Adding Synchronization and Rolling Shutter in Multi-Camera Bundle Adjustment

Thanh-Tin Nguyen

Maxime Lhuillier
http://maxime.lhuillier.free.fr

Institut Pascal
CNRS UMR 6602, Université Blaise Pascal, IFMA
Aubière, FR

Abstract

Multi-cameras built by fixing together several consumer cameras become popular and are convenient for applications like 360 videos. However, their self-calibration is not easy since they are composed of several unsynchronized and rolling shutter cameras. This paper introduces a new bundle adjustment for these multi-cameras that estimates not only the usual parameters (camera poses and 3D points) but also the synchronization and the rolling shutter of the cameras. We experiment using videos taken by GoPro cameras mounted on a helmet, moving along trajectories of several hundreds of meters or kilometers, and compare our results to ground truth.

1 Introduction

Multi-cameras built by fixing together several consumer cameras become popular thanks to their prices, high resolutions, their growing applications including 360 videos (*e.g.* in YouTube) and generation of virtual reality content [1, 2]. However such a multi-camera also has drawbacks. A first problem is the lack of accurate synchronization between the cameras. In the usual cases (*e.g.* GoPro), the camera manufacturer provides a wifi-based synchronization: the user starts all videos by a single click. However it is too inaccurate for applications such as 360 video and 3D modeling. Secondly, a low price camera implies that the camera is rolling shutter or RS. This means that two different lines of pixels of a frame are acquired at different instants (in a global shutter or GS camera, all pixels of a frame have the same time). Both inaccurate synchronization and RS complicate the self-calibration for the same reason: they act as time varying relative pose between the cameras, *i.e.* the multi-camera has a varying non-central calibration [5].

This paper introduces a new bundle adjustment (or BA) for these multi-cameras, that simultaneously estimates not only the usual parameters (camera poses and 3D points) but also the synchronization and the RS coefficient. We start from an initial calibration with the simplest camera model (GS) and a frame-accurate synchronization (FA) provided by previous self-calibration methods [10]. FA means that we skip the first frames in the videos such that the sequels of the videos have the following property: the frames with the same frame index are taken at the same time up to the inverse of fps (frames-per-second). Our BA provides subframe-accurate synchronization (SFA), *i.e.* it estimates the residual time offsets

Pages 62.1-62.11
DOI: https://dx.doi.org/10.5244/C.30.62

between a reference camera and the others. It also estimate the RS coefficient, *i.e.* the time delay between two adjacent lines of a frame.

2 Previous work

In contrast to our multi-camera BA, the previous ones estimate neither synchronization nor RS. They [9, 10, 14] assume that the cameras are synchronized and GS, only [8] deals with known RS but needs other sensors. Previous monocular BA estimates the RS assuming that the 3D points are known in a calibration pattern [13] or enforce a known RS coefficient [3, 6]. In the context of visual SLAM [7], a GS BA is applied to a RS (monocular) camera thanks to a RS compensation: this method corrects beforehand the RS effects on the feature tracks by estimating instantaneous velocities of the camera.

Each RS BA has a model of the camera trajectory, which provides the camera pose at each instant corresponding to each line of a frame, and which should have a moderated number of parameters to be estimated. In [6], one pose is estimated at each frame by BA and the poses between two consecutive frames are interpolated from the poses of these two frames. An assumption on the inter-frame motion is required. The BA in [3] adds extra parameters to avoid this assumption: it not only optimizes a pose but also rotational and translation speeds at every keyframe (not all frames). The translation speed is optional if its RS effect is negligible compared to that of rotation speed. In [13], a continuous-time trajectory model is used using B-splines and the BA optimizes the knots of the splines. The method chooses the number of knots and initializes their distribution along the trajectory sequence. In [8], the relative pose between an inter-frame pose and an optimized frame pose is provided by IMU at high frequency. The visual only RS approaches [3, 6, 7, 13] are experimented on few meters long camera trajectories. Ours is experimented on longer trajectories (hundreds of meters, kilometers) since it only estimates poses at keyframes.

In the context of a general multi-sensor, [4] simultaneously estimates the temporal and spatial registrations between sensors. In the experiments, the multi-sensor is composed of a camera and IMU. The best accuracy is obtained thanks to the use of all measurements at once, a continuous-time representation (a B-spline for IMU poses) and maximum likelihood estimation of the parameters (time offset, transformation between IMU and camera, IMU poses, and others). In [11], a camera-inertial multi-sensor is self-calibrated (synchronization, spatial registration, intrinsic parameters) by a sliding window visual odometry. Thanks to an adequate continuous-time motion parametrization, it also deals with RS cameras and has a better parametrization of the rotations. Indeed, it avoids the singularities of the global and minimal parametrization of rotations (*e.g.* in [4]), but assumes that the time between consecutive keyframes is uniform. Our work introduces a global minimal rotation parametrization and deals with non-uniform distribution of keyframes provided by standard Structure-from-Motion (SfM). This is done thanks to an assumption on the multi-camera motion, which is tenable for helmet-held cameras in most cases.

3 Proposed Method

3.1 Initialization

First we assume that the monocular videos are approximately synchronized by removing few frames at their beginning (FA synchronization), *i.e.* the videos are synchronized up to the inverse of fps. The fps is assumed to be the same for all videos. Then we define the i-th frame of the multi-camera by a concatenation of sub-images, every of them is the i-th frame of a monocular camera. From now on, we use word *frame* for "frame of the multi-camera" and the *video* is the sequence defined by all frames. Last we use standard SfM based on keyframe subsampling of the video and local BA [12], followed by global BA. The camera is self-calibrated assuming GS [10]. We remind that the *keyframes* are the only frames whose poses are refined by the BAs (this is useful for both time computation and accuracy).

