Ajustement de faisceaux multi-caméra intégrant l'estimation de la synchronisation et du rolling-shutter

Thanh-Tin Nguyen Maxime Lhuillier

Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000 Clermont-Ferrand, France
maxime.lhuillier.free.fr

Résumé

Les caméras omnidirectionelles obtenues en fixant rigidement ensemble plusieurs caméras grand public deviennent populaires et sont utilisées pour des applications de vidéos 360 et de réalité virtuelle. Cependant, leur auto-étalonnage n'est pas simple car elles sont composées de plusieurs caméras qui ne sont pas synchronisées précisément et/ou sont rolling shutter. Cet article décrit un nouvel ajustement de faisceaux pour ces multi-caméras qui estime non seulement les paramètres classiques (intrinsèques, distortion radiale, poses, points 3D) mais aussi la synchronisation et le rolling shutter des caméras. On expérimente à partir de vidéos prises par des caméras Gopro ou par une caméra sphérique, montées sur un casque et se déplaçant sur plusieurs centaines de mètres ou des kilomètres, puis on compare les résultats avec la vérité terrain.

Mots Clef

Ajustement de faisceaux, multi-caméra, auto-étalonnage, synchronisation, rolling shutter.

Abstract

Omnidirectional cameras built by fixing together several consumer cameras become popular and are convenient for applications like 360 videos and virtual reality. However their self-calibration is not easy since they are composed of several inaccurately synchronized and/or rolling shutter cameras. This paper describes a new bundle adjustment for these multi-cameras that estimates not only the usual parameters (intrinsic, radial distortion, poses, 3D points) but also the synchronisation and the rolling shutter of the cameras. We experiment using videos taken by GoPro cameras or a spherical camera, mounted on a helmet and moving several hundreds of meters or kilometers, then we compare the results to ground truth.

Keywords

Bundle adjustment, multi-camera, self-calibration, synchronization, rolling shutter.

1 Introduction

Les caméras omnidirectionnelles obtenues en fixant rigidement ensemble plusieurs caméras grand public deviennent populaires grâce à leur prix faible et leur résolution en augmentation, ainsi que leur utilisation dans des applications comme les vidéos 360 et la génération de contenu pour la réalité virtuelle [1, 2]. Cependant, ces multi-caméras ont aussi des inconvénients. La synchronisation entre caméras peut être approximative, par exemple lorsque les vidéos démarrent grâce à un signal envoyé par wifi dans le cas des caméras Gopro. Ensuite un prix faible implique que les caméras sont rolling shutter ou RS, ce qui signifie que des lignes distinctes de pixels d'une même image sont acquises à des instants différents (contrairement au cas global shutter ou GS, pour lequel tous les pixels d'une image ont la même date). Une synchronisation imprécise et le RS complique l'auto-étalonnage d'une multi-caméra se déplaçant dans son environnement, comme si la multi-caméra avait une calibration GS qui varie dans le temps [10].

Cet article décrit un nouvel ajustement de faisceaux (AF) pour ces multi-caméras, qui estime non seulement les paramètres classiques (intrinsèques, distortion radiales, poses, points 3D), mais aussi la synchronisation et le coefficient de rolling shutter. On part d'une calibration initiale avec un modèle de caméra plus simple (central et GS) et une synchronisation précise à l'image près (abrégée en FA, ou frame-accurate) obtenues avec une méthode d'auto-étalonnage précédente [21]. Une synchronisation FA signifie que l'on saute des premières images des vidéos de sorte que les restes des vidéos ont la propriété suivante : les images de même index sont prises au même instant modulo l'inverse du FR (frame rate), i.e. modulo la période d'acquisition. Notre AF estime une synchronisation à précision sous-trame (abrégée en SFA, ou subframe-accurate), i.e. il estime les décalages résiduels en temps entre une caméra de référence et les autres. Il estime aussi le coefficient de RS (le délai de ligne), i.e. le temps écoulé entre deux lignes adjacentes d'une image.

On se restreint ici au cas où les caméras ont les mêmes FR et délai de ligne. Ces hypothèses sont adaptées au cas très fréquent d'une multi-caméra omnidirectionelle (et caméra sphérique) sans direction privilégiée.

2 Travaux précédents

Contrairement à notre AF multi-caméra, les AF multi-caméras précédents n'estiment ni la synchronisation ni le

coefficient de RS. Notamment [16, 24, 17] supposent que les caméras sont synchronisées et GS, et seul [14] traite le cas RS (de coefficient connu) mais utilise d'autres capteurs. Un AF monoculaire estime le coefficient de RS en supposant que les points 3D sont connus sur une mire de calibration [23], ou bien force une valeur donnée de ce coefficient [12, 7]. Dans le contexte de SLAM visuel [13], un AF pour GS est appliqué à une caméra monoculaire RS grâce à une compensation faite avant l'AF : cette méthode corrige au préalable les effets RS sur les suivis de primitives en estimant les vitesses instantanées de la caméra.

Tout AF pour RS a un modèle de mouvement de la (multi-) caméra, qui donne la pose de la caméra à tout instant correspondant à une ligne d'une image, et qui doit avoir un nombre modéré de paramètres à estimer. Dans [12], l'AF estime une pose pour chaque image et les poses entre deux images successives sont des interpolations des poses des deux images, ce qui nécessite une hypothèse sur le mouvement entre images. L'AF dans [7] ajoute des paramètres supplémentaires pour éviter cette hypothèse : il estime non seulement une pose mais aussi des vitesses de translation et rotation à chaque image clef (pas toutes les images). Dans [23], le modèle de mouvement continu en temps est basé sur des B-splines et l'AF optimise les noeuds des splines. Cette méthode choisit le nombre de noeuds et initialise leur distribution le long de la trajectoire. Dans [14], la pose relative entre une pose inter-image et une pose d'image optimisée est fournie par un capteur IMU à haute fréquence. Les approches visuelles pures [13, 12, 7, 23] sont expérimentées sur des trajectoires longues de seulement quelques mètres.

