2009-07-12

Construction of Large Geo-Referenced Mosaics from MAV Video and Telemetry Data

Benjamin Kurt Heiner
Brigham Young University - Provo

CONSTRUCTION OF LARGE GEO-REFERENCED MOSAICS FROM MAV VIDEO AND TELEMETRY DATA

by

Benjamin K. Heiner

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Brigham Young University

August 2009

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Benjamin K. Heiner

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date Clark N. Taylor, Chair

Date Randal W. Beard

Date Bryan S. Morse

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Benjamin K. Heiner in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

______________________ ______________________________

Date

Clark N. Taylor
Chair, Graduate Committee

Accepted for the Department

____

Michael J. Wirthlin
Graduate Coordinator

Accepted for the College

____

Alan R. Parkinson
Dean, Ira A. Fulton College of
Engineering and Technology

ABSTRACT

CONSTRUCTION OF LARGE GEO-REFERENCED MOSAICS FROM MAV VIDEO AND TELEMETRY DATA

Benjamin K. Heiner

Department of Electrical and Computer Engineering

Master of Science

Miniature Aerial Vehicles (MAVs) are quickly gaining acceptance as a platform for performing remote sensing or surveillance of remote areas. However, because MAVs are typically flown close to the ground (1000 feet or less in altitude), their field of view for any one image is relatively small. In addition, the context of the video (where and at what orientation are the objects being observed, the relationship between images) is unclear from any one image. To overcome these problems, we propose a geo-referenced mosaicing method that creates a mosaic from the captured images and geo-references the mosaic using information from the MAV IMU/GPS unit. Our method utilizes bundle adjustment within a constrained optimization framework and topology refinement. Using real MAV video, we have demonstrated our mosaic creation process on over 900 frames. Our method has been shown to produce the high quality mosaics to within 7m using tightly synchronized MAV telemetry data and to within 30m using only GPS information (i.e. no roll and pitch information).

ACKNOWLEDGMENTS

I would like to acknowledge the committee members, friends, and family who provided assistance in this research and made this thesis possible. First, I would like to thank my committee members, Dr. Taylor, Dr. Beard, and Dr. Morse, for taking the time to review this work and for helping me complete my research. In specific, I would like to thank Dr. Taylor for offering to be my graduate advisor and allowing me the opportunity to be in the MAGICC lab. His guidance and assistance have been essential to completion of this research and thesis. Furthermore, I would like to thank him for his assistance in improving my writing skills.

I am also very thankful to all the students in the MAGICC lab for helping me complete this research. They made this work and time spent at BYU enjoyable and fun. In specific, I would like to thank Bryce Ready, Evan Andersen, Travis Millet, and Greg Droge. First, Bryce thank you for always taking the time to answer my question, assist in this research, and being a friend. You are definitely a great asset in the MAGICC lab and a friend to each member in the lab. Second, Evan thank you for your many hours helping me get equipment working, understanding computer vision algorithms, completing class work and flight tests. Third, Travis thank you for your invaluable knowledge with the repairs of aircraft, expert piloting skills, assisting in class work, and for being a friend in the lab. Lastly, Greg thank you for providing me with the opportunity to help, for your assistance with this work, and reviewing this thesis.

I would also like to thank the people at the Air Force Research Laboratory (AFRL) in Rome New York for their part in this research. They provided me with a great opportunity to learn and greatly assisted in the selection of this work. In

specific, I would like to thank Ralph Kohler for getting me started on this topic and Dr. Linderman for his mentorship while at AFRL.

Finally, I would like to thank my family for their help and support. First, I would like to thank my wife Taryn for all of her support. She above all has been most understanding of the long hours spent completing this research, class work, and other activities. She has assisted in more way than can be stated here. Second, I would like to thank my brother, Jonathan Heiner, for his encouragement and help throughout my schooling here at BYU and at OIT. He has been an invaluable resource and friend.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, civilian and military agencies have increased their utilization of miniature unmanned aerial vehicles (MAVs - less than a 2m wingspan) in many information gathering missions, which include rural search and rescue, agricultural information gathering, and reconnaissance and surveillance. Due to their small size, MAVs are attractive platforms for executing these missions. MAVs can be deployed quickly and repeatedly, their small size simplifies storage and reduces their detectability, they have lower costs when compared to larger UAVs and human-operated aircraft, they enable operators to explore hazardous environments without risk of life, and they can obtain high-resolution sensor data (such as imagery) with low-cost sensors due to their low flight altitude.

While high-resolution imagery is easily obtained through the flight of a MAV, the utilization of the collected imagery can become a cumbersome task. During the flight of an MAV, hundreds of pictures can be collected by the MAV that must be analyzed by a user. For example, a ten-minute flight with imagery collected once every second will produce 600 images. Depending upon the amount of information present in the imagery and the desired task, the process of analyzing this small sequence of imagery can become overwhelming for a single user.

Furthermore, even when the user is looking at an image, the “context” of each image is not immediately apparent. In other words, the geo-location, orientation, and relationship of the image to other imagery are not always apparent. In order to overcome these problems, a single, large, integrated image (mosaic) can be created from the collected imagery. This mosaic provides a context rich view that allows the user to quickly analyze all of the visual data collected and reveals the spatial relationship between images.

This spatial relation can be improved by orthorectifying[1] the aerial imagery and relating each pixel in the mosaic to geodetic coordinates (geo-referencing the mosaic.) This allows measurements in the mosaic to relate directly to those in the world. This thesis focuses on the task of creating and orthorectified geo-referenced mosaics that are "visually appealing" and context rich using MAV imagery and sensor data (GPS and IMU readings) from an onboard autopilot.

The remainder of this chapter is divided into four subsections which are as follows. In Section 1.1, we present a survey of literature relating to the work presented in this thesis. In Section 1.2, we discuss the novel contribution of the work presented in this thesis. In Section 1.3, we present a overview of the proposed system. In Section 1.4, we detail the organization of the remaining chapters in the thesis.

1.1 Related Work

The discussion of literature related to this thesis is divided into three subsections, which discuss the creation of geo-referenced mosaics using IMU and GPS sensors (i.e. pose information), reference imagery, and a bundle adjustment (BA) process.

1.1.1 Creation of Geo-referenced Mosaics Using Pose Data

One method of producing the georeferencing information for the mosaic is to use the initial pose estimates, from the IMU and GPS, to project the MAV imagery onto a common coordinate system. This method has been shown to produce highly accurate mosaics [2]. This method, however, is not well suited to MAVs. The small size of an MAV airframe limits the weight, size, and power available for payloads, necessitating the use of low-quality sensors for the IMU, leading to noisy pose estimates.

[1] Orthorectify - process of geometrically correcting an aerial photograph so that its scale is uniform and can be measured like on a map.

The effects of noisy pose estimation can be seen in Figure 1.1. In this figure, several MAV images are placed in a geo-referenced mosaic using only the information obtained from the IMU and GPS unit. The placement of the images varies significantly with noise in the pose estimates, causing discontinuities between consecutive video frames. This is especially apparent in two particular areas of this figure, which are the runway and base complex. Notice how a single, continuous runway is no longer continuous and how the base buildings are duplicated and disjoint.

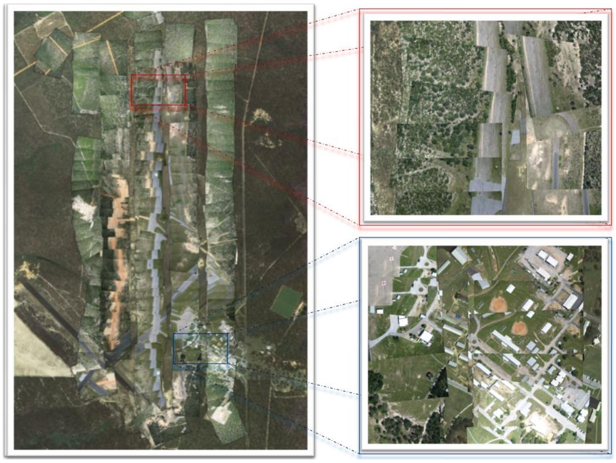


Figure 1.1: This figure shows a sequence of MAV images that have been placed in a geo-referenced mosaic using only the information from the IMU and GPS unit. Note how inaccuracies in the pose estimates cause discontinuities in the global mosaic. This is especially apparent in two particular areas of this figure, which are the runway and base complex.

Other techniques which create a continuous mosaic from the imagery, but geo-reference from the pose data of a single frame ([3, 4]) similarly suffer from poor performance due to inaccurate pose estimates from the IMU/GPS system on-board the MAV. This is illustrated in Figure 1.2. In [5, 1], this is addressed via the use of a Unscented Kalman Filtering (UKF) framework that fuses GPS/IMU pose estimates with perspective transformations computed between images in an MAV sequence. By using the perspective transformations computed between images, they are able to improve the accuracy of MAV pose estimates coming from a GPS/IMU system. This framework has been used to produce geo-referenced mosaics in real time. However, this framework can only create small mosaics, durations of 30 to 60 seconds. This is due to the accumulation of errors in the estimated frame-to-frame perspective transformations.

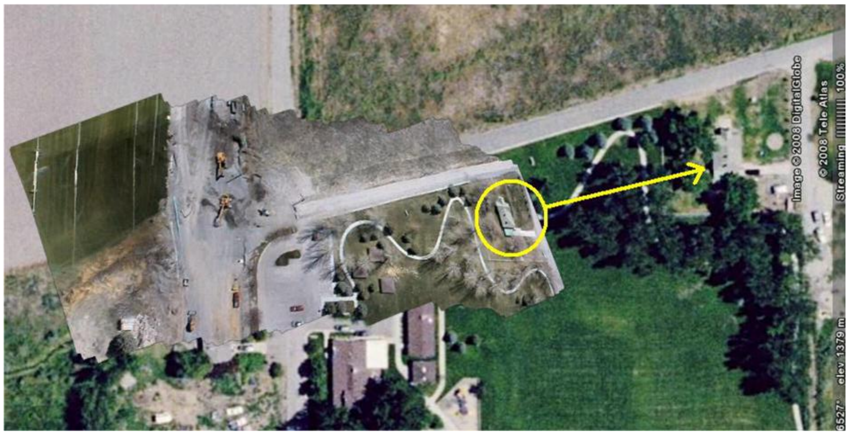
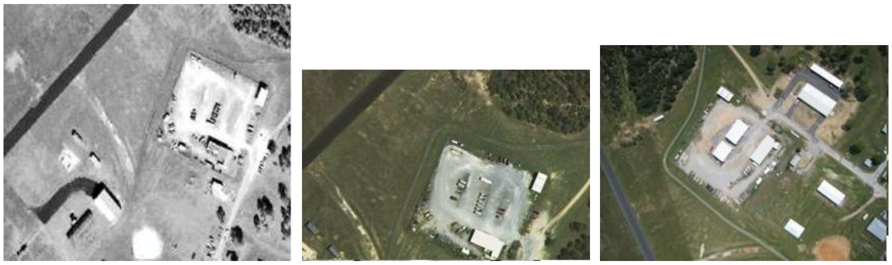


Figure 1.2: This figure shows a geo-referenced mosaic created using a single pose to geo-locate the mosaic. As can be seen the process is sensitive to errors in the used pose estimate.

1.1.2 Creation of Geo-referenced Mosaics Using Reference Imagery

In order to overcome the effects of noisy pose estimates, [6, 7] creates a consistent frame-to-frame mosaic from the video stream and then registers the mosaic to a preexisting geo-referenced image. In [8], a series of perspective transformations are used to relate images in the video sequence to reference imagery. These perspective transformations are made up of frame-to-frame, frame-to-local, local-to-local, and local-to-reference registration. However, these methods are dependent on the availability of reference imagery, which is a significant assumption. In addition, when reference imagery is available for the region of interest, it may be out of date or have significant differences in appearance due to differing environmental conditions (i.e. lighting, structures, season of the year, etc), resolution, or imaging technologies (i.e. IR, EO, etc), significantly impairing the image-to-map registration process.



(a) CIB Imagery (b) Google Imagery (c) MAV Sample Image

Figure 1.3: This figure shows reference imagery that is out of date and has significant differences in appearance due to differing environmental conditions, resolution, or imaging technologies between the current image and the geo-referenced MAV image. (a) 1m CIB (Controlled Image Base - dataset of orthophotos, made from rectified grayscale aerial images.) Imagery distributed by National Geospatial Agency, dated 25 Aug 2001 (b) Imagery collected by Florida Department of Environmental Protection, Distributed by Google, dtd 2003 (c) Imagery collected on 9 August, 2008 at Ranger Camp Six

These issues are illustrated in Figure 1.3 using the two most recent reference images available for a given MAV image. It is quickly apparent that the reference imagery, Subfigures 1.3(a) and 1.3(b), are out of date. These images are missing four main structures and appear to have dirt roads instead of paved roads leading to the center complex. Furthermore, Subfigure 1.3(a) has been taken using a different imaging device and at a reduced resolution. As can be seen, it is a significant assumption that reference imagery existences that is up to date and compatible with captured imagery.

1.1.3 Creation of Geo-referenced Mosaics Using BA Process

An alternate approach to building the mosaic is to simultaneously estimate the camera poses and geo-referenced feature locations from overlapping images in the video sequence, a process known as Bundle Adjustment (BA) [9]. In a general sense, this process functions by identifying a set of parameters to be estimated, and a set of measured values which are to be modeled as nonlinear functions of those parameters. Most BA methods use the location of salient feature points as the parameter space to be estimated. This approach minimizes the reprojection error between the image locations of several observed and predicted image points and has been shown to produce high quality mosaics [10, 11].

Despite the high-quality mosaics product produced by these methods, they have a high computational cost associated with the solving of their normal equations, equations of the form $A\vec{x} = \vec{b}$. In these equations, the $\vec{x}$ is the parameter space and $\vec{b}$ is measurement space. Furthermore they do not directly make use of geographic information obtained from IMU/GPS systems attached to the camera in the BA method.

In order to reduce the cost of solving the normal equations present in the BA process, [12] modified the BA algorithm so that it does not explicitly estimate the location of each feature. Instead, they define a minimum distance between a pair of rays emanating from two cameras in the direction of an observed feature. This reduces the number of parameters that need to be estimated and subsequently the computational cost of solving the normal equations. Alternately in [13], the BA algorithm is modified so that it estimates point locations generated using projective transformations between images instead of direct feature locations. In other words, they distill the information contained in point locations into perspective transformation that are estimated using feature correspondences. These perspective transformations are then used to generate control points in the world space. This approach is based on the existence of a frame-to-frame mapping between images (referred to as a homography [14]) and assumes that the scene is planar (a universal assumption among aerial mosaicing applications). While both of these approaches reduce the computational complexity associated with solving the normal equations, they do not directly make use of geographic information obtained from IMU/GPS systems in the BA method, which can improve the accuracy of the final product.

1.2 Contribution of This Thesis

In this thesis, we present a novel modification to the traditional BA method that creates geo-referenced mosaics that are comparable in quality to prior methods. This method makes direct use of pose information obtained from an IMU/GPS unit thereby allowing us to accurately geo-reference the mosaic to the world. Our approach is similar to [13] in that we assume a planar scene and that information contained in point correspondences are distiled into homographies.

The key insight used to set up our bundle adjustment problem is that the frame-to-frame homography mapping is also a function of the poses of the camera at the time the images were taken. Because they define a “visually appealing” mosaic, the computed perspective mappings can be treated as constraints on the true pose estimates. A measurement of this pose is directly computed by the IMU/GPS system on-board the MAV. If we assume that for each frame an estimate of the camera location is returned by the IMU/GPS system, then a constrained optimization routine can be used to determine the set of camera poses that are closest to the measured pose values while meeting the constraints imposed by the frame-to-frame homographies. This approach has three main advantages which are discussed bellow.

First, similar to that of [12, 13], our method reduces the cost associated with solving the normal equations. Traditional bundle adjustment requires a parameter space that is of size $3M + 7N$, where M is the number of features throughout the entire video that were tracked, and N is the number of images being used to create the mosaic. Our method's parameter vector is of size $7N + 7$, a significant reduction in size. Similarly, the measurement space in a traditional BA is $\sum 2k_{ii}$ where k_i is the number of images in which feature i appears. Because $M > N$, our method's measurement space of size $8(N - 1) + 7N + 7$ is significantly smaller than traditional BA. This reduction in the size of the measurement and parameter space reduces the computational cost associated with solving the BA normal equations.

Second, our method uses a two stage filter process to remove outliers. This comes from using homographies instead of feature locations in our BA cost function. As stated in Section 1.1.3, traditional BA methods use the location of salient feature points as the parameter space to be estimated. These feature locations can contain outliers, which can degrade the accuracy of the BA method. While a RANSAC algorithm can be applied to the parameter space of traditional BA methods, it can become costly as the number of features in the parameter space increase. However, our algorithm explicitly addresses the problem of outliers during the selection of homographies to be used in the BA cost function. This is accomplished using a two-stage filter process. The first stage uses a RANSAC [15] process to remove outliers during the calculation of candidate homographies, which minimizes the possibility of calculating bad homographies. The second stage evaluates the registration of image pairs generated via the candidate homographies. If a registration is blurred beyond a predetermined threshold, determined using a focus metric, then it is removed from consideration. This allows us to remove homographies that are still dominated by outliers not detected in the first stage thereby further reducing the possibility of outliers in the BA cost function.

The third benefit comes from the incorporation of IMU/GPS pose information in the BA cost function. By including the global location information that is available from the MAV's IMU/GPS system in our BA, the most probable geo-location of the mosaic, as a function of pose parameters, is explicitly computed in our constrained optimization framework.

1.3 Process Overview

The proposed process for creating geo-referenced mosaics, as shown in Figure 1.4, is divided into four segments: data collection, initial mosaic creation, iterative refinement of mosaic, and post processing. These segments are summarized in the following subsections.

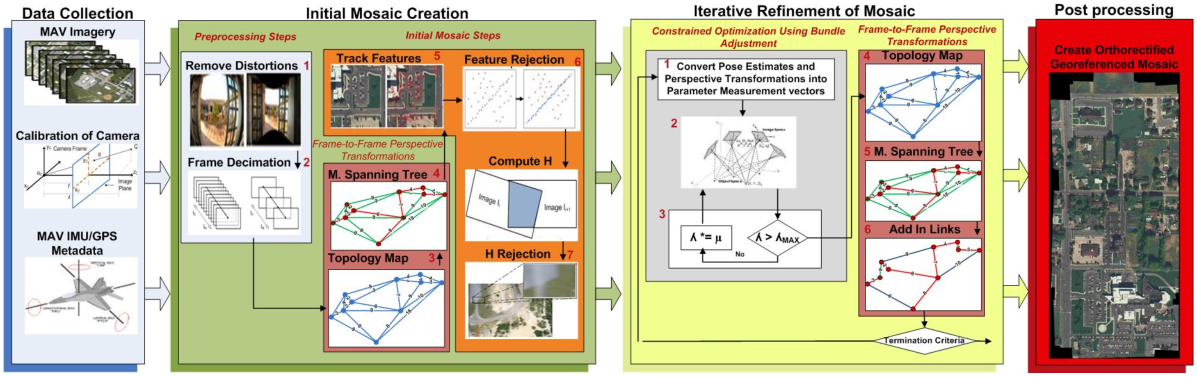


Figure 1.4: This figure shows a flow digram of the mosaic creation process.

1.3.1 Data Collection

The data collection segment begins the process and encompasses the collection of IMU/GPS pose information and imagery from an MAV flight. Once a flight is complete, the imagery is associated with the pose information and tagged. The association process can be accomplished by synchronizing each telemetry packet (contains pose information) GPS time stamp with an images received GPS time stamp or EXIF[2] creation time. In addition to the tagging of imagery, a camera calibration is generated for the current camera setup if one does not already exist.

1.3.2 Initial Mosaic Creation

The initial mosaic creation segment begins with the removal of lens distortions from GPS tagged imagery using function calls from OpenCV [16]. Once the lens distortions are removed, the images are sent to a frame decimation unit which removes other sources of noise caused by the acquisition process (i.e. transmision noise and blurring) and redundant imagery which provide little to no new information about the environment. The frame decimation unit then returns a subset of the MAV imagery which best represents the area of interest. These preprocessing steps are highlighted in blue and are shown in Figure 1.4.

Given this imagery subset, we then build a minimum spanning tree[3] (MST) that maximizes the overlap between candidate image pairs. To compute the MST, we first create a fully connected graph using pose information, imagery, and a reference frame. The nodes in the graph represent the projected centers of each image onto the reference frame. The edges in the graph represent possible frame-to-frame registrations. Each edge is inversely weighted according to its overlap percentage (i.e. $w \propto \%^{-1}$). Once the graph is created, we use Prim's algorithm [17] to compute the MST. The units used in the selection of image pairs are highlighted in purple and are shown in Figure 1.4.

[2] Exchangeable Image File Format (EXIF) - standard for storing interchange information in image files.

[3] Minimum Spanning Tree (MST)- a spanning tree, tree connecting all nodes in a graph, with weight less than or equal to the weight of every other spanning tree.

We then generate the candidate frame-to-frame transformations for each edge