3.2 Parametrization of the Multi-Camera Trajectory

Let $\mathcal{R}$ be a $\mathcal{C}^1$ continuous function that maps $\Omega \subseteq \mathbb{R}^k$ to the set $SO(3)$ of rotations of $\mathbb{R}^3$. We assume that there is a $\mathcal{C}^3$ continuous function $M : [0,1] \subset \mathbb{R} \to \mathbb{R}^3 \times \Omega$ that parametrizes the motion of the multi-camera. More precisely, $M(t)^T = \begin{pmatrix} T_M(t)^T & E_M(t)^T \end{pmatrix}$ where $T_M(t) \in \mathbb{R}^3$ is the translation and $\mathcal{R}(E_M(t)) \in SO(3)$ is the rotation. The columns of matrix $\mathcal{R}(E_M(t))$ and $T_M(t)$ are the vectors of the multi-camera coordinate system at time t expressed in world coordinates. The choice of $\mathcal{R}$ is detailed in Sec. 4 for the paper clarity.

Thanks to these notations and assumptions, we will approximate M at every time $t \in [0,1]$ by using values of M at few times $t_0, t_1, \cdots t_n$ where $t_0 = 0, t_i < t_{i+1}$ and $t_n = 1$. The $M(t_i)$ are the only parameters of the multi-camera trajectory estimated by our BA. Sec. 3.3 defines t_i and Sec. 3.4 describes our approximation of $M(t)$ by using the $M(t_i)$.

3.3 Time, Rolling-Shutter and Synchronization Parameters

The i-th keyframe is composed of sub-images taken by the monocular cameras. Every line of every sub-image is taken at its own time, which is described now. The 0-th line of the 0-th sub-image in the i-th keyframe is taken at time t_i, assuming that the time exposure of a line is instantaneous [5]. Thus $t_{i+1} - t_i$ is a multiple of the inverse of fps. Since the cameras are RS, line delay τ is such that the y-th line of the 0-th sub-image in the i-th keyframe is taken at time $t_i + y\tau$. Let $\Delta_j \in \mathbb{R}$ be the time offset between the j-th camera and the 0-th camera: the 0-th line of the j-th sub-image in the i-th keyframe is taken at time $t_i + \Delta_j$. Since we assume that all cameras have the same fps and same (and constant) τ, the y-th line of the j-th sub-image in the i-th keyframe is taken at time $t_i + \Delta_j + y\tau$.

3.4 Approximations for the Multi-Camera Trajectory

First we have Taylor's expansion

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

Second we provide a relation between derivative $M'(t_i)$ and all $M(t_i)$. Let D be function

$$D(\mathbf{x}, \mathbf{y}, \mathbf{z}, a, b) = \frac{b\mathbf{z}}{a(a+b)} - \frac{a\mathbf{x}}{b(a+b)} + \frac{(a-b)\mathbf{y}}{ab}, \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^{k+3}, a > 0, b > 0. \tag{2}$$

Let $\Delta = \max_i(t_{i+1} - t_i)$ and shortened notation $\mathbf{m}_i = M(t_i)$. Thanks to a linear combination of Taylor expansions of M at t_i (more details in the supplementary material),

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

Third we approximate $M(t)$ from the $\mathbf{m}_i$ by neglecting all remainders expressed by "$\mathcal{O}$" above. We compute $M(t)$ for the y-th line of the j-th camera/sub-image in the i-th keyframe using $t = t_i + \Delta_j + y\tau$ (Sec. 3.3). If $0 < i < n$,

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

If $i = 0$ (similarly if $i = n$), we use $M(t) = \mathbf{m}_0 + (t - t_0)(\mathbf{m}_1 - \mathbf{m}_0)/(t_1 - t_0)$.

Last we provide conditions that reduce the remainders of our Taylor developments, *i.e.* $\mathcal{O}(|t - t_i|^2)$ and $\mathcal{O}(\Delta^2)$. If the density of keyframes in the video increases, Δ decreases. Furthermore, $|y\tau| \leq 1/fps$ and the FA synchronization provides $|\Delta_j| \leq 1/fps$. Thus $|t - t_i| = |\Delta_j + y\tau|$ decreases if the fps increases. If $M'' = 0$, both remainders are exactly 0.

3.5 Reprojection Error of the Multi-Camera

Since our BA minimizes the sum of squared modulus of reprojection error for every inlier, this section describes the computation of a reprojection error for 3D point $\mathbf{x} \in \mathbb{R}^3$ (in world coordinates) and its inlier observation $\tilde{\mathbf{p}} \in \mathbb{R}^2$ in the j-th sub-image of the i-th keyframe.