Dans le contexte d'un multi-capteur général, [8] estime simultanément les recalages en temps et espace entre les capteurs. Dans les expériences, le multi-capteur est composé d'une caméra et d'un capteur IMU. La meilleure précision est obtenue grâce à l'utilisation simultanée de toutes les mesures, à la représentation continue en temps (une B-spline) pour les poses IMU, et à une estimation du maximum de vraisemblance des paramètres (décalage en temps, transformation entre capteurs IMU et caméra, les poses IMU, et d'autres). Dans [18], un multi-capteur caméra-inertiel est auto-étalonné (synchronisation, recalage spatial, paramètres intrinsèques) par odométrie visuelle et un AF local. Grâce à un paramétrage du mouvement continu en temps adéquat, cette méthode traite aussi le cas d'une caméra RS et a un meilleur paramétrage des rotations. En effet, elle évite les singularités des paramétrages globaux et minimaux des rotations [8], mais suppose que le temps entre images clefs successives est constant.

Nos contributions sont les suivantes. On propose le premier AF multi-caméra estimant à la fois la synchronisation et le coefficient de RS en plus des paramètres habituels. Comme les poses sont seulement estimées en des images clefs, on traite des séquences plus longues que les AF RS précédents : des trajectoires de quelques centaines de mètres ou kilomètres. On propose aussi un paramétrage global et minimal des rotations et traite le cas de distributions d'images clefs non uniformes dans le temps (contrairement à [18]), comme celles fournies par les méthodes de structure-from-motion standard [19]. Ceci est possible grâce à une hypothèse sur le mouvement de caméra, qui est vérifiée dans la plupart des cas pour une caméra fixée sur un casque ou même sur un véhicle. Cet article est une introduction à l'article (deux fois plus) long [22], qui décrit plus en détails l'ensemble de toute notre méthode d'auto-étalonnage, y compris l'initialisation GS et les calculs d'erreurs de reprojections non explicites, et a plus d'expérimentations. On se focalise ici sur l'AF pour RS dans sa version la plus simple, et on ajoute une nouvelle expérience sur les auto-étalonnages GS et RS en fonction de la vitesse de la caméra. De plus, [20] ne raffine ni les paramètres intrinsèques ni les distortion radiales, contrairement à ici.

3 Méthode proposée

3.1 Initialisation

On suppose d'abord que les vidéos monoculaires sont approximativement synchronisées en supprimant quelques images au début (synchronisation FA), i.e. elles sont synchronisées à l'inverse du FR près. On suppose aussi que le FR est le même partout. Ensuite on définit l'image numéro i de la multi-caméra par une concaténation de sous-images, chacune d'elle étant l'image numéro i d'une caméra monoculaire. Dorénavant, on utilise le mot *image* pour "image de la multi-caméra" et le mot *vidéo* est la séquence définie par toute ces images. Enfin on utilise un structure-from-motion standard basé sur un sous-échantillonnage en image clefs et un AF local [19], suivi d'un AF global. La multi-caméra est auto-étalonnée [21] avec l'approximation GS et la synchronisation FA. On rappelle que les images clefs sont les seules images pour lesquelles les poses sont raffinées par les AFs (c'est utile pour le temps de calcul et la précision).

3.2 Paramétrage du mouvement

Soit $\mathcal{R}$ une fonction $\mathcal{C}^1$ continue qui envoie $\mathbb{R}^k$ sur l'ensemble $SO(3)$ des rotations de $\mathbb{R}^3$. Le choix de $\mathcal{R}$, qui inclue k, est détaillé plus tard en section 3.6 pour plus de clarté (on prendra $k = 3$). On suppose qu'il existe une fonction $\mathcal{C}^3$ continue $M : \mathbb{R} \to \mathbb{R}^3 \times \mathbb{R}^k$ qui paramétrise le mouvement de la multi-caméra. Plus précisement, $M(t)^T = \begin{pmatrix} T_M(t)^T & E_M(t)^T \end{pmatrix}$ avec la translation $T_M(t) \in \mathbb{R}^3$ et la rotation $\mathcal{R}(E_M(t)) \in SO(3)$. Les vecteurs colonnes de $\mathcal{R}(E_M(t))$ et $T_M(t)$ sont les vecteurs du repère multi-caméra au temps t exprimés dans le repère monde.

Grâce à ces notations et hypothèses, on va approcher $M(t)$ en utilisant des valeurs de M en quelques instants $t_0, t_1, \cdots t_n$ tels que $t_0 < t_1 \cdots < t_n$. Les $M(t_i)$ sont les seuls paramètres du mouvement multi-caméra estimés par notre AF. La section 3.3 définit t_i, et la section 3.4 décrit notre approximation de $M(t)$ en utilisant les $M(t_i)$. Notons que l'on n'a pas besoin de connaitre M exactement, on suppose juste qu'elle existe.

