Flight Dynamics-based Recovery of a UAV Trajectory using Ground Cameras

Artem Rozantsev[a] Sudipta N. Sinha[b] Debadeepta Dey[b] Pascal Fua[a]

[a]Computer Vision Laboratory, EPFL
{artem.rozantsev, pascal.fua}@epfl.ch

[b]Microsoft Research
{sudipta.sinha, dedey}@microsoft.com

Abstract

We propose a new method to estimate the 6-dof trajectory of a flying object such as a quadrotor UAV within a 3D airspace monitored using multiple fixed ground cameras. It is based on a new structure from motion formulation for the 3D reconstruction of a single moving point with known motion dynamics. Our main contribution is a new bundle adjustment procedure which in addition to optimizing the camera poses, regularizes the point trajectory using a prior based on motion dynamics (or specifically flight dynamics). Furthermore, we can infer the underlying control input sent to the UAV's autopilot that determined its flight trajectory.

Our method requires neither perfect single-view tracking nor appearance matching across views. For robustness, we allow the tracker to generate multiple detections per frame in each video. The true detections and the data association across videos is estimated using robust multi-view triangulation and subsequently refined during our bundle adjustment procedure. Quantitative evaluation on simulated data and experiments on real videos from indoor and outdoor scenes demonstrates the effectiveness of our method.

1. Introduction

Rapid adoption of unmanned aerial vehicles (UAV) and drones for civilian applications will create demand for low-cost aerial drone surveillance technology in the near future. Although, acoustics [1], radar [2] and radio frequency (RF) detection [3] have shown promise, they are expensive and often ineffective at detecting small, autonomous UAVs [4]. Motion capture systems such as Vicon [5] and OptiTrack [6] work for moderate sized scenes. However, the use of active sensing and special markers on the target makes them ineffective for tracking non-cooperative drones in large and bright outdoor scenes. With the exception of some recent works [7, 8], visual detection and tracking of drones using passive video cameras remains a largely unexplored topic.

Existing single-camera detection and tracking methods are mostly unsuitable for drone surveillance due to their limited field of view and the fact that it is difficult to accurately estimate the distance of objects far from the camera that occupy very few pixels in the video frame. Using multiple overlapping cameras can address these limitations. However, existing multi-camera tracking methods are designed to track people, vehicles to address indoor and outdoor surveillance tasks, where the targets are often on the ground. In contrast, small drones must be tracked within a 3D volume that is orders of magnitude larger. As a result

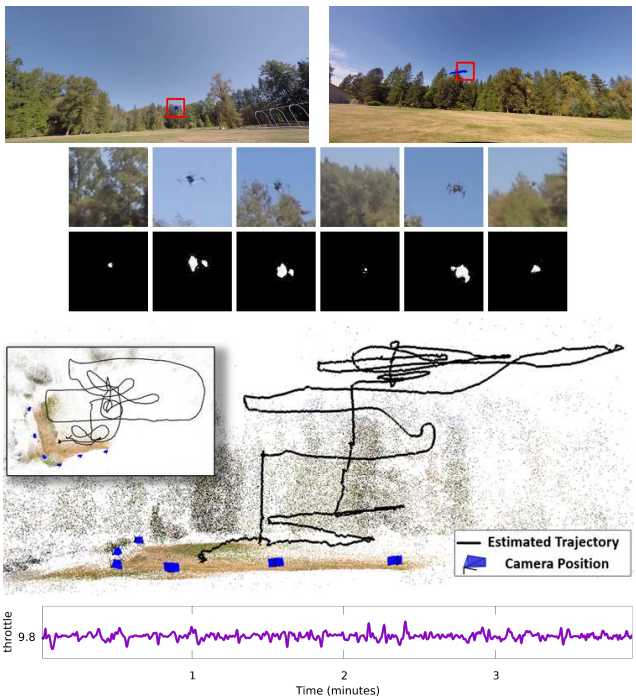


Figure 1: A quadrotor UAV was manually flown to a height of 45 meters above a farm within a $100 \times 50 m^2$ area with six cameras on the ground. [TOP] Two input frames along with the detections and zoomed-in views of the UAV are shown. [MIDDLE] 3D trajectory for a 4 minute flight and camera locations estimated by our method. The inset figure shows the top-down view. [BOTTOM] Estimated throttle signal (one of the control inputs sent to the autopilot).

its image may occupy less then 20 sq. pixels in a 4K UHD resolution video stream. Most existing multi-camera systems also rely on accurate camera calibration that requires someone to collect calibration data by walking around in the scene. A drone detection system is difficult to calibrate in this way because the effective 3D working volume is large and extends high above the ground.

In this paper, we present a new structure from motion (SfM) formulation to recover the 6-dof motion trajectory of a quadrotor observed by multiple fixed cameras as shown in Fig. 1. We model the drone as a single moving point and assume that its underlying flight dynamics model is known. Our contributions are three fold:

- We propose a novel bundle adjustment (BA) procedure that not only optimizes the camera poses and the 3D trajectory, but also regularizes the trajectory using a prior based on the known flight dynamics model.
- This method lets us explicitly infer the underlying control inputs sent to the UAV's autopilot that determined its trajectory. This could provide analytics to drone pilots or enable learning controllers from demonstration [9].
- Finally, our BA procedure uses a new cost function. It is based on traditional image reprojection error but does not depend on explicit data association derived from image correspondences, which is typically considered a prerequisite in classical point-based SfM.

Briefly, the latter lets us keep multiple 2D detections per frame instead of a single one. The true detection is indexed using a per-frame assignment variable. These variables are initialized using a RANSAC-based multi-view triangulation step and further optimized during our bundle adjustment procedure. This makes the estimation less reliant on either perfect single-view tracking or cross-view appearance matching both of which can often be inaccurate.

Our method runs batch optimization over all the videos, which can be viewed as a camera calibration technique that uses the drone as a calibration object. In this work, we assume that the videos are synchronized, have known framerates and the cameras intrinsics and lens parameters are also known whereas an initial guess of the camera poses are available. Finally, we assume only a single drone in the scene. We evaluate our method extensively on data from a realistic quadrotor flight simulator and real videos captured in both indoor and outdoor scenes. We demonstrate that the method is robust to noise and poor initialization and consistently outperforms baseline methods in terms of accuracy.

2. Related work

We are not aware of any existing method that can accurately recover a UAV's 3D trajectory from ground cameras and infer the underlying control inputs that determined its trajectory. However, we review closely related works that address single and multi-camera tracking, dynamic scene reconstruction and constrained bundle adjustment.

