

Pré-estimation et analyse de la précision pour la cartographie par drone

■ Laurent Valentin JOSPIN

L'arrivée sur le marché de drones à usages civil à la fois légers et à bon marché a révolutionné les méthodes de cartographie aérienne, tant pour la télédétection que pour la topométrie. Un géomètre peut désormais employer un drone pour facilement acquérir un ensemble d'images aériennes d'un objet ou d'une zone et obtenir un modèle en trois dimensions en utilisant des logiciels commerciaux dédiés. Néanmoins, il est difficile de prévoir exactement la précision finale de ces modèles avant de réaliser le vol et le post-traitement : si les résultats ne sont pas satisfaisants, tout le processus doit être repris à zéro. Dans ce projet de master, un logiciel a été développé pour simuler facilement un relevé aérien en prenant en compte la géométrie du terrain, la couverture GPS et les points de calage présents au sol. Cet outil se base sur les dernières avancées dans la résolution d'ajustement de problèmes photogrammétriques, notamment des solveurs dédiés, les algèbres de Lie et, une fois intégré dans les logiciels de gestion de missions, permettra de réduire le temps et le coût de la planification des relevés par drones.

MOTS-CLÉS

Drones, cartographie aérienne, photogrammétrie, vision par ordinateur, optimisation, optimisation sur les graphes, groupes de Lie, Algèbre de Lie

Applications

À l'heure où les drones permettent une démocratisation des prises de vues aériennes, il devient de plus en plus rapide de créer des modèles 3D photogrammétriques. Cependant, la précision de ces modèles peut être en deçà des attentes, due souvent à un mauvais choix de matériel, de plan de vol, ou de placement de points de calage. Une bonne planification permettra de faire "bien du premier coup" plutôt que de devoir retourner sur le terrain pour effectuer de nouvelles photos ou ajouter des points de calage, ce qui augmenterait le coût du relevé.

Plus que le simple contrôle, il deviendra possible d'optimiser automatiquement les plans de vol pour maximiser la précision, le temps de vol ou le temps de post-traitement qui peuvent être non négligeables. Ce genre de logiciel pourra aussi être utile dans le choix du matériel, optimisant ainsi la qualité des mesures tout en limitant les coûts. Une fois intégré dans l'interface d'un gestionnaire de mission, il deviendra facile pour l'opérateur d'explorer interactivement les effets de ses choix sur la précision attendue.

Le logiciel se limite pour l'instant à la prédiction de précision de réseaux photogrammétriques purs. Cependant, le logiciel a été développé de manière à être facilement adapté pour traiter d'autres types de mesures, tels que des mesures LIDAR. La technologie du LIDAR aéroportée par drone est en effet très prometteuse, grâce à la miniaturisation des capteurs, et le développement de chaînes de traitement multi-capteurs [Leroux 2017].

Introduction

La photogrammétrie est l'ensemble des techniques de mesures basées sur des prises de vues photographiques. Il est devenu courant de croiser ces techniques appliquées à l'archéologie, la muséologie, l'infographie et bien d'autres domaines pour digitaliser des objets en 3 dimensions.

Dans le domaine de la géomatique, la photogrammétrie permet la création de nombreux produits, orthophotos, modèles de terrains... Malheureusement il n'est pas toujours aisé de prédire la précision de ces produits finaux, les problèmes d'optimisation impliqués dans les étapes intermédiaires étant souvent très grands et pas toujours simples à appréhender, surtout avant de disposer de toutes les mesures. Cette difficulté peut limiter l'utilisation de la photogrammétrie pour certains travaux, la reléguant

au rang de complément pour des méthodes plus classiques.

Mais si un humain ne peut pas analyser seul ce genre de problèmes, l'ordinateur le peut. Du moins, il peut donner une estimation a priori moyennant quelques hypothèses sur les mesures qui ne seront disponibles qu'après le vol (par exemple la disposition des points homologues ou leur distribution).

Le but du projet de master présenté dans cet article est d'implémenter un logiciel capable de recevoir en entrée les données du plan de vol et du terrain, ainsi que certaines informations pour formuler les hypothèses nécessaires. En sortie, ce logiciel devra fournir une estimation réaliste de la précision, notamment du positionnement des points homologues et des positions et orientations de la plate-forme aérienne lors de la prise de vue.