3.3 Temps, rolling shutter et synchronisation

Toute image clef est composée de sous-images prises par les caméras monoculaires. Chaque ligne de chaque sous-image est prise à un instant qui lui est propre. La 0-ième ligne de la 0-ième sous-image de la i-ième image clef est prise à l'instant t_i. Donc $t_{i+1} - t_i$ est un multiple de l'inverse du FR. Comme les caméras sont RS, le délai de ligne τ est tel que la y-ième ligne de la 0-ième sous-image dans la i-ième image clef est prise à l'instant $t_i + y\tau$. Soit $\Delta_j \in \mathbb{R}$ le décalage en temps entre les j-ième et 0-ième caméras. Comme on suppose que toutes les caméras ont les mêmes FR et τ, la y-ième ligne de la j-ième sous-image dans la i-ième image clef est prise à l'instant $t_i + \Delta_j + y\tau$.

3.4 Approximations pour le mouvement

On a un développement de Taylor

$$M(t) = M(t_i) + (t - t_i)M'(t_i) + \mathcal{O}(|t - t_i|^2) \qquad (1)$$

et on donne une relation entre la dérivée $M'(t_i)$ et les $M(t_i)$. Soient $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^{k+3}, a > 0, b > 0$ et

$$D(\mathbf{x}, \mathbf{y}, \mathbf{z}, a, b) = \frac{b\mathbf{z}}{a(a+b)} - \frac{a\mathbf{x}}{b(a+b)} + \frac{(a-b)\mathbf{y}}{ab}. \quad (2)$$

Soit $\Delta = \max_i(t_{i+1} - t_i)$ et utilisons la notation abrégée $\mathbf{m}_i = M(t_i)$. Grâce à une combinaison linéaire de développements de Taylor de M en t_i [22], $M'(t_i)$ est égal à

$$D(\mathbf{m}_{i-1}, \mathbf{m}_i, \mathbf{m}_{i+1}, t_{i+1} - t_i, t_i - t_{i-1}) + \mathcal{O}(\Delta^2). \quad (3)$$

Enfin, on approxime $M(t)$ avec les $\mathbf{m}_i$ en négligeant les restes en "$\mathcal{O}$". On calcule $M(t)$ pour la y-ième ligne de la j-ième caméra dans la i-ième image clef avec $t = t_i + \Delta_j + y\tau$ (sec. 3.3) et les Eqs. 1 et 3 : $M(t)$ est égal à

$$\mathbf{m}_i + (t - t_i)D(\mathbf{m}_{i-1}, \mathbf{m}_i, \mathbf{m}_{i+1}, t_{i+1} - t_i, t_i - t_{i-1}) \quad (4)$$

si $0 < i < n$. Si $i = 0$, et de façon similaire si $i = n$, on utilise $M(t) = \mathbf{m}_0 + (t - t_0)(\mathbf{m}_1 - \mathbf{m}_0)/(t_1 - t_0)$.

3.5 Erreur de reprojection

Notre AF minimise la somme des modules au carrés des erreurs de reprojections pour chaque point 2D détecté (inlier). On décrit maintenant le calcul de l'erreur de reprojection pour un point $\mathbf{x} \in \mathbb{R}^3$ dans le repère monde et son observation $\tilde{\mathbf{p}} \in \mathbb{R}^2$ par la j-ième caméra dans la i-ième image clef ($\tilde{\mathbf{p}}$ est un des points de Harris reconstruisant $\mathbf{x}$). On introduit d'abord des notations. Soit $\mathbf{p} \in \mathbb{R}^2$ la projection de $\mathbf{x}$ par la j-ième caméra dans la i-ième image clef. L'erreur de reprojection est $\mathbf{p} - \tilde{\mathbf{p}}$. Soit $(\mathtt{R}_j, \mathbf{t}_j)$ la pose de la j-ième caméra dans le repère multi-caméra. Soit $K_j : \mathbb{R}^3 \setminus \{\mathbf{0}\} \to \mathbb{R}^2$ la fonction de projection de la j-ième caméra. On suppose que $K_j, \mathtt{R}_j, \mathbf{t}_j$ sont constants. Les instants d'acquisition de $\mathbf{p} = (x, y)$ et $\tilde{\mathbf{p}} = (\tilde{x}, \tilde{y})$ sont $t_{\mathbf{p}} = t_i + \Delta_j + y\tau$ et $t_{\tilde{\mathbf{p}}} = t_i + \Delta_j + \tilde{y}\tau$.
Ensuite on détaille la relation entre $\mathbf{p}$ et $\mathbf{x}$. On a $E_M(t_{\mathbf{p}})$ et $T_M(t_{\mathbf{p}})$, i.e. $M(t_{\mathbf{p}})$, qui sont définis par l'Eq. 4 avec l'index i d'image clef et $t = t_{\mathbf{p}}$. Les coordonnées de $\mathbf{x}$ dans le repère multi-caméra sont

$$\mathbf{x}_M = \mathcal{R}(E_M(t_{\mathbf{p}}))^\top(\mathbf{x} - T_M(t_{\mathbf{p}})). \qquad (5)$$

Les coordonnées de $\mathbf{x}$ dans le repère de la j-ième caméra, ainsi que $\mathbf{p}$, sont

$$\mathbf{x}_j = \mathtt{R}_j^\top(\mathbf{x}_M - \mathbf{t}_j) \text{ et } \mathbf{p} = K_j(\mathbf{x}_j). \qquad (6)$$

Enfin on voit que $\mathbf{p}$ a besoin du calcul de $\mathbf{x}_M$, qui à son tour a besoin du calcul de (la coordonnée y de) $\mathbf{p}$. Ce problème est résolu grâce à une approximation [14] : $t_{\mathbf{p}}$ est remplacé par $t_{\tilde{\mathbf{p}}}$ dans l'expression de $\mathbf{x}_M$, i.e. on suppose que la pose multi-caméra est la même aux instants $t_{\tilde{\mathbf{p}}}$ et $t_{\mathbf{p}}$. Nous pensons que c'est acceptable puisque $|t_{\tilde{\mathbf{p}}} - t_{\mathbf{p}}| \leq \tau||\mathbf{p} - \tilde{\mathbf{p}}||_2$ et que l'ordre de grandeur de τ est 10^{-5} s/pixel et $\tilde{\mathbf{p}}$ est un inlier (i.e. $||\mathbf{p} - \tilde{\mathbf{p}}||_2 \leq 4$ pixels).

