / Multi-shot	0.001	0.003	0.004	0.008
Distance (7-9m) / Multi-shot	0.002	0.004	0.007	0.011

Ecarts maximum entre distances réelles et calculées

Test 4: Le *BLK3D* étant un outil servant à prendre des mesures il pourrait être utile par exemple pour mesurer rapidement le diamètre d'un arbre à partir d'une photo. Cette solution pourrait répondre par exemple à un besoin exprimé par le laboratoire d'inventaire forestier (LIF) de l'IGN. Nous avons donc effectué une série de mesures sur différents arbres sous trois formes. Une première mesure au décamètre puis une seconde en prenant un jeu de photos via le *BLK3D* et enfin une dernière mesure avec deux jeux de photos. L'étude a été menée sur 10 arbres. Voici les écarts moyens en fonction des types de mesures (tableau complet en annexe 5) :

Décamètre - 1 jeu de photos	Décamètre - 2 jeux de photos	1 jeu de photos - 2 jeux de photos
1.55 cm	1.51 cm	0.043 cm



Les mesures au décamètre étant peu précises (on mesure le périmètre de l'arbre puis nous divisons par *Pi* en supposant qu'il est totalement circulaire) elles ne servent qu'à nous donner un ordre de grandeur du diamètre des arbres. Cependant les précédents tests effectués sur le *BLK3D* ont prouvé sa fiabilité à courte distance (moins de 10m), les mesures effectuées n'étant pas aberrantes nous pouvons les supposer précises au cm. Le *BLK3D* pourrait donc être utilisé pour de la mesure de diamètre d'arbre. Le seul problème résiderait dans la vitesse de prise de photos, il faut compter au moins 10 secondes entre deux acquisitions.

Photo diamètre arbre avec *BLK3D*

Pour la *Mi Sphere Camera*, il a été entrepris de calculer la valeur des focales des caméras manuellement afin de les comparer aux données du constructeur. Pour cela, nous placions une règle quelque part puis nous la photographiions avec chacune des caméras depuis une distance connues précisément. Néanmoins, la résolution de ces caméras étant faible, il n'était pas possible de repérer de façon précise les échelles sur la règle. Ainsi, nous avons considéré comme vraie la valeur de focale des caméras fournie par le constructeur.

De plus, la caméra de *Xiaomi* est normalement dotée d'une Wi-Fi détectable jusqu'à 50 mètres, ce qui s'est avéré faux lors de nos multiples utilisations de cet appareil. Cela peut poser problème lors de l'acquisition de photos car dans certaines situations le photographe sera forcément visible sur les clichés. Or cela peut gêner lors du traitement photogrammétrique.

Enfin, cet appareil chauffe énormément. Ainsi, son utilisation avec le trépied fourni est indispensable car la *Mi Sphere Camera* ne peut être tenue directement avec les mains. De plus, cela a une influence notable sur sa batterie qui a une autonomie d'une heure et trente minutes.

III. Calibration

La calibration de l'appareil est le processus permettant d'obtenir le modèle de distorsion et les coordonnées du point principal de symétrie (PPS). Cette méthode est représentée par deux types de paramètres : les paramètres intrinsèques qui sont les paramètres de l'objectif d'une seule caméra et les extrinsèques qui concernent l'orientation de l'appareil. De plus, nos appareils étant tous deux équipés de deux caméras, il est possible de déterminer la base fixe entre celles-ci, c'est-à-dire le vecteur et la matrice de rotation entre les centres optiques des caméras.

1. Chantiers mis en place

Pour la calibration des appareils mis à disposition, nous avons décidé de commencer les mesures nécessaires sur le polygone de calibration du centre IGN dont les coordonnées terrains des ciblettes étaient déjà connues grâce à des mesures topométriques faites les années précédentes.



Pour la calibration du *BLK3D*, nous avons pris en photo un angle du polygone sous différents points de vue. Puis nous avons traité les données à l'aide de *MicMac*.

Polygone d'étalonnage du centre IGN

Pour la calibration de l'appareil de *Xiaomi*, nous avons travaillé sur trois chantiers différents, à savoir le polygone de calibration du centre IGN, la salle de rattrapage du centre IGN et l'intérieur de la chapelle de Pierrerue.

Après trois acquisitions, le polygone de l'IGN nous a donné une calibration de la *Mi Sphere Camera* mais celle-ci nous a posée des problèmes car elle ne s'adaptait pas aux autres mesures réalisées.

Nous avons donc décidé de faire des photos dans une salle du centre afin d'avoir un endroit fermé où l'on pouvait exploiter entièrement les capacités des capteurs *fisheye*. Nous avons collé des cibles de photogrammétrie sur les murs de la salle de façon homogène pour faire de la topométrie à l'intérieur afin de pouvoir traiter les images après la calibration. Malheureusement la calibration n'a pas été réussie car *MicMac* ne trouvait toujours pas assez de points homologues et cela en raison des murs trop homogènes et n'ayant pas de texture.



De ce fait, nous sommes allés faire des photos à l'intérieur de la chapelle de Pierrerue. Nous avons utilisé les cibles déjà mises en place par nos prédécesseurs comme des points d'appui et grâce aux G1 de l'ENSG nous avons pu gagner du temps en exploitant leurs mesures fournissant les coordonnées des cibles.

Acquisition photos Mi Sphere Camera à la chapelle de Pierrerue

2. Traitement des données via MicMac

Pour le *BLK3D*, nous avons, dans un premier temps, effectué les calculs de calibration sur les deux appareils séparément à partir des photos propres à chacun. Dans un second temps, la calibration a été faite sur les deux appareils simultanément.

Ces calibrations ont toutes été effectuées à partir du logiciel *MicMac*. Pour cela, nous avons tapé manuellement certaines commandes (voir annexe 6) dans un terminal ouvert dans le dossier comprenant le lot de photos à traiter.

