WAGENINGEN UNIVERSITY
WAGENINGEN UR

Environmental Sciences
Laboratory of Geo-Information Science and Remote Sensing
MMRS - Multimodal Remote Sensing Group

ETH zürich

IGP
Institute of Geodesy and Photogrammetry

Unmanned Aerial System Trajectory Reconstruction using Un-synchronized, Multi-Camera Networks

Jesse Murray

Thesis Supervisors

Supervisor:
Prof.dr D. Tuia
Professor of Geo-information Science and Remote Sensing
Wageningen University

Supervisor:
Dr. C. Albl
Assistant Professor, Institute of Geodesy and Photogrammetry
ETH, Zürich

Unmanned Aerial System Trajectory Reconstruction using Un-synchronized, Multi-Camera Networks

Jesse Murray

Thesis

submitted in partial fulfilment of the requirements for the degree of Master of Science at Wageningen University

Wageningen, NL (2019)

Abstract

Advancements in unmanned aerial vehicle (UAV) technology have made UAVs easier to operate, more affordable, and capable of performing a broader range of tasks. This trend has led to a significant expansion of UAV applications in society. However, the use of UAVs for tasks that require high precision measurements is still an area of development due to challenges in determining the precise location at which a UAV collects the desired data. While on-board, high-precision Global Navigation Satellite System (GNSS) sensors can be employed to provide such information, this adds significant cost and operational complexity to the operation of a UAV. Moreover, such GNSS sensors cannot be used in indoor or complex urban or industrial environments where the GNSS signals become unreliable or completely fail. The objective of this thesis is to produce a highly precise localization of a moving UAV using a network of static, non time-synchronized video surveillance cameras. The pipeline uses pixel-wise detection sequences of a moving UAV captured in the image space of each network camera. A robust solver is implemented to simultaneously estimate the geometry of the camera network and the relative temporal offsets between the video sequences of the moving drone. An incremental bundle adjustment procedure is used to jointly optimize the relative camera geometries and UAV trajectory. The bundle adjustment procedure integrates rolling shutter correction and several motion constraint methods including B-splines and physical motion priors. Experiments using various models of the bundle adjustment procedure were conducted on synthetic and outdoor video data to determine the most robust and precise configuration. The results of the reconstructed trajectories obtained on the outdoor data show that the implemented approach can obtain decimeter accuracy as compared to Real-time kinematic (RTK) ground truth measurements collected during the flight.

Contents

List of Figures

List of Tables

1 Introduction

This chapter is an introduction to the main ideas covered in this thesis. It gives a brief overview of the purpose and challenges of the thesis. The structure of the thesis is introduced at the end of this chapter.

1.1 Motivation

Digital video surveillance networks mounted on urban infrastructure, unmanned aerial vehicles (UAV), cars, smart phones, and other devices have established a ubiquitous presence in society due to rapid advancements in performance, ease of use, and reduction in cost, which have improved overall utility. The proliferation of digital video surveillance networks has in turn led to an increase in the development of a broad range of applications that utilize digital surveillance data to provide real-time, dynamic information on traffic, security, infrastructure, weather, agriculture, forests, and other physical spaces (Collins et al., 2000) (Frey et al., 2018) (Rosca et al., 2018). As these networks become more widely deployed and the data they produce becomes richer, efforts to enhance these applications through the integration of automated object detection and tracking, as well as 3D surface reconstruction, have been gaining interest (López-Araquistain et al., 2017) (Zhang et al., 2017) (Liang et al., 2018).

Computer vision and pattern recognition approaches have been widely researched to produce efficient and accurate results in the automatic detection, tracking, and 3D geometry reconstruction of objects within image and video scenes (Jebara et al., 1999) (Ess et al., 2010). However, maintaining this accuracy becomes challenging as image scenes become more complex (i.e. multiple moving objects and occlusions present) and multiple cameras with low scene overlap and large physical spacing (large baseline) are introduced. Recently, novel deep learning approaches have made great progress in addressing these challenges (Feichtenhofer et al., 2017),(Keuper et al., 2018). However, such approaches require carefully calibrated and time synchronized camera setups to produce reliable results (Patino et al., 2016) (Tang et al., 2019) (Feichtenhofer et al., 2017) (Ristani et al., 2016). Such limitations add

significant cost and effort to the data collection process and limit their ability to be extended to non-laboratory settings where camera synchronization is not possible or the camera calibration details are unknown. Additionally, feature localization methods applied to these approaches still lack the precision required for use in 3D geometry reconstruction (Dai et al., 2018) where accurate information of the camera location/orientation at the time of image acquisition plays a critical role.

The goal of this thesis is to develop a framework to reconstruct the 3D trajectory of moving objects detected in the data of multi-camera video systems. The framework will be designed flexibly to be applied to a broad range of applications, and utilize the detected 2D trajectory of a given moving target across all cameras to jointly optimize the geometric calibration (orientation and location) and the time synchronization of input image sequences to obtain the reconstructed 3D trajectory of the target.

1.2 Problem Formulation

1.2.1 Research Questions

In this thesis, the proposed synchronization and geometry calibration approach to reconstruct the 3D trajectory of moving objects captured by a multi-camera video system will be implemented to answer the following research questions:

1. To what extent does the approach proposed by (Albl et al., 2017) help to improve the 3D trajectory reconstruction accuracy and synchronization capacity as compared to previous state-of-the-art reconstruction results?

2. What effect do different optimization constraints (spline, linear motion approximation) have on the accuracy of the proposed trajectory reconstruction framework for different scene scenarios (i.e. planar vs 3D motion)?

3. What is the best approach (pair-wise synchronization vs. iterative sequential synchronization) to optimize the time synchronization and geometric calibration of a camera network of three or more cameras?

1.3 Thesis Structure

The remainder of this thesis is organized as follows: Chapter 2 summarizes previous work relating to video synchronization, multi-view 3D reconstruction and bundle adjustment. Chapter 3 gives an overview of the concepts that will be utilized in this thesis to develop the reconstruction pipeline. Chapter 4 provides a detailed overview of the implementation process of the concepts described in Chapter 3. Finally, the experiments and results developed in the trajectory reconstruction pipeline are displayed in Chapter 5 and further assessed in Chapter 6. A conclusion of the completed work is provided in Chapter 7.

2 Related Work

Interest in the ability to model, map, and visualize the world in three dimensional (3D) space for applications in autonomous navigation, security, augmented and virtual reality (Liang et al., 2018) (Brahmbhatt et al., 2018), and 3D surface reconstruction (Gindraux et al., 2017) (He et al., 2018) has increased recently due to the ease and relatively accessible cost of collecting the optical data necessary for such tasks. However, these efforts become difficult to implement in complex urban environments where the tracked targets undergo significant occlusions and travel over large distances thus making the task of accurately tracking and reconstructing an object's trajectory non-trivial. Implementing such tasks within large 3D spatial volumes (for example, reconstructing the flight trajectory of a UAV, airplane, or bird) and across multiple cameras introduces further complexity as one must then consider tracking objects and synchronizing events across multiple cameras and frame sequences. This thesis will investigate mechanisms to address such challenges in an attempt to reconstruct the 3D trajectory of objects viewed in a multiple camera network where the image sequences are not time-synchronized and the camera location and orientation (geometric camera calibration) information is unknown.

While few works exist combining 2D object tracking with 3D trajectory reconstruction from multiple un-calibrated and asynchronous cameras, the individual components associated with the task are well studied.

2.1 Multi-View object detection and tracking

Object detection and tracking between image sequences has received a great deal of research interest within the computer vision community Luo et al., 2014. Some of the earliest and most widely implemented methods include background subtraction in which the static parts of a scene are modeled with a Gaussian mixture model (GMM), thus enabling successive scenes to be subtracted from one another and taking the difference as the moving object within each scene Haque et al., 2008 Mukherjee et al., 2014. Such methods are prone to producing high numbers of false

positive object proposals that require further processing approaches such as template matching and applying thresholds to accept only object proposals that are consistent across several scenes Dong et al., 2018 Rozantsev et al., 2017. Another commonly employed multi-object tracking method is optical flow Beauchemin et al., 1995, which computes the differential image intensities between successive images and associating moving objects with pixel neighborhoods in the image where this differential is high. Other approaches employ key point detection methods (SIFT/SURF) and track these key points across images sequences using methods such as KLT Buddubariki et al., 2015. Recently, many deep-learning based approaches have been developed that are more robust and accurate in tracking multiple objects across multiple views Baque et al., 2017 Chavdarova et al., 2017. However, these approaches require significant amounts of training data, the majority of which is tailored to specific applications, e.g. pedestrian, car tracking. Given the variety of approaches available to achieve object tracking, the goal of this thesis is to build sufficient flexibility and robustness into the framework developed to ensure it can produce sufficient results from the tracking approach that best suits the use case.

2.2 3D trajectory reconstruction

One of the most well studied tasks within photogrammetry and computer vision is the recovery of 3D information from the 2D information captured in multiple images. For static objects captured in images this is a well defined problem for which standard solutions exist to accurately derive 3D point coordinates given a set of camera projection matrices and 2D image points Hartley et al., 2003. The process, known as triangulation, becomes non-trivial when one considers moving objects and cameras that need to be calibrated not only spatially but also in time. For such scenarios the parameters that define the camera matrix and time offsets need to be accounted for jointly in the optimization process. Avidan et al. Avidan et al., 2000 were some of the earliest to introduce motion constraints to provide a solution for the trajectory triangulation problem. They assumed that correctly triangulated points would fall on a path approximated by linear motion between frames as an optimization constraint to the problem. Rozantsev et al., 2017 addressed the trajectory triangulation of moving objects in synchronized multi-view sequences through injecting the 2D point detections of moving objects into a physical motion model to constrain their optimization approach to derive

3D trajectories. Vo et al., 2016a addresses the task of trajectory triangulation in non-synchronized sequences of moving cameras through assuming a constant velocity motion constraint in their optimization process. While the above approaches produce good synchronization and trajectory reconstruction results for small time offsets (< 5 image frames), they rely on good initial estimates of the time synchronization offset to obtain model convergence and are this limited in the length of offset that they can synchronize. Additionally, both methods use a maximal number of trajectory point correspondences between camera pairs to estimate the orientation and location (camera geometry), making them prone to potential noise in the trajectory signal and resulting in less precise 3D reconstruction results.

In this thesis, the approach proposed by Albl et al., 2017 will be employed to improve upon both of the limitations described above. Albl et al., 2017 showed that a minimal algebraic solver along with random sampling and consensus (RANSAC) inspired by Fischler et al., 1981 could be used to jointly estimate the camera geometry and time offset between two video feeds. Using the minimum number of points necessary to solve for the camera geometry and time-offset limits the effect that noise in the 2D input object trajectory can have on the optimization process thereby enabling for a more stable estimation. Combining iteratively the solutions of the minimal solver along the 2D trajectory further enables for time offsets of up to hundreds of frames to be solved for. Within this thesis the approach proposed by Albl et al., 2017 will be implemented and integrated into a 3D trajectory reconstruction framework. The established framework will be compared to results obtained by previously published state of the art solutions Noguchi et al., 2007, Nischt et al., 2009, and Rozantsev et al., 2017.

3 Theoretical Foundations

This chapter provides an overview of the fundamental concepts upon which the 3D trajectory reconstruction procedure is based.