3.6 Paramétrage des rotations

On décrit ici notre choix de la fonction $\mathcal{R} : \mathbb{R}^k \to SO(3)$. On dit que $\mathbf{x} \in \mathbb{R}^k$ est une singularité si la jacobienne de $\mathcal{R}$ n'est pas de rang 3. Le contexte est le suivant :

(a). on veut un minimum de paramètres k pour $SO(3)$

(b). notre AF est basé sur un paramétrage global $\mathcal{R}$ de $SO(3)$ (non local et non incrémental)

(c). le tangage et le roulis sont faibles pour les mouvements habituels d'une caméra fixée sur un casque

(d). pas d'AF au voisinage des singularités de $\mathcal{R}$ [26].

D'après (a) et (b), $k = 3$ et $\mathcal{R}$ est surjective. Donc $\mathcal{R}$ a des singularités [25]. On veut s'en éloigner le plus possible pour satisfaire (d), et c'est possible grâce à (c).
On examine maintenant les singularités de plusieurs $\mathcal{R}$ candidats, et choisit $\mathcal{R}$ si l'ensemble des $E_M(t)$ pour un mouvement habituel sont distants de toutes ses singularités.

Application exponentielle Avec exp l'exponentielle de matrice et $[\mathbf{x}]_\times$ la matrice telle que $[\mathbf{x}]_\times\mathbf{y} = \mathbf{x} \wedge \mathbf{y}$ et $\{\mathbf{x}, \mathbf{y}\} \subset \mathbb{R}^3$, on a $\mathcal{R}(E_M) = \exp([E_M]_\times)$. Les singularités E_M forment des sphères concentriques de centre 0 et de rayons multiples de 2π [11]. Donc l'angle de rotation $||E_M||$ doit rester inférieur à 2π : la caméra doit donc éviter de tourner pour ne pas se retrouver sur une trajectoire (rectiligne) avec $||E_M||$ constante et égale à 2π.

Euler classique On a $\mathcal{R}(\alpha, \beta, \gamma) = R_z(\gamma)R_y(\beta)R_x(\alpha)$ avec $R_z(\gamma)$, $R_y(\beta)$ et $R_x(\alpha)$ les rotations d'angles γ, β et α autour de $\begin{pmatrix}0 & 0 & 1\end{pmatrix}^\top$, $\begin{pmatrix}0 & 1 & 0\end{pmatrix}^\top$ et $\begin{pmatrix}1 & 0 & 0\end{pmatrix}^\top$ respectivement. Les singularités $\begin{pmatrix}\alpha & \beta & \gamma\end{pmatrix}^\top$ forment des plans parallèles et équidistants d'équations $\beta = \pi/2 + p\pi$ avec $p \in \mathbb{Z}$ [25]. Il y a deux cas extrêmes possibles pour un mouvement vérifiant (c). Si les repères monde et multi-caméra sont tels que $\forall i, \mathcal{R}(E_M(t_i)) \approx R_z(\gamma_i)$, on est loin des singularités. Si les repères monde et multi-caméra sont tels que $\forall i, \mathcal{R}(E_M(t_i)) \approx R_y(\pi/2)R_x(\alpha_i)$, on est proche des singularités.

Euler corrigé On propose

$$\mathcal{R}(\alpha,\beta,\gamma) = \mathtt{A} R_z(\gamma) R_y(\beta) R_x(\alpha) \mathtt{B} \quad (7)$$

avec des rotations $\mathtt{A}$ et $\mathtt{B}$ constantes. Pour un mouvement vérifiant (c), on calcule $\mathtt{A}$ et $\mathtt{B}$ telles que $\mathtt{A}^{-1}\mathcal{R}(E_M(t_i))\mathtt{B}^{-1}$ est proche d'une rotation autour de $\begin{pmatrix}0 & 0 & 1\end{pmatrix}^\top$ pour tout i (détails dans [22]). Cela revient à tourner les repères mondes et multi-caméra de sorte que l'axe (oz) du repère multi-caméra reste presque vertical, et on se retrouve dans le cas favorable de "Euler classique" (avec β proche de 0). Rappelons que l'on a une estimation initiale de $\mathcal{R}(E_M(t_i))$ car notre AF prend en entrée une reconstruction complète avec les approximations GS et FA.

4 Expériences

4.1 Multi-caméras

On expérimente plusieurs caméras omnidirectionnelles en supposant que tous les paramètres de calibration sont constants pendant chaque acquisition. Le gain de caméra évolue de façon indépendante pour chaque caméra.
La première est composée de quatre Gopro Hero 3 [3] qui sont démarrées en appuyant une fois sur le bouton d'une télécommande wifi. L'utilisateur peut choisir la pose relative des caméras : dans une boite en carton pour avoir une distance minimale entre les centres des caméras, ou bien fixées à l'aide de coques en plastique fournies par le constructeur. La seconde est une caméra sphérique modélisée par deux fish-eyes de directions opposées, avec une très faible distance entre les deux centres et qui sont synchronisées exactement (à la ligne près) : la caméra Ricoh Theta S [4]. On expérimente aussi avec une multi-caméra professionnelle (Ladybug2 [5]) pour expérimenter une multi-caméra idéale GS et parfaitement synchronisée, dont la vérité terrain est donnée par le constructeur sous la forme d'une table de rayons. A l'exception d'un cas synthétique, les autres caméras ont une vérité terrain seulement partielle (un stroboscope donne τ dans tous les cas).