Pour la *Mi Sphere Camera*, le lot de photos utilisé pour l'étalonnage des caméras est celui de la chapelle de Pierrerue. Comme pour le BLK3D, nous avons utilisé dans un premier temps le logiciel *MicMac* pour traiter les clichés. La commande *Tapioca* a été utilisée de la même façon. De plus, cette commande était parfois complétée par un *SFS*, qui est une commande permettant de rehausser les contrastes notamment sur des murs homogènes. Le *Schnaps*, quant à lui, ne pouvait pas être utilisé car il y avait trop peu de points de liaison trouvés par la commande précédente. Enfin, la commande *Tapas* faisait diverger les calculs et ne permettait donc pas d'obtenir la calibration des caméras. Il était alors nécessaire de compléter cette commande et de la décomposer en plusieurs étapes :

Etape 1: mm3d Tapas FishEyeBasic .*JPG PropDiag=0.65 DRMax=1 Out=RMax1Reg RegulDist=[1,10,100]

Etape 2: mm3d Tapas FishEyeBasic .*JPG PropDiag=0.65 Out=Reg InOri=RMax1Reg RegulDist=[1,10,100]

Etape 3: mm3d Tapas FishEyeBasic .* JPG PropDiag=0.65 Out=MeP InOri=Reg

Pour toutes ces étapes, le modèle de calibration choisi fut le *FishEyeBasic* étant donné le type de caméra utilisée. La commande *PropDiag* permet d'ignorer la zone noire autour des images et donc de garder uniquement un cercle de rayon utile de 65%, correspondant au cercle des images fournies par les caméras.

Lors de l'étape 1, le *DRMax=1* permet de limiter le polynôme de distorsion au premier terme de celui-ci.

Ensuite, lors des étapes 2 et 3, nous fournissons une solution initiale établie lors des étapes précédentes afin de fournir une approximation des calibrations des appareils et de l'orientation des photos. Cela permet d'aider la mise en place des caméras à chaque étape.

Cependant, nous n'avons pas d'idée sur la fonction de la commande *RegulDist* car celle-ci n'apparaît jamais sur la documentation de *MicMac* et nous n'avons pas eu de réponse des développeurs sur le sujet.

3. Résultats obtenus

Pour la calibration du BLK3D, nous n'avons pas rencontré de problèmes particuliers une fois l'appareil pris en main. En effet, les premières commandes *MicMac* étaient déjà connu et le modèle de calibration aussi *RadialStd*. De plus, grâce au *LocalChantierDescripteur.xml*, nous obtenions directement le résultat de calibration des deux caméras.

Ces résultats donnent des écarts entre PPA ou PPS et le centre du capteur d'une dizaine de pixels. De plus, ces écarts sont similaires d'un capteur à l'autre. De même, au niveau des différents coefficients de distorsion, ils varient peu d'un capteur à l'autre. Par conséquent, nous en avons conclu que les calibrations étaient bonnes et qu'il n'y avait pas de capteur défectueux.

Nous avons ensuite exploité les données de calibration que fournies *Leica* dans ses fichiers d'exports. Elles étaient sous la forme d'une calibration *OpenCV*. Nous avons réussi à exporter les valeurs du coefficient du polynôme de distorsion. Grâce à cela, nous avons tracé le graphique de la calibration calculée à l'aide de *MicMac* avec les données *Leica* :



D'après ce graphique, nous avons une différence de calibration importante avec des valeurs élevées de correction de distorsion. Nous avons donc pris en photo un quadrillage pour lui appliquer notre modèle à l'aide de *MicMac* afin de valider ou non ce dernier.



Nous en concluons alors que notre calibration semble correcte et que le modèle de *Leica* est décalé.

Pour la *Mi Sphere* Camera, malgré le fichier *pipeline* pour la commande *Tapas* fournie généreusement par des développeurs *MicMac*, le calcul résultant de cette dernière n'arrivait jamais à son terme. En effet, lors de la troisième étape du *Tapas*, le calcul divergeait quel que soit le lot de photos utilisé. Ainsi, nous avons à notre disposition uniquement une approximation de la calibration des deux caméras de l'appareil et des orientations des clichés.



AperiCloud de la mise en place de la Mi Sphere Camera

Cette illustration ci-contre montre que le chantier semble être bien retranscris par *MicMac*. Néanmoins, nous pouvons remarquer rapidement que la mise en place des caméras semble fausse, notamment à cause de la base de l'appareil qui ne semble pas être respectée. De plus, on retrouve parfois des bouts de murs courbés prouvant que la mise en place s'est mal déroulée.

Une fois la calibration des deux caméras obtenue pour chaque appareil, il est possible de déterminer leurs bases. Pour cela, nous utilisions la commande *Blinis*. De plus, cette commande fixe le référentiel au niveau de l'axe optique d'une caméra puis calcul la distance vers l'autre axe optique dans ce même référentiel.

Exemple de commande : mm3d Blinis .* jpg Ori-Absolue/Loc-Assoc-Im2Block Blinis.xml

Ainsi, à partir de l'orientation absolue de chaque caméra pour le même lot d'images prises simultanément, cette commande permet de calculer la distance en mètre entre les deux centres optiques des caméras, ainsi que la matrice de rotation de chacune des caméras.

Cela a été utilisé sur le BLK3D et nous avons obtenu de nouveaux résultats intéressants. En effet, en cherchant à l'intérieur des fichiers en sortie du BLK3D, nous pouvons repérer des fichiers de calibration avec positions et matrices de rotation des deux caméras. Les différentes valeurs étant semblables aux nôtres, cela a permis de valider notre calibration. Nous avons alors pu conserver ces données afin de les utiliser sur les chantiers suivants.

Base fournie par Leica	0.151162	-0.046216	-0.000278
(m)			
Base calculée par	0.153143	-0.047414	0.000290
photogrammétrie (m)			
Différence (m)	0.001981	0.001198	0.000568

Pour la *Mi Sphere Camera*, le problème venait certainement du modèle de calibration utilisé dans *MicMac*, qui semble ne pas pouvoir prendre en compte toutes les déformations des images de la *Xiaomi Mi Sphere Camera*. En effet, le modèle de distorsion des caméras de la *Mi Sphere Camera* semble plus complexe que celui d'un appareil *fisheye* classique.