Approche proposée

L'utilisateur dispose en général d'une bonne idée du vol effectif de son drone (ou avion), qui doit correspondre (à quelques perturbations près) au plan de vol prévu. Il a aussi une idée approximative du terrain, de nombreux modèles numériques de terrain ou d'élévation étant disponibles en tout point de la surface de la terre [Nasa and METI, 2011]. Selon les pays l'utilisateur a aussi une assez bonne idée de la nature de l'utilisation du sol et de la présence de plans d'eau. Enfin l'utilisateur a une idée approximative des points de calage qu'il va relever sur le terrain et/ou des technologies de positionnement qu'il va embarquer sur sa plate-forme aérienne.

L'idée est d'utiliser toutes ces données en entrée pour simuler la forme probable de l'ajustement par faisceaux, notamment en simulant la distribution des points homologues au sol à partir de la nature du sol.

Dans les applications, les observations sont la position de points de calage et de points homologues dans les images. Les paramètres sont les poses du drone au moment de la prise de vue et la positions des points au sol. On dispose aussi d'une estimation de la position des points de calage et éventuellement de mesures GPS/INS de la pose du drone. Tous ces paramètres sont reliés aux observations par les équations de colinéarité :

$$\begin{pmatrix} x_h \\ y_h \\ 1 \end{pmatrix} = \frac{1}{\lambda} \begin{pmatrix} c & 0 & pp_x \\ 0 & c & pp_y \\ 0 & 0 & 1 \end{pmatrix} R_m^c (x_G - x_D)$$

Où x_G représente la position du point au sol, x_D la position de la caméra du drone, R la matrice de rotation entre le repère local de la carte et le repère de la caméra, pp est le point principal, généralement au centre de l'image, c est la longueur focale et λ est le facteur d'échelle.

On ajoute en général comme paramètres les coefficients de distorsion, k pour les coefficients radiaux et p pour les coefficients tangentiels, avec r , la distance du point au point principal de l'image. On obtient ainsi nos observations (entachées d'erreurs, dont la variance dépend des algorithmes qui vont chercher des points homologues dans les images) :

$$\begin{aligned} \ell_{i,x}^h + e_{i,x} &= x_h(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2x_h) + 2p_2 x_h y_h \\ \ell_{i,y}^h + e_{i,y} &= y_h(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_2(r^2 + 2y_h) + 2p_1 x_h y_h \end{aligned}$$

La solution exacte du problème simulé est connue. Admettons à présent qu'on introduise sur les paramètres x une petite perturbation Δx . On peut retrouver la solution initiale en linéarisant le problème et en résolvant le système linéaire associé par moindres carrés. L'équation, très connue, est :

$$A^T P A \cdot \Delta x = A^T P \cdot e$$

Où A est la matrice jacobienne du système linéarisé, P est une matrice de poids qui est égale à l'inverse de la matrice de covariance des observations (dont on a une bonne idée a priori), Δx est la perturbation sur les paramètres, soit les points au sol et les positions et orientations de la plate-forme lors des prises de vues (connue, mais c'est sa matrice de covariance qui nous intéresse) et e est la perturbation sur les observations (mesures de positions des points homologues dans les images, virtuelles ici).

En appliquant la règle de propagation de variance on peut calculer la matrice de covariance assez facilement :

$$\Sigma_{xx} = (A^T P A)^{-1}$$

Dans le cas non linéaire, cette matrice dépend du point autour duquel le problème est linéarisé. Dans la pratique nous verrons que la distribution exacte des points homologues n'a pas beaucoup d'influence sur la covariance des paramètres, et donc la précision finale des nuages de points obtenus, ce qui nous permettra de simuler ces points.

Notons que même si on repose en général sur des algorithmes relativement plus complexes pour converger vers la solution, le fait qu'on s'intéresse aux perturbations autour de cette dernière conduit à ce que l'analyse basée sur les moindres carrés pondérés ordinaires reste valide. Par contre, cela ne nous informe pas sur la capacité de la méthode choisie de détecter et résister aux fautes de mesures.

Le problème (et pas des moindres), il s'avère que les matrices en jeu ici sont très grandes, de l'ordre de plusieurs milliers, voir dizaines de milliers de lignes. À ce niveau-là, la plupart des ordinateurs modernes commencent à ne plus pouvoir réaliser l'inversion des dites matrices en temps réel. Néanmoins nous ne nous intéressons en fait qu'à quelques blocs proches de la diagonale, qui représentent les matrices de covariances des triplets de coordonnées de nos points au sol. Par chance, il existe des algorithmes capables d'extraire uniquement ces coefficients avec une complexité asymptotique moindre [Kaess and Dellaert, 2009].