4.2 Jeux de données

Il y a trois vidéos multi-caméras prises avec les 4 Gopro dans des conditions variables montrées dans la Fig. 1 (BC1 ou BikeCity1 : à vélo dans une ville, WT ou WalkTown : marche dans un village, FH ou FlyHill : vol en parapente à très basse altitude en haut du Puy de Dôme). WT et BC1 ont la même configuration de caméras (boite en carton). Pour FH, la distance entre les centres est plus grande, le FR est plus petit et la résolution angulaire est plus grande. BC2 ou BikeCity2 est générée par ray-tracing d'une scène urbaine synthétique avec des textures réelles et en déplaçant la caméra le long d'une trajectoire imitant celle de BC1. On obtient une vidéo pour chaque caméra en compressant les images obtenues à l'aide de la commande "ffmpeg" [6] et les options "-c :v libx264 -preset slow -crf 18". BC2 a une vérité terrain : $f\Delta_1 = 0.25, f\Delta_2 = 0.50, f\Delta_3 = 0.75$ et un τ similaire à celui de BC1 (rappel : si $f\Delta_j = 1$ et f est le FR, Δ_j est le temps séparant deux images successives). Il y a aussi une vidéo WU ou WalkUniv (marche dans le campus de l'UCA) avec la caméra sphérique Ricoh Theta S. Enfin, CC ou CarCity est prise avec Ladybug2 et a une trajectoire similaire à BC1. Cette caméra est montée sur un véhicule à l'aide d'un mât, est à une hauteur d'environ 4m, et fournit des images non compressées (toutes les autres sont compressées avec H.264). La table 1 donne plus d'informations sur les jeux de données.

4.3 Notations

On utilise les notations suivantes. #2D est le nombre d'inliers 2D (points dont l'erreur de reprojection est inférieur à 4 pixels). GT signifie vérité terrain (ground truth). La fréquence d'images ou FR est f. RMS est l'erreur RMS en pixels. AF signifie ajustement de faisceaux. gs.sfa est une méthode alternative de synchronisation SFA (sans AF) sous l'hypothèse GS, et partant d'une synchronisation FA [22]. y_{max} est le nombre de lignes d'une image monoculaire ($y_{max} = 768$ pour CC, $y_{max} = 1440$ pour FH, $y_{max} = 960$ pour BC1/WT/BC2/WU).

Ajustement de faisceaux Une combinaison de notations décrit les paramètres estimés par notre AF. On a C (approximation centrale) si on estime tous les $\mathtt{R}_j$ et on fixe tous les $\mathbf{t}_j = \mathbf{0}$, ou bien NC (non-central) si on estime tous les $\mathtt{R}_j$ et $\mathbf{t}_j$. On a RS (rolling shutter) si on estime τ ou bien GS (global shutter) si on fixe $\tau = 0$. On a SFA (subframe-accurate) si on estime tous les Δ_j ou bien FA si on fixe tous les $\Delta_j = 0$. On a INT si on raffine tous les paramètres intrinsèques et distortion radiales (chaque caméra a ses propres paramètres intrinsèques) ; on utilise un modèle polynomial [15] (pour les Gopro et Ladybug2) ou unifié [9] (pour Ricoh Theta S). On donne un exemple avec ces notations : GS.NC.SFA.INT (ou bien gs.nc.sfa.int) est un AF qui fixe $\tau = 0$ et estime simultanément tous les Δ_j, $\mathtt{R}_j$, $\mathbf{t}_j$ et paramètres intrinsèques et les poses $\mathbf{m}_i$ dans le repère monde et les points 3D. Chaque AF alterne trois fois une mise à jour d'inliers et une minimisation avec la méthode de Levenberg-Marquardt pour ces inliers. Une succession de deux AF est possible, par exemple gs.c.fa.int+rs.c.sfa pour faire d'abord gs.c.fa.int puis rs.c.sfa.

Erreurs L'erreur $e(\Delta)$ est la somme des erreurs absolues des $f\Delta_j$. L'erreur $e(\tau)$ est l'erreur relative de τ. On a aussi l'erreur de calibration d, qui est le RMS pour chaque pixel de la multi-caméra de l'angle entre deux rayons partant de ce pixel pour les deux calibrations (les calibrations estimée et vérité terrain). Elle est calculée comme dans [21, 22] et convertie en pixels avec la résolution angulaire des caméras (r dans la table 1).

4.4 Comparaisons avec la vérité terrain

On donne ici toutes les erreurs de plusieurs AFs pour BC2 et CC qui ont une vérité terrain complète.
La table 2 donne les erreurs $e(\Delta)$, $e(\tau)$ et d pour plusieurs AFs estimant à la fois les décalages en temps Δ_j et les délais de ligne τ. On compare GS.C.FA.INT+RS.C.SFA et

FIGURE 1 – Deux multi-caméras définies avec quatre Gopro Hero 3 et des images prises en un point pour chaque jeu de données. En haut et au milieu : les caméras sont dans une boite en carton (pour avoir une distance faible entre les centres). En bas : les coques fournies avec les caméras sont utilisées.