Pour l'appareil de *Xiaomi*, le vecteur donné par cette commande confirme que notre modèle de calibration est faux pour cet appareil. En effet, pour chaque valeur comprise dans le vecteur, celle-ci est plus grande que les dimensions de l'appareil. Ainsi, *MicMac* n'est apparemment pas adapté pour ce type d'appareil photo.



Vecteur de base de la *Mi Sphere Camera* en mètre

4. Solutions envisagées

Pour résoudre ce problème de distorsion, nous avons voulu travailler sur des images corrigées de la distorsion, obtenues à partir d'une commande *Drunk*. Lorsque les images étaient corrigées de cette distorsion, la commande *Tapioca* trouvait un plus grand nombre de points de liaison.



Illustration de la commande Drunk sur une image issue de la Mi Sphere Camera

Cependant le problème de cette commande est qu'elle corrige les images de la distorsion mais elle coupe ensuite une partie de l'image pour en fournir une en sortie. Cela pose problème quant à l'utilisation d'un appareil 360° car la commande supprime des zones de recouvrement importantes. Nous avons donc entrepris d'agrandir la taille de l'image en ajoutant des bordures noires autour des images d'origine avec un script avec un script *Python* (voir annexe 10). Ainsi, lors de la découpe de celles-ci par le *Drunk*, nous conserverions les zones de recouvrement.



Illustration de la solution apportée avec l'ajout de bandes noires

Le problème de l'ajout de ces bandes manuellement est le décalage des coordonnées de toutes les données. Il a donc fallu créer un script *Python* permettant de translater toutes les coordonnées contenues dans les fichiers au format xml de l'appareil.

Une fois cette étape terminée, il aurait fallu faire une commande *Tapioca* sur ces images modifiées afin d'obtenir des coordonnées des points de liaison. Ensuite, un script *Python* aurait été codé afin de repositionner ces coordonnées dans un repère image. En lisant les données générées par Tapioca (Pastis), nous aurions obtenu la position des points de liaison dans le repère de l'image après correction de la distorsion par *Drunk*. Ces coordonnées auraient ensuite été transformées, en connaissant le modèle de distorsion et la calibration, vers le repère image de l'image distordue. Pour ensuite les traiter avec une commande *Tapas*. D'ailleurs, nous soupçonnons la commande *Drunk* de ne pas prendre en compte tous les paramètres de distorsion pour corriger les images.

Néanmoins, ce dernier script n'a pas été finalisé car le modèle de calibration de l'appareil n'était pas connu, la documentation *MicMac* du modèle de distorsion ne semble pas être celle implantée dans *FishEyeBasic* (nombre de paramètres différents) et la lecture des points homogènes nécessite des détails sur la structure de ces fichiers. Cependant après avoir fait de nouvelles acquisitions sur la chapelle de Pierrerue, nous nous sommes aperçus que la quantité de points de liaison n'était pas le problème. En effet, cette dernière acquisition n'aboutissait pas alors que les images avaient beaucoup de points de liaisons. Cette solution a donc été mise de côté.

IV. Dérive

Notre objectif était de quantifier la dérive c'est-à-dire la déformation de la mise en place. En effet, ce phénomène traduit le fait qu'une mise en place photogrammétrique basée uniquement sur des points de liaison peut ne pas être précise et le modèle peut être impacter par des déformations. Dans notre cas, nous avons utilisé des points d'appuis pour se géoréférencer et des points de contrôle pour contrôler la mise en place.

1. Chantiers mis en place



Afin de calculer la dérive des deux appareils, nous avons effectué un premier jeu de mesures avec le BLK3D sur le polygone d'étalonnage du centre IGN dont les ciblettes sont connues en coordonnées. Grâce à cela, le calcul d'une première dérive du BLK3D a pu être réalisé.

Une seconde mesure a été effectuée sur un chantier plus conséquent dans une partie de la rue Saint Pierre à Forcalquier. Ce chantier a été prévu dans cette rue spécifiquement car elle est assez étroite, éloignée du centre de Forcalquier afin de limiter l'afflux de personnes et de par sa proximité du centre IGN. Ainsi, cette rue avait pour objectif de mettre en valeur l'intérêt d'utiliser une caméra 360° telle que la nôtre.

Nous y avons installé 4 stations et 8 ciblettes et y avons réalisé une polygonale. Cela nous a permis de connaître les coordonnées des ciblettes dans un référentiel local. Après traitement des données de la polygonale, le résidu le plus élevé obtenu entre les stations était de 9mm et celui pour les ciblettes était de 3mm, ce qui est suffisant pour la précision des appareils étudiés.



Plan de la polygonale de la rue

Nous noterons que faire la topographie d'une rue étroite n'est pas évident car les stations tachéomètres sont positionnées de manière linéaire, ce qui n'est pas optimal. De plus, l'inter visibilité étant faible, la redondance dans les mesures l'est aussi ce qui ajoute à l'imprécision des mesures.

Photo de la rue Saint Pierre à Forcalquier

Nous avons ensuite réalisé deux acquisitions photographiques de la rue, une première au *BLK3D* et une seconde avec la *Mi Sphere Camera*. Les photos avec la *Mi Sphere Camera* ont été prises de deux manières différentes : d'une part avec les caméras orientées face aux murs et d'autres part avec les caméras dirigées vers la rue. Cela nous permet d'avoir des lots de photos différents à traiter. Pour le *BLK3D*, les photos ont été prises en *single-shot* afin de limiter le nombre de données en post-traitement. De plus, les photographes essayaient d'avoir une distance de 2 mètres entre chaque prise de vue.

2. Traitement des données

Les données fournies par le *BLK3D* étaient traitées avec le logiciel *MicMac*. Ainsi, les premières étapes des traitements de données étaient les mêmes que lors du paragraphe sur la calibration. Cependant, la commande du *Tapas* était complétée par un *InCal* permettant de donner en entrée du calcul les fichiers de calibration des caméras calculés lors de l'étalonnage des appareils. Les commandes utilisées peuvent être trouvées en annexe 7.