Pour faciliter la visualisation de ces milliers de petites matrices de covariance, un algorithme a été développé pour interpoler des cartes représentant différentes mesures telles que les écarts-types planimétriques, altimétriques, spatiaux ou la LERM (*log-Euclidean Riemannian metric*) [Cherian et al., 2011], mesure de la "distance" entre

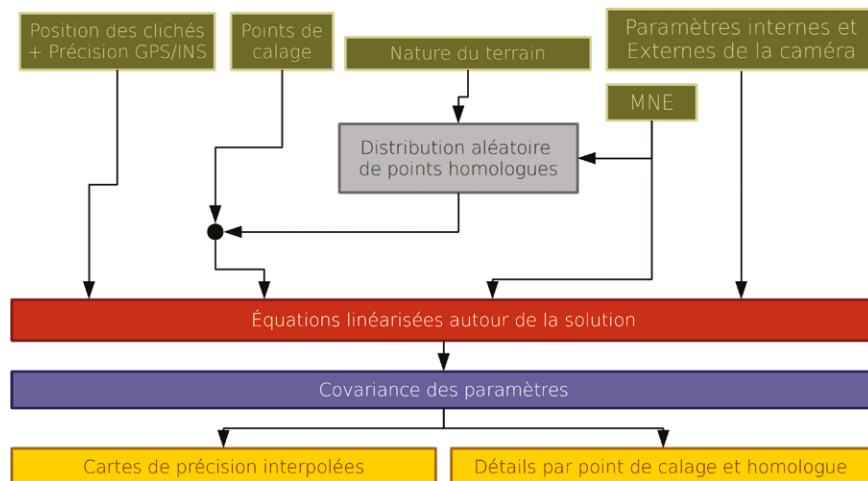


Figure 1. Flux dans le programme, depuis les entrées, en passant par la simulation et jusqu'aux sorties finales pour l'utilisateur.

deux matrices de covariances basée notamment sur les groupes et algèbres de Lie [Alexander Kirillov, 2008].

Quelques notes sur l'implémentation

Sans entrer dans les détails, qui intéresseront surtout les informaticiens et sont disponibles dans la thèse [Jospin, 2017], le système qui a été développé se base sur g2o (*general graph optimisation*) [Giorgio Grisetti, 2012], un ensemble de bibliothèques utilisées en robotique, vision par ordinateur et photogrammétrie pour résoudre des problèmes d'optimisation sur des fonctions pouvant être représentées comme des graphes entre les paramètres. L'idée importante derrière ce concept est que si on considère un graphe dont les côtés représentent les observations, qui relient les coins, figurant les paramètres, dont elles dépendent, alors la structure de ce graphe nous renseigne sur la structure des matrices impliquées dans les problèmes d'optimisation associés. Cette structure est particulièrement adaptée aux problèmes de compensation par faisceaux, qui sont représentés par des graphes bipartis (*figure 2*). On peut ainsi facilement isoler 4 blocs, un (petit) bloc représentant les orientations du drone, un bloc (le plus important) pour les points au sol et deux blocs rectangulaires sur les côtés qui synthétisent les relations entre les points au sol et en l'air.

Afin d'utiliser des techniques employées dans les programmes de photogrammétrie modernes, le modèle d'optimisation mis en place utilise les groupes et les algèbres de Lie pour traiter les jeux de paramètres non-euclidiens (notamment l'orientation du drone). Une introduction complète sur ces objets mathématiques est donnée dans [Alexander Kirillov, 2008]. Ce qui est important de noter ici, c'est qu'il est possible d'associer à chaque groupe de Lie (par exemple les matrices de rotation dans l'espace 3d $SO(3)$, le groupe des orientations rigides en 3d $SE(3)$ ou le groupe des matrices symétriques définies positives de taille 3×3) un algèbre de Lie euclidien qui représente le plan tangent au groupe de Lie autour de son élément neutre.