First we introduce notations. Let $\mathbf{p} \in \mathbb{R}^2$ be the projection of $\mathbf{x}$ in the j-th sub-image of the i-th keyframe. The reprojection error is $\mathbf{p} - \tilde{\mathbf{p}}$. Let $(\mathtt{R}_j, \mathbf{t}_j)$ be the pose of the j-th camera in the multi-camera frame. Let $K_j : \mathbb{R}^3 \setminus \{\mathbf{0}\} \rightarrow \mathbb{R}^2$ be the projection function of the j-th camera. We assume that $K_j, \mathtt{R}_j, \mathbf{t}_j$ are constant. The acquisition times of $\mathbf{p} = (x, y)$ and $\tilde{\mathbf{p}} = (\tilde{x}, \tilde{y})$ are $t_{\mathbf{p}} = t_i + \Delta_j + y\tau$ and $t_{\tilde{\mathbf{p}}} = t_i + \Delta_j + \tilde{y}\tau$.

Second we detail the relation between $\mathbf{p}$ and $\mathbf{x}$. Both $E_M(t_{\mathbf{p}})$ and $T_M(t_{\mathbf{p}})$, *i.e.* $M(t_{\mathbf{p}})$, are defined by Eq. 4 using index i of the keyframe and $t = t_{\mathbf{p}}$. The coordinates of $\mathbf{x}$ in the multi-camera coordinate system is $\mathbf{x}_M = \mathcal{R}(E_M(t_{\mathbf{p}}))^\top(\mathbf{x} - T_M(t_{\mathbf{p}}))$. The coordinates of $\mathbf{x}$ in the j-th camera coordinate system is $\mathbf{x}_j = \mathtt{R}_j^\top(\mathbf{x}_M - \mathbf{t}_j)$. We also have $\mathbf{p} = K_j(\mathbf{x}_j)$.

Third we estimate $\mathbf{p}$. We see that $\mathbf{p}$ needs the computation of $\mathbf{x}_M$, which in turn needs the computation of (the y coordinate of) $\mathbf{p}$. This problem is solved thanks to an approximation in [8]: $t_{\mathbf{p}}$ is replaced by $t_{\tilde{\mathbf{p}}}$ in the expression of $\mathbf{x}_M$, *i.e.* we assume that the multi-camera pose is the same at times $t_{\tilde{\mathbf{p}}}$ and $t_{\mathbf{p}}$. We think that this is acceptable since $|t_{\tilde{\mathbf{p}}} - t_{\mathbf{p}}| \leq \tau||\mathbf{p} - \tilde{\mathbf{p}}||_2$ and the magnitude order of τ is 10^{-5} s/pixel and $\tilde{\mathbf{p}}$ is an inlier (*i.e.* $||\mathbf{p} - \tilde{\mathbf{p}}||_2 \leq 4$ pixels).

3.6 Summary

First keyframes are selected and standard BA initializes the 3D points and the keyframe poses assuming GS cameras and FA synchronization (Sec. 3.1). Then the $\mathbf{m}_i$ are computed from the poses. We also have $\tau = 0$ (GS assumption) and $\Delta_j = 0$ (FA assumption). Last we apply our BA which refines not only the $\mathbf{m}_i$ and the 3D points, but also line delay τ (RS assumption) and/or the time offsets Δ_j (SFA assumption). It is based on Levenberg-Marquardt method.

4 Parametrization of Rotations

According to Sec. 3.2, $\mathcal{R}$ is a $\mathcal{C}^1$ continuous function such that $\mathcal{R}(\Omega) = SO(3)$ and $\Omega \subseteq \mathbb{R}^k$. Following [13], we prefer a minimal (non-redundant) parametrization $\mathcal{R}$ to avoid any constraints on the $\mathcal{R}$ entry and limit the number of estimated parameters. Sec. 4.1 is a reminder of these parametrizations and Sec. 4.2 details our choice.

4.1 Minimal Parametrizations of $SO(3)$ for BA

During BA, the set of all rotations in a neighborhood of a current estimate of a rotation should be reachable by parametrization $\mathcal{R}$ [16]. Since $SO(3)$ is a 3D manifold, such a neighborhood is 3D and jacobian $\partial\mathcal{R}$ of $\mathcal{R}$ should have rank 3. Thus a minimal parametrization $\mathcal{R}$ meets $k=3$, *i.e.* $\Omega \subseteq \mathbb{R}^3$. Unfortunately, all 3D parametrizations of $SO(3)$ have singularities [15].

We detail the case of Euler parametrization $\mathcal{E}(\alpha,\beta,\gamma) = R_z(\gamma)R_y(\beta)R_x(\alpha)$ where $R_x(\alpha)$, $R_y(\beta)$ and $R_z(\gamma)$ are the rotations about axes x-y-z with angles α-β-γ. The singularities of $\mathcal{E}$ are the points $(\alpha,\beta,\gamma)\in\mathbb{R}^3$ where $\partial\mathcal{E}$ is rank deficient, *i.e.* planes $\beta=\pi/2+p\pi$ such that $p\in\mathbb{Z}$ (a detailed proof is in the supplementary material). Parametrization $\mathcal{E}$ can be used in BA if β is as far as possible to $\pi/2+\pi\mathbb{Z}$, *e.g.* local Euler angles [16] such that $|\beta|\ll 1$.

The angle-axis parametrization used in [13] has a bounded angle domain due to singularities at angles $2\pi\mathbb{Z}^*$. This restricts the continuous camera motion: rotations around a fixed axis must have angle range in $]-2\pi,2\pi[$, *e.g.* no more than two full turns around a building.