Nom (abrégé)	Caméra	f	r (mr)	b (cm)	τ (μs)	$f\Delta_j$	l (m)	fr	kfr	#Tracks	$\|\|\beta_i\|\|_\infty$
BikeCity1 (BC1)	4*Gopro 3	100	1.56	7.5	9.10	?	2500	50.4k	2047	343k	0.223
WalkTown (WT)	4*Gopro 3	100	1.56	7.5	9.10	?	900	70.3k	1363	240k	0.268
FlyHill (FH)	4*Gopro 3	48	1.06	18	11.3	?	1250	8.6k	627	432k	0.494
BikeCity2 (BC2)	4*Gopro 3	100	1.56	7.5	9.12	j/4	615	12.5k	225	51k	0.074
CarCity (CC)	Ladybug 2	15	1.90	6	0	0	2500	7.7k	891	282k	0.068
WalkUniv (WU)	Theta S	30	3.85	1.5	-32.1	0	1260	29.4k	1287	154k	0.129

TABLE 1 – Jeux de données : nombre d'images par seconde f (ou FR), résolution angulaire r (milliradian), diamètre b des centres des caméras, délai de ligne τ (vérité terrain), décalage en temps $f\Delta_j$ (vérité terrain), longueur approximative l de la trajectoire, nombre d'images fr et d'images clefs kfr, maximum des angles $|\beta_i|$ (en radians) du paramétrage en section 3.6

AFs appliqués à BC2	$e(\Delta)$	$e(\tau)$	d
gs.c.fa.int+rs.c.sfa	0.057	14.6%	1.970
rs.c.sfa.int	0.097	2.7%	1.476
gs.nc.fa.int+rs.nc.sfa	0.051	12.2%	1.312
rs.nc.sfa.int	0.111	3.7%	0.366
gs.sfa (in Sec. 4.1)	0.215	na	na
AFs appliqués à CC	$e(\Delta)$	$y_{max}f\tau$	d
gs.c.fa.int+rs.c.sfa	0.052	-0.0052	1.176
rs.c.sfa.int	0.055	-0.0069	1.167
gs.nc.fa.int+rs.nc.sfa	0.034	-0.0020	1.322
rs.nc.sfa.int	0.039	0.0086	1.313
gs.sfa (in Sec. 4.1)	6e-3	na	na

TABLE 2 – Erreurs pour les séquences BC2 et CC.

RS.C.SFA.INT, i.e. on compare les estimations séparées et simultanées des paramètres INT et des paramètres RS.SFA. On compare aussi les versions C et NC de ces AFs, et la synchronisation SFA sans AF (gs.sfa).

D'abord on expérimente la seule séquence RS avec une vérité terrain complète : BC2. On voit que les estimations simultanées de INT et RS.SFA ont des $e(\tau)$ et d plus petites que les estimations séparées (à la fois C et NC). Cependant, le cas séparé a une $e(\Delta)$ deux fois plus petite que dans le cas simultané, qui à son tour est deux fois plus petite que celle du SFA sans AF. On rappelle que $e(\Delta)$ cumule les erreurs absolues de la synchronisation SFA : une valeur de 0.1 (pour le cas simultané) signifie que la moyenne des erreurs pour n caméras est $0.1/(n-1)$. De plus, les AFs NC donnent aussi des d plus petits.

Ensuite, on expérimente CC, la seule séquence réelle avec une vérité terrain complète. Puisque CC est GS, l'erreur relative $e(\tau)$ est infinie et on la remplace par $y_{max}f\tau$ (plus $|y_{max}f\tau|$ est petit, meilleur est le résultat). On voit que tous les AFs donnent un $|y_{max}f\tau|$ faible comparé à celui d'une caméra grand public qui est de l'ordre de 0.8-0.9. De plus, tous les $e(\Delta)$ sont plus petites que 0.055 pour 5 caméras ; d s'accroit et $e(\Delta)$ décroit par les AFs NC. La synchronisation SFA sans AF donne ici la plus petite $e(\Delta)$.

4.5 Décalages en temps et délais de ligne

On donne ici les décalages en temps et délais de ligne pour tous les jeux de données obtenus avec RS.C.SFA.INT.

La table 3 montre les décalages normalisés $f\Delta_j$, les délais normalisés $y_{max}f\tau$, et les erreurs $e(\tau)$ pour toutes les vidéos en appliquant l'AF RS.C.SFA.INT. On voit que $e(\tau)$ est inférieure à 7.2% sauf pour WT. Pour WT, τ est sur-estimé ($y_{max}f\tau$ est même plus grand que 1, sa valeur maximale théorique) avec une grande $e(\tau)$ égale à 16%. Comparé à cela, $e(\tau)$ de BC1 est vraiment faible. En fait, la valeur de τ sur une seule expérience est à tempérer car elle dépend du choix des images clefs (voir la section 4.6). On note que τ peut être négatif (pour WU), cela signifie simplement que la date de la y-ième ligne augmente lorsque y diminue. Enfin, les valeurs de $f\Delta_j$ sont très proches de leur vérité terrain que l'on connait pour BC2, CC et WU (voir la colonne $f\Delta_j$ dans la table 1).

	$f\Delta_1$	$f\Delta_2$	$f\Delta_3$	$y_{max}f\tau$ (GT)	$e(\tau)$
BC1	-0.334	-0.153	0.132	0.8755(0.873)	0.2%
WT	-0.583	-0.320	-0.795	1.013(0.873)	16.0%
FH	0.287	0.203	-0.326	0.8372(0.781)	7.2%
BC2	0.246	0.546	0.797	0.8989(0.875)	2.7%
CC	-0.017	-0.013	-0.006	-0.0069(0)	nan
WU	0.001	na	na	-0.8882(-0.924)	3.9%

TABLE 3 – décalages en temps (SFA) et délais de lignes pour tous les jeux de données estimés par l'AF rs.c.sfa.int. Ici, GT est la vérité terrain de $y_{max}f\tau$.