Pour les groupes de Lie pouvant être représentés par des matrices, il est possible de lier groupe et algèbre de Lie via la fonction exponentielle (de matrice), le log de tout élément du groupe de Lie étant dans l'algèbre de Lie, respectivement l'exponentielle de tout élément de l'algèbre de Lie se trouvant dans le groupe. Il est donc possible de définir par exemple des "matrices de rotation infinitésimale", à partir d'accroissements infinitésimaux dans le groupe de Lie et de définir une règle de composition noté \boxplus telle que :

$$\exp(\Delta \log(x)) * x_n \stackrel{\text{def}}{=} x_n \boxplus \Delta \log(x)$$

Avec x et Δx dans le groupe de Lie, mais en réalité c'est $\log(\Delta x)$ qui est connu, par exemple comme pertur-

bation dans l'algèbre de Lie. Cet opérateur nous permet de redéfinir la dérivée dans nos groupes de Lie :

$$\frac{\partial f(x)}{\partial \log(x)_i} = \lim_{\epsilon_i \rightarrow 0} \frac{f(x \boxplus \epsilon_i) - f(x)}{\epsilon_i}$$

La division étant pour un incrément sur une seule direction de l'algèbre de Lie, qui est euclidien on le rappelle. Plus de détails sur la méthodologie sont donnés dans [Giorgio Grisetti, 2012].

L'avantage de cette technique est d'éviter les problèmes de singularité que peuvent impliquer par exemple, une représentation naïve avec les angles d'Euler, tout en évitant le problème de devoir ajouter des contraintes à notre problème initial si on optimisait par exemple directement les 9 paramètres d'une matrice de rotation ou les 4 paramètres d'un quaternion. C'est aussi sur cette base qu'on peut calculer la LERM (*log-Euclidean Riemannian metric*) [Cherian et al., 2011], qui n'est autre que la distance basée sur la norme de Frobenius dans l'algèbre de Lie associée aux matrices symétriques définies positives (soit les matrices de covariances non dégénérées).

$$LERM(M_1, M_2) = \|\log(M_1) - \log(M_2)\|_F$$

La LERM mesure à quel point, le produit de la première matrice par l'inverse de l'autre est proche de l'identité. Pour en faciliter la compréhension, on peut utiliser la LERM relative, qui mesure à quel point l'erreur est grande par rapport à la "taille" de la première matrice :

$$\frac{\|\log(M_1) - \log(M_2)\|_F}{\|\log(M_1)\|_F}$$

L'avantage de la LERM, est qu'elle permet aisément de définir un critère de comparaison entre les matrices de covariance, ce qui permet de mesurer à quel point la prédiction du logiciel correspond à un certain objectif. La LERM étant en échelle logarithmique, elle se prête aussi assez bien à la mesure des précisions, que l'on donne généralement sous forme d'ordre de grandeur (mm, cm, m...).

Validations sur le terrain

Le programme a été d'abord soigneusement testé avec plusieurs jeux de données synthétiques, puis mis en pratique sur une série de vols déjà

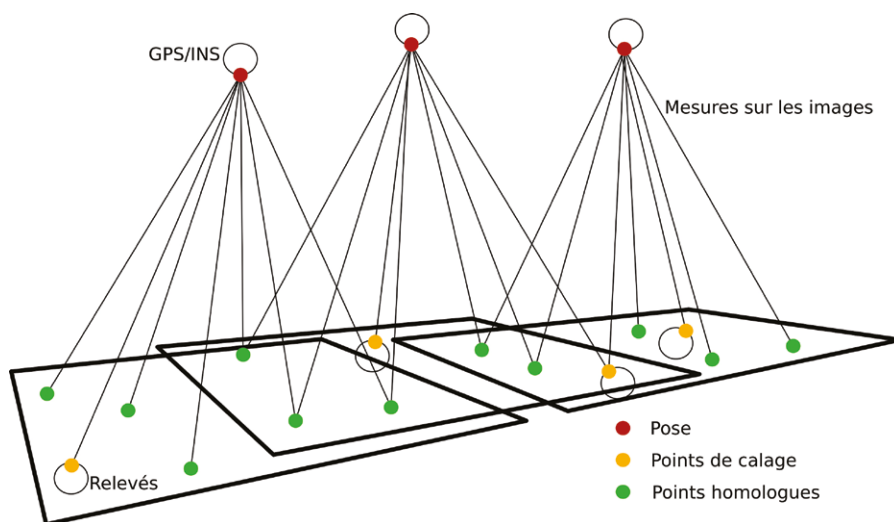


Figure 2. Un problème de compensation par faisceaux, vu comme un graphe reliant les différents paramètres par des observations.