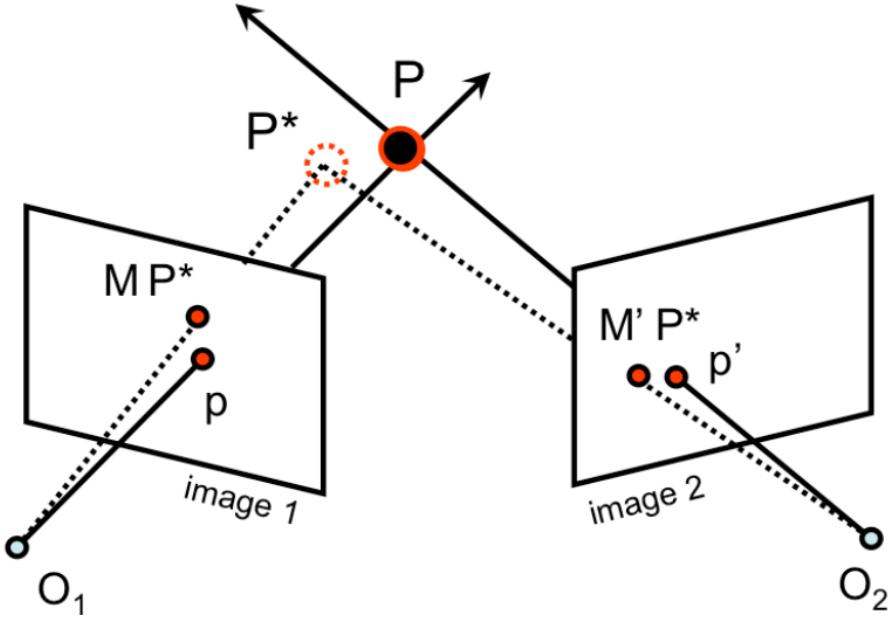


Figure 3.1: The triangulation of a point P in 3D space is accomplished by minimizing the reprojection error between points p and p' in the image space given a camera projection matrix M.

3.1 3D Trajectory Reconstruction

As illustrated in Figure 3.1 from (*Stanford CS231A Course Notes, 2019*), the basic relationship by which a point p in a 2D image space is related to a point, P in 3D space is formulated as follows:

$$p = \mathsf{M}P \tag{3.1}$$

where M is the camera projection matrix that defines the internal and external op-

tical properties of the camera. When P is viewed by two cameras with overlapping views and different optical centers, one can solve 3.1 for M using a least squares optimization approach on the error between the true 3D point, P and the re-projected point, P^* given corresponding 2D points, p and p'. However the scenario shown in Figure 3.1 assumes that the corresponding 2D points are viewed at the exact same time which is rarely the case in most real world settings. As illustrated in Figure 3.2 from (Albl et al., 2017), when two cameras capture images a-synchronously, objects are captured at different times and therefore at different locations in space. Determining the temporal offset between the respective camera frames as well as the sub-frame, corresponding 2D locations of objects projected into each camera image are then required to solve the optimization problem described above. The considerations required to determine this time offset and continuous object location in order to accurately reconstruct an object's movement in 3D space forms the basis of this thesis.

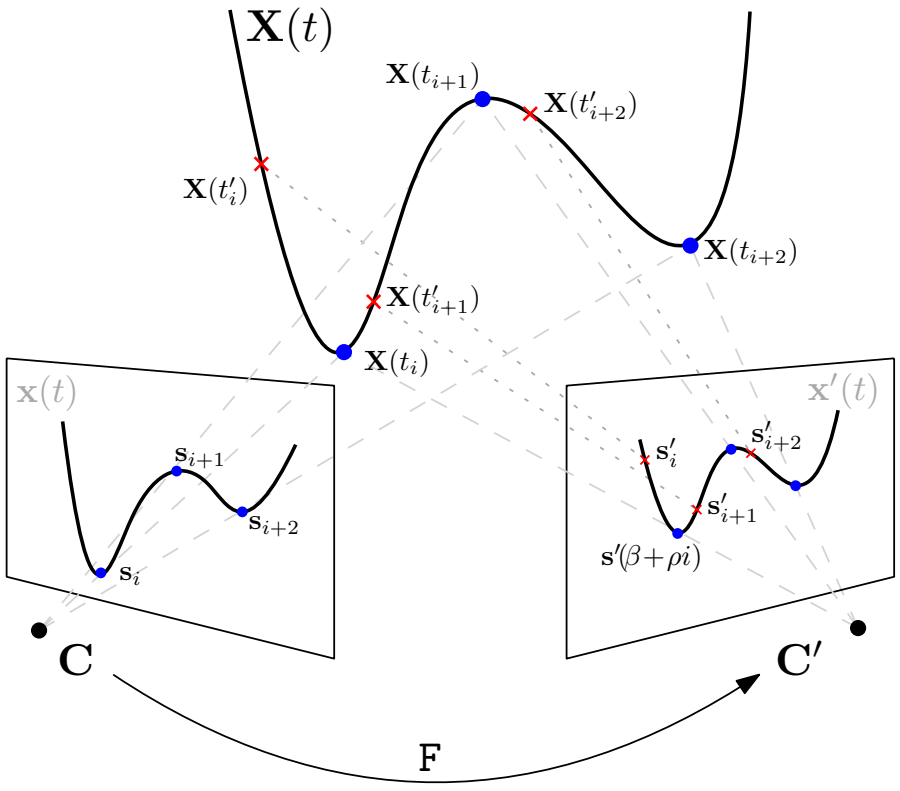


Figure 3.2: An object imaged by two cameras at different points in time will be in different locations in space within each image. This spatio-temporal offset must be determined in order to triangulate the object's location in 3D space.

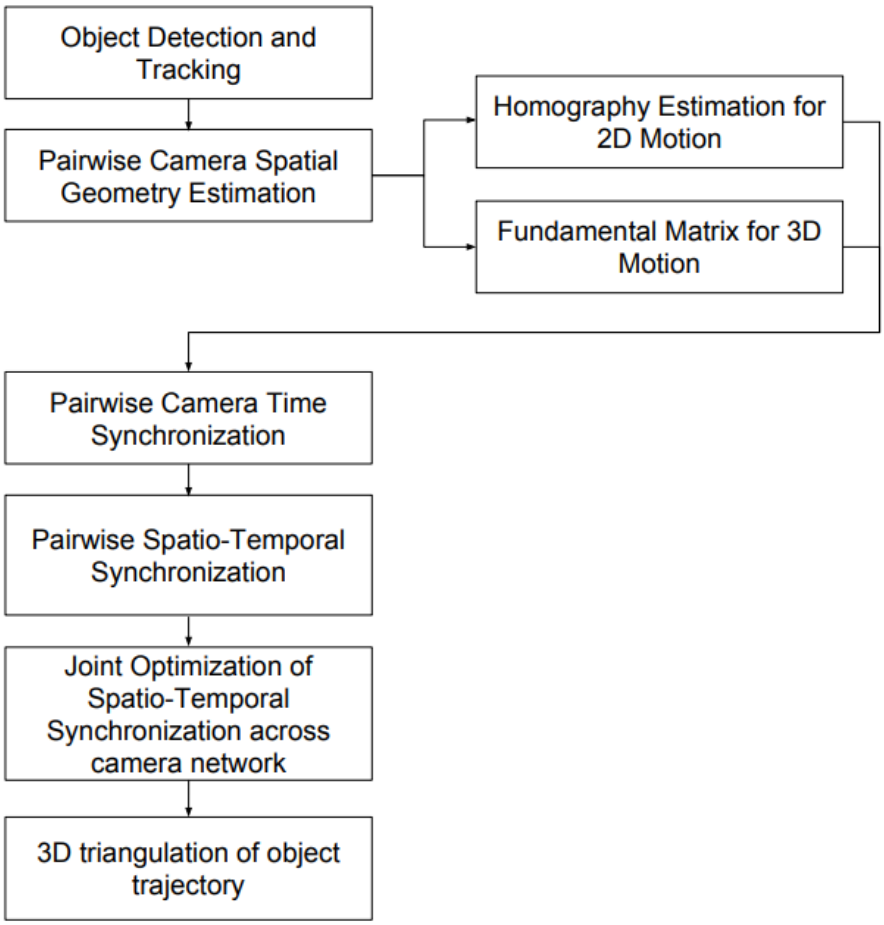


Figure 3.3: Spatio-Temporal Calibration and Triangulation Pipeline

3.1.1 3D Reconstruction Framework

The components of the proposed 3D reconstruction framework that are implemented within this thesis are shown in Figure 3.3. The implementation steps involved for each component are further described in the following sections.

3.1.2 Object Detection and Tracking

As defined in 2, several methods ranging from background subtraction, which is used to remove static pixels from an image thereby simplifying the detection task, to optical flow, which utilizes changes in the pixel histograms to locate moving objects in the image space are utilized in the object detection framework to provide the basic detection results required for the reconstruction process. The reconstruc-

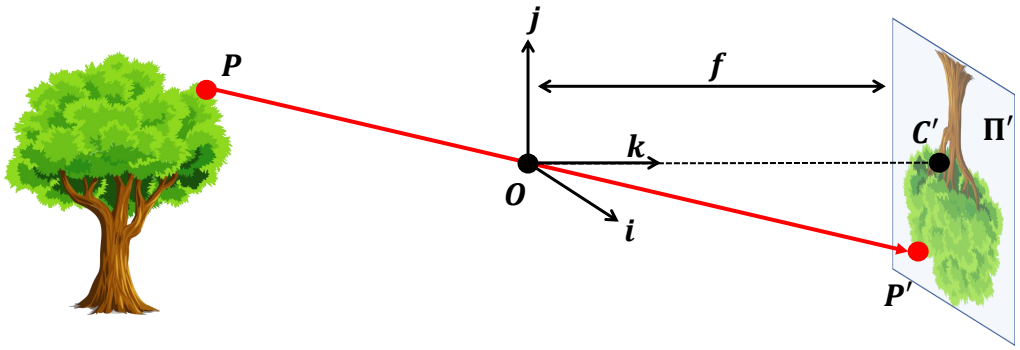


Figure 3.4: Perspective camera model

tion pipeline will be designed to be robust to noisy and false detections through implementing RANSAC and regularization terms so as to reduce the dependency of the results on the accuracy of the detections.

3.1.3 Camera Calibration

Camera calibration is the intrinsic and extrinsic optical parameters that define how an object is projected from the world space into the image space of the camera sensor. The intrinsic parameters define the parameters that govern the projection of objects in the 3D world space into the camera's 2D image space, while the extrinsic parameters describe the camera pose (location and orientation) in the 3D world space. The standard pinhole camera model, as shown in 3.4 (*Stanford CS231A Course Notes, 2019*), defines the camera calibration parameters through the following equations.

$$\lambda x = K[R|t]X \tag{3.2}$$

$$P = K[R|t] \tag{3.3}$$

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$\begin{aligned} x_{distorted} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{distorted} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \tag{3.5}$$

Equation 3.2 defines a point X in the 3D world space and x defines the coordinates of its projection into the image space both in homogeneous coordinates. P in 3.3

defines the 3×4 camera projection matrix. P is comprised of the calibration matrix which describes the *intrinsic* camera parameters K and the *extrinsic* rotation (R) and translation (t) defined above. Within the calibration matrix K, f_x and f_y define the focal length in pixels, s is the skew coefficient (assumed to be zero for modern cameras), and c_x, c_y are the optical center in pixels.

Radial Distortion

Radial distortion is a phenomena in which light rays entering the camera lens bend more at the edge of the lens than at the center due to the large angle of incidence at lens edge. This causes a warping effect in the projected image that needs to be corrected for to ensure an accurate projection model. The standard radial distortion model as described in 3.5, is applied in this work to determine the distortion coefficients k_1 - k_3

Chessboard Calibration Estimation

In this work the standard chessboard calibration procedure in *OpenCV* is employed to determine the *intrinsic* and distortion coefficients that describe each network camera. The method implemented in *OpenCV* as proposed by (Zhang et al., 2013) uses key-points detected on a planar calibration surface and relates them to 3D coordinates in the world space to estimate the camera *intrinsic* and *extrinsic* parameters by means of a closed form solution. Non-linear optimization is then performed over all camera parameters and distortion coefficients to obtain the final calibration solution. The details of the optimization procedure are described below.