Les mesures ont été effectuées sur les deux capteurs du *BLK3D* séparément, puis sur les deux simultanément en modifiant : la calibration, la base ou la résolution des images. Comme précisé précédemment, nous avons traité les données sous *MicMac* et il a fallu exploiter les données issues du *GCPCtrl*. Pour cela, nous avons fait un script sous *Python* pour automatiser le tri des données et la conception de graphique pour améliorer la lisibilité des résultats.

Etant donné les problèmes obtenus avec *MicMac* dans le traitement des données de la *Mi Sphere Camera*, nous avons décidé de traiter les données à partir d'un nouveau logiciel. A l'aide du groupe P2 s'occupant du parangonnage des logiciels photogrammétriques, nous avons pu nous apercevoir que la plupart de leurs logiciels testés ne supportaient pas le type de données, c'est-à-dire les images circulaires, que l'on devait traiter. Cependant, d'après un rapport cité en bibliographie, certains logiciels comme *Metashape* traitaient les images *stitched*.



Image stitched d'une prise de vue faite avec l'appareil de Xiaomi

Ainsi, nous avons décidé de traiter les données de la dérive avec *Metashape*. Il était alors nécessaire de transformer les images fournies par la *Mi Sphere Camera* en images *stitched*, c'est-à-dire la combinaison des photos prises par chaque caméra indépendamment. Cela était possible grâce au logiciel *Mi Sphere Camera* fourni par *Xiaomi*.

Deux versions de *Metashape* étaient disponibles pour une période d'essai, nous avons choisi la version professionnelle car celle-ci était plus complète. En effet, elle possédait un modèle de calibration sphérique que ne possédait aucun autre logiciel testé. Or ce type de calibration convenait plus à notre appareil qu'un modèle *fisheye* classique.

Les photos fournies étant des images *stitched,* nous avons obtenu une image panoramique 360°. *Metashape* n'avait donc pas besoin de paramètres de calibration pour obtenir la mise en place de la prise de vue.

Tout comme *MicMac*, ce logiciel nous permettait de calculer les orientations des clichés dans un repère relatif puis de les transformer grâce à des points d'appuis vers un repère absolu. La bascule était aussi possible après avoir pointé la position des points d'appuis sur certaines images.

3. Résultats obtenus

La première dérive obtenue fut celle calculée à partir du polygone d'étalonnage du centre IGN avec le *BLK3D*. Ainsi, ce polygone possédait 51 points de contrôle et nous n'en fixions que trois en points d'appui en début d'acquisition. La dérive alors obtenue pour les photos gauche et droite est la suivante :



Nous remarquons que l'approximation en un polynôme de degré deux a un sens. Pour simplifier la comparaison des différents jeux de données, nous avons donc utilisé cette interpolation :



Nous avons ensuite déterminé la dérive des appareils sur une zone de 120 mètres de long afin de pouvoir observer le phénomène de dérive sur une distance plus significative.

Au niveau des résultats pour le *BLK3D*, nous avons commencé par comparer les deux capteurs entre eux. Nous remarquons que nous obtenons de meilleurs résultats sur le capteur gauche que sur le capteur droit. La combinaison des images issues de ces deux capteurs semble nous donner, au niveau des résultats, une moyenne des deux dérives. L'avantage de la stéréoscopie réside donc dans la connaissance de la base et non dans la multiplication des points homologues.

Concernant les bases, l'utilisation de celle trouvée dans les données *Leica* ou celle que nous avons calculée offre des résultats similaires. Les valeurs des deux bases sont alors assez proches pour que nous puissions négliger la différence et considérer les deux justes. Dans tous les cas, le renseignement de la base fixe dans le traitement permet une amélioration notable des résultats.

Pour la calibration, nous remarquons que laisser les paramètres de calibration libre lors de la dernière mise en place permet d'obtenir de meilleurs résultats. Cela montre alors que notre calibration initiale n'était pas optimale.

Concernant les résultats numériques, nous avons sur le polygone d'étalonnage une dérive qui semble croître en distance au carré alors qu'au niveau de la dérive étudiée sur la rue, cela ressemble plutôt à une courbe linéaire. Néanmoins, la rue avait une configuration complexe et nous avons donc limité le nombre de cibles à installer.

Cela ne nous permet donc pas de trouver une fonction pour approximer cette dérive. De plus, nous ne pouvons pas appliquer les fonctions interpolées sur le polygone à la rue, les erreurs ne correspondent pas. Nous n'avons donc pas de modélisation finale de la dérive du *BLK3D*.



Le graphique résumant l'ensemble de ces remarques est le suivant :

Ensuite, le type de données obtenu avec *Metashape* depuis les données de la *Mi Sphere Camera* était de cette forme :



Mise en place des sommets de prise de vue depuis Metashape avec les données de la Mi Sphere Camera

Ainsi, le fait que la caméra de *Xiaomi* possède une vue 360° devrait permettre d'avoir une faible dérive : le grand angle de vision permettrait d'avoir des points de liaison sur un plus grand angle. L'orientation entre deux clichés successifs devrait donc être meilleure. Cependant, nous constatons que les images restent fortement déformées, notamment au centre de l'image et à ses bords. Nous pouvons supposer que l'algorithme utilisé pour fusionner les deux photos n'est pas optimal, et que l'apport de distorsion est nonnégligeable. C'est pourquoi, nous apercevons sur les résultats que la dérive reste importante même pour ce type d'appareil photo.



Ce graphique permet de confirmer le fait que l'utilisation de deux photos par prises de vue avec la *Mi Sphere Camera* diminue le phénomène de dérive, ce qui notamment due à l'agrandissement des zones de recouvrement entre les clichés. Ici, le lot d'une photo correspond aux photos prises avec les objectifs orientés vers les façades. Ainsi, pour les lots deux photos, nous sommes venus ajouter le lot de photos avec les objectifs orientés vers la rue au lot précédent. De ce fait, le logiciel parvient à trouver plus aisément des points de liaison ce qui rigidifie l'ensemble de notre bloc d'acquisition. D'où la diminution des écarts dans la dérive.