N_3	$e(\Delta)$	$e(\tau)$	d
400	0.089	1.9%	1.466
425	0.131	3.0%	1.570
450	0.097	2.7%	1.476
475	0.191	7.4%	1.867
500	0.199	2.7%	1.664
mean	0.141	3.5%	1.609
max/min	2.23	4.0	1.27

TABLE 4 – Stabilité des erreurs de RS.C.SFA.INT vis à vis du choix des images clefs pour BC2.

4.6 Stabilité par rapport aux images clefs

Ici on expérimente la stabilité de nos résultat vis à vis d'un changement modéré du choix des images clefs. Ce choix ne dépend pas de la calibration et est réglé à l'aide d'un seuil N_3, qui est une borne inférieure sur le nombre de points appariés entre trois images clefs successives. Pour chaque valeur $N_3 \in \{400, 425, 450, 475, 500\}$, on applique d'abord un structure-from-motion multi-caméra [19] qui choisit les images clefs avec N_3 puis l'AF RS.C.SFA.INT (la valeur utilisée dans les expériences précédentes avec 4 Gopro est $N_3 = 450$). La calibration multi-caméra initiale et la synchronisation FA est la même pour tous les N_3.

La table 4 donne les erreurs $e(\Delta)$, $e(\tau)$ et d pour BC2. Il y a 206 images clefs si $N_3 = 400$ et 248 images clefs si $N_3 = 500$. Les variations d'erreurs sont importantes : du simple au double pour $e(\Delta)$, du simple au quadruple pour $e(\tau)$, et d'environ 30% pour d. La table 5 donne les décalages normalisés $f\Delta_j$ et l'erreur $e(\tau)$ pour la séquence la plus longue BC1. Les variations de $e(\tau)$ sont importants ; les variations de $f\Delta_j$ sont inférieures à 0.032.

4.7 Vitesse variable et calibrations GS-RS

Enfin, on expérimente l'effet de la vitesse de la multi-caméra sur les erreurs de calibrations GS et RS. On re-synthétise pour cela la séquence BC2 pour plusieurs vitesses moyennes de 20km/h à 100km/h (la vitesse de la séquence originale BC2 est d'environ 18km/h comme BC1). La trajectoire est la même, mais avec des ralentissements dans des virages (c'est plus réaliste, et aussi utile

N_3	kfr	$f\Delta_1$	$f\Delta_2$	$f\Delta_3$	$y_{max}f\tau$	$e(\tau)$
400	1813	-0.341	-0.171	0.131	0.8920	2.1%
425	1929	-0.337	-0.155	0.131	0.8882	1.6%
450	2047	-0.334	-0.153	0.132	0.8755	0.2%
475	2166	-0.366	-0.156	0.134	0.9399	7.5%
500	2256	-0.337	-0.141	0.130	0.8786	0.5%
$\bar{m}$	2042	-0.343	-0.155	0.132	0.8948	2.4%
$\|m\|$	443	0.032	0.030	0.004	0.0644	7.3%

TABLE 5 – Stabilité des décalages en temps et délais de ligne de RS.C.SFA.INT vis à vis du choix des images clefs pour BC1. Le nombre des images clefs est kfr, $\bar{m}$ est la moyenne et $|m|$ la différence entre maximum et minimum.

pour l'étape de mise en correspondance de [19]). La figure 2 montre les résultats des AFs GS.(N)C.FA.INT et RS.(N)C.SFA.INT. L'erreur d de GS augmente avec la vitesse, ce qui est compréhensible car les effets RS (non modélisés par GS) s'accroissent avec la vitesse. Les erreurs d, $e(\Delta)$ et $e(\tau)$ de RS s'accroissent également avec la vitesse lorsque celle ci est supérieure ou égale à 60 (les approximations pour calculer $M(t)$ sont peut être moins tenables pour les grandes vitesses). Plus important, l'erreur d de RS.X.SFA.INT est meilleure que celle de GS.X.FA.INT pour X=C et X=NC. Elle reste inférieure à 1.8 pixels pour une vitesse inférieure ou égale à 80km/h. Si on utilise RS, NC apporte la plupart du temps (pour toutes les erreurs) un léger gain par rapport à C, et c'est souvent le contraire si on utilise GS. Notons que tous les AFs utilisent les mêmes images clefs et mise en correspondance pour une vitesse donnée.

5 Conclusion

Cet article décrit le premier ajustement de faisceaux pour auto-étalonner une multi-caméra, qui estime les décalages en temps et le délai de ligne en plus des paramètres habituels (paramètres intrinsèques, poses multi-caméra et entre caméras, points 3D). On part du résultat obtenu à l'aide d'une méthode précédante d'auto-étalonnage sous l'hypothèse que les caméras sont global shutter et donnant une synchronisation à l'image près.

On expérimente dans un contexte que nous pensons utile pour des applications comme la modélisation 3D, la réalité virtuelle et la vidéo 360 : plusieurs caméras grand public ou une caméra sphérique montées sur un casque. Les trajectoires longues sont permises car notre méthode estime seulement des paramètres liés aux images clefs fournies par une méthode classique de structure-from-motion. On donne les erreurs de calibration, de délais de lignes et de décalages en temps grâce à la vérité terrain.