3.1.4 Pairwise 3D point Triangulation

The 3D points (X_i) corresponding to a set of detected 2D points (x_i) viewed by each calibrated camera pair can be determined through a process of triangulation using the relationship described in Equation 3.6 and reformulated below:

$$x_i = PX_i \tag{3.6}$$

Applying the definition of the cross product simplifies Equation 3.6 to:

$$x_i \times PX_i = 0 \tag{3.7}$$

$$x_i \times PX_i = 0 \tag{3.8}$$

The result of this cross product operation allows us to define the following constraints:

$$\begin{aligned} x(P_3X) - (P_1X) &= 0 \\ y(P_3X) - (P_2X) &= 0 \\ x(P_2X) - y(P_1X) &= 0 \end{aligned} \tag{3.9}$$

The 3D point corresponding to a set of 2D points can be linearly approximated by applying SVD to a linear equation in the form $AP = 0$ where A is a matrix of constraints defined by Equation 3.9 for two observed sets of points $p_i^1(x, y)$ and $p_i^2(x, y)$ as:

$$A = \begin{bmatrix} x^1P_3^1 - P_1 \\ y^1P_3^1 - P_2 \\ x^2P_3^2 - P_1^2 \\ y^2P_3^2 - P_2^2 \end{bmatrix} \tag{3.10}$$

The linearly estimated 3D points, $\hat{X}$, determined in Equation 3.10 can be refined with a non-linear least squares approach as described above. Using the reprojection error between the 2D point determined by projecting $\hat{X}$ with the projection matrix $\mathsf{P} = K[RT]$ and the detected points, $\hat{X}$ can be optimized using the Gauss-Newton or Levenberg-Marquardt algorithm to perform a non-linear least squares minimization approach with the following objective:

$$\min_{\hat{X}} \|P\hat{X} - x\|^2 + \|P'\hat{X} - x'\|^2 \tag{3.11}$$

3.1.5 Epipolar Geometry

As discussed in 3.1, the relationship between a 3D point in a given scene and the corresponding 2D point observed in an image is defined by a camera's projection matrix, P which defines a camera's optical and geometric properties. For the case that P is unknown, it can be solved for using the relative projections of an observed point within two images. The relationship defining the projection properties of two cameras viewing the same scene is defined by the *epipolar geometry constraint* for objects situated in 3D space (e.g. flying objects) and by the *homography constraint* for objects constrained within a plane (e.g. pedestrians, cars) (Hartley et al., 2003).

As illustrated in Figure 3.2, considering an observed point x for a given camera C, we would like to solve for the corresponding point x' in a camera C'. The $fundamental\ matrix$, F, which encodes both the optical and geometric (relative translation and rotation) parameters between a pair of cameras can be used to relate x to x' by

$$x^T \mathsf{F} x' = 0 \tag{3.12}$$

For objects positioned in a 2D plane, this relation can be formulated by the *homography* constraint, H, by:

$$\lambda x = \mathsf{H} \cdot x' \tag{3.13}$$

where λ is the scale factor encoding the image pixel size. The matrices defined by F and H can then be used in 3.1 to determine the desired 3D information of a point X viewed by the two cameras.

Methods to estimate both F and H as described in (Hartley et al., 2003) are implemented within this thesis.

Essential Matrix The fundamental matrix can be defined for image coordinates that are normalized by the calibration matrix. This matrix is known as the

essential matrix E. The essential matrix is particularly useful for multi-camera geometry reconstruction as the image coordinates obtained from it are dimensionless therefore allowing cameras with differing optical properties to be related to one another. The essential matrix can be related to the fundamental matrix through the following relations.

$$\hat{x}'^{\top} E \hat{x} = 0 \tag{3.14}$$

$$E = K'^{\top} F K \tag{3.15}$$

Another useful property of the essential matrix is that the relative camera pose from two projected points can be extracted from it. Assuming the first camera projection matrix is identity $[I \mid 0]$, the rotation and translation of the second camera can be derived from E up to a four-fold ambiguity, i.e. four possible solutions. The correct solution is the one that projects the image scene into the positive image space.

3.1.6 Perspective-n-Point

Perspective-n-Point (or PnP) is a standard geometry problem to estimate the relative *rotation* and *translation* (described in 3.1.3) of a calibrated camera from a set of n 3D↔2D point correspondences.

The method proposed by (Lepetit et al., 2009) known as EP3P is implemented in this work through (Bradski, 2000) to determine the pose of each camera added to the camera network within the incremental bundle adjustment procedure. Outlier solutions from this method are reduced through employing RANSAC 3.2 and the pose is further jointly optimized in the bundle adjustment procedure to minimize the reprojection error produced by the estimated pose of each camera.

3.1.7 Pairwise Camera time Synchronization

The point correspondences, x and x' defined above hold for image sequences where the respective cameras capture images synchronously in time. However, as portrayed in 3.2, when two cameras capture images asynchronously, objects are captured at different times and thus different locations in space. In this case, the process of solving for 3.19 or 3.20, requires the relative corresponding points x and

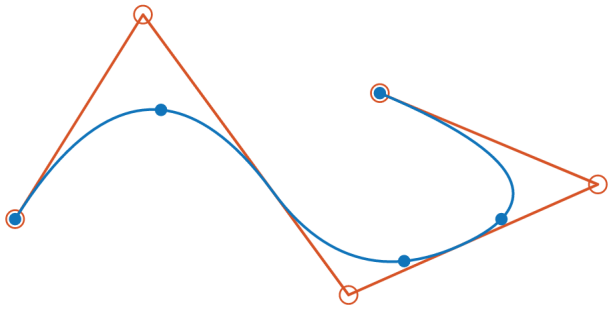


Figure 3.5: An example of B-spline [1]with control points (red) and the parameterized curve fit to the knot points (blue)

x' used in **??** and **??** to be estimated with respect to one another. Both the spatial and temporal offsets that exist between the image sequences must be determined to make these estimates. The approach that will be taken to determine these offsets is described below.

3.1.8 2D Spatial Motion Models

As illustrated in Figure 3.2, the corresponding 2D points between two a-synchronous image sequences are not guaranteed to be captured in a frame of both sequences. Therefore the corresponding points must be estimated by approximating an object's location in one camera sequence as a continuous function of the sequence image frames. Within this thesis two methods will be compared to obtain the continuous object location between image frames:

- **Linear Interpolation**
  As motion of physical objects is relatively constant within the timescale of a single video frame, linear interpolation will be used to obtain the sub-frame location of an object between the two successive image frames.

- **Spline Fitting**
  As in (Nischt et al., 2009) and (Rozantsev et al., 2017) the position of an object detected in each camera sequence frame will also be modeled as a continuous function by fitting a spline to the detected object points in the image sequence. Spline functions are widely used in computer vision for their ability to describe complex shapes with a small number of parameters.

[1]Source: bsplines.org

In this work, a B-spline representation is applied to interpolate virtual correspondences between cameras and to describe the resulting triangulated trajectories at each step of the bundle adjustment.

- **Physical motion prior regularization**
  As described in (Vo et al., 2016b), this work exploits the fact that the motion of the UAV is constrained by fundamental Newtonian mechanics and thus the trajectory must follow a path that minimizes the cost of these properties. Within this work the least force and least kinetic energy motion priors are included as a last step within the trajectory optimization procedure to ensure that the result conforms as best as possible to these physical laws. The least kinetic energy motion prior 3.17 promotes results that maintain constant velocity between successive image frames while the least force motion prior 3.16 promotes results that maintain constant acceleration.

$$\int mv(t)^2 dt \tag{3.16}$$

$$\int ma(t)^2 dt \tag{3.17}$$

Relative Time Estimation

The motion models described in 3.1.8 provide the object location as a function of the image frame. However, as illustrated in Figure 3.2, corresponding frame indices within an image sequence is unknown within in an asynchronous camera pair. Thus in order to determine the corresponding frames/subframes between a non synchronous camera pair, the frame offset between the cameras must be determined as a function of global time. The frame offset with respect to global time is a function of both the initial time offset at which each sequence was started and the ratio of the frame rate of each cameras. In this work the time t at which frame i in one camera is captured will be related to the time t' at which the same frame is captured in a second camera through the following relation as described in (Rozantsev et al., 2017) (Vo et al., 2016a) (Albl et al., 2017):

$$\frac{t_i - t_o}{\rho'} = \frac{t + i\alpha - t_o}{\rho'} = \frac{t_o - t'_o}{\rho'} + \frac{\alpha}{\rho'} i = \beta + \alpha i \tag{3.18}$$

Where β corresponds to the frame offset between the cameras and α corresponds to the ratio of the frame rate between a camera pair for a given frame i. The method on how to apply the time offset defined in 3.18 as proposed by (Albl et al., 2017) will be described in detail

Time and Motion Synchronization Camera Models

The 2D motion model and time synchronization model defined in 3.1.8 and 3.1.8 can be combined and integrated into the *epipolar geometry* and *homography constraint* defined in 3.1.5 as:

$$x^T \mathsf{F} s(\beta + \rho i) = 0 \tag{3.19}$$

and

$$\lambda x = \mathsf{H} \cdot s(\beta + \rho i) \tag{3.20}$$

respectively. These relations are used within the optimization procedure of the reconstruction pipeline to jointly determine the relative camera geometries and frame offsets of each camera pair using the trajectory of the tracked UAV observed by each camera pair.

3.1.9 Spatio-Temporal Generalized Eigenvalue Solution

The process of jointly solving for the camera geometry matrices, F or H and the temporal off-set between the camera sequences in 3.19 and 3.20 can be approached in several ways as described in 3.1.8. Within this thesis, the approach proposed by (Albl et al., 2017) to formulate 3.19 and 3.20 as a generalized eigenvalue problem (GEP) is used. Solving 3.19 and 3.20 to estimate corresponding object points between a camera pair to determine the corresponding camera matrices can be formualted asa standard generalized eigenvalue implementation available in (Jones et al., 2001–) for example. The solutions to the GEP provide estimates for the time-shift as the eigenvalues of the solution and the camera geometry (F or H) as the eigenvectors.

[2]Source: bhp photo

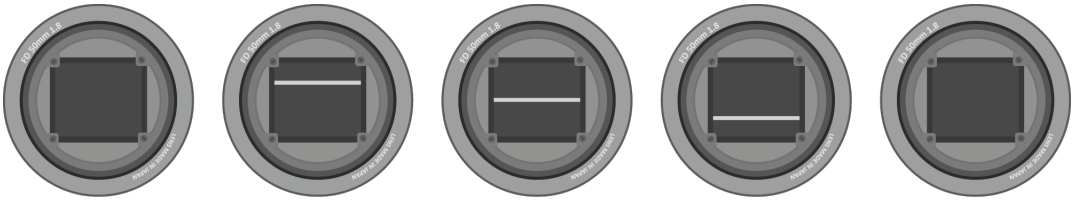


Figure 3.6: Perspective-n-Point model [2]

3.1.10 Rolling Shutter Correction

Inexpensive and low powered mobile devices employ Complementary Metal Oxide Semiconductor (CMOS) sensor cameras that employ a method referred to as rolling shutter exposure. As shown in 3.6, rolling shutter cameras readout each horizontal line of the image sensor sequentially. This causes each row of pixels on the image sensor to be exposed at a slightly different time. The rolling shutter effect therefore introduces positional errors of objects observed in the video sequence. Thus for a fast moving object, its potential location as detected in the image space can be several centimeters off from its physical location in the world space as the image sensor is only partially exposed at any given time. To account for this difference an extra term will be added to the time synchronization correction by the formulation in 3.21

$$tr_{ij} = t_{0j} + \rho_i \omega_i \frac{y_{ij}}{n_i} \tag{3.21}$$

3.21 accounts for the global rolling shutter time offset for a given camera i at frame j and initial frame time t_{0j}. Here ω is the scan time of the sensor normalized to one, y_{ij} is the vertical pixel coordinate of a given detection in camera i and frame j and n_i is the total vertical dimension in pixels of the image sensor of camera i. ω is included and initialized as zero within the synchronization and optimization steps of the bundle adjustment to obtain an optimal value for each camera.