Single-View Tracking. This topic has been well studied in computer vision [10]. However, most trackers struggle with tiny objects such as flying birds [11] and tracking multiple tiny targets remains very difficult even with infrared cameras [12]. Some recent works [7, 13] proposed practical sense-and-avoid systems for distant flying objects using passive cameras that can handle low-resolution imagery and moving cameras. However, these methods cannot recover accurate 3D UAV trajectories.

Multi-View Tracking. Synchronized passive multi-camera systems are much more robust at tracking objects within a 3D scene [14, 15]. Traditionally, they have been proposed for understanding human activities, analyzing sports scenes and for indoor, outdoor and traffic surveillance [16, 17, 18, 19, 20, 21]. These methods need calibrated cameras and often assume targets are on the ground, and exploit this fact by proposing efficient optimization techniques such as bipartite graph matching [20, 16, 19], dynamic programming [17, 18] and min-cost network flow [21]. These methods have rarely been used to track tiny objects in large 3D volume, where the aforementioned optimization methods are impractical. Furthermore, conventional calibration methods are unsuitable in large scenes, especially when much of the scene is high above ground level.

Multi-view 3D reconstruction. Synchronized multi-camera systems have also been popular for dynamic scene reconstruction. While most existing techniques require careful pre-calibration, some techniques make it possible to calibrate cameras on the fly [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]. Avidan *et al*. [22] proposed a method for simple linear or conical object motion which was later extended to curved and general planar trajectories [23, 25]. More recent methods have exploited other geometric constraints for joint tracking and camera calibration [24, 26, 27, 28]. However, these methods require accurate feature tracking and matching across views and are not suitable for tiny objects. Sinha *et al*. [29] use silhouettes correspondence and Puwein *et al*. [31] used human pose estimates to calibrate cameras. They do not require cross view feature matching but only work on small scenes with human actors.

Vo *et al*. [32] need accurate feature tracking and matching but can handle unsynchronized and moving cameras. They reconstruct 3D trajectories on the moving targets using motion priors that favor motion with constant velocity or constant acceleration. While our motivation is similar, our physics-based motion dynamics prior is more realistic for UAVs and enables explicit recovery of underlying parameters such as the control inputs given by the pilot.

Constrained Bundle Adjustment. In conventional bundle adjustment [33], camera parameters are optimized along

with the 3D structure which is often represented as a 3D point cloud. Sometimes, regularization is incorporated into bundle adjustment via soft geometric constraints on the 3D structure, including planarity [34], 3D symmetry [35], bound constraints [36] and prior knowledge of 3D shape [37]. These priors can add significant overhead to the Levenberg-Marquardt nonlinear least squares optimization [38]. In our problem setting, the sequential nature of the dynamics-based prior introduces a dependency between all structure variables i.e. those representing sampled 3D points on the trajectory. This leads to a dense Jacobian and makes the nonlinear least square problem infeasible for long trajectories. In this paper, we propose an alternative approach that retains the sparsity structure in regular BA. Our idea is based on generating an intermediate trajectory and then adding soft constraints to the variables associated with 3D points during optimization. We discuss it in Section 4.

3. Problem formulation

Consider the bundle adjustment (BA) problem for point-based SfM [39]. Given image observations $\mathbf{O} = \{\mathbf{o}_{jt}\} : \mathbf{o}_{jt} \in \mathbb{R}^2$ of T 3D points in M static cameras, one seeks to estimate the coordinates of the 3D point $\mathbf{X} = \{\mathbf{x}_t\} : \mathbf{x}_t \in \mathbb{R}^3, t \in [1..T]$ and the camera poses $\mathbf{C} = \{\mathbf{c}_j\}, j \in [1..M]$, where each $\mathbf{c}_j = [\mathbf{R_j}|\mathbf{t_j}] \in \mathbb{R}^{3\times4}$. For our trajectories, let $\mathbf{x}_t$ denote each 3D point on the trajectory at time t. When we have unique observations of $\mathbf{x}_t$ in all the cameras where it is visible (denoted by $\mathbf{o}_{jt}$ for the j-th camera at time t), the problem can be solved by minimizing an objective based on the 2D image reprojection error

$$E_{BA}(\mathbf{C}, \mathbf{X}, \mathbf{O}) = \sum_{t=1}^{T} \sum_{j \in \mathbf{\Omega}_t} \rho(\pi(\mathbf{c}_j, \mathbf{x}_t) - \mathbf{o}_{jt}), \qquad (1)$$

where $\mathbf{\Omega}_t \subseteq \mathbf{C}$ is the set of cameras where the 3D point $\mathbf{x}_t$ is visible at time t, $\pi(\mathbf{c}_j, \mathbf{x}_t) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is the function that projects $\mathbf{x}_t$ into camera $\mathbf{c}_j$ and the function $\rho(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}$ robustly penalizes reprojections of $\mathbf{x}_t$ that deviate from $\mathbf{o}_{jt}$.

Now, let us relax the assumption that unique observations of $\mathbf{x}_t$ are available in the camera views where it is visible. Instead, we will assume that multiple candidate observations are given in each camera at time t, amongst which at most one is the true observation. To handle this situation, we propose using a new objective of the following form:

$$E(\mathbf{C}, \mathbf{X}, \mathbf{O}) = \sum_{t=1}^{T} \sum_{j \in \mathbf{\Omega}_t} \min_k \rho\left(\pi(\mathbf{c}_j, \mathbf{x}_t) - \mathbf{o}_{jtk}\right), \qquad (2)$$

where $\mathbf{o}_{jtk}$ is the k-th amongst K_{tj} candidate 2D observations at time t in camera j. This objective is motivated by the fact that many object detectors naturally produce multiple hypotheses but accurately suppressing all the false detections in a single view can be quite difficult.

So far, we have treated $\mathbf{X}$ as an independent 3D point cloud and ignored the fact that the points lie on a UAV's flight trajectory. Since consecutive points on the trajectory can be predicted from a suitable motion dynamics model (given additional information about the user inputs), we propose using such a regularizer in our BA formulation for higher robustness to erroneous or noisy observations. Our objective function therefore takes the following form:

$$\underset{\mathbf{C},\mathbf{X},\mathbf{\Gamma}}{\arg\min} \left(E(\mathbf{C}, \mathbf{X}, \mathbf{O}) + \lambda R(\mathbf{X}, \mathbf{\Gamma})\right). \qquad (3)$$

Here, the regularizer $R(\mathbf{X}, \mathbf{\Gamma})$ favors trajectories that can be explained by *good* motion models. $\mathbf{\Gamma} : \{\gamma_t\}$ denotes the set of latent variables γ_t for the motion model at time t and λ is a scalar weight. Typically, such regularization, where structure variables $\{\mathbf{x}_t\}$ depend on one another destroys the sparsity in the problem, which is key to efficiently solving large BA instances. However, in our work, we avoid that issue by using regularizers of the following type.

$$R(\mathbf{X}, \mathbf{\Gamma}) = \sum_{t=1}^{T} (\mathbf{x}_t - \hat{\mathbf{x}}_t(\gamma_t))^2, \qquad (4)$$

where $\{\hat{\mathbf{x}}_t\}$ are 3D points predicted by a motion model. As a simple example, one could smooth the trajectory estimate from a previous iteration of BA by setting $\hat{\mathbf{x}} = (g * \mathbf{x})$ with Gaussian kernel g and $(\cdot * \cdot)$ the convolution operator, to favor a smooth trajectory in the current iteration. There are no latent variables for this simple case and so $\mathbf{\Gamma} = \emptyset$. Next, we discuss a more realistic case, involving a flight dynamics model for a quadrotor UAV and based on it derive an appropriate regularizer $R(\mathbf{X}, \mathbf{\Gamma})$.