4. Comparaison des données

De plus, comme nous nous sommes occupés des données issues de la *Mi Sphere Camera* sur *Metashape*, nous en avons profité pour comparer les données traitées par *MicMac* avec celles traitées par *Metashape* pour le lot de photos fourni par le *BLK3D*.



Ainsi, *Metashape* semble plus précis pour calculer la dérive de cet appareil. D'ailleurs, l'utilisation de *Metashape* semblait plus aisée en comparaison à *MicMac*. Néanmoins, nous n'avions aucun contrôle sur *Metashape* alors que *MicMac* laisse libre accès à plus de paramètres dans les actions effectuées.

Ensuite, nous avons voulu comparer les dérives entre le *BLK3D* et la *Mi Sphere Camera*. Pour cela, nous avons fait une comparaison sur le logiciel *Metashape* afin une dérive comparable entre les deux jeux de données. Les lots de photos choisis pour cela correspondent alors aux dérives minimales acquises pour chaque appareil.



Ainsi, nous pouvons remarquer que la dérive du *BLK3D* est nettement moins importante que la *Mi Sphere Camera*. Cette différence est certainement due au mauvais modèle de calibration utilisé par *Metashape* pour l'appareil de *Xiaomi*. En effet, les images *stitched* sont traitées avec un modèle de calibration sphérique. Cela ne correspond peut-être pas parfaitement à notre modèle.

Enfin, nous avons voulu comparer le meilleur de nos deux appareils d'après les résultats obtenus, c'est-à-dire le *BLK3D*, avec un appareil photo classique de la marque *Sony*.



Ainsi, nous apercevons qu'à partir d'une certaine distance, la dérive du *BLK3D* a tendance à diminuer vis-à-vis de la dérive l'appareil Sony. De plus, la courbe de dérive de l'appareil de *Leica* correspond à la pire dérive obtenue. De ce fait, lorsque l'on ajoute la base fixe, la dérive du *BLK3D* diminue grandement, ce qui prouve que l'apport d'une base fixe et donc de la stéréoscopie apporte grandement à la photogrammétrie.

V. Conclusion

Ces trois de semaines nous ont permis de réutiliser les compétences acquises durant cette année scolaire mais également durant les six semaines de stage précédent le projet. Ce dernier a mélangé plusieurs facettes de la géomatique telles que la photogrammétrie, la topométrie et la programmation.

Néanmoins, l'inconvénient de ce projet était la séparation des tâches en deux groupes afin traiter les appareils séparément.

Cependant, cela ne nous a pas empêchés de nous entraider lorsque des problèmes survenaient sur les appareils ou le traitement de données.

De plus, l'ensemble des tests effectués sur le *BLK3D* se sont déroulés globalement sans accroc. Cela a permis une avancée rapide et une diversification des procédés dans l'acquisition des données et dans leurs traitements. Tandis que pour la *Mi Sphere Camera*, beaucoup de temps et de motivations ont été perdus dans les tentatives de calibration de l'appareil qui ont donc toutes échouées. Néanmoins, la découverte du logiciel *Metashape* parvenant à traiter nos données a réussi à ramener en engouement au sein du groupe. Nous sommes alors parvenus aux objectifs fixés en début de projet (mis à part la calibration de la *Mi Sphere Camera*) qui étaient de qualifier le *BLK3D* et vérifier la compatibilité des appareils avec la photogrammétrie.

Ainsi, de nombreux tests ont été réalisés sur le *BLK3D* puis comparés avec les données fournies par *Leica*. D'ailleurs, ce dernier étant intéressé par nos tests, il sera heureux de constater que ceux-ci ont été à la hauteur voire meilleur que leurs espérances.

Ensuite, nous avons pu comparer la qualité de nos appareils avec un appareil photo classique dans le domaine de la photogrammétrie. Notre comparaison se basait uniquement sur les dérives des différents appareils.

De ce fait, le *BLK3D* parvient à concurrencer ce type d'appareil classique, notamment avec l'utilisation de la base fixe et donc de la stéréoscopie. Néanmoins, nous pouvons dire que l'utilisation de la *Mi Sphere Camera* n'est a priori pas conseillée dans ce domaine vis-à-vis d'un appareil photo classique.

Cependant, il ne faut pas délaisser la *Mi Sphere Camera* dans le domaine de la photogrammétrie. En effet, il serait nécessaire de trouver un logiciel capable de traiter les données correctement, cela permettrait alors d'utiliser de façon optimale les qualités que peut apporter cette caméra notamment dans des lieux clos ou exiguës.

VI. Annexe

Annexe 1

Test 1, single shot:

Distance	a	Réel	Écart
1,5	0,171	0,171	0
2	0,17	0,171	0,001
2,5	0,172	0,171	0,001
3	0,172	0,171	0,001
3,5	0,172	0,171	0,001
4	0,17	0,171	0,001
4,5	0,171	0,171	0
5	0,171	0,171	0
5,5	0,171	0,171	0
6	0,173	0,171	0,002
6,5	0,171	0,171	0
7	0,179	0,171	0,008
7,5	0,174	0,171	0,003
8	0,21	0,171	0,039
8,5	0,175	0,171	0,004
9	0,171	0,171	0

Tableau des mesures de distance « a » en fonction de la distance à la cible

Distance m	Bm	Réel m	Écart m
1,5	0,479	0,48	0,001
2	0,48	0,48	0
2,5	0,48	0,48	0
3	0,48	0,48	0
3,5	0,486	0,48	0,006
4	0,481	0,48	0,001
4,5	0,479	0,48	0,001
5	0,483	0,48	0,003
5,5	0,485	0,48	0,005
6	0,49	0,48	0,01
6,5	0,482	0,48	0,002
7	0,481	0,48	0,001
7,5	0,481	0,48	0,001
8	0,49	0,48	0,01
8,5	0,491	0,48	0,011
9	0,487	0,48	0,007