3.1.11 Bundle Adjustment

The process described in 3.1.8 provides the spatio-temporal synchronization for a pair of cameras. The next task then becomes to jointly optimize this synchronization across the entire camera network. This process is commonly referred to as

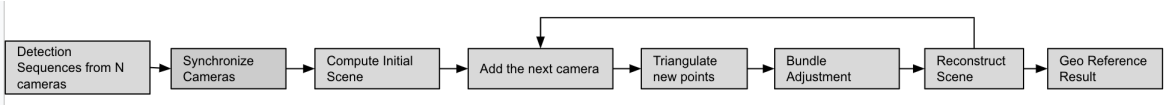


Figure 3.7: A basic workflow of incremental Structure from Motion

bundle adjustment for which standard frameworks exist (Agarwal et al., 2012) and have been adopted within this thesis. The cost function that is implemented in the bundle adjustment procedure in this thesis is based on minimizing the sum of the squared reprojection errors between the detection sequences and the triangulated 3D points derived through the geometry and time shift procedure described in 3.1.8. The objective function of the bundle adjustment is shown below.

$$\arg\min_{P_i,\mathbf{X}_j} \sum_{i=1}^{m}\sum_{j=1}^{n} \|\mathbf{x}_{ij} - P_i\mathbf{X}_j\|^2 \tag{3.22}$$

P_i denotes the individual camera matrices, $\mathbf{X}_j$ the reconstructed 3D trajectory point and $\mathbf{x}_{ij}$ the detected object projection of point $\mathbf{X}_j$ on the camera P_i. Eq. 3.22 is minimized using nonlinear least square algorithms, which require a good initialization and can be computationally very expensive because of the large number of parameters involved. The Levenberg-Marquardt algorithm (LM) (Marquardt, 1963), is employed in this thesis with a sparse Jacobian matrix to increase the computational efficiency.

3.1.12 Structure from Motion

Structure from Motion (SfM) is a general computer vision procedure used to jointly construct a 3D scene (structure) and the cameras that describe the scences from a set of 2D images (motion).

The basic procedure of SfM includes all of the steps as describe in 3.1.1. First the corresponding points in the scene are initialized from a pair of images using the fundamental matrix or the essential matrix. The corresponding points are then used to triangulate a 3D scene. Each additional image view is registered to the scene by solving the PnP problem using the known 3D points visible by the camera to obtain its pose as described in 3.1.6. Additional 3D points can be added to the scene by triangulating new points from the correspondences between each pair in the camera network. Finally the camera parameters and 3D points are jointly optimized in a bundle adjustment procedure. After all cameras have been

added to the network a final bundle adjustment run is applied to obtain an optimal solution that minimizes the total reprojection error across all of the cameras.

3.2 Random Sample Consensus Algorithm

The Random sample consensus (RANSAC) is a robust model fitting method that estimates a mathematical model from a set of observed data. RANSAC works by sampling a subset of points from a data set and fitting a model to it. The model is then used to determine other points in the set that conform to the model within a certain error tolerance. The algorithm randomly samples through subsets in the data to find the subset which fits the largest number of points in the data set. In this thesis a variant of RANSAC is employed called local optimized RANSAC (LO-RANSAC) in which an optimization method (e.g. non-linear least-squares) is applied to the set of final model inliers in order to further refine them. RANSAC serves as a means of robust estimation that is employed throughout many of the SfM and Bundle adjustment steps described above to reduce the influence of noise the procedures.

4 Methodology

This chapter provides a detailed description of how the methods described in Chapter 3 were implemented to obtain the final reconstructed trajectories obtained in this thesis. Section 4.1 describes the synchronization of a given camera pair using the relationships between their relative geometries. Section **??** describes the incremental structure from motion (SfM) and bundle adjustment procedure employed to reconstruct a set of globally synchronized cameras.

4.1 Camera Pair Synchronization

A method for the simultaneous estimation of the time synchronization between two cameras and the fundamental matrix that relates them, as presented by (Albl et al., 2017), was employed in this thesis. The *quasi-minimal solver* that they introduced described in section 4.1.2 estimates the time shift between a given camera pair using the minimal set of point correspondences needed to satisfy the set of linear equations used.

4.1.1 Relation between two un-synchronized cameras

In this thesis the location of the UAV is considered to be the 3D point corresponding to its center of mass at a given time t as follows:

$$X(t) = [X_1(t), X_2(t), X_3(t), 1]^T \tag{4.1}$$

. Projecting $X(t)$ into image planes of two distinct cameras produces two 2D image sequences. As each cameras captures images at discrete frames of $X(t)$, the corresponding image sequences can be described accordingly:

$$x_i = \mathsf{P}(X(t_i)), \quad i = 1, 2, \dots n \tag{4.2}$$

$$x'_j = \mathsf{P}'(X(t'_j)), \quad j = 1, 2, \dots n' \tag{4.3}$$

where i and j are frame indices, x_i and x'_j are 2D image points of the UAV and P and P' are the projection matrices of the two cameras as described in 3.6.

As described in 3.2 the global time at which the image sequences are captured do not agree and therefore they cannot be used as direct correspondences. To solve for the corresponding image point between the two sequences one must account for the relative time offset between the cameras as defined in 3.1.8, and the relative time offset at which two cameras capture image frames can be described by 3.18. As the time offset is not an integer value, the corresponding detection location must be approximated by means of linear interpolation between successive frames. The b spline function described in 4.2.2 is employed in this process to define virtual correspondences between a given camera pair.

The synchronization between cameras therefore reduces to the estimation of the two parameters (α, β) using correspondences defined above. In this thesis we assume that α is known or can be simply calculated. The time shift β can be any arbitrary value and thus there in not a straightforward way to estimate it with any degree of precision. Therefore, the following section presents an overview of the solver that was implemented within this thesis as proposed by (Albl et al., 2017).

4.1.2 A quasi-minimal solver of epipolar geometry

As described in 3.1.8, the 2D correspondences defined above and the assumption of known alpha, the general epipolar constraint between two cameras (see Eq. 3.12) is modified into 3.19. (Albl et al., 2017) employs the assumption that motion between small numbers of points can be linearly interpolated as an object's velocity between these points is constant for short time intervals. Thus a given point $x'(i + \beta)$ can be approximated by leveraging the sample point x'_i and a vector $v(d)$ as

$$v(d) = \frac{x'_{i+d} - x'_i}{d} \tag{4.4}$$

$$x'(i + \beta) = x'_i + v_i(d) \cdot \beta \tag{4.5}$$

where $v(d)$ is an approximation of the 2D trajectory over the next d samples, and d is number of frames used in the interpolation.

Based on this model the epipolar geometry in Eq. 3.19 can be solved through the

following system of linear equations:

$$
\begin{gathered}
\begin{pmatrix} x' + v_x\beta & y' + v_y\beta & 1 \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{11} & f_{12} & f_{13} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \\
\Downarrow \\
\begin{gathered} x(x' + v_x\beta)f_1 + y(x' + v_x\beta)f_2 + (x' + v_x\beta)f_3 \\ +x(y' + v_y\beta)f_4 + y(y' + v_y\beta)f_5 + (y' + v_y\beta)f_6 \\ +xf_7 + yf_8 + 1 \end{gathered} = 0 \\
\Downarrow \\
\begin{gathered} \begin{pmatrix} xx' & yx' & x' & xy' & yy' & y' & x & y & 1 \end{pmatrix} f \\ +\beta \begin{pmatrix} xv_x & yv_x & v_x & xv_y & yv_y & v_y & 0 & 0 & 0 \end{pmatrix} f \end{gathered} = 0
\end{gathered}
\tag{4.6}
$$

As β needs to be solved for in addition to the 8 points required for the fundamental matrix estimation, 9 point correspondences are necessary to solve 4.6. This results in the following linear system of equations

$$
(M_1 + \beta M_2)\, f = 0 \tag{4.7}
$$

where M_1 and M_2 are 9×9 matrices from 2D point correspondences and approximated tangent vectors. Eq. 4.7 can be solved as a generalized eigenvalue problem, which can be solved efficiently using singular value decomposition (SVD). As three columns of M_2 contain only zeros, M_2 has a rank of six as three of the nine eigenvalues are zero. Thus the solution for 4.7 produces six possible solutions where the eigenvalues represent the time shifts and eigenvectors represent the corresponding fundamental matrices. Similar to the Eight-point algorithm, the resulting fundamental matrix does not necessarily have rank 2, which can be enforced by letting its third singular value be zero using SVD. Note a similar solution for the *homography* constraint presented in 3.20 can also be derived for planar scenes as described by (Albl et al., 2017).

As the above described solver operates on small numbers of correspondences, it was implemented in a LORANSAC framework to provide robustness to noise and correspondence outliers that are created due to poor calibration and uncertainty in the detection locations. In practice, the detection uncertainties will be higher for objects traveling at high speeds and whose size in the image space varies a great deal. Additionally, the assumed constant velocity of an object between detections as utilized in 4.5 fails when an object such as a UAV un-

dergoes large changes in acceleration or non linear motion. Detections at such points in the trajectory would then be more prone to generating outlier correspondences. RANSAC helps to eliminate such points and focus the solution on parts of the trajectory in which the object's motion conforms to the constant velocity assumption. RANSAC also helps to eliminate non-real solutions for $\beta produced in the six possible eigenvalues obtained in Eq. 4.7, only one of which is the correct solution.$

4.1.3 Iterative algorithm of the minimal solver

This section presents an overview of the algorithm presented in (Albl et al., 2017) and implemented within this thesis to employ the quasi minimal solver as described in 4.1.2 to enable the for large time shifts to be estimated between image sequences. As shown in (Albl et al., 2017), the accuracy of the minimal solver decreases as the time shift increases. However, they also show that as the interpolation distance used in the velocity vector described in 4.5 is increased, the performance improves for time shifts that are close to the interpolation distance being used. With this observation an iterative method is employed that searches for the correct time shift by selecting the interpolation distance used in the minimal solver that produces the most valid points from the corresponding estimate of F from 4.7. The iterative algorithm as presented by (Albl et al., 2017) is described in algorithm 1.

As displayed in algorithm 1, in the initial step in the algorithm, a range of d interpolation distances are defined. A feasible choice is the exponential numbers, i.e. $d \in \{\pm 2^0, \pm 2^1, \cdots, \pm 2^m\}$. At each iteration, the time shift β and the fundamental matrix F are estimated at each d defined in the search interval. Each successive d is evaluated to determine the one that generates the highest ratio of coresponding points that have a reprojection error below a predefined error threshold. These points are labeled as inliers. The search is repeated until the maximum number of inlier points does not change.