3.1. Flight dynamics model

$\mathbf{V} : \{\mathbf{v}_t\}$	: velocity	$\mathbf{\Phi} : \{\phi_t\}$	: roll
m	: mass	$\mathbf{\Theta} : \{\theta_t\}$	: pitch
$\mathbf{U} : \{u_t\}$	: throttle	$\mathbf{\Psi} : \{\psi_t\}$	: yaw
$\mathbf{B} : \{\mathbf{b}_t\}$	: angular velocity	$\mathbf{U}_\phi : \{u_\phi(t)\}$	: control inputs
I_x, I_y, I_z	: moments of inertia	$\mathbf{U}_\theta : \{u_\theta(t)\}$	
J_{tp}	: propeller's inertia	$\mathbf{U}_\psi : \{u_\psi(t)\}$	

Table 1: Notations for the physics-based model [40].

While several flight dynamics models for quadrotors are known, we use the one proposed by Webb *et al.* [40, 41]. Table 1 presents the relevant notation. Here, we only include a subset of the equations that are required for deriving the prior or computing the control inputs. $(\mathbf{U}, \mathbf{\Phi}, \mathbf{\Theta})$ denote the thrust and the angles for the complete trajectory. The control inputs $[\mathbf{U}, \mathbf{U}_\phi, \mathbf{U}_\theta, \mathbf{U}_\psi]$ in Table 1 denote the joystick positions in the RC controller. In our case, we need to assume that the yaw angle is zero $\mathbf{\Psi} = 0, \mathbf{U}_\psi = 0$. This

implies that quadrotor is always "looking" in a certain direction, regardless of the motion direction. Since propeller inertia J_{tp} is often very small ($\sim 10^{-4}$), we set it to zero to reduce the model complexity without losing much accuracy. From the basic equations of motion, we have

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_t dt, \quad \mathbf{v}_{t+1} = \mathbf{v}_t + \mathbf{a}_t dt, \tag{5}$$

where $\mathbf{a}_t = (a_x(t), a_y(t), a_z(t))$ is the acceleration of the quadrotor at time t. From the principles of rigid body dynamics, we have the following equation.

$$\begin{bmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \begin{bmatrix} \sin\theta_t \cos\phi_t \\ -\sin\phi_t \\ \cos\theta_t \cos\phi_t \end{bmatrix} \frac{u_t}{m}, \tag{6}$$

where g is the standard gravitational acceleration. From Eq. 6 we obtain the following expression for (ϕ_t, θ_t, u_t):

$$\begin{aligned} u_t &= m\sqrt{a_x(t)^2 + a_y(t)^2 + (a_z(t)+g)^2}, \\ \phi_t &= \arcsin\left(-a_y(t)m/u_t\right), \\ \theta_t &= \arcsin\left((a_x(t)m/u_t)\cos\phi_t\right), \\ & \phi_t \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right), \theta_t \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right) \end{aligned} \tag{7}$$

Here, u_t must be greater than zero which is always satisfied by a quadrotor in flight. Finally, we can estimate the UAV's angular velocity $\mathbf{b}_t = [\mathbf{b}_p(t), \mathbf{b}_q(t), \mathbf{b}_r(t)]$ as follows:

$$\begin{bmatrix} \mathbf{b}_p(t) \\ \mathbf{b}_q(t) \\ \mathbf{b}_r(t) \end{bmatrix} = \begin{bmatrix} (\phi_t - \phi_{t-1})/dt \\ ((\theta_t - \theta_{t-1})/dt)(\cos\phi_t / \sin^2\phi_t) \\ -\theta_t / \sin\phi_t \end{bmatrix}, \tag{8}$$

which can be used to compute control inputs as follows:

$$\begin{aligned} u_\phi(t) &= I_x \tfrac{\mathbf{b}_p(t) - \mathbf{b}_p(t-1)}{dt} - (I_y - I_z)\,\mathbf{b}_q(t)\mathbf{b}_r(t) \\ u_\theta(t) &= I_y \tfrac{\mathbf{b}_q(t) - \mathbf{b}_q(t-1)}{dt} - (I_z - I_x)\,\mathbf{b}_p(t)\mathbf{b}_r(t) \end{aligned}. \tag{9}$$

Next, we describe the flight dynamics based regularizer.

3.2. Flight dynamics prior

In the rest of the paper, we denote $\mathbf{\Gamma} = [\mathbf{\Phi}, \mathbf{\Theta}, \mathbf{U}]^\tau$. These variables will serve as the latent variables in the flight dynamics based prior (we use the term regularizer and prior interchangeably). The dynamics model provides us two transformations denoted $\mathcal{F}$ and $\mathcal{G}$ below.

$$\mathcal{G} : \mathbf{X} \to \mathbf{\Gamma}, \mathcal{F} : \mathbf{\Gamma} \to \mathbf{X}. \tag{10}$$

Equations 5 and 7 are used to obtain $\mathbf{\Gamma}$ from a trajectory estimate. Similarly, the values of $\mathbf{X}$ can be derived from $\mathbf{\Gamma} = (\mathbf{U}, \mathbf{\Phi}, \mathbf{\Theta})$ by recursively using Eqs. 5 and 6. This is equivalent to performing integration with respect to time which uniquely determines the quadrotor's internal state variables (position, velocity, acceleration etc.) up to constant unknown namely the quadrotor's state at time $t = 0$.