4.2 Keep Away from the Singularities of Euler Parametrization

Here we define $\mathcal{R}$ using $\mathcal{E}$ and keep away from all singularities. According to Sec. 3.2, $\mathcal{R}$ is a global parametrization for all rotations of the continuous camera motion $M(t)$. Note that there is a single $\mathcal{R}$ for all times, in contrast to the local methods in [15, 16] which switch $\mathcal{R}$ over time. Furthermore, $\mathcal{R}$ has singularities since it is minimal (Sec. 4.1). Thus we do an assumption on the camera motion to keep away from the singularities.

Now we remind that the multi-camera is helmet-held. All yaw motions of the head are possible since the user can move in all horizontal directions. We assume that the pitch and roll of the head are small, *i.e.* the viewing direction of the user is roughly pointing toward the horizon without odd roll rotations. We believe that this assumption is reasonable for an user exploring the environment without a special objective like grasping at objects on the ground.

Let $\mathtt{R}_i^0$ be the rotation of the initial $\mathbf{m}_i$ computed by standard BA. Thus all $(\mathtt{R}_i^0)^T\mathtt{R}_j^0$ are roughly rotations sharing a same axis $\mathbf{v}\in\mathbb{R}^3$. Let $R(\mathbf{v},\theta)$ be the rotation with axis $\mathbf{v}$ and angle θ. There are rotation $\mathtt{R}$ and angles θ_i such that $\mathtt{R}_i^0 \approx \mathtt{R}R(\mathbf{v},\theta_i)$. Let $\mathbf{k}=\begin{pmatrix}0 & 0 & 1\end{pmatrix}^T$. Let rotations $\mathtt{A}$ and $\mathtt{B}$ be such that $\forall i$, $\mathtt{A}\mathtt{R}_i^0\mathtt{B}\approx R(\mathbf{k},\gamma_i)$. Angles $(\alpha_i,\beta_i,\gamma_i)$ meet $\mathcal{E}(\alpha_i,\beta_i,\gamma_i)=\mathtt{A}\mathtt{R}_i^0\mathtt{B}$ and 2π multiples can be added to them. We would like that $\alpha_i,\beta_i,\gamma_i$ are coordinates of $M(t_i)$. We choose β_i that has the smallest $|\beta_i|$. Since $\mathcal{E}(\alpha_i,\beta_i,\gamma_i)\approx R(\mathbf{k},\gamma_i)$, $|\beta_i|$ is small enough to keep away from $\mathcal{E}$ singularities. We also remind that M is continuous and $|t_i-t_{i+1}|$ is small thanks to the keyframe sampling. Thus the γ_i series is chosen such that $|\gamma_i-\gamma_{i-1}|$ is as small as possible ($|\beta_i-\beta_{i-1}|$ is also small, and we do similarly for α_i). Last we define $\mathcal{R}(\alpha,\beta,\gamma)=\mathtt{A}^{-1}\mathcal{E}(\alpha,\beta,\gamma)\mathtt{B}^{-1}$ to obtain $\mathcal{R}(\alpha_i,\beta_i,\gamma_i)=\mathtt{R}_i^0$. Since $\mathcal{R}$ has the same singularities than $\mathcal{E}$ (supplementary material) and $\beta\approx\beta_i$ during our BA, (α,β,γ) is far from the $\mathcal{R}$ singularities.

4.3 Technical Details: Estimate $\mathtt{A}$ and $\mathtt{B}$

For all i and j, $(\mathtt{R}_i^0)^T\mathtt{R}_j^0\approx R(\mathbf{v},\theta_j-\theta_i)$. Let $\mathbf{v}_{i,j}$ be the axis of $(\mathtt{R}_i^0)^T\mathtt{R}_j^0$. First we search $\mathbf{v}$ as the most colinear vector to all $\mathbf{v}_{i,j}$, *i.e.* $\mathbf{v}$ maximizes $\sum_{i,j}(\mathbf{v}_{i,j}^T\mathbf{v})^2$. Thus $\mathbf{v}$ is the eigen vector of the largest eigen value of the symmetric matrix $\sum_{i,j}\mathbf{v}_{i,j}\mathbf{v}_{i,j}^T$. Second we estimate rotation $\tilde{\mathtt{R}}$ such that $\tilde{\mathtt{R}}\mathtt{R}_i^0\approx R(\mathbf{v},\theta_i')$. Since $\mathtt{R}_i^0\approx\mathtt{R}R(\mathbf{v},\theta_i)$, $\mathtt{R}_i^0\mathbf{v}\approx\mathtt{R}\mathbf{v}$. Let $\tilde{\mathbf{v}}=\sum_i\mathtt{R}_i^0\mathbf{v}/||\sum_i\mathtt{R}_i^0\mathbf{v}||$. Thus $\tilde{\mathbf{v}}\approx\mathtt{R}\mathbf{v}\approx\mathtt{R}_i^0\mathbf{v}$. Let $\tilde{\mathtt{R}}$ be a rotation such that $\tilde{\mathtt{R}}\tilde{\mathbf{v}}=\mathbf{v}$. Since $\tilde{\mathtt{R}}\mathtt{R}_i^0\mathbf{v}\approx\tilde{\mathtt{R}}\tilde{\mathbf{v}}=\mathbf{v}$, $\tilde{\mathtt{R}}\mathtt{R}_i^0\approx R(\mathbf{v},\theta_i')$. Let $\mathtt{R}'$ be a rotation such that $\mathtt{R}'\mathbf{v}=\mathbf{k}$. We obtain $\mathtt{R}'\tilde{\mathtt{R}}\mathtt{R}_i^0\mathtt{R}'^T\approx R(\mathbf{k},\gamma_i)$. Thus $\mathtt{A}=\mathtt{R}'\tilde{\mathtt{R}}$ and $\mathtt{B}=\mathtt{R}'^T$.