Algorithm 1 Iterative algorithm

Input:
Image trajectories from two camera s, s', maximal exponent p_{max}
Output:
Time shift β, fundamental matrix F

Let $ratio_{max}$=0, $\beta = 0$, $p = 0$
while $p <= p_{max}$ **do**
  Apply *quasi-minimal solver* with $d = 2^p$, obtain β_1, F_1, $ratio_1$
  Apply *quasi-minimal solver* with $d = -2^p$, obtain β_2, F_2, $ratio_2$
  if $ratio_1 > ratio_2$ **then**
    $\beta_{temp} = \beta_1$, $F_{temp} = F_1$, $ratio_{temp} = ratio_1$
  else
    $\beta_{temp} = \beta_2$, $F_{temp} = F_2$, $ratio_{temp} = ratio_2$
  end if
  if $ratio_{temp} > ratio_{max}$ **then**
    Update s' according to β_{temp}
    $\beta = \beta + \beta_{temp}$, $F = F_{temp}$
    $ratio_{max} = ratio_{temp}$, $p = 0$
  else
    $p = p + 1$
  end if
end while
return β, F;

4.2 Incremental Trajectory Reconstruction

The process to synchronize the cameras through the estimation of the time shift *beta* allows for correspondences to be accurately determined and subsequently the initial geometry of a camera pair and triangulated UAV trajectory to be determined. The synchronization step serves as the first step in the incremental SfM pipleline that is introduced in this thesis. The remaining steps involved in the pipeline will be discussed in further detail below.

4.2.1 Trajectory initialization

The incremental SfM approach is a standard approach in computer vision that is widely employed to build 3D scences in the compute vision domain. The fundamental steps in the procedure are shown in 3.3. Within this thesis we assume that the cameras used in the SfM pipeline are well calibrated by the procedure as described in 3.1.3. The scene is initialized by defining a synchronized camera pair and estimating their pose and generating 3D trajectory as described in 3.1.4 from the correspondences between the UAV detection sequences obtained from each camera video. The initial camera geometry and triangulated trajectory are then used to estimate the pose and triangulate additional points with a third camera using the PnP approach defined in 3.1.6. The initial camera triplet is then jointly optimized for geometry, synchronization and their corresponding triangulated trajectory. Additional cameras are added in a similar fashion and the scene is jointly optimized after each addition. The details of the methods implemented in the pipeline are described in 3.1.1. The areas where the pipeline implemented in this thesis differ from the standard incremental approach will be highlighted in the sections that follow.

As described in 3.1.8 the SfM pipeline implemented in this thesis takes as input detection sequences of a moving UAV from a network of n cameras. The method and algorithm used to determine the synchronization offset for a given camera pair in the network are described in 1 above. The synchronization of each camera pair in the network is computed and initial scene and 3D trajectory is determined for each pair. The pair that produces the most number of initialized points is then selected as the pair to start the optimization process.

4.2.2 Spline regularization scene detections and trajectory

The desired reconstruction of the UAV trajectory should represent a continuous function describing the UAV motion over time. As the detections from each camera are unlikely to be continuous given that the UAV can move in and out of each camera view, the resulting triangulated 3D points obtained from their detections will also not be continuous. To account for the discontinuities between the detected UAV locations and the true trajectory, a B-spline as described in 4.2.2 is implemented within the reconstruction to ensure that the optimization of all of the scene parameters is based on a continuous set of points within each camera sequence. The spline serves multiple benefits within the SfM pipeline. Firstly, the spline can be fit to the existing set of 3D points triangulated between each camera and represent the curve that they represent with many fewer parameters. This serves to reduce the overhead required in the optimization phase in the bundle adjustment procedure and allows the optimized spline parameters to be used to describe the true measured 3D points of the UAV. Second, as each camera is synchronized with non integer time offsets as defined in 3.1.8 and 3.1.10, a spline can be fit to each detection sequence during the triangulation and optimization phases to define virtual correspondences between camera pairs. The spline helps to constrain the reconstructed trajectory to maintain a uniform geometry. Individual optimized points could otherwise be moved to nonsensical locations that only serve to minimize the reprojection error and not the physical geometry of the real UAV motion. Another useful feature of the spline representation is the sparsity property. As a spline is defined as a sparse piecewise combination of polynomial functions describing a set of points using a sparse set of knots, its parameters can be optimized without altering the entire distribution of the fitted curve.

4.2.3 Physical Motion priors

As described above, the spline representation is useful to efficiently describe the set of 3D points defined by each camera pair. However the spline function also applies smoothing to the true measured data points and there are no controls other than maintaining geometric symmetry that ensures that the spline representation is as close as possible to the true movement of the UAV. An alternative method of constraining the optimized trajectory to conform to the true motion of the UAV is to inject a prior into the objective function that forces the optimization to move

points to locations that agree with physical laws of motion. As described in 3.1.8, the least force and least kinetic energy motion priors are added as a final step in the bundle adjustment. In this step, the original synchronized detections are used to interpolate 3D points via a spline from the final optimized trajectory that is produced after all cameras have been added to the scene. This set of 3D points is then run through a final bundle adjustment step together with the motion prior regularization terms as defined in 3.1.8. The objective function for this final step is defined in 4.2.4.

4.2.4 Spatiotemporal bundle adjustment

As defined in 3.1.11, bundle adjustment is a standard optimization procedure that is used to jointly optimize a complex set of parameters that describe a set of data points. In this thesis, as well as in works such as (Vo et al., 2016b), the bundle adjustment procedure is used to optimize the 3D trajectory point, the relative camera poses, the β time offset and rolling shutter synchronization parameters of each camera. Through testing it was determined that additionally including the optical parameters of each camera led to worse results, so it was concluded that the results of the calibration were already optimal and they were removed from the bundle adjustment procedure. The cost function which minimizes the reprojection error of the reconstructed points accounting for the time synchronization parameters is defined as:

$$\arg\min_{P,X,\beta} \sum_{t=1}^{T} \sum_{c=1}^{C} V_c^t \left\| \pi(P_c, X(t)) - x_c(t_c + \beta_c + \omega_c) \right\|^2 \tag{4.8}$$

where P_c denotes camera parameters, $X(t)$ the 3D UAV location at time t, and V_c^t the binary indicator of point-camera visibility. t_c is the local time of camera c at which the UAV is projected and β_c and ω_c are the time shift and rolling shutter offset of a given camera c with respect to the global timeline, such that $t = t_c + \beta_c + \omega_c$. Thus $x_c(t_c)$ is the original detection and $x_c(t_c + \beta_c + \omega_c)$ is the actual projection of the UAV at time t. This variant of the optimization procedure will be labeled as S_{Points} as described in the experiments section.

In section 4.2.2 the concept of fitting a spline to the set of reconstructed 3D points is introduced. The resulting parameterized representation of the UAV trajectory can then be used in place of the individual 3D point locations of the UAV within the

spatiotemporal bundle adjustment procedure. The cost function which minimizes the reprojection error of the reconstructed trajectory can then be defined as follows:

$$\arg\min_{P,S,\beta} \sum_{t=1}^{T} \sum_{c=1}^{C} V_c^t \left\| \pi(P_c, S(k_t)) - x_c(t_c + \beta_c + \omega_c) \right\|^2 \tag{4.9}$$

where $S(k_t)$ denotes the 3D point sampled from the spline S at the global time t, which is parameterized by the spline coefficients k_t. Only the spline coefficients are optimized in this case in order to maintain the number of spline parameters as constant and reduce the complexity of the optimization. This optimization variant will be described as S_{spline} in the experiments section.

As described in 4.2.3, an additional cost is employed as a final step in the bundle adjustment procedure to constrain the optimized cost to comply with physical motion models. The least kinetic energy cost function which enforces constant velocity is described in 4.10 while the least force cost which enforces constant acceleration is described in 4.11. These cost functions are added to the reprojection error cost function in 4.8 without optimizing the time synchronization terms to obtain a final optimized trajectory directly from the original time period that the UAV was observed by each camera.

$$\arg\min_{X_t} \sum_{t=1}^{T} \sum_{c=1}^{C} \frac{w^t}{2} \left\| \frac{(X_c^{i+1}(t_c^{i+1}) - X_c^i(t^i)}{(t^{i+1} - t^i)} \right\|^2 (t^{i+1} - t^i) \tag{4.10}$$

$$\arg\min_{\delta X_t} \sum_{t=1}^{T} \sum_{c=1}^{C} \frac{w^t}{2} \left\| \frac{(\delta X_c^{i+1}(t_c^{i+1}) - \delta X_c^i(t^i)}{(\delta t^{i+1} - \delta t^i)} \right\|^2 (\delta t^{i+1} - \Delta t^i) \tag{4.11}$$

4.2.5 Georeferencing of Reconstruction

While the resulting constructed trajectory can be inspected visually to assess the effectiveness of the reconstruction, a true validation of the accuracy of the reconstruction requires comparison to ground truth measurements. In order to compare the accuracy of the resulting reconstructed trajectory, a Euclidian similarity transform and uniform scaling was applied to the trajectory to project the trajectory into the East North Up coordinate reference projection that the ground truth measurements were collected in.

4.2.6 Measurement Alignment with Ground Truth

As the collection frequency of the on-board RTK sensor was much lower than the frame rates of the cameras, the reconstructed trajectory was much more dense than the ground truth. As such a method had to be employed to determine the temporal alignment between the reconstructed trajectory and the ground truth. This was achieved by densely sampling the resulting trajectory reconstructions and shifting the trajectory by small increments across the dense samples. During each shift across the trajectory the Euclidian transform was applied and the euclidean distance between the ground truth and the shifted reconstructed trajectory was computed. The shifted trajectory with the lowest mean euclidean distance between the reconstructed trajectory and ground truth was selected as the best result. Additionally, the iterative closest point method was employed as a final post processing step to optimize the results of the Euclidean similarity transformation.

5 Experimental Results

This chapter describes the setup and results of the experiments conducted on the methods implemented within this thesis. Validation was conducted on both synthetic and real data.

5.1 Camera Synchronization on Synthetic Data

Experiments were run to validate the iterative time synchronization method proposed by (Albl et al., 2017) and presented in algorithm 1.

5.1.1 Synthetic Data

Synthetic trajectories were generated randomly for the synthetic data tests. An example of such trajectory is displayed in 5.1. The three dimensional trajectory was projected into two known cameras to simulate two dimensional point sequences to serve as ground truth data for the subsequent synchronization experiments.

5.1.2 Performance of iterative algorithm

The iterative algorithm presented in algorithm 1 was tested on the simulated 2D trajectory tracks described above. Based on the observations made by (Albl et al., 2017) that the performance of the time shift estimation achieved by the quasi minimal solver increases as the velocity vector interpolation distance approaches the true timeshift, an experiment was setup to validate this across two randomly generated trajectories. As the estimation of large time shifts is a desired feature of the algorithm, the extent of ground truth time shifts were set to be evaluated from $\in [-200, 200]$ frames, which corresponds to a time span of $\in [-6.7, 6.7]$ seconds. The interpolation distance was tested at intervals between $\in \{\pm 2^0, \pm 2^1, \cdots, \pm 2^8\}$. For each ground truth time shift, the algorithm was run 10 times. A successful