Figure 3. Vol 1, juste un aller et retour



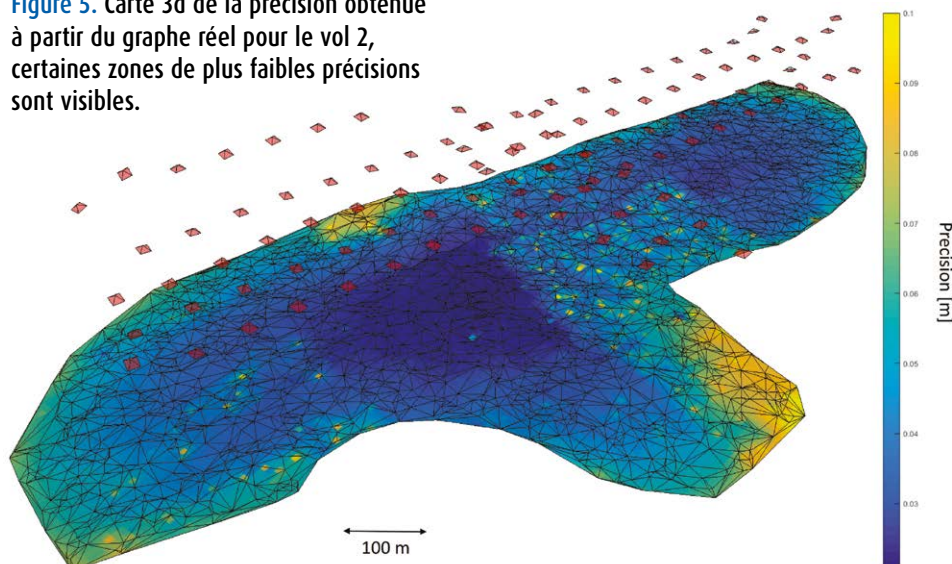
Figure 4. Vol 2, un peu plus complexe



réalisés et post-traités. Ces vols ont été réalisés en 2015 à Vuflens-la-Ville, en Suisse, dans le canton de Vaud [Rehak, 2017]. L'un des deux représente un vol simple avec seulement trois points de calage (figure 3), le second, un peu plus conséquent, a été réalisé au même endroit mais avec plus de points de calage et un recouvrement plus important (figure 4). Pour le premier vol, il est évident que la répartition choisie des points de calage est hautement défavorable, et qu'aucun photogrammètre n'aurait choisi une telle configuration. Cependant, le but ici est d'observer le comportement de notre logiciel face à une telle configuration. Bien que très simples, et réalisés sur terrain relativement plat (où la visibilité des points au sol est rarement un problème), ces vols sont très intéressants. Premièrement ils n'ont pas été réalisés au même moment, les conditions de visibilité étant légèrement différentes, et deuxièmement ils vont permettre de mettre en valeur un des problèmes récurrents de notre approche et comment le résoudre. Cet article va se concentrer sur la comparaison entre la prédiction et la référence obtenue à partir du graphe tel que reconnu et optimisé par le programme Pix4D, utilisé pour l'ajustement.

Les cartes principales ont été réalisées avec des simulations ayant à peu près le même nombre de points homologues que dans le cas réel. Mais des simulations ont aussi été réalisées pour différents nombres de points homologues, vu qu'il est difficile de prédire la densité de points homologues qu'on obtiendra effectivement.

Figure 5. Carte 3d de la précision obtenue à partir du graphe réel pour le vol 2, certaines zones de plus faibles précisions sont visibles.



LERM relative pour les vols 1 & 2

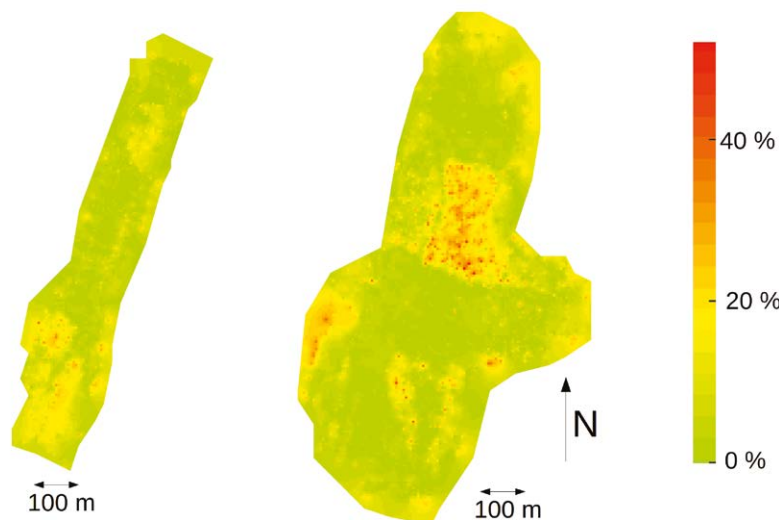


Figure 6. Carte des LERM relatives entre matrices de covariances réelles et simulées pour les vols 1 et 2. Les LERM sont normés par rapport à la norme du log de la matrice de covariance réelle.