Algorithm 1 Bundle Adjustment with motion dynamics

1: **Inputs**:
   - Initial trajectory $\mathbf{X}^0$ and camera parameters $\mathbf{C}^0$
   - Observations in camera views $\mathbf{O}$
2: **Outputs**: Final estimates $(\mathbf{X}^*, \mathbf{C}^*, \mathbf{\Gamma}^*)$ and $(\mathbf{U}_\theta, \mathbf{U}_\phi)$
3: **for** iteration $s \in [1..S]$ **do**
4: $\quad \mathbf{\Gamma}^{s-1} \leftarrow \mathcal{G}(\mathbf{X}^{s-1})$
5: $\quad \hat{\mathbf{\Gamma}}^{s-1} \leftarrow \mathcal{H}(\mathbf{\Gamma}^{s-1})$, $h(\cdot)$ defined in Eq. 11
6: $\quad (\mathbf{X}^s, \mathbf{C}^s, \mathbf{\Gamma}^s) \leftarrow$ run one step of LM to solve Eq. 3
7: **end for**
8: $(\mathbf{X}^*, \mathbf{C}^*, \mathbf{\Gamma}^*) \leftarrow (\mathbf{X}^s, \mathbf{C}^s, \mathbf{\Gamma}^s)$
9: $(\mathbf{U}_\theta, \mathbf{U}_\phi) \leftarrow (\mathbf{\Theta}, \mathbf{\Phi})$ from Eqs. 8 and 9

The general idea of the regularizer will be to add appropriate constraints to the latent variables $(\mathbf{U}, \mathbf{\Phi}, \mathbf{\Theta})$ during the intermediate steps of the bundle adjustment procedure. Below, we use $\mathcal{H}$ to denote such a function.

$$\hat{\mathbf{\Gamma}} = [\hat{\mathbf{\Phi}}, \hat{\mathbf{\Theta}}, \hat{\mathbf{U}}] = \mathcal{H}([\mathbf{\Phi}, \mathbf{\Theta}, \mathbf{U}]) = \mathcal{H}(\mathbf{\Gamma}), \tag{11}$$

In other words, we first recover the latent variables using $\mathcal{F}$ and then apply a suitable amount of smoothing to them to obtain $\hat{\mathbf{\Gamma}}$. Finally, we apply $\mathcal{G}$ on $\hat{\mathbf{\Gamma}}$ to obtain a new trajectory which then serves as a soft constraint during the next iteration of bundle adjustment.

In our experiments, we expect the UAV to move slowly and smoothly. Therefore in our current implementation, we used $\mathcal{H}(\mathbf{\Gamma}) = (g * \mathbf{\Gamma})$, where g denotes a Gaussian kernel. Other more sophisticated forms of $\mathcal{H}(\cdot)$ are worth exploring in the future. We can now write down the expression for the dynamics-based prior or regularizer.

$$R(\mathbf{X}, \mathbf{\Gamma}) = \begin{bmatrix} 1 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}^\intercal \begin{bmatrix} (\mathbf{X} - \mathcal{F}(\mathbf{\Gamma}))^2 \\ \rho_1(\mathbf{\Phi} - \hat{\mathbf{\Phi}}) \\ \rho_2(\mathbf{\Theta} - \hat{\mathbf{\Theta}}) \\ \rho_3(\mathbf{U} - \hat{\mathbf{U}}) \end{bmatrix} \tag{12}$$

4. Optimization

We now describe the steps needed to solve the regularized bundle adjustment problem stated in Eqs. 3 and 12. This is done using an efficient nonlinear least squares solver [42] and suitable initialization. In conventional SfM, the 3D points are treated independently resulting in a sparse problem that can be solved using a sparse Levenberg-Marquardt (LM) method [38]. As we discussed, imposing our dynamics-based regularizer directly would lead to a dense linear systems within each iteration of LM [39].

Here, we describe our proposed technique to impose the regularization indirectly. We will use the trajectory estimate from the previous BA iteration to generate the soft constraints for the dynamics-based prior. Formally these

Algorithm 2 RANSAC-based multi-view triangulation

Inputs :
- Cameras $\mathbf{C} : \{\mathbf{c}_j\}, j \in [1..M]$
- Sets of observations $\mathbf{o}_{jt} : \{\mathbf{o}_{jtk}\}, k \in [1..K_{jt}]$ for camera j at time t

Outputs: 3D Point $\mathbf{x}_t$

for $i \in [1..N]$ **do**
  Randomly pick 2 cameras: $\mathbf{c}_m, \mathbf{c}_n$
  Randomly pick observation $\mathbf{o}_{mtk}$ from $\mathbf{o}_{mt}$
  Randomly pick observation $\mathbf{o}_{ntl}$ from $\mathbf{o}_{nt}$
  $x_i \leftarrow \texttt{triangulate-2view}(\mathbf{o}_{mtk}, \mathbf{c}_m, \mathbf{o}_{ntl}, \mathbf{c}_n)$
  $e(x_i) \leftarrow$ evaluate score for hypothesis x_i using Eq. 2
end for
$\mathbf{x}_t = \arg\min_{x_i}(e(x_i))$

constraints are represented by $\hat{\mathbf{\Gamma}}^{s-1} = \mathcal{H}(\mathbf{\Gamma}^{s-1})$, therefore we rewrite Eq. 12 as:

$$R(\mathbf{X}^s, \mathbf{\Gamma}^s) = \begin{bmatrix} 1 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}^\intercal \begin{bmatrix} (\mathbf{X}^s - \mathcal{F}(\mathbf{\Gamma}^s))^2 \\ \rho_1(\mathbf{\Phi}^s - \hat{\mathbf{\Phi}}^{s-1}) \\ \rho_2(\mathbf{\Theta}^s - \hat{\mathbf{\Theta}}^{s-1}) \\ \rho_3(\mathbf{U}^s - \hat{\mathbf{U}}^{s-1}) \end{bmatrix} \tag{13}$$

where s is the iteration index in the LM technique for solving Eq. 3, $\lambda_i \in \mathbb{R}, i \in \{1,2,3\}$ are scalar weights and $\rho_i(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}, i \in \{1,2,3\}$ are robust cost functions. This allows us to make the points $\mathbf{x}_t^s$ independent of each other in the current iteration. However, there is an indirect dependence on the associated points from the previous iteration $\mathbf{x}_t^{s-1}$. Algorithm 1 depicts the exact steps of our method. So far, we have not discussed initialization of the camera poses and parameters. This is described next.

Trajectory and Camera Pose Initialization. First, we estimate camera poses $\mathbf{C}^0$ using a traditional point-based SfM pipeline [43]. Because the cameras are often far apart and the visible backgrounds do not overlap substantially, we have used an additional camera to recover the initial calibration. We capture a video walking around the scene using an additional hand-held camera. Keyframes were extracted from this video and added to the frames selected from the fixed ground cameras. After running SfM on these frames, we extract the poses for the fixed cameras.