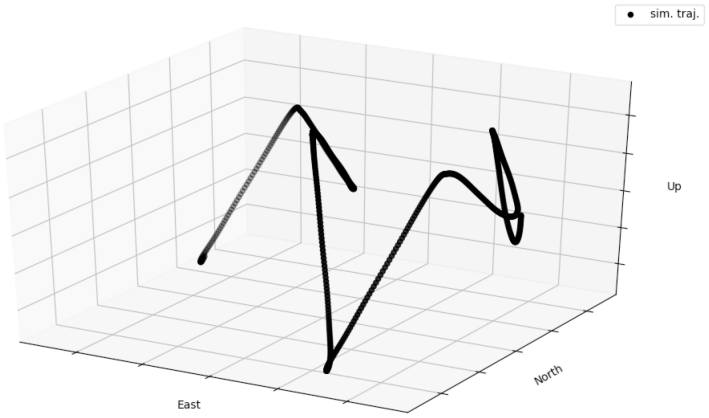


Figure 5.1: Synthetic trajectory

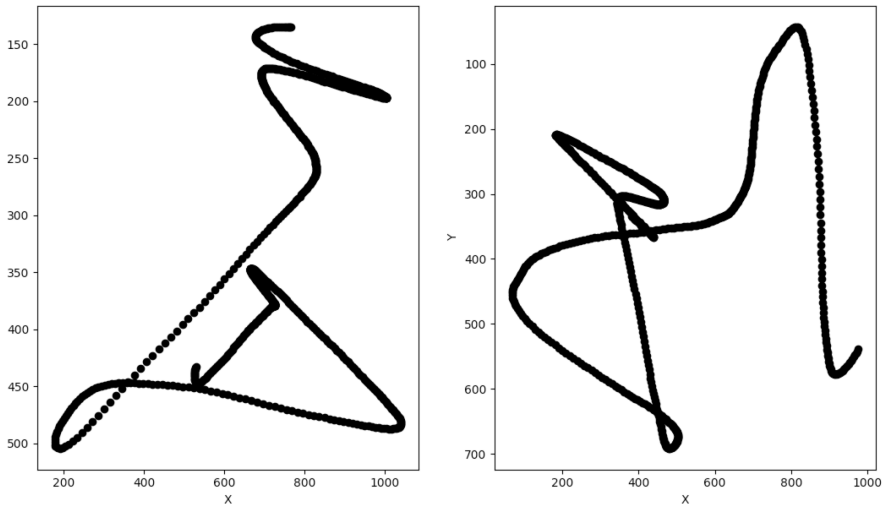


Figure 5.2: 2D trajectory projections

estimation was counted if the estimated time shift was within one frame of the ground truth. The results of this analysis are shown in Fig 5.3. From the figure, it can be observed that for smaller time shifts the algorithm performs relatively better than as the time shift increases beyond 5 seconds. One can also observe that the time shifts for the negative and positive intervals perform nearly the same, which indicates that the algorithm performs well in terms of determining in which direction the velocity vector should be interpolating. The time shift estimates are rather inconsistent across the time interval even for lower values. This indicates that more runs should be conducted for any given time shift in order to ensure a higher chance of success in estimating a valid time shift.

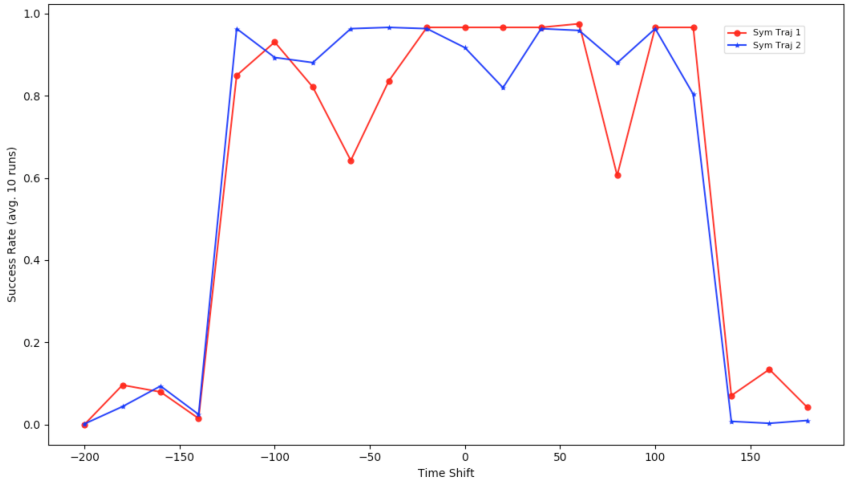


Figure 5.3: Success rates of the iterative algorithm for two randomly simulated trajectories. Success rates are averaged across ten runs evaluated at each time shift interval. Success is determined if the time shift is estimated correctly estimated up to one frame.

5.2 Trajectory Reconstruction on Real Data

This section presents the performance of the SfM trajectory reconstruction pipeline.

5.2.1 Data Description

The real data sets were collected for two flights in an outdoor setting of a hexicopter UAV. Each flight was conducted for approximately two minutes. A network of four cameras was established in a semicircle surrounding the flight zone. The UAV was provided by (*Company Fixposition*) and was fitted with a Real-time Kinematic (RTK) positioning sensor to provide ground truth location data for both flights. The raw RTK measurements were provided longitude, latitude and height coordinates which were transformed into local ENU (East, North, up) coordinates for comparison with the reconstruction results. The RTK data collected was considered as ground truth throughout the analysis as the manufacturer guarantees the accuracy of their sensor to be on the order of centimeters (see Tab. 5.1).

Horizontal accuracy	Vertical accuracy	Measurements rate
1cm + ppm	1.5cm + ppm	5Hz

Table 5.1: Specifications of RTK data according to Fixposition

5.2.2 Camera Network

The camera network used in the experiment consisted of four unsynchronized cameras mounted statically on tripods surrounding the flight zone. The network consisted of two smart phone cameras and two handheld photography cameras. The cameras remained in the same position for both flights. The distance of each camera to the UAV center was approximated to be less than 100m at any time during the flight. The frame rates of each camera were determined in evaluating the video sequences and were used to create virtually interpolated video sequences normalized to 30 frames per second. The video sequences were started by hand and thus the time offset between the sequences is unknown.

Detection sequences were collected from each camera video using the methods described in 3.1.2 and provided as input for the reconstruction pipeline (see Fig 5.4). As the UAV flight was completely random if flew in and out of the camera views and at different distances from each camera at any given time. The discontinuities and varying distances provide a very challenging data set for the detection pipeline. Therefore it is assumed that there is a high degree of uncertainty within each detection sequence due to inaccurate detection location and discontinuous tracks. This means that the robust features built into the reconstruction pipeline needed to be fully utilized.

Figure 5.4: Example of input UAV detections

5.2.3 Experimental configuration

As there was no ground truth time synchronization collected for the video sequences, a process of visual and audio analysis was done to obtain an initial estimate of the offset between each camera. After the cameras were globally synchronized, the incremental SfM pipeline was started with the camera pair that

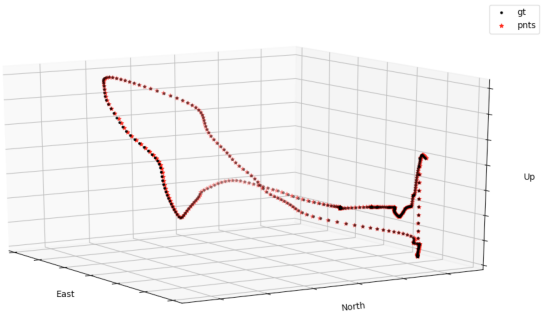
shared the most overlapping frames. The geometry and initial trajectory were evaluated and the additional two cameras were sequentially added into the pipe as described in 3.7. The non-linear optimization that was included in the bundle adjustment procedure uses the *least_squares* function from the Scipy Python library. The maximum number of iterations for each optimization step was set to 100. A large number of iterations was used as the cost function employed within the pipeline is complex and therefore there has a high chance of getting stuck in a local minimum.

Two base experimental configurations were used within the pipeline to evaluate the optimized trajectory as individual points as a spline. Additional tests were run to compare the results of the reconstruction with and without rolling shutter correction and through employing the motion prior cost function as defined in 4.11 and 4.10. As described in 3.1.3, it is assumed that the calibration parameters derived from standard chessboard calibration frameworks provided in python and matlab obtaion are able to obtain an optimal solution, and therefore the calibration parameters were left out of the bundle adjustment procedure after verification that including them did not add to the accuracy of the trajectory reconstruction.

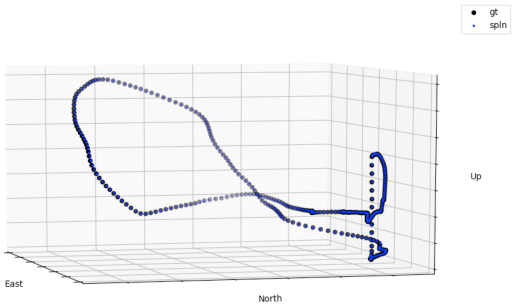
5.2.4 Qualitative and quantitative results

The following section provides the results of the reconstruction of the UAV trajectory for the various model configurations defined above.

Figures 5.5 and 5.7 show the reconstructed trajectories obtained for flight 1 for the optimized points and spline reconstruction models. It can be seen visually that the results agree generally well with the ground truth RTK measurements. While difficult to see the differences between the results visually, observing the reconstruction errors in 5.2 reveals the differences in the reconstruction method. First, we see that for both methods employing rolling shutter correction provides a reduction in error. This makes sense for fast motions of a UAV, where even a few pixels of inaccuracy in the detection location translates to tens of centimeters in the world space. Correcting for these small offsets imposed by the rolling shutter scan time helps to reduce some of this error. Secondly, we see that the mean error for the optimized spline representation is much lower than the points representation. This was to be expected because, as described in 4.2.2, the spline serves to ensure geometric symmetry within the optimized points. Without this constraint, points



(a) Points



(b) Spline

Figure 5.5: Reconstruction of 1st Flight compared with RTK ground truth without (left) and with (right) spline representation.

can be moved to locations that minimize the reprojection error but don't agree with natural motion of the UAV. One can observe this effect in the distribution of the data in 5.6. Here we see that generally the spline representation reconstructs many more points with low error than the points model, while at the same time having fewer points with high error. This can again be attributed to the geometry constraint that agrees with the smooth nature of a UAV in flight.

	Mean (cm)	RMSE (cm)	Max (cm)	Min (cm)
Points	6.64	5.2	24.8	0.6
Points$_R S$	6.60	5.0	24.8	0.6
Spline	5.8	4.6	23.3	0.5
Spline$_R S$	5.6	4.6	22.8	0.5

Table 5.2: Mean, RMSE and max errors of the 1st Flight reconstruction

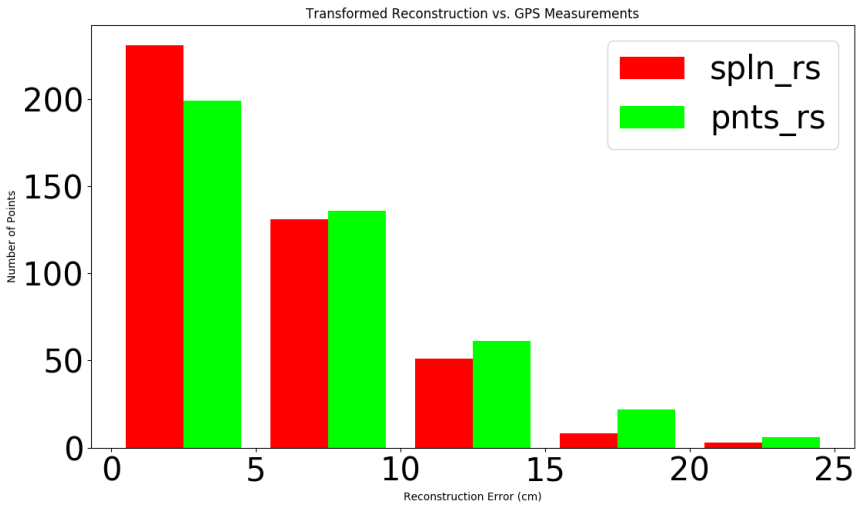


Figure 5.6: Error histogram of the 1st Flight

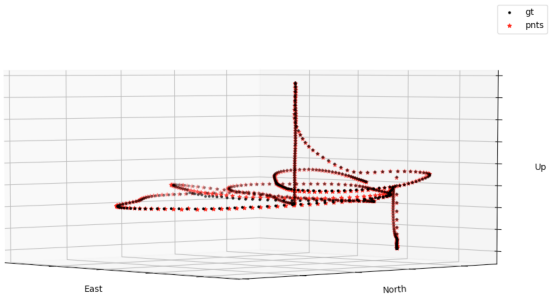
	Mean (cm)	RMSE (cm)	Max (cm)	Min (cm)
Points	13.6	9.0	39.8	1.7
Points$_R S$	13.3	8.9	37.1	1.3
Spline	11.2	7.6	31.4	1.2
Spline$_R S$	10.9	7.6	30.3	1.1

Table 5.3: Mean, RMSE and max errors of the 2nd Flight reconstruction

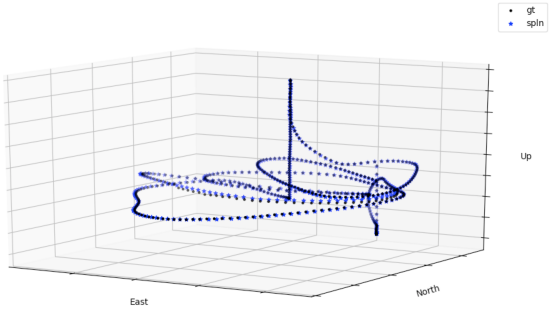
Results of 2nd Flight The results obtained from the second flight are similar to the trends observed in the first flight. Fig. 5.7 shows the results of the trajectory reconstruction for both the point and spline representations. One observation to note is that the trajectory of the second flight is much more complex and has more portions where the UAV is moving in a planer motion. This can potentially introduce degeneracies in the reconstruction result as the estimation of the Fundamental matrix fails for planar scenes. Tab. 5.3 and Fig 5.8 display the error between the ground truth measurement and the reconstruction results for the second camera. Though the errors are higher than those of the first flight, again we see that the spline configuration produces the lowest errors and more of the points are situated in the lower error margins of the error distribution.