Tableau des mesures de distance « b » en fonction de la distance à la cible

Distance	С	Réel	Écart
1,5	0,51	0,511	0,001
2	0,511	0,511	0
2,5	0,512	0,511	0,001
3	0,513	0,511	0,002
3,5	0,515	0,511	0,004
4	0,513	0,511	0,002
4,5	0,511	0,511	0
5	0,513	0,511	0,002
5,5	0,512	0,511	0,001
6	0,516	0,511	0,005
6,5	0,514	0,511	0,003
7	0,52	0,511	0,009
7,5	0,514	0,511	0,003
8	0,514	0,511	0,003
8,5	0,518	0,511	0,007
9	0,517	0,511	0,006

Tableau des mesures de distance « c » en fonction de la distance à la cible

Distance	d	Réel	Écart
1,5	-	-	-
2	-	-	-
2,5	1,579	1,575	0,004
3	1,582	1,575	0,007
3,5	1,579	1,575	0,004
4	1,58	1,575	0,005
4,5	1,581	1,575	0,006
5	1,58	1,575	0,005
5,5	1,585	1,575	0,01
6	1,592	1,575	0,017
6,5	1,583	1,575	0,008
7	1,58	1,575	0,005
7,5	1,583	1,575	0,008
8	1,597	1,575	0,022
8,5	1,599	1,575	0,024
9	1,601	1,575	0,026

Tableau des mesures de distance « d » en fonction de la distance à la cible

Point	a	b	С
Moyenne des mesures m	0,1745625	0,4834375	0,5139375
Différence m	0,04	0,012	0,01

Tableau des moyennes et différences maximales sur les mesures effectuées

Annexe 2

Test 1, multi-shot :

Distance	a	Réel	Écart
1,5	0,171	0,171	0
2	0,17	0,171	0,001
2,5	0,171	0,171	0
3	0,17	0,171	0,001
3,5	0,171	0,171	0
4	0,17	0,171	0,001
4,5	0,171	0,171	0
5	0,17	0,171	0,001
5,5	0,172	0,171	0,001
6	0,172	0,171	0,001
6,5	0,171	0,171	0
7	0,172	0,171	0,001
7,5	0,173	0,171	0,002
8	0,17	0,171	0,001
8,5	0,171	0,171	0
9	0,17	0,171	0,001

Tableau des mesures de distance « a » en fonction de la distance à la cible

Distance	b	Réel	Écart
1,5	0,48	0,48	0
2	0,48	0,48	0
2,5	0,481	0,48	0,001
3	0,482	0,48	0,002
3,5	0,482	0,48	0,002
4	0,48	0,48	0
4,5	0,48	0,48	0
5	0,48	0,48	0
5,5	0,483	0,48	0,003
6	0,482	0,48	0,002
6,5	0,481	0,48	0,001
7	0,484	0,48	0,004
7,5	0,483	0,48	0,003
8	0,482	0,48	0,002
8,5	0,481	0,48	0,001
9	0,482	0,48	0,002

Tableau des mesures de distance « b » en fonction de la distance à la cible

Distance	С	Réel	Écart
1,5	0,51	0,511	0,001
2	0,511	0,511	0
2,5	0,512	0,511	0,001
3	0,511	0,511	0
3,5	0,513	0,511	0,002
4	0,511	0,511	0
4,5	0,514	0,511	0,003
5	0,515	0,511	0,004
5,5	0,514	0,511	0,003
6	0,51	0,511	0,001
6,5	0,511	0,511	0
7	0,518	0,511	0,007
7,5	0,513	0,511	0,002
8	0,51	0,511	0,001
8,5	0,514	0,511	0,003
9	0,515	0,511	0,004

Tableau des mesures de distance « c » en fonction de la distance à la cible

Distance	d	Réel	Écart
1,5	1,576	1,575	0,001
2	1,58	1,575	0,005
2,5	1,58	1,575	0,005
3	1,583	1,575	0,008
3,5	1,583	1,575	0,008
4	1,579	1,575	0,004
4,5	1,583	1,575	0,008
5	1,581	1,575	0,006
5,5	1,581	1,575	0,006
6	1,582	1,575	0,007
6,5	1,58	1,575	0,005
7	-	-	-
7,5	1,586	1,575	0,011
8	1,577	1,575	0,002
8,5	1,583	1,575	0,008
9	1,578	1,575	0,003

Tableau des mesures de distance « d » en fonction de la distance à la cible

Distance	а	b	С
Moyenne m	0,1709375	0,4814375	0,512625
Différence m	0,003	0,004	0,008

Tableau des moyennes et différences maximales sur les mesures effectuées

Annexe 3

Test 3, Angles verticaux :

Angles	a	Réel	Écart
0	0,171	0,171	0
	0,173	0,171	0,002
	0,169	0,171	0,002
	0,171	0,171	0
	0,172	0,171	0,001
	0,171	0,171	0
70	0,17	0,171	0,001

Tableau de l'évolution de la mesure « a » en fonction de l'angle vertical

Angles	b	Réel	Écart
0	0,482	0,48	0,002
	0,481	0,48	0,001
	0,481	0,48	0,001
	0,482	0,48	0,002
	0,48	0,48	0
	0,482	0,48	0,002
70	0,481	0,48	0,001

Tableau de l'évolution de la mesure « b » en fonction de l'angle vertical

Angles	С	Réel	Écart
0	0,511	0,511	0
	0,512	0,511	0,001
	0,512	0,511	0,001
	0,51	0,511	0,001
	0,509	0,511	0,002
	0,512	0,511	0,001
70	0.511	0.511	0

Tableau de l'évolution de la mesure « c » en fonction de l'angle vertical

Angles	MoyEcarts	
0	0 0,000666666666666	
	0,001333333333333	
	0,001333333333333	
	0,001	
	0,001	
	0,001	
70	0,000666666666666	

Tableau des écarts moyen sur les mesures de « a », »b », et « c » en fonction de l'angle vertical