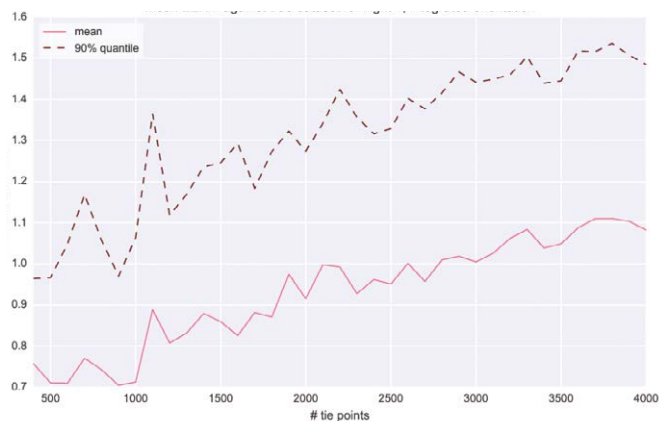


Figure 7. LERm moyenne et quantile 90 % par rapport au nombre de points homologues dans la simulation pour le vol 1.

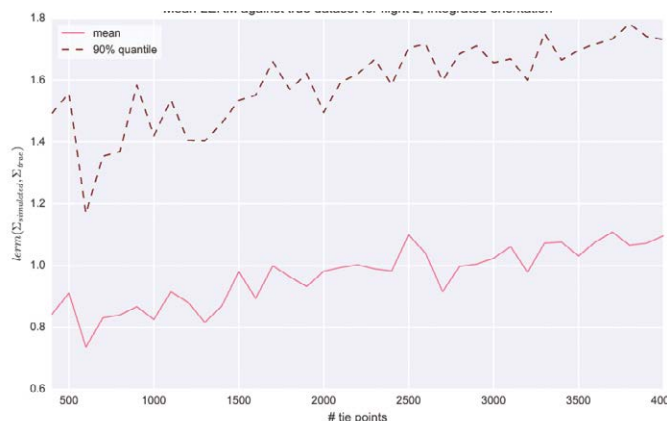


Figure 8. LERm moyenne et quantile 90 % par rapport au nombre de points homologues dans la simulation pour le vol 2.

On remarque immédiatement que certains points avec de grandes différences entre la prédiction et la réalité apparaissent isolément (ce qui n'est pas trop surprenant, il s'agit d'erreurs des algorithmes de vision par ordinateur, que l'on peut retirer de manière automatique), mais que des différences assez systématiques apparaissent dans certaines zones, semblant suivre la forme des champs, et sont corrélées avec des zones moins précises dans la simulation finale (figure 5).

Entre les deux vols, les champs qui posent problème ont changé, ce qui prouve que la nature de l'utilisation du sol ne suffit pas, seule, à prédire efficacement le nombre et la disposition des points homologues.

On remarque en revanche, en faisant varier le nombre de points homologues dans la simulation, le nombre de points dans la réalité, étant fixé, ne change pas, on obtient des résultats qui restent globalement cohérents (Figures 7 et 8). Au vu de la structure du graphe du problème, à partir d'une densité suffisante de points, qui est très vite atteinte, c'est plus le nombre d'images avec les mêmes points homologues que le nombre de points homologues, ou leurs dispositions précises, qui a de l'influence sur la précision. Cela permet donc au logiciel de fournir une prédiction relativement bonne de la précision tout en employant un nuage de points homologues simulé d'une taille potentiellement différente que la taille effective du nuage final et ce, sans faire une erreur trop importante.

Limite de la méthode

Il n'en reste pas moins que, au vu des cartes présentées dans la figure 6, le logiciel donne des résultats trop optimistes dans certaines zones qui semblent soudainement moins précises que la prédiction. Nous avons vu que la densité de points homologues n'a pas énormément d'importance.