Given $\mathbf{C}^0$, we triangulate the trajectory points from detections obtained from background subtraction. We propose a robust RANSAC-based triangulation method to obtain $\mathbf{X}^0$ (see Algorithm 2). This involves randomly picking a camera pair and triangulating two random detections in these cameras using a fast triangulation routine [44] to obtain a 3D point hypothesis. Amongst all the hypotheses, we select the 3D point that has the lowest total residual error in all the views (as defined in Eq. 2. We use Algorithm 2 on all timesteps to compute the initial trajectory $\mathbf{X}^0$.

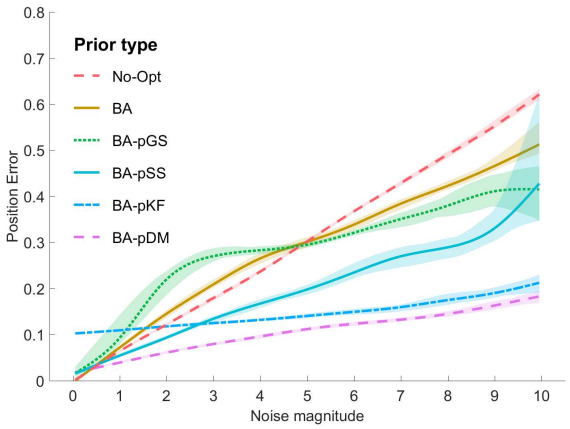


Figure 2: [Evaluation on synthetic data] with respect to different amount of noise added to the initial point locations, camera parameters and point observations in camera views. For each method plot above shows mean and standard deviation of the resulting trajectory from the ground truth across 10 different runs of our algorithm. (best seen in color)

Implementation details. Our method is implemented in C++, using the Ceres non linear least squares minimizer [42]. In order to detect the UAV we have used the OpenCV [45] implementation of Gaussian Mixture Models (GMM)-based background subtraction [46]. Briefly, it creates a GMM model for the background and updates it with each incoming frame. Further, the regions of the image that are not consistent with the model are considered to be foreground and we use them as detections. This, however, leads to large amounts of false positives for the outdoor videos, therefore we processed the resulting detections with a Kalman Filter (KF) [47] with a constant acceleration model. Detections from the resulting tracks are used in our BA optimization. To reduce the amount of false positives we only considered tracks that are longer than 3 time steps. As a result we ended up with 0-8 detections per frame in all the camera views (see Fig. 7).

5. Experiments

We first evaluated our proposed method on synthetic data obtained from a realistic quadrotor flight simulator to analyze accuracy and robustness of our estimated trajectories and control inputs in the presence of image noise, outliers in tracking and errors in the initial camera pose parameters. We also present several results on real data captured indoors and a large outdoor scene where we have access to ground truth trajectory information.

We measure the accuracy of our estimated trajectories by robustly aligning our trajectory estimate to the ground truth trajectory [48]. In the ground truth coordinate system, we then calculate the root mean squared error (RMSE). We

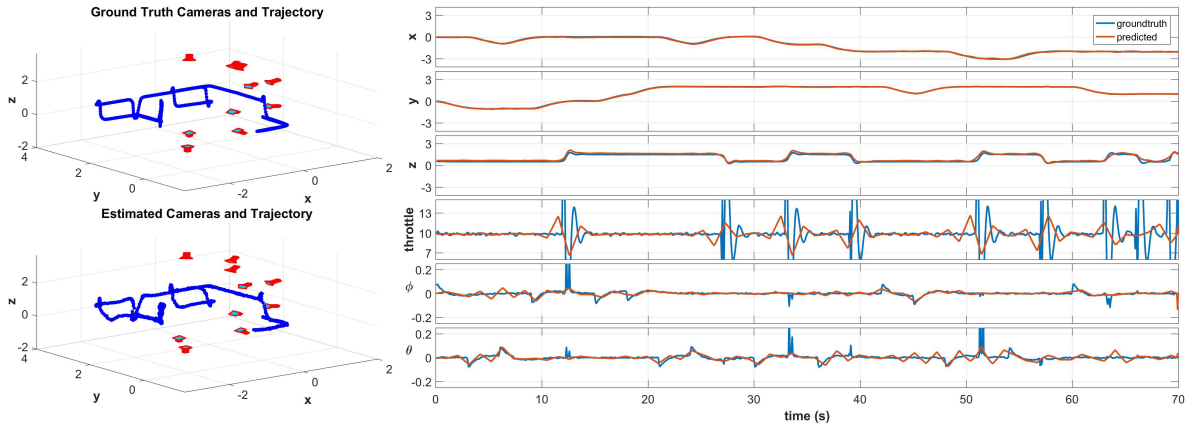


Figure 3: Comparison between the predicted and ground truth positions and internal parameters of the quadrotor. In the left part of the figure you can see the visualization of trajectory point locations. On the right we can see the difference between 3D coordinates ($\mathbf{X}$) of the quadrotor and its internal parameters ($\mathbf{U}, \mathbf{\Phi}, \mathbf{\Theta}$). (best seen in color)

have compared many variants of BA. The suffix '-p' below denotes the type of regularizer (prior).

- **No-Opt**: has no bundle adjustment optimization.
- **BA**: does regular point-based BA without regularization.
- **BA-pGS**: uses a Gaussian smoothing based regularizer.
- **BA-pSS**: uses a spline smoothing based regularizer.
- **BA-pKF**: This denotes the method with a Kalman filter based motion regularizer.
- **BA-pDM**: This denotes our proposed method with the flight dynamics-based regularizer.

Datasets. We evaluated our method on three sets of data. We used an existing quadrotor simulator[1] to generate trajectories with random waypoints. Each simulated trajectory contains 510 points or time steps that was used to simulate 17 seconds of video at 30Hz from 10 cameras. The camera locations were randomly generated around the flight volume and the cameras were oriented towards the center of the flight volume. We generated 100 sequences with varying degree of noise in the camera pose, initial trajectories as well as image noise and outliers.