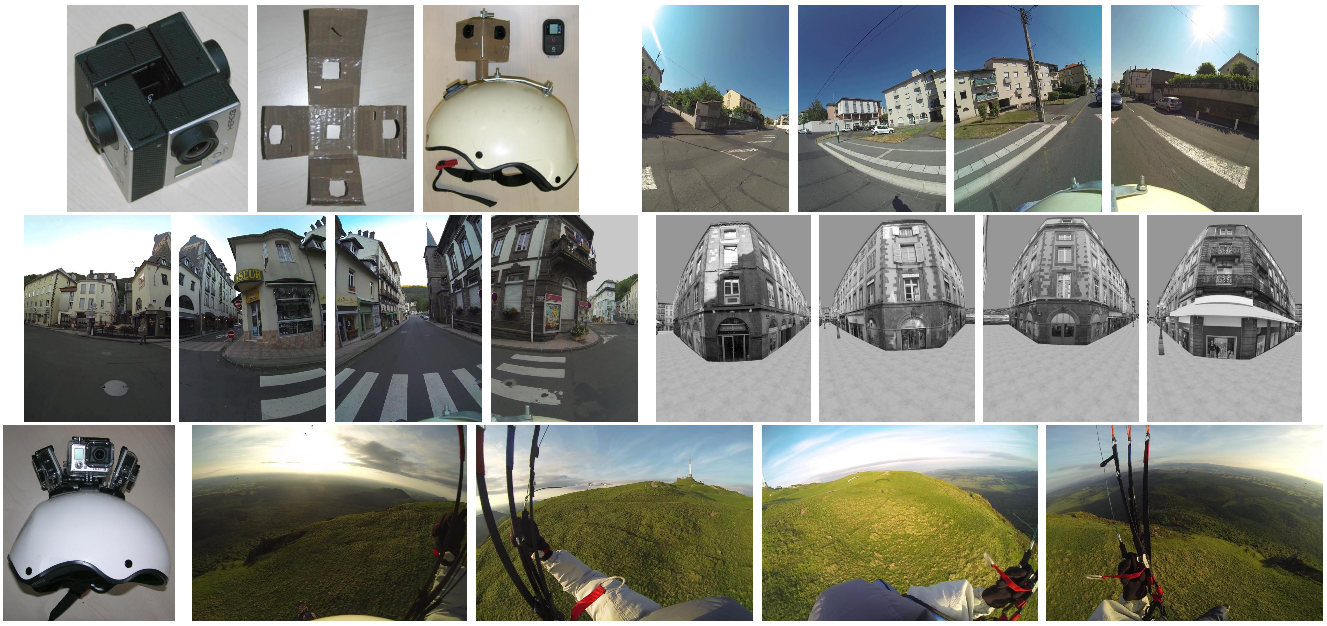

Figure 1: Two multi-cameras formed by four GoPro Hero3 cameras and images taken at a viewpoint for every dataset. Top and middle: the cameras are enclosed in a cardboard (for small baseline). Bottom: the housings provided with the cameras are used.

5 Experiments

5.1 Datasets and Notations

The multi-camera is defined by four GoPro Hero3 cameras (Fig. 1) that have the same setting at a time, except the camera gain that evolves independently for every camera. We assume that time offsets and calibrations do not change in a video. Tab. 1 summarizes our datasets: three real (multi-camera) videos under various conditions (bike riding in a city, walking in a town [10], paragliding flying at very low height above a hill) and one synthetic video. In all cases, a 360° field of view around the head is obtained. The ground truth of line delay τ is available thanks to a strobe. Furthermore, we use global shutter and central approximations [10] to obtain the initial self-calibration (FA synchronization, intrinsic parameters, 3D points, multi-camera poses and relative poses) and a concurrent SFA synchronization based on instantaneous angular velocity (we call it *Sync*).

BikeCity2 is generated by ray-tracing of a synthetic urban scene having real textures and by moving the camera along a trajectory that mimics that of BikeCity1 (the "pose noise", *i.e.* relative poses between consecutive frames, are similar in both videos). We obtain a video for each camera by compressing the output images using ffmpeg and options "-c:v libx264 -preset slow -crf 18". BikeCity2 has ground truth: $f\Delta_0 = 0$, $f\Delta_1 = 0.25$, $f\Delta_2 = 0.5$, and $f\Delta_3 = 0.75$ where $f = fps$ (if $f\Delta_j = 1$, Δ_j is the time between two consecutive frames).

We note that the setting of the cameras of FlyHill is different to those of the other videos (frequency, resolution, orientations, baseline). The baseline in the others is as small as possible for the central approximation. Furthermore, FlyHill is the most difficult case due to rolling shutter. Indeed, its fps is twice smaller and it includes faster head turns.