Visualizations of position errors along the trajectory for both flights can be found in Appendix **??**.

Motion Prior Analysis As mentioned in section 5.2.3, a final step was added to the bundle adjustment to optimize the raw detections. 3D points are triangulated



(a) Points



(b) Spline

Figure 5.7: Reconstruction of 2nd Flight compared with RTK ground truth without (left) and with (right) spline representation.

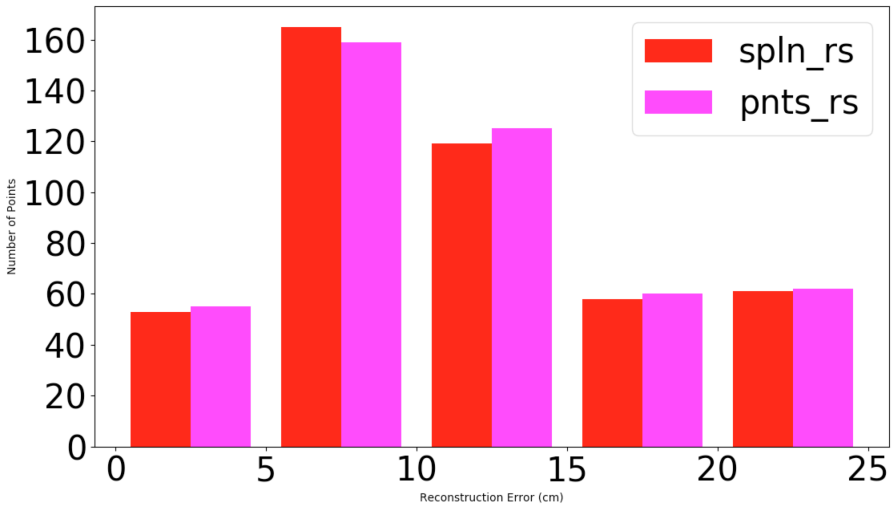


Figure 5.8: Error histogram of the 2nd Flight

Figure 5.9: Reconstruction Error sensitivity on Motion Prior Weight Factor

from the final result of the optimized point reconstruction which correspond to the times at which to original raw UAV detections were made after synchronization. As shown in 4.10 and 4.11 a weight factor is applied to the constant velocity and acceleration terms respectively to add importance to points in the trajectory where the motion does not abide by physical laws of motion by a large margin and forces the optimization to constrain those points. It can be seen in 5.9 that for low values of the weight factor the raw detection 3D points are unconstrained and are therefore reprojected in correctly. As the weight factor approaches values that are on par with the number points in the trajectory the prior starts to constrain the trajectory and the final error produced is on par with the optimized point solution though still higher than the result obtained by the spline representation. A second experiment was made to compare the whether the regularization provided by the motion priors is similar to that of geometric symmetry constraint provided by the spline. To compare the effect that each of these regularization methods has on the resulting trajectory reconstruction the motion prior weight factor and spline smoothing factors were applied at different values for each run of the bundle adjustment. The results of the sensitivity analysis is shown in 5.10. A similar trend in the motion prior weight factor and spline smoothing parameter can be observed. As the number of knot defined in the spline representation is increased, we see that the reconstruction error reaches a plateau. A good approach would then be to determine the sufficient value of knot points that allows the spline to describe the

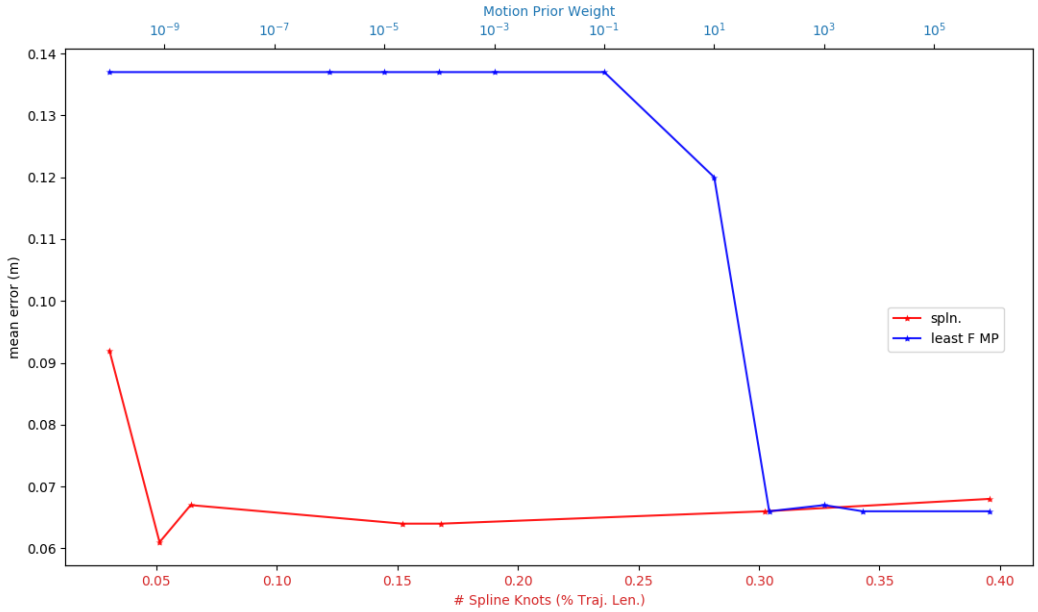


Figure 5.10: Motion Prior Weight compared to Spline Smoothing factor

trajectory even for complex motion but not so many that the spline can over-fit the distribution and create invalid geometries. From 5.10 we can see that around using defining a spline with a number of knot points equal to 20 percent of the total trajectory length helps to maintain a low reprojection error while containing a sufficient number of points to allow the distribution of the trajectory to be well defined.

Local Error Distribution Analysis A final investigation was made to gain and understanding of specific features with the reconstructed trajectories contribute trends in the error distribution of the reconstruction. Figure 5.11 shows the distribution of the error of the optimized point trajectory. Zooming into a point where high errors occur in 5.12 we can see in the ground truth signals that points that are closely spaced low errors occur where as regions where high variation in the ground truth points occurs the errors are high. This makes sense as points that are spaced far apart this indicates that the UAV is accelerating and therefore it is more likely that mis-detections can occur in image sequences and invalid time correspondences can be determined withiin the time synchronization procedure and rolling shutter distortion on the location of the UAV detection in the image space.

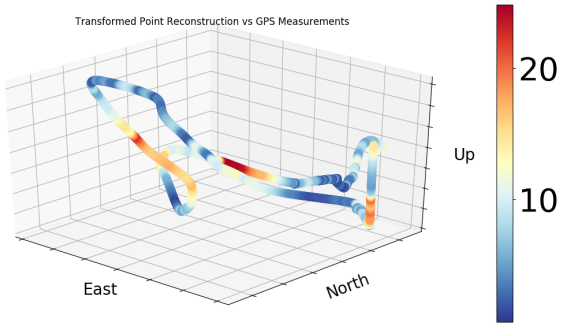


Figure 5.11: Local Error Distribution Across The Optimized Point Trajectory

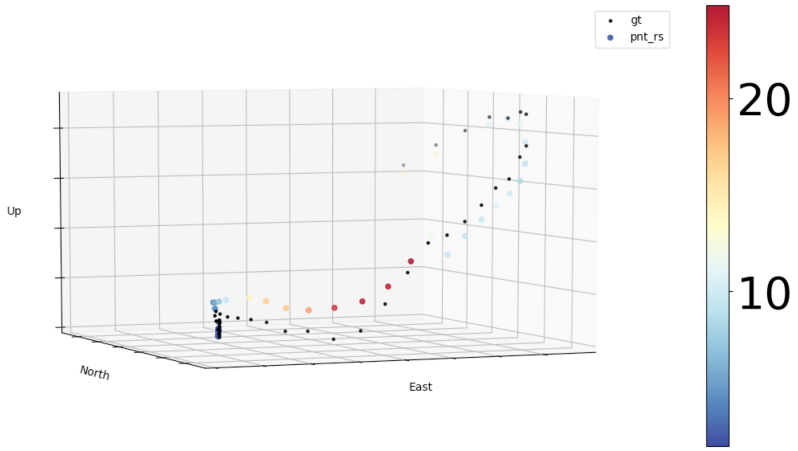


Figure 5.12: Local Error Distribution Across High error region of the point trajectory

Figure **??** shows a section of the reconstruction from flight one where the errors are on the lower half of the distribution. The points in this part of the trajectory are more uniformly space indicating that UAV is moving at a more uniform rate. This results in more accurate detections and less rolling shutter effect and ultimately are lower reprojection error.

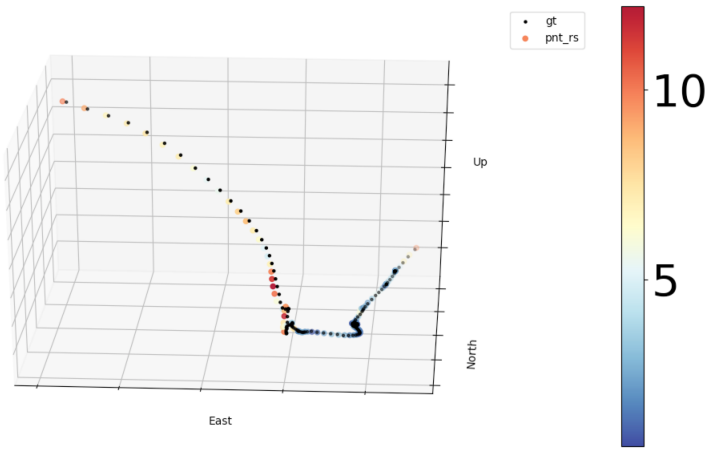


Figure 5.13: MLocal Error Distribution Across low error region of the point trajectory

6 Discussion

The results of the experimental evaluation of the implemented trajectory reconstruction pipeline show that the methods investigated within this thesis allow for a valid time synchronization and scene geometry to be determined; these are the necessary components required to obtain the final desired reconstructed trajectory. The following was validated: through jointly accounting for time synchronization parameters such as rolling shutter distortion and parameterizing the data through interpolation methods such as a spline, valid image correspondences can be determined to allow for the geometry of the scene to be determined. Though the final trajectory agrees well with the measured ground truth trajectory, no ground truth data was collected to validate the results of the time synchronization. Several post processing steps were involved to obtain the final result; therefore, it is difficult to confirm that these approaches are not masking some deficiencies in the time synchronization method. Further data should be collected with ground truth synchronization to enable the deficiencies or improvement areas to be more thoroughly explored.