Le paramètre qui semble avoir le plus d'importance, sachant que c'est bien le nombre de vues sur un même point homologue qui devient vite le paramètre déterminant pour la précision, est bien la capacité du logiciel utilisé pour le post-traitement à trouver des points homologues dans les différentes images. La figure 9 nous montre que, dans le vol tel que effectivement réalisé, les zones problématiques n'ont en effet pas été correctement connectées par les

algorithmes de vision par ordinateur et les points homologues ne lient pas autant d'images qu'ailleurs sur la carte. En pratique, si l'on intègre ces paramètres dans le modèle (ce qu'il est possible de faire, le logiciel acceptant une carte de nombre de vues moyenne par points et limitant la connectivité du graphe en conséquence) on améliore grandement l'adéquation entre le modèle et la réalité. Pas parfaitement, car le logiciel ne prend pas encore en compte le niveau de corrélation entre les vues mais préfère choisir parfaitement au hasard les caméras qui peuvent voir ou ne pas voir un point homologue donné. Cela imposerait d'introduire encore plus de données mais il serait aussi possible de prendre en compte le fait que si une image voit un point donné, alors elle a plus de chance d'en voir certains plutôt que d'autres.

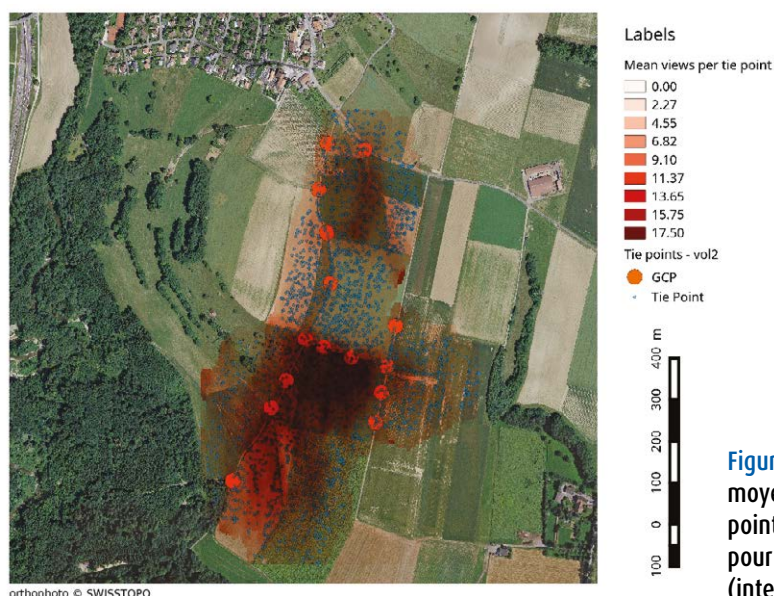


Figure 9. Nombre moyen de vues par point homologue pour le vol 2 (interpolation).



Ainsi, il s'avère qu'un paramètre difficile à prendre en compte peut passablement modifier les résultats. Cela prouve en premier lieu que les ordinateurs ne sont pas encore totalement prêts à remplacer les professionnels humains. Mais ces mêmes ordinateurs peuvent être une aide appréciable, si ces mêmes hommes et femmes d'expérience s'en servent judicieusement. Il reste en effet possible à l'utilisateur d'entrer les zones à problèmes à la main, s'il sait par exemple que le type et l'état des cultures dans un champ posera vraisemblablement quelques complications au moment du vol, le logiciel pourra ensuite les prendre en compte. Un ingénieur prudent pourra aussi se dire *"better safe than sorry"*, et demander à son logiciel de faire une hypothèse pessimiste partout, ou d'évaluer l'impact de différents *patchworks* aléatoires de zones à problèmes.

Développements futurs et conclusion

En l'état actuel, le logiciel qui a été développé est capable de fournir une prédiction cohérente avec les incertitudes obtenues en post-traitement avec d'autres *softwares*, pour autant qu'on lui donne le maximum d'informations (la forme complète du graphe et tous les écarts-types a priori). Une fois privé des informations indisponibles avant le vol il donne de bons résultats, mais est incapable de deviner seul la possible émergence de zones où les algorithmes de vision par ordinateur n'arriveront pas, ou mal, à lier les images (en exceptant les plans d'eau, que l'on peut exclure des zones où des points homologues seront trouvés). Un possible développement futur est de concevoir un modèle empirique basé sur un grand nombre de vols pour prédire la probabilité de ce genre de problème en fonction de la nature du sol, et ainsi avertir l'utilisateur et/ou être capable de simuler une distribution probable de ces zones à problèmes. Un développement important qui reste à réaliser pour transposer cet outil du monde de la recherche au monde professionnel est le développement d'une interface capable d'exploiter cet outil tout en orientant efficacement l'utilisateur (déplacement des lignes de vols,

ajouts de lignes transversales, choix du recouvrement entre les images) afin de tirer le maximum des interactions homme/machine. ●