We use notations: C (central approximation) estimates $\mathtt{R}_j$ and fixes $\mathbf{t}_j = 0$, NC (non-central) estimates both $\mathtt{R}_j$ and $\mathbf{t}_j$, RS estimates τ, SFA estimates Δ_j, GS means $\tau = 0$ and FA means $\Delta_j = 0$. For example, a GS+SFA+NC bundle adjustment fixes $\tau = 0$ and estimates

Name	fps	r (mr)	b (cm)	τ (μs)	l (m)	fr	kfr	#3D	$\|\|\beta_i\|\|_\infty$
BikeCity1	100	1.56	7.5	9.12	2500	50.4k	1701	354k	0.184
WalkTown	100	1.56	7.5	9.12	900	70.3k	1329	400k	0.176
FlyHill	48	1.06	18	11.3	1250	8.6k	593	565k	0.434
BikeCity2	100	1.56	7.5	9.12	615	12.5k	372	110k	0.049

Table 1: Our videos: angular resolution r (milliradians), diameter b of multi-camera centers, line delay τ (ground truth), trajectory length l, numbers of frames fr and keyframes kfr and 3D points #3D, maximum of angles $|\beta_i|$ (radians) of our parametrization in Sec. 4.2.

simultaneously all Δ_j, $\mathtt{R}_j$, $\mathbf{t}_j$, the keyframe poses $\mathbf{m}_i$ and the 3D points. The threshold for inlier selections is set to 4 pixels in all videos.

5.2 One New Assumption at Once

There are several new assumptions NC, SFA and RS. Thus we first examine what they provide separately: we experiment GS+NC+FA, GS+C+SFA and RS+C+FA bundle adjustments. The number of independent and optimized parameters of these BAs are $x+9$, $x+3$ and $x+1$ (respectively), where x is the number of parameters of the initial GS+C+FA BA.

In Tab. 2, the inlier set is fixed in every video to compare the improvements in term of RMS of reprojection errors (in pixels). The RMS decreases are small (less than 1.7%), except for FlyHill: GS+C+SFA has 3.9% and RS+C+FA has 3.2%. This confirms that the RS and SFA effects are non negligible for FlyHill due to the fast image motion (it is faster than in other videos). We also see that the NC assumption has the lowest impact on the RMS in spite of its larger number of parameters. At this point, the relative error of τ is quite large for FlyHill (58%), it also important for the others (3.8%-13.3%). The error of $f\Delta_j$ (BikeCity2), or the difference between our $f\Delta_j$ and those of Sync (others), can reach 0.14 in BikeCity2 or 0.24 in WalkTown. Such discrepancies look large since we expect that $|f\Delta_j| \in [0,1]$, but the resulting discrepancies for the 3D locations of the multi-camera are small. For example, the mean distance between multi-camera poses for consecutive images of WalkTown is $900/70300m$, thus $f\Delta_j = 0.24$ implies a 3D discrepancy of only $2.9mm$.

We continue these experiments by alternating inlier updates and BAs in Tab. 3. Then τ is improved: the relative error is less than 7.9% except for FlyHill (42%). The $f\Delta_j$ estimation is similar to that in Tab. 2 (the RMS of differences of $f\Delta_j$ is less than 0.05). The inlier sets increases slightly: 0.6-0.9% for RS+C+FA and GS+C+SFA of FlyHill and less than 0.1% elsewhere; their main computations have been done before by initialization BA (GS+C+FA).

5.3 Several New Assumptions at Once

Now we try RS+SFA simultaneously and study differences between these results and the previous ones in Tab. 3 and Sync. More precisely, we compute RS+NC+SFA and RS+C+SFA like this. Get the GS+NC+FA result in Tab. 3, apply RS+NC+FA and then RS+NC+SFA. Get the GS+C+SFA result in Tab. 3, apply RS+C+SFA (all BAs include inlier updates). Fig. 2 shows views of the RS+NC+SFA results.

We first focus on BikeCity2 (Tab. 4). The accuracy of Δ_j is better than that in Tab. 3 and Sync (the RMS of $f\Delta_j$ difference between Tab. 4 and ground truth is 0.027 for RS+NC+SFA or 0.019 for RS+C+SFA). However the relative error of τ is worse: 6.8%-8.5% (0.31% in Tab. 3). Thus the simultaneous use of RS+SFA improves Δ_j but provides a bias for τ.

	Method	i-RMS	f-RMS	$f\Delta_1$	$f\Delta_2$	$f\Delta_3$	$10000f\tau$
BikeCity1	Sync or G.T.			-0.042	-0.163	0.306	9.120
	GS+**NC**+FA	0.9550	0.9534				
	GS+C+**SFA**	0.9550	0.9520	-0.122	-0.147	0.314	
	RS+C+FA	0.9550	0.9484				7.898
WalkTown	Sync or G.T.			0.517	0.474	0.443	9.120
	GS+**NC**+FA	0.9452	0.9406				
	GS+C+**SFA**	0.9452	0.9391	0.715	0.714	0.510	
	RS+C+FA	0.9452	0.9391				8.714
FlyHill	Sync or G.T.			0.207	-5e-4	-0.358	5.424
	GS+**NC**+FA	1.3643	1.3537				
	GS+C+**SFA**	1.3643	1.3107	0.058	-0.024	-0.231	
	RS+C+FA	1.3643	1.3207				2.270
BikeCity2	G.T.			0.25	0.5	0.75	9.120
	GS+**NC**+FA	0.8124	0.8121				
	GS+C+**SFA**	0.8124	0.7985	0.388	0.476	0.788	
	RS+C+FA	0.8124	0.8009				8.777

Table 2: BA results for a fixed set of inliers for every video. i-RMS and f-RMS are RMS before and after BA.