We evaluated our method on two datasets captured in real scenes – LAB and FARM. In both cases, six GoPro tripod-mounted cameras were pointed in the direction of the flight volume. We recorded video at 2704×1536 pixel resolution and 30 fps. The LAB-SCENE was approximately 6×8 meters and contains an OptiTrack motion capture system [6] that was used to collect the ground truth quadrotor trajectory and camera poses. In this case we flew an off-the-shelf quadrotor (44.5 cm diameter) for 35 seconds. In order to achieve precise tracking accuracy we equipped UAV with a bright LED and an OptiTrack marker, placed close to each other. LED helps us to facilitate the detection process, as now it is narrowed down to searching for the brightest point

[1] github.com/OMARI1988/Quadrotor_MATLAB_simulation

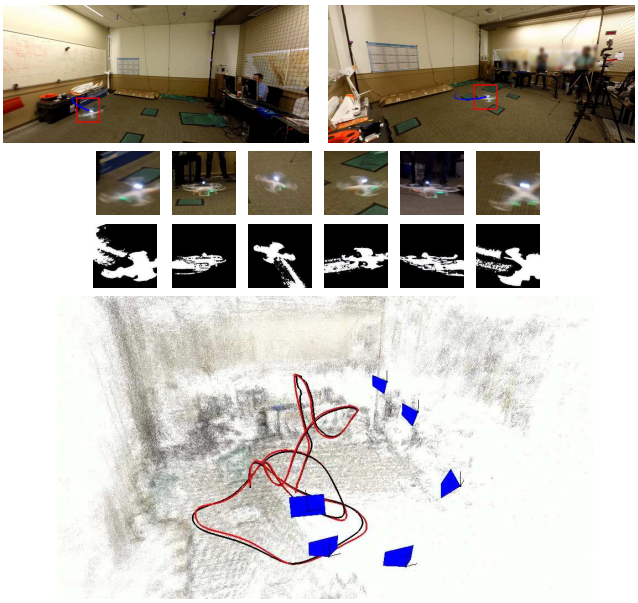
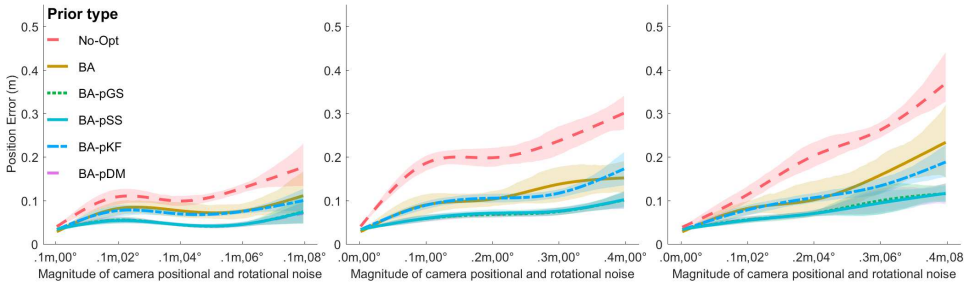


Figure 4: Qualitative results (LAB dataset): [TOP] Two of the six camera viewpoints. [MIDDLE] Zoomed in patches of the trajectory point from all camera views with corresponding background segmentation results. [BOTTOM] A 3D visualization of the estimated trajectory (in black) and the ground truth trajectory (in red). (best seen in color)

in the frame. Despite the simple scenario, people that were moving around in the room and reflectance from the walls were frequently detected as moving objects.

The FARM dataset was captured in a large outdoor scene (see Figure 1) with a bigger UAV (1.5 m in diameter). Footage of a 4-minute flight was recorded using the six cameras, placed $\sim$ 20 meters apart. The videos were captured at the frame-rate of 15 FPS, which results in 3600 trajectory points. We have also collected drone GPS data and used it as ground truth for evaluation.



method	Position Error(m)
No-Opt	.2045±.015
BA	.1165±.014
BA-pGS	.0760±.007
BA-pSS	.0761±.007
BA-pKF	.1141±.009
BA-pDM	**.0757±.007**

Figure 5: Comparing the various methods on the LAB dataset. The initial camera poses are perturbed with increasing position and orientation noise. The plots show the mean and std. dev. of the final error from 10 runs for the various methods. The table lists the error statistics (mean and std. dev.) for all the methods across all the different runs. (best seen in color)

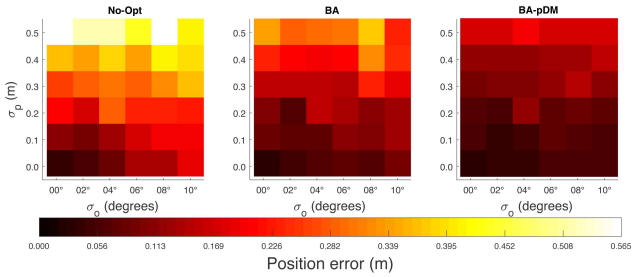


Figure 6: Sensitivity analysis on LAB dataset. [LEFT] Triangulation result. [MIDDLE] Results of standard BA [43] without any prior information. [RIGHT] Results of our method with a dynamics-based prior. Each of these plots show the mean position error across 10 runs, for different amounts of noise added to the camera positions (σ_p) and orientations (σ_o). The amount of error is color-coded and the colorbar below the plots show the correspondence between colors and actual values in meters. (best seen in color)

5.1. Experiments on Synthetic Data

Here we first describe the influence of noise on the performance of our method. We then show that our approach is capable of inferring the underlying control inputs that define the motion of the drone.

Sensitivity Analysis. Fig. 2 summarizes the results on the synthetic dataset. We see that prior information mostly helps to improve reconstruction accuracy. Of all the priors, motion-based ones show a significant improvement over conventional smoothing methods. Overall, the dynamics-based prior is the most accurate. This is probably because the trajectory generated by the quadrotor simulator complies with the same model used to develop the dynamics-based prior (Section 3.2).

Inferring Control Inputs. We have evaluated the quality of prediction of the internal parameters $(\mathbf{U}, \mathbf{\Phi}, \mathbf{\Theta})$ of the quadrotor on the synthetic dataset, as it provides an easy way of collecting ground truth information for these parameters. Fig. 3 summarizes this comparison. In the left part of the figure we can see the predicted and ground truth locations of the 3D trajectory point locations. The right part of the figure depicts the difference between predicted and ground truth (x, y, z) positions of the quadrotor in the environment and the comparison of the predicted internal parameters of the quadrotor with the ground truth.

Fig. 3 shows that 3D locations of the quadrotor were predicted quite well with very small deviations from the ground truth. Regarding the internal parameters, our method is able to predict them very well for the parts of the sequence, where these parameters vary smoothly. This behavior is introduced by Algorithm 1, where we constrain the values of $(\mathbf{U}, \mathbf{\Phi}, \mathbf{\Theta})$ to vary smoothly through time. In our future research we would like to investigate other constraints on the internal quadrotor parameters that will allow us to recover their sharp changes.

5.2. Results on LAB Dataset

Fig. 4 depicts an example of our indoor experiment. Fig. 4(top) depicts sample camera views. Fig. 4(middle) illustrates the cropped and zoomed in patches around the tracked object. Fig. 4(bottom) shows the 3D reconstruction of the trajectory compared to the OptiTrack ground truth.