Annexe 4

Test 2, Angles horizontaux :

Angles	a	Réel	Écart
10	10 0,171		0
	0,172	0,171	0,001
	0,17	0,171	0,001
	0,171	0,171	0
	0,172	0,171	0,001
	0,17	0,171	0,001
	0,172	0,171	0,001
	0,17	0,171	0,001
	0,17	0,171	0,001
	0,171	0,171	0
170	0,17	0,171	0,001

Tableau de l'évolution de la mesure « a » en fonction de l'angle horizontal

Angles	b	Réel	Écart
10	0,485	0,48	0,005
	0,483	0,48	0,003
	0,484	0,48	0,004
	0,482	0,48	0,002
	0,481	0,48	0,001
	0,48	0,48	0
	0,482	0,48	0,002
	0,48	0,48	0
	0,483	0,48	0,003
	0,485	0,48	0,005
170	0,482	0,48	0,002

Tableau de l'évolution de la mesure « b » en fonction de l'angle horizontal

Angles	С	Réel	Écart
10	0,514	0,511	0,003
	0,512	0,511	0,001
	0,512	0,511	0,001
	0,511	0,511	0
	0,511	0,511	0
	0,509	0,511	0,002
	0,513	0,511	0,002
	0,508	0,511	0,003
	0,511	0,511	0
	0,512	0,511	0,001
170	0,509	0,511	0,002

Tableau de l'évolution de la mesure « c » en fonction de l'angle horizontal

Angles	Ecarts moyen
10	0,002666666666666
	0,001666666666666
	0,002
	0,000666666666666
	0,000666666666666
	0,001
	0,001666666666666
	0,001333333333333
	0,001333333333333
	0,002
170	0,001666666666666

Tableau des écarts moyens sur les mesures de « a », « b » et « c » en fonction de l'angle vertical

Annexe 5

Test 4, diamètre des arbres :

décamètre : périmètre, cm	décamètre : diamètre, cm	1 photo	2 photos	déca- 1photo	déca- 2photo	1 photo-2 photo
90	28,65	28,1	27,4	0,55	1,25	0,70
110	35,02	33,7	33,4	1,32	1,6	0,30
103	32,79	33,8	33,7	-1,01	-0,91	0,01
126	40,11	39,4	39,2	0,71	0,91	0,20
154	49,02	50,1	50,6	-1,08	-1,58	-0,5
108	34,38	32,3	32,9	2,08	1,48	-0,60
130	41,38	40	38,9	1,38	2,48	1,1
229	72,90	67,9	68,3	5,00	4,60	-0,40
257	81,81	77,77	78,2	4,047	3,61	-0,43
181	57,62	55,1	56	2,52	1,62	-0,89

Tableau des mesures et comparaison des diamètres des arbres au décamètre, en single-shot et en multi-shot

Mesures comparées	déca-1photo	déca-2photo	1 photo-2 photo
Moyennes (cm)	1,55	1,51	-0,043

Tableau des différences moyennes entre les mesures (cm)

Annexe 6

Commandes MicMac pour le processus de calibration :

Etape 1, Tapioca : Cette première étape permet de lire chaque photo et d'y détecter des points repérables. Elle permet aussi d'associer ces points entre les différentes photos et de créer des points homologues. Si la commande utilisée est un *Tapioca All* alors le logiciel effectuera cette association de points repérables sur toutes les images, cependant cette méthode étant très chronophage la commande *Tapioca Line* peut être préférable suivant l'acquisition. Elle associera alors les points repérables seulement entre un nombre d'images prédéfini. Les images peuvent aussi être sous-échantillonnées si trop volumineuses afin d'accélérer le traitement.

Exemple de commande : mm3d Tapioca Line .* jpg 1500 4

Etape facultative, SEL : Cette commande permet d'afficher les points homologues entre 2 images choisies.

Etape 2, Schnaps : Les points homologues étant très nombreux, réduire leur nombre pourrait aussi permettre un gain de temps. La commande *Schnaps* est alors optimale pour cela. Dans un premier temps, elle permet de réduire le nombre de points homologues entre les différentes images en supprimant les points avec une fréquence d'apparition moins importante et en privilégiant, à l'inverse, les points visibles sur le plus d'images possible. Cette commande permet aussi d'obtenir des points homologues uniformément répartis entre chaque couple d'images.

Exemple de commande : mm3d Schnaps .*.jpg

Etape 3, Tapas : La commande Tapas permet d'obtenir deux choses: la calibration interne de l'appareil ainsi que l'orientation relative des photos les unes par rapport aux autres. Pour cela, nous lui fournissons si possible un modèle de calibration prédéfini dans le logiciel afin qu'il ait un modèle de distorsion pour corriger les photos. De plus, la commande *SH* permet de choisir le lot de photos à prendre pour effectuer le calcul.

Exemple de commande : mm3d Tapas RadialStd .*.jpg Out=Ori-RadialStd SH=_mini

Annexe 7

Commandes MicMac pour le processus de la dérive :

Etape 4, SaisieAppuisInitQT: Cette commande permet de sélectionner les ciblettes, donc les points d'appui, sur chaque photo afin d'obtenir leurs coordonnées images.

Exemple de commande : *mm3d SaisieAppuisInitQT*.*jpg Calib Appuis.txt Mesures.xml

Etape 5, Bascule : Une fois ces étapes effectuées nous pouvons réaliser une « bascule » qui permet de réunir les informations obtenues grâce à la topométrie avec les données de la photogrammétrie. Nous obtenons alors la mise en place dans un référentiel absolue. Cela ajoute notamment une dimension métrique au jeu de données.

Exemple de commande : *mm3d GCPBascule .*.jpg Calib Absolu Mesures-S2D.xml Coordonnees.xml*

Etape 6, SaisieAppuisPredicQT : Cette commande permet de sélectionner les autres ciblettes sur chaque photo afin d'obtenir leurs coordonnées dans le référentiel global déterminé via la bascule.