	Method	#3D	$f\Delta_1$	$f\Delta_2$	$f\Delta_3$	$10000f\tau$
BikeCity1	GS+**NC**+FA	+87				
	GS+C+**SFA**	+130	-0.137	-0.156	0.336	
	RS+C+FA	+323				8.401 (7.9%)
WalkTown	GS+**NC**+FA	+165				
	GS+C+**SFA**	+235	0.750	0.746	0.535	
	RS+C+FA	+242				9.020 (1.1%)
FlyHill	GS+**NC**+FA	+448				
	GS+C+**SFA**	+5231	0.089	-0.028	-0.312	
	RS+C+FA	+3627				3.153 (42%)
BikeCity2	GS+**NC**+FA	+9				
	GS+C+**SFA**	+38	0.403	0.495	0.836	
	RS+C+FA	+21				9.092 (0.31%)

Table 3: Results of our BAs with increasing set of inliers for every video (#3D is the number of 3D inlier points added to those in Tab 2, *i.e*. Tab 1). Percentages are relative errors.

For FlyHill, the relative error of τ is less that 5.1% and is quite better than that in Tab. 3; offsets Δ_j of RS+NC+SFA and RS+C+SFA are similar. Once more, the largest increase of inlier set (3.5%) is observed in this video. The relative error of τ is better for BikeCity1 (less than 5.3%), but it is worse for WalkTown (less than 9.8%). For FlyHill and BikeCity1-2, the RMS of $f\Delta_j$ diff. between Tab. 4 and Sync (range from 0.1 to 0.2) is greater than the RMS of $f\Delta_j$ diff. between Tab. 3 and Sync (range from 0.05 to 0.08). Both Sync and GS+C+SFA (Tab. 3) use the same GS+C assumption and we believe that they provide similar Δ_j for this reason (this does not mean that the Δ_j in Tab. 3 are better than the Δ_j in Tab. 4).

	Method	#3D	$f\Delta_1$	$f\Delta_2$	$f\Delta_3$	$10000f\tau$
BikeCity1	**RS+NC+SFA**	+492	-0.357	-0.155	0.151	8.922 (2.1%)
	RS+C+SFA	+379	-0.355	-0.148	0.156	8.636 (5.3%)
WalkTown	**RS+NC+SFA**	+489	0.543	0.807	0.339	10.016 (9.8%)
	RS+C+SFA	+343	0.567	0.795	0.343	9.480 (3.5%)
FlyHill	**RS+NC+SFA**	+19659	0.286	0.200	-0.326	5.700 (5.1%)
	RS+C+SFA	+19608	0.287	0.200	-0.330	5.595 (3.1%)
BikeCity2	**RS+NC+SFA**	+53	0.256	0.542	0.772	8.497 (6.8%)
	RS+C+SFA	+50	0.251	0.530	0.762	8.342 (8.5%)
	Sync		0.404	0.398	0.829	

Table 4: Results of our BAs with increasing set of inliers for every video.

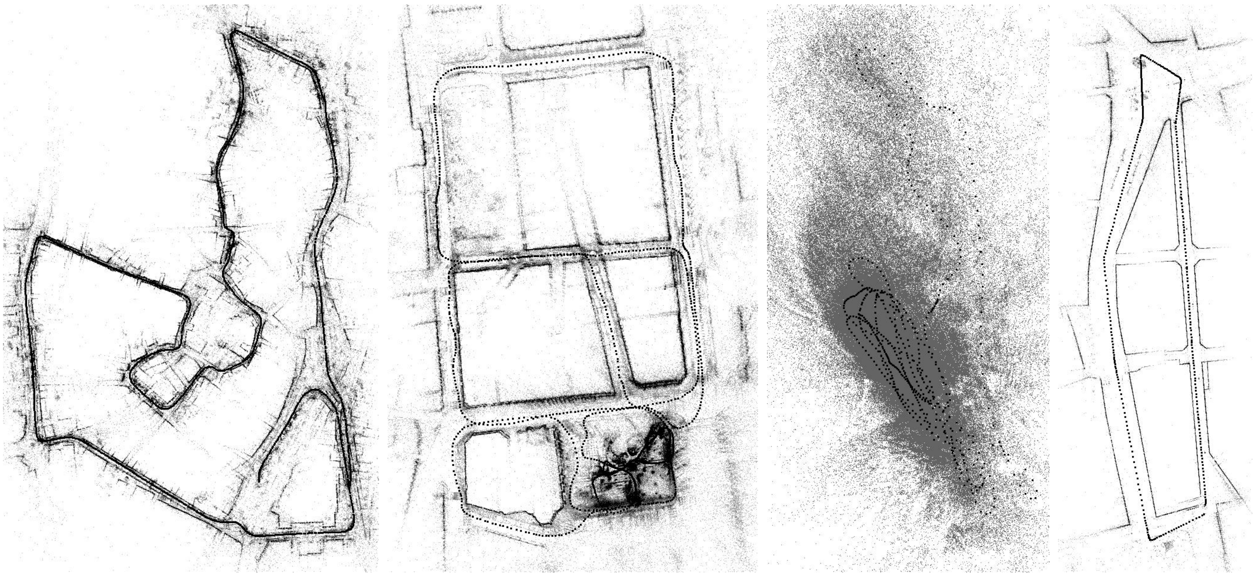

Figure 2: From left to right: reconstructions of BikeCity1, WalkTown, FlyHill and BikeCity2 by RS+NC+SFA without loop closure. The FlyHill trajectory has a lot of sharp S turns.