Remerciements

Il semble important de remercier en tout premier lieu le laboratoire de topométrie de l'EPFL, qui a proposé et organisé ce projet de master. Plus particulièrement Jan Skaloud, Davide Cucci et Emmanuel Clédât qui ont supervisé ce projet. Bonne chance aussi à Emmanuel Clédât, dont la thèse de doctorat consiste à continuer le développement et surtout les applications de cet outil. Il me semble encore important de remercier ici les nombreux professeurs, chargés de cours et assistants qui ont jalonné mon parcours à l'EPFL et m'ont permis d'apprendre en 5 ans assez de choses dans des sujets variés pour finalement réaliser ce projet de master.

Contact

Laurent Valentin JOSPIN
laurent.jospin@alumni.epfl.ch

Bibliographie

- [Alexander Kirillov, 2008] Alexander Kirillov, J. (2008). *An Introduction to Lie Groups and Lie Algebras*. Stony Brook.
- [Cherian et al., 2011] Cherian, A., Sra, S., Banerjee, A., and Papanikolopoulos, N. (2011). *Efficient similarity search for covariance matrices via the jensen-bregman logdet divergence*. In 2011 International Conference on Computer Vision, pages 2399–2406.
- [Förstner and Wrobel, 2016] Förstner, W. and Wrobel, B. (2016). *Photogrammetric Computer Vision: Geometry, Orientation and Reconstruction*. Geometry and Computing. Springer International Publishing.
- [Förstner and Moonen, 1999] Förstner, W. and Moonen, B. (1999). *A metric for covariance matrices*.
- [Gandor, 2015] Gandor, F. (2015). *Mission planner for micro unmanned aerial vehicles*. Master's thesis, epfl.
- [Giorgio Grisetti, 2012] Giorgio Grisetti, Rainer Kümmerle, H. S. K. K. (2012). *g 2 o: A general framework for (hyper) graph optimization*. Uni-freiburg.
- [Jospin, 2017] Jospin, L. (2017) *Accuracy Prediction in Aerial Mapping*. Master's thesis, epfl.
- [Kaess and Dellaert, 2009] Kaess, M. and

Dellaert, F. (2009). *Covariance recovery from a square root information matrix for data association*. Robotics and Autonomous Systems, 57:1198–1210.

[Leroux 2017] B. Leroux B., Cali J., Morel L., He H. (2017). *Assessing the reliability and the accuracy of attitude extracted from visual odometry for lidar data georeferencing*. ISPRS Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2/W6.

[Nasa and METI, 2011] Nasa and METI (2011). Aster <https://asterweb.jpl.nasa.gov/gdem.asp>. Accessed: 2017-06-13.

[Pascale, 2016] Pascale, R. (2016). *Quality management in flight planning*. Master's thesis, epfl.

[Rehak, 2017] Rehak, M. (2017). *Integrated sensor orientation on micro aerial vehicles*. PhD thesis, epfl.

[Strasdat, 2012] Strasdat, H. (2012). *Local Accuracy and Global Consistency for Efficient Visual SLAM*. PhD thesis, Imperial College London.

ABSTRACT

Key words: UAV, aerial mapping, photogrammetry, computer vision, optimization, graph optimization, Lie groups, Lie Algebra.

The advent of lightweight and affordable unmanned aerial vehicles is radically revolutionizing the methods of aerial mapping, both in remote sensing and in the surveying industry. A surveyor can now employ easy-to-use drones to automatically acquire close range aerial pictures of an object or a field and obtain a three dimensional model using commercial photogrammetry softwares. However, it is difficult to foresee and assess the accuracy of such models before the actual data acquisition and processing: if the results are not satisfactory, the flight has to be repeated. In this master project a software was developed to simulate an aerial survey and formulate a prediction of the expected quality given the mission plan, considering the geometry of the terrain and the availability of GPS receivers and ground control points. This tool is built on the latest advancements in the solution of photogrammetric adjustment problems, including efficient least-square solvers and Lie algebra, and, if integrated in commercial flight planning softwares, will improve duration and cost efficiency of drone based surveys.