Exemple de commande : *mm3d SaisieAppuisPredicQT .*JPG Ori-FraserBasic_26_l2 Appuis.txt Mesures.xml*

Etape 7, Bascule : Cette commande est là même que précédemment mais est faite à partir du deuxième *SaisieAppuis*.

Etape optionnelle, Campari : Cette commande est la dernière compensation permettant de ré-estimer l'orientation des images. A l'inverse de *GCPBascule, Campari* ne garde pas forcément un bloc rigide. De plus, grâce au paramètre *BlocGlob*, nous pouvons utiliser le fichier en sortie du *Blinis* afin de préciser la base avec certaines précisions. Cette commande peut aussi ré-estimer la calibration des caméras en utilisant le paramètre *AllFree*.

Etape 8, GCPCtrl : Cette commande permet d'obtenir les erreurs entre les positions des points d'appuis et de contrôle réelles et celles calculées à partir de la bascule. Exemple de commande : *mm3d GCPCtrl .*JPG Ori-Bascule/TsPoints.xml Mesures-S2D.xml*

Etape optionnelle, Apericloud : Cette commande permet d'obtenir un nuage de points calculé à partir de la bascule.

Exemple de commande : mm3d AperiCloud .*JPG Ori-Out

Annexe 8 Script déchargement Leica :



Annexe 9

Script déchargement Mi Sphere Camera des fichiers JPG vers des fichiers JPG

```
import imageio
import math as m
import os
recupère les images .JPG, les coupes en deux et les transforme en .TIF
dir_path = os.getcwd()
os.chdir(dir_path)
try:
    os.mkdir("JPG/")
except:
    Dass
for file in os.listdir(dir_path):
    if file.endswith(".JPG"):
         raw = imageio.imread(file)
         #rgb = raw.postprocess(output_bps=8, no_auto_scale=True, no_auto_bright=True)
         im1 = raw[:,:m.floor(raw.shape[1]/2),:]
         im2 = raw[:,m.floor(raw.shape[1]/2):,:]
         imageio.imsave("JPG/"+file.replace(".JPG","_1.JPG"), im1, quality = 100)
imageio.imsave("JPG/"+file.replace(".JPG","_2.JPG"), im2, quality = 100)
```

Script déchargement Mi Sphere Camera des fichiers DNG vers des fichiers JPG

```
import imageio
import rawpy as rp
import math as m
import os
"" Récupère des images sous format .DNG, les coupe en deux et les transforme en .TIF
....
dir_path = os.getcwd()
os.chdir(dir_path)
try:
    os.mkdir("TIFF/")
except:
    pass
for file in os.listdir(dir_path):
    if file.endswith(".DNG"):
         with rp.imread(file) as raw:
              rgb = raw.postprocess(output_bps=8, no_auto_scale=True, no_auto_bright=True)
              im1 = rgb[:,:m.floor(rgb.shape[1]/2),:]
              im2 = rgb[:,m.floor(rgb.shape[1]/2):,:]
              imageio.imsave("TIFF/"+file.replace(".DNG","_1.JPG"), im1, quality = 100)
imageio.imsave("TIFF/"+file.replace(".DNG","_2.JPG"), im2, quality = 100)
```

Annexe 10

Script ajoutant les bordures aux images issues de la Mi Sphere Camera :

```
from PIL import Image
import os
dir_path = os.getcwd()
os.chdir(dir_path)
fonction: Ajoute une bordure noire aux images du la mi-sphère camera.
try:
    os.mkdir("PADD/")
except:
    pass
for file in os.listdir(dir path):
    if file.endswith(".JPG"):
        im = Image.open(file)
        im_size = im.size
        Add_Padd = 1728
        new_im = Image.new('RGB', (im_size[0]+2*Add_Padd,im_size[1]+2*Add_Padd))
        new_im.paste(im, (Add_Padd,Add_Padd))
        new_im.save("PADD/" + file.split(".")[0] + ".PNG", "PNG")
```

VII. Bibliographie

L. Barazzetti, M. Previtali, F. Roncoroni. (2018). "Can we use low-cost 360 degree cameras to create accurate 3D models ? ". Disponible sur : <

https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2/69/2018/isprsarchives-XLII-2-69-2018.pdf >

(consulté le 8 juillet 2019)

A.Girard, L.Gonnetan, T.Harling, M.Margolle, F.Thill. (2017). "Qualification de la caméra stereo ZED". Disponible sur : < http://Dias.ensg.eu/Terrain/Data/ZED/ING16 ProjetForca ZED rapport.pdf > (consulté le 8 juillet 2019)

OpenCV, "Camera Calibration and 3D Reconstruction". Disponible sur : < https://docs.poencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction https://docs.poencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction https://docs.poencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction https://docs.poencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction (consulté le 11 juillet 2019)

Leica BLK3D : <u>https://lasers.leica-geosystems.com/eu/fr/blk3d</u>

MicMac : https://micmac.ensg.eu

Agisoft Metashape : <u>https://www.agisoft.com</u>

Remerciements

Nous tenions à remercier sincèrement notre professeur encadrant Antoine Pinte pour le temps qu'il nous a accordé ainsi que pour ses précieux conseils tout au long de ces semaines de projet.

Nous tenions également à remercier le personnel de la mairie de Pierrerue pour nous avoir prêté avec autant de bienveillance la clé de leur chapelle.

De plus, nous remercions fortement les géomètres de premières années pour nous avoir fourni les données topométriques de la chapelle de Pierrerue.

Un grand merci aux développeurs *MicMac* pour avoir gracieusement essayer de traiter nos données et de nous avoir généreusement fournis un script devant nous aider à avancer.

De plus, un grand merci à nos chers camarades du groupe P2, traitant le parangonnage des logiciels photogrammétriques, sans qui nous n'aurions pu avancer dans le traitement des données de *Xiaomi*.

Enfin, une attention toute particulière pour *Leica* qui nous a prêté en toute confiance leur appareil. En espérant que les traitements que nous avons effectué leurs conviendront.

Petite dédicace à Pageot qui apporte la bonne humeur tout au long de ces semaines de stage.