We also performed a quantitative evaluation of our method. In the LAB dataset, the detections in individual frames were quite reliable. Therefore to perform the noise sensitivity analysis, we have added noise to our initial camera pose estimates before running the optimization.

Figures 5 and 6 show the quantitative evaluation results. Note that as we increase the noise added to the initial camera poses, the triangulation accuracy steadily decreases. This is because the point correspondence between different camera views progressively become inaccurate. The use of standard bundle adjustment improves the reconstruction accuracy. However, the quadrotor dynamics-based prior produces even higher accuracy especially when the noise is quite significant (see Fig. 6).

In Fig. 5, the plots for BA-pDM, BA-pGS and BA-pSS methods are almost indistinguishable from each other, due to very similar performance. This is because in the indoor experiments, the UAV is always clearly visible and not too

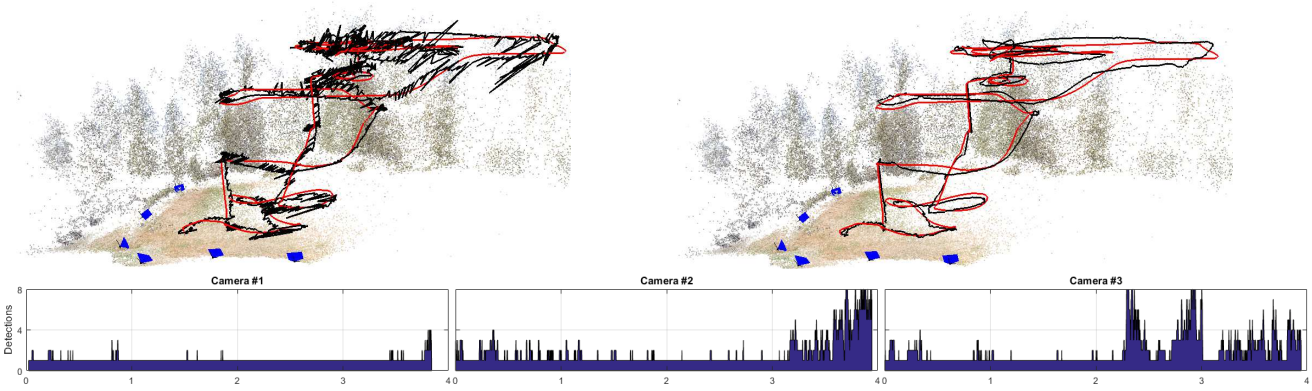


Figure 7: Qualitative Results (FARM dataset): The **black** and **red** curves denote the estimated and ground truth (GPS) trajectory respectively. [LEFT] Initialization trajectory estimate after the triangulation step. [RIGHT] The result obtained after bundle adjustment (BA-pDM). The trajectory is smoother and more accurate compared to the initial trajectory. [BOTTOM] The number of per-frame candidate detections for 3 videos. The middle and right plots show where tracking was difficult.

far from all the cameras. This results in high quality detections with few false positives and allows even BA with simple smoothing-based priors to achieve good accuracy.

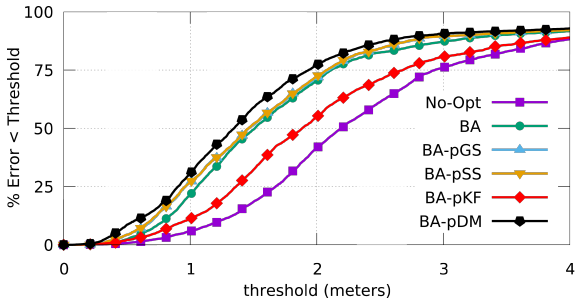
The general trend we noticed in these experiments was that when the initial camera parameters are relatively accurate even simple triangulation can produce quite reliable trajectories. However, larger errors in initial camera pose quickly degrades the performance of both standard triangulation and bundle adjustment methods, whereas our approach is robust to relatively larger amounts of camera pose error due to the effective use of quadrotor dynamics.

5.3. Results on FARM Dataset

Finally we have evaluated our methods on the outdoor dataset. Fig. 1 depicts an example of our outdoor experiment. We can see that compared to the indoor case the drone is much further away from the camera, which results in some challenges not just for the optimization, but also for its detection with background subtraction algorithms. For this experiment we have used Algorithm 2 to set the initial values for the camera parameters and trajectory point locations. Fig. 7(left) shows the result obtained using Algorithm 2, which we use as an initialization. Fig. 7(right) shows the final result of our approach. Fig. 7(bottom) depicts the number of detections per frame for three out of six camera views. We can see that towards the end of the sequence more false positives appear, due to the UAV entering a complicated area of the environment. Nevertheless, our approach allows successfully tracking the drone.

Fig. 8 summarizes the accuracy of the various methods. Similar to the other experiments, the prior information helps to recover a more accurate trajectory. Moreover, using an appropriate dynamics model prior produces the most accurate result amongst all the priors.

The Kalman filter-based prior is not very effective in this case due to noise in the initial trajectory (see Fig. 7(left)).



No-Opt	BA	BA-pGS	BA-pSS	BA-pKF	BA-pDM
2.551	1.910	1.785	1.781	2.275	**1.636**

Figure 8: Comparing various methods on the FARM dataset. The percentage of 3D points with position error less than a threshold is shown for a range of thresholds. The RMSE error in meters for all the methods are reported in the table.

The initialization noise causes the Kalman filter to be unstable and it fails to recover the correct motion model parameters unlike the other priors which are more robust.

6. Conclusion

We have presented a new technique for accurately reconstructing the 3D trajectory of a quadrotor UAV observed from multiple cameras. We have shown that using motion information significantly improves the accuracy of reconstructed trajectory of the object in the 3D environment. Furthermore our method allows inferring the internal parameters of the quadrotor, such as roll, pitch angles and thrust, that is being commanded by the operator. Inferring these parameters has a broad variety of applications, ranging from reinforcement learning to providing analytics for pilots in air race competitions and making feasible research on UAVs in large outdoor and indoor spaces without depending on expensive motion capture systems.