5.4 Parametrization of the Multi-Camera Orientations

Here we examine the Euler angles involved in our rotation parametrization in Sec. 4.2. Fig. 3 illustrates function E that maps keyframe number i to $(\alpha_i, \beta_i, \gamma_i)$ for BikeCity1. Function E looks continuous (zoom in to see the blue crosses); the largest value of $|\gamma_i - \gamma_{i-1}|$ is equal to 0.61 rad. Such a result is expected since E_M is assumed to be $\mathcal{C}^3$ continuous and the keyframe sampling t_i is dense enough to obtain a successful SfM result. Here $t_{i+1} - t_i$ ranges from 0.1s to 2.3s. Furthermore, $|\beta_i|$ is as small as possible to keep away from the singularities $\beta \in \pi/2 + \pi\mathbb{Z}$. According to Tab. 1, all $|\beta_i|$ are less than 0.44 rad. The $|\beta_i|$ RMS is about 0.3-0.6 times the $|\beta_i|$ maximum for every video.

Last we detail consequences of a naive use of Euler angles ignoring singularities (using notations in Sec. 4). Assume that the initial multi-camera poses meet $\mathtt{R}_i^0 \approx R_z(\gamma_i)R_y(\pi/2)$. This is possible because of a “bad” choice of coordinate systems: if the world coordinate system is rotated by $\mathtt{A}$ and the multi-camera coordinate system is rotated by $\mathtt{B}R_y(\pi/2)$, $\mathtt{R}_i^0$ is replaced by $\mathtt{A}\mathtt{R}_i^0\mathtt{B}R_y(\pi/2) \approx R_z(\gamma_i)R_y(\pi/2)$. Now we naively set $\mathcal{R} = \mathcal{E}$ and redo the experiments of BikeCity1. Then $\beta_i \approx \pi/2$ (this is very close to a singularity). Angles α_i and γ_i are quite more perturbed although they are chosen such that function E is as continuous as possible ($\max_i |\gamma_i - \gamma_{i-1}| = 3.11$ and $\max_i |\alpha_i - \alpha_{i-1}| = 3.13$). The new values of $10000f\tau$

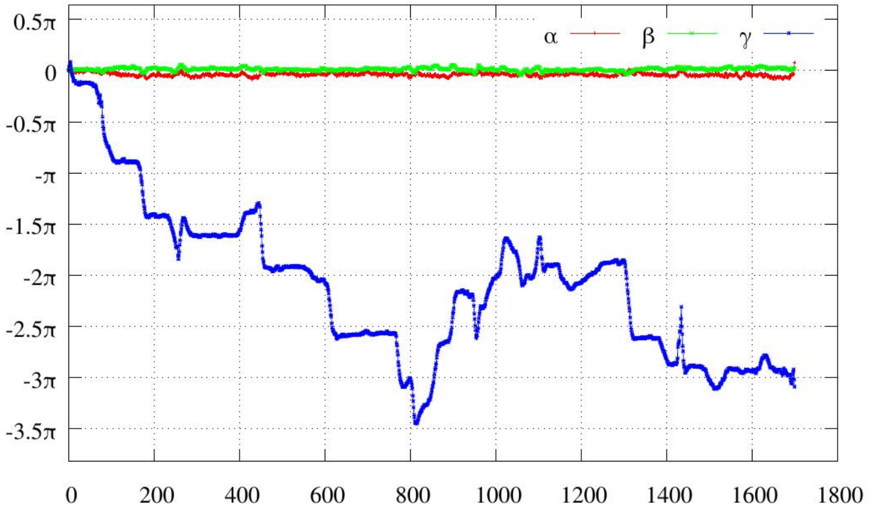

Figure 3: Euler angles for BikeCity1.

are 7.338, 7.957, 7.962 and 7.858 in respective conditions of Tabs. 2, 3, 4. Thus the relative errors of τ (%) increase by 6.1, 4.8, 10 and 8.5. The inlier sets are similar (slightly worse).

6 Conclusion

We present the first bundle adjustment for multi-cameras that estimate not only rolling shutter (line delay) but also synchronization (time offsets), in addition to the usual 3D parameters (points, camera and multi-camera poses). In contrast to the previous Structure-from-Motion methods involving rolling shutter, only keyframes are involved and we deal with larger trajectories (600m-2.5km). The multi-camera motion is modeled at all times thanks to Taylor approximations and a careful use of Euler angles avoiding singularities. We experiment in cases that we believe useful: several and identical consumer cameras mounted on a helmet.

At first glance, our approximations seem hazardous if the user does a motion that is not consistent with the neighboring keyframes. Anyway, the majority of keyframes provides accurate enough approximation to obtain the following results in our non trivial datasets. The relative error of the estimated line delay is less than 7.9% except in the most difficult case with faster head motions; the simultaneous estimation of line delay and time offsets can provide bias but it also provides the best result (5.1%) for the most difficult case. The best (subframe-accurate) time offsets are given by the simultaneous estimation.

Several extensions are possible: adding estimated parameters only for keyframes where our model of multi-camera motion is not accurate, adding parameters by taking care of overfitting (*e.g.* intrinsic parameters, one line delay and frame rate per camera), trying alternative camera models and rotation parametrizations, improving applications like 3D modeling and 360 video.

References

[1] http://www.360heros.com/.

[2] http://www.video-stitch.com.

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

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

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

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

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

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

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

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

[11] 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*.

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

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

[14] 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.

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

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