Another area lacking is the ability to validate the estimated camera poses. Ground truth data regarding the camera locations and orientations should be collected in future data sets to validate that ability of the pipeline to accurately derive valid camera geometries that describe the scene.

The motion prior regularization term applied in the final step of the bundle adjustment did not perform as expected. It was assumed that the physical motion laws that these parameters constrained the optimization with would outperform the spline approximation, but this wasn't the case. It could be that the motion of a UAV is too complex to be well described by such simple relations and a more complex model which accounts for the dynamic motion of the UAV should be adopted to achieve better results. The spline representation being a piece wise polynomial function is well-suited to describe more complex motion and perhaps this is why the results obtained were so much better.

Further work that needs to be tested within this pipeline is the effect of the uncertainty in the detection sequences that are used within the reconstruction. Uncer-

tainties included in the detection need to be built into the reconstructed trajectory. If a model of these uncertainties is known, they can perhaps be accounted for within the pipeline and a better and easier-to-validate reconstruction can be achieved.

Finally, work should be done to investigate the sensitivity of the pipeline to the size of the detection sequences that are be used to perform the reconstruction. This would provide details on the limitations of the approach that would need to be accounted for to allow it to perform the reconstruction in real time. The degenerate geometry configurations such as planar motion should also be investigated to determine if there is novel way to automatically employ the homography constraint to provide robustness to the reconstruction pipeline.

7 Conclusion & Outlook

As UAV technology becomes more advanced and further integrated into our society, robust and accessible methods to accurately localize them as they move through the environment are necessary. In this work, a novel approach to achieve this goal was presented. It was demonstrated that a valid camera geometry and time synchronization could be jointly estimated by way of a minimal solver. Furthermore, it was shown that through integrating geometry and physical modeling parameters into the optimization procedure better reconstruction results can be achieved. Furthermore in accounting for hardware specific properties of the camera network, such as rolling shutter distortion, the resulting synchronization can be more accurately estimated and subsequently a better reconstruction result can be achieved. More extensive and carefully designed experimental datasets should be collected in the future to further refine the methods explored within this thesis. However, the results achieved confirm that the basic concepts upon which the trajectory reconstruction framework developed within this thesis are based provide the fundamental foundation for future developments in the field of 3D geometry and object detection.

Reference

Agarwal, S., K. Mierle, et al. (2012). "Ceres solver". In:

Albl, C., Z. Kukelova, A. W. Fitzgibbon, J. Heller, M. Smíd, and T. Pajdla (2017). "On the Two-View Geometry of Unsynchronized Cameras". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5593–5602.

Avidan, S. and A. Shashua (2000). "Trajectory Triangulation: 3D Reconstruction of Moving Points from a Monocular Image Sequence". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22, pp. 348 –357. DOI: 10.1109/34.845377.

Baque, P., F. Fleuret, and P. Fua (2017). "Deep Occlusion Reasoning for Multi-camera Multi-target Detection". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. DOI: 10.1109/iccv.2017.38. URL: http://dx.doi.org/10.1109/ICCV.2017.38.

Beauchemin, S. S. and J. L. Barron (1995). "The computation of optical flow". In: *ACM computing surveys (CSUR)* 27.3, pp. 433–466.

Bradski, G. (2000). *The OpenCV Library*.

Brahmbhatt, S., J. Gu, K. Kim, J. Hays, and J. Kautz (2018). "Geometry-Aware Learning of Maps for Camera Localization". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. DOI: 10.1109/cvpr.2018.00277. URL: http://dx.doi.org/10.1109/CVPR.2018.00277.

Buddubariki, V., S. G. Tulluri, and S. Mukherjee (2015). "Multiple object tracking by improved KLT tracker over SURF features". In: *2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pp. 1–4. DOI: 10.1109/NCVPRIPG.2015.7490012.

Chavdarova, T. and F. Fleuret (2017). "Deep Multi-camera People Detection". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. DOI: 10.1109/icmla.2017.00-50. URL: http://dx.doi.org/10.1109/ICMLA.2017.00-50.

Collins, R., A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa (2000). *A System for Video Surveillance and*

Monitoring. Tech. rep. CMU-RI-TR-00-12. Pittsburgh, PA: Carnegie Mellon University.

Dai, A. and M. Niessner (2018). "3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation". In: *The European Conference on Computer Vision (ECCV)*.

Dong, H. and X. Zhang (2018). "Moving Object Detection Algorithm Using Gaussian Mixture Model and SIFT Keypoint Match". In: DOI: 10.1007/978-3-319-73564-1_3.

Ess, A., K. Schindler, B. Leibe, and L. Van Gool (2010). "Object Detection and Tracking for Autonomous Navigation in Dynamic Environments". In: *I. J. Robotic Res.* 29, pp. 1707–1725. DOI: 10.1177/0278364910365417.

Feichtenhofer, C., A. Pinz, and A. Zisserman (2017). "Detect to Track and Track to Detect". In: *CoRR* abs/1710.03958. arXiv: 1710.03958. URL: http://arxiv.org/abs/1710.03958.

Fischler, M. A. and R. C. Bolles (1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6, pp. 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: http://doi.acm.org/10.1145/358669.358692.

Frey, J., K. Kovach, S. Stemmler, and B. Koch (2018). "UAV Photogrammetry of Forests as a Vulnerable Process. A Sensitivity Analysis for a Structure from Motion RGB-Image Pipeline". In: *Remote Sensing* 10.6. ISSN: 2072-4292. DOI: 10.3390/rs10060912. URL: http://www.mdpi.com/2072-4292/10/6/912.

Gindraux, S., R. Boesch, and D. Farinotti (2017). "Accuracy Assessment of Digital Surface Models from Unmanned Aerial Vehicles' Imagery on Glaciers". In: *Remote Sensing* 9.2. ISSN: 2072-4292. DOI: 10.3390/rs9020186. URL: http://www.mdpi.com/2072-4292/9/2/186.

Haque, M., M. Murshed, and M. Paul (2008). "A hybrid object detection technique from dynamic background using Gaussian mixture models". In: pp. 915 –920. DOI: 10.1109/MMSP.2008.4665205.

Hartley, R. and A. Zisserman (2003). *Multiple view geometry in computer vision*. Cambridge university press.

He, F., T. Zhou, W. Xiong, S. Hasheminnasab, and A. Habib (2018). "Automated Aerial Triangulation for UAV-Based Mapping". In: *Remote Sensing* 10.12, p. 1952.

Jebara, T., A. Azarbayejani, and A. Pentland (1999). "3D structure from 2D motion". In: *IEEE Signal processing magazine* 16.3, pp. 66–84.

Jones, E., T. Oliphant, P. Peterson, et al. (2001–). *SciPy: Open source scientific tools for Python*. [Online; accessed ¡today¿]. URL: http://www.scipy.org/.

Keuper, M., S. Tang, B. Andres, T. Brox, and B. Schiele (2018). "Motion Segmentation amp; Multiple Object Tracking by Correlation Co-Clustering". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2018.2876253.

Lepetit, V., F. Moreno-Noguer, and P. Fua (2009). "Epnp: An accurate o (n) solution to the pnp problem". In: *International journal of computer vision* 81.2, p. 155.

Liang, J., J. Gong, and Y. Liu (2018). "Introduction to the Special Issue: "State-of-the-Art Virtual/Augmented Reality and 3D Modeling Techniques for Virtual Urban Geographic Experiments"". In: *ISPRS International Journal of Geo-Information* 7.9. ISSN: 2220-9964. DOI: 10.3390/ijgi7090366. URL: http://www.mdpi.com/2220-9964/7/9/366.

Luo, W., J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim (2014). *Multiple Object Tracking: A Literature Review*. arXiv: 1409.7618 [cs.CV].

López-Araquistain, J., I. Campaña, L. Bergesio, and J. A. Besada (2017). "Experimental prototype for remote tower systems design". In: *2017 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pp. 1D2–1–1D2–9. DOI: 10.1109/ICNSURV.2017.8011891.

Marquardt, D. W. (1963). "An algorithm for least-squares estimation of nonlinear parameters". In: *Journal of the society for Industrial and Applied Mathematics* 11.2, pp. 431–441.

Mukherjee, D., Q. M. J. Wu, and T. Nguyen (2014). "Gaussian Mixture Model With Advanced Distance Measure Based on Support Weights and Histogram of Gradients for Background Suppression". In: *Industrial Informatics, IEEE Transactions on* 10, pp. 1086–1096. DOI: 10.1109/TII.2013.2294134.

Nischt, M. and R. Swaminathan (2009). "Self-calibration of asynchronized camera networks". In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 2164–2171. DOI: 10.1109/ICCVW.2009.5457548.

Noguchi, M. and T. Kato (2007). "Geometric and Timing Calibration for Unsynchronized Cameras Using Trajectories of a Moving Marker". In: *2007 IEEE Workshop on Applications of Computer Vision (WACV '07)*, pp. 20–20. DOI: 10.1109/WACV.2007.27.

Patino, L., T. Cane, A. Vallee, and J. Ferryman (2016). "PETS 2016: Dataset and Challenge". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1240–1247. DOI: 10.1109/CVPRW.2016.157.

Ristani, E., F. Solera, R. Zou, R. Cucchiara, and C. Tomasi (2016). "Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking". In: *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*.

Rosca, S., J. Suomalainen, H. Bartholomeus, and M. Herold (2018). "Comparing terrestrial laser scanning and unmanned aerial vehicle structure from motion to assess top of canopy structure in tropical forests". In: *Interface Focus* 8, p. 20170038. DOI: 10.1098/rsfs.2017.0038.

Rozantsev, A., S. N. Sinha, D. Dey, and P. Fua (2017). "Flight Dynamics-Based Recovery of a UAV Trajectory Using Ground Cameras". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: 10.1109/cvpr.2017.266. URL: http://dx.doi.org/10.1109/CVPR.2017.266.

Stanford CS231A Course Notes, 2019.

Tang, Z., M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. C. Anastasiu, and J.-N. Hwang (2019). "CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification". In: *CVPR 2019: IEEE Conference on Computer Vision and Pattern Recognition*.

Vo, M., S. G. Narasimhan, and Y. Sheikh (2016a). "Spatiotemporal Bundle Adjustment for Dynamic 3D Reconstruction". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vo, M., S. G. Narasimhan, and Y. Sheikh (2016b). "Spatiotemporal bundle adjustment for dynamic 3d reconstruction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1710–1718.

Zhang, X., Y. Chen, L. Yu, W. Wang, and Q. Wu (2017). "Three-Dimensional Modeling and Indoor Positioning for Urban Emergency Response". In: *ISPRS International Journal of Geo-Information* 6.7. ISSN: 2220-9964. DOI: 10.3390/ijgi6070214. URL: http://www.mdpi.com/2220-9964/6/7/214.

Zhang, Z., T. Tan, K. Huang, and Y. Wang (2013). "Practical Camera Calibration From Moving Objects for Traffic Scene Surveillance". In: *Circuits and Systems for Video Technology, IEEE Transactions on* 23, pp. 518–533. DOI: 10.1109/TCSVT.2012.2210670.

Zhenzhong Su. *Company Fixposition*. `https://www.fixposition.ch/`. Accessed: 2019-06-30.