References

[1] J. Busset, F. Perrodin, P. Wellig, B. Ott, K. Heutschi, T. Rühl, and T. Nussbaumer, "Detection and tracking of drones using advanced acoustic cameras," in *SPIE Security+ Defence*. International Society for Optics and Photonics, 2015, pp. 96 470F–96 470F. 1

[2] D. K. Barton, *Radar system analysis and modeling*. Artech House, 2004, vol. 1. 1

[3] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, "Investigating cost-effective rf-based detection of drones," in *Proceedings of the 2Nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ser. DroNet '16, 2016, pp. 17–22. 1

[4] U. Böniger, B. Ott, P. Wellig, U. Aulenbacher, J. Klare, T. Nussbaumer, and Y. Leblebici, "Detection of mini-uavs in the presence of strong topographic relief: a multisensor perspective," in *Proc. SPIE*, 2016. 1

[5] "Vicon Motion Systems," http://www.vicon.com/. 1

[6] "OptiTrack Motion Capture Systems," http://optitrack.com/. 1, 6

[7] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4128–4136. 1, 2

[8] C. Martínez, I. F. Mondragón, M. A. Olivares-Méndez, and P. Campoy, "On-board and ground visual pose estimation techniques for uav control," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 301–320, 2011. 1

[9] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, 2010. 2

[10] W. Luo, J. Xing, X. Zhang, X. Zhao, and T.-K. Kim, "Multiple Object Tracking: A Literature Review," in *arXiv Preprint*, no. 212, 2015. 2

[11] Y. Huang, H. Zheng, H. Ling, E. Blasch, and H. Yang, "A comparative study of object trackers for infrared flying bird tracking," *CoRR*, vol. abs/1601.04386, 2016. 2

[12] M. Betke, D. E. Hirsh, A. Bagchi, N. I. Hristov, N. C. Makris, and T. H. Kunz, "Tracking large variable numbers of objects in clutter," in *Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8. 2

[13] D. Dey, C. Geyer, S. Singh, and M. Digioia, "A cascaded method to detect aircraft in video imagery," *International Journal of Robotics Research*, 2011. 2

[14] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt *et al.*, "A system for video surveillance and monitoring," Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Tech. Rep., 2000. 2

[15] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proc. of the IEEE*, vol. 89, no. 10, pp. 1456–1477, 2001. 2

[16] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L.Van Gool, "Robust Tracking-By-Detection Using a Detector Confidence Particle Filter," in *International Conference on Computer Vision*, 2009, pp. 1515–1522. 2

[17] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple Object Tracking Using K-Shortest Paths Optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 1806–1819, September 2011. 2

[18] A. Andriyenko and K. Schindler, "Multi-Target Tracking by Continuous Energy Minimization," in *Conference on Computer Vision and Pattern Recognition*, 2011. 2

[19] Z. Qin and C. Shelton, "Improving Multi-Target Tracking via Social Grouping," in *Conference on Computer Vision and Pattern Recognition*, 2012. 2

[20] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-Based Multiple-Person Tracking with Partial Occlusion Handling," in *Conference on Computer Vision and Pattern Recognition*, 2012. 2

[21] A. A. Butt and R. T. Collins, "Multi-Target Tracking by Lagrangian Relaxation to Min-Cost Network Flow," in *Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1846–1853. 2

[22] S. Avidan and A. Shashua, "Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 348–357, 2000. 2

[23] J. Kaminski and M. Teicher, "A General Framework for Trajectory Triangulation," *Journal of Mathematical Imaging and Vision*, vol. 21, no. 1, pp. 27–41, 2004. 2

[24] D. Wedge, D. Huynh, and P. Kovesi, "Motion Guided Video Sequence Synchronization," in *Asian Conference on Computer Vision*, 2006, pp. 832–841. 2

[25] C. Yuan and G. Medioni, "3D Reconstruction of background and objects moving on ground plane viewed from a moving camera," in *Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2261–2268. 2

[26] Y. Caspi, D. Simakov, and M. Irani, "Feature-Based Sequence-to-Sequence Matching," *International Journal of Computer Vision*, vol. 68, no. 1, pp. 53–64, 2006. 2

[27] F. Padua, R. Carceroni, G. Santos, and K. Kutulakos, "Linear Sequence-to-Sequence Alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 304–320, 2010. 2

[28] J. Valmadre and S. Lucey, "General trajectory prior for non-rigid reconstruction," in *Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1394–1401. 2

[29] S. N. Sinha and M. Pollefeys, "Camera network calibration and synchronization from silhouettes in archived video," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 266–283, 2010. 2

[30] S. N. Sinha and M. Pollefeys, "Visual-hull reconstruction from uncalibrated and unsynchronized video streams," in *3D Data Processing, Visualization and Transmission*. IEEE, 2004, pp. 349–356. 2

[31] J. Puwein, L. Ballan, R. Ziegler, and M. Pollefeys, "Joint camera pose estimation and 3d human pose estimation in a multi-camera setup," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 473–487. 2

[32] M. Vo, S. Narasimhan, and Y. Sheikh, "Spatiotemporal Bundle Adjustment for Dynamic 3D Reconstruction," *Conference on Computer Vision and Pattern Recognition*, 2016. 2

[33] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, *Bundle Adjustment — A Modern Synthesis*. Springer Berlin Heidelberg, 2000, pp. 298–372. 2

[34] A. Bartoli and P. Sturm, "Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene," *International Journal of Computer Vision*, vol. 52, no. 1, pp. 45–64, 2003. 3

[35] A. Cohen, C. Zach, S. N. Sinha, and M. Pollefeys, "Discovering and exploiting 3d symmetries in structure from motion," in *Conference on Computer Vision and Pattern Recognition*, 2012. 3

[36] Y. Gong, D. Meng, and E. J. Seibel, "Bound constrained bundle adjustment for reliable 3d reconstruction," *Opt. Express*, vol. 23, no. 8, pp. 10 771–10 785, Apr 2015. 3

[37] P. Fua, "Regularized bundle-adjustment to model heads from image sequences without calibration data," *International Journal of Computer Vision*, vol. 38, no. 2, pp. 153–171, 2000. 3

[38] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963. 3, 4

[39] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 3, 4

[40] D. Webb and J. van den Berg, "Kinodynamic RRT*: Optimal Motion Planning for Systems with Linear Differential Constraints," *arXiv Preprint*, vol. abs/1205.5088, 2012. 3

[41] J. Li and Y. Li, "Dynamic analysis and PID control for a quadrotor," in *2011 IEEE International Conference on Mechatronics and Automation*, August 2011, pp. 573–578. 3

[42] S. Agarwal, K. Mierle, and Others, "Ceres Solver," http://ceres-solver.org. 4, 5

[43] S. Agarwal, N. Snavely, S. Seitz, and R. Szeliski, "Bundle Adjustment in the Large," in *European Conference on Computer Vision*, 2010, pp. 29–42. 5, 7

[44] P. Lindstrom, "Triangulation made easy," in *Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1554–1561. 5

[45] "Open Source Computer Vision Library," http://opencv.org. 5

[46] P. KaewTraKulPong and R. Bowden, *An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection*. Springer US, 2002. 5

[47] G. Welch and G. Bishop, "An Introduction to Kalman Filter," Department of Computer Science, University of North Carolina, Technical Report, 1995. 5

[48] B. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987. 5