Cependant, la méthode a une limitation : les déformations d'images dues au rolling shutter doivent être modérées, non seulement pour l'initialisation par structure-from-motion sous l'approximation global shutter, mais aussi pour la définition continue en temps du mouvement de multi-caméra

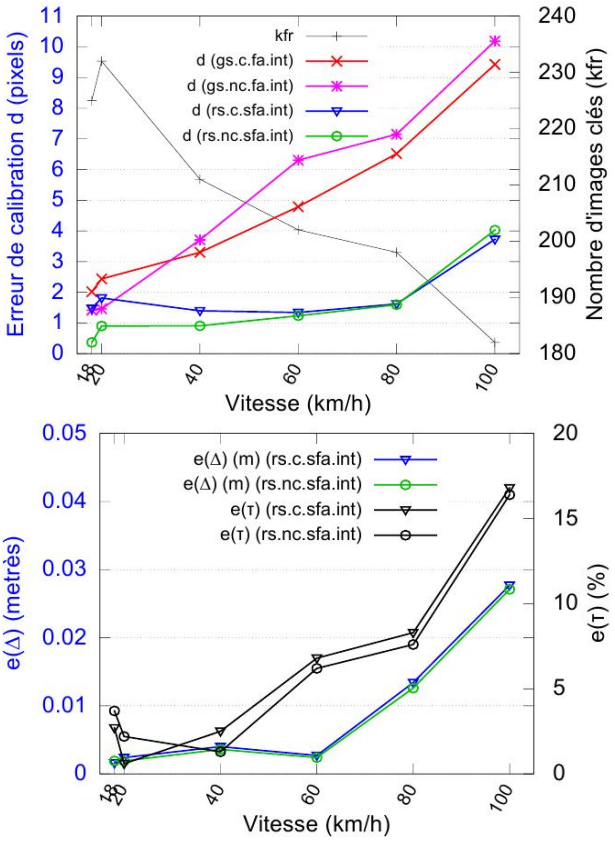

FIGURE 2 – Erreurs de calibration pour BC2 et plusieurs vitesses. En haut : d et le nombre d'images clefs. En bas : $e(\Delta)$ converties en mètres (c'est à dire $\frac{e(\Delta)v}{3f}$) et $e(\tau)$.

à partir des images clefs. De plus, plusieurs améliorations sont possibles. L'initialisation des paramètres intrinsèques et extrinsèques peut être améliorée grâce à des travaux précédents. Un prétraitement peut sélectionner des segments de vidéos ou l'on peut appliquer sans problème le structure-from-motion. Des alternatives sont possibles pour paramétrer les rotations, sélectionner les images clefs, et modéliser les caméras. Enfin, il faudrait examiner l'apport dans les applications des estimations de calibration dans notre contexte.

Références

[1] www.360rize.com.

[2] www.orah.co.

[3] gopro.com.

[4] theta360.com.

[5] www.ptgrey.com.

[6] ffmpeg.org.

[7] G. Duchamp, O. Ait-Aider, E. Royer, and J. Lavest. Multiple view 3D reconstruction with rolling shutter cameras. In *VISIGRAPP'15*.

[8] P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IROS'13*.

[9] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical applications. In *ECCV'00*.

[10] C. Geyer, M. Meingast, and S. Sastry. Geometric models of rolling-shutter cameras. In *OMNIVIS'05*.

[11] F. S. Grassia. Practical parametrization of rotations using the exponential map. *Journal of graphics tools*, 3(3), 1998.

[12] J. Hedborg, P. Forseen, M. Felsberg, and R. Ringaby. Rolling shutter bundle adjustment. In *CVPR'12*.

[13] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *ISMAR'09*.

[14] B. Klingner, D. Martin, and J. Roseborough. Street view motion-from-structure-from-motion. In *ICCV'13*.

[15] J. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration ? In *ECCV'98*.

[16] P. Lebraly, E. Royer, O. Ait-Aider, C. Deymier, and M.Dhome. Fast calibration of embedded non-overlapping cameras. In *ICRA'11*.

[17] M. Lhuillier and T. Nguyen. Synchronization and self-calibration for helmet-held consumer cameras, applications to immersive 3d modeling and 360 videos. In *3DV'15*.

[18] S. Lovegrove, A. Patron-Perez, and G. Sibley. Spline fusion : a continuous-time representation for visual-intertial fusion with application to rolling shutter cameras. In *BMVC'13*.

[19] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion. In *BMVC'07*.

[20] T. Nguyen and M. Lhuillier. Adding synchronization and rolling shutter in multi-camera bundle adjustment. In *BMVC'16*.

[21] T. Nguyen and M. Lhuillier. Synchronisation et auto-étalonnage de plusieurs caméras fixées sur un casque. In *RFIA'16*.

[22] T. Nguyen and M. Lhuillier. Self-calibration of omnidirectional multi-camera including synchronization and rolling shutter. *CVIU*, 162, 2017.

[23] L. Oth, P. Furgale, L. Kneip, and R. Siegwart. Rolling shutter camera calibration. In *CVPR'13*.

[24] J. Schneider and W. Forstner. Bundle adjustment and system calibration with points at infinity for omnidirectional cameras. Technical Report TR-IGG-P-2013-1, Institute of Geodesy and Geoinformation, University of Bonn, 2013.

[25] P. Singla, D.Mortari, and J.L.Junkins. How to avoid singularity when using Euler angles ? In *AAS Space Flight Mechanics Conference*, 2004.

[26] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms : Theory and Practice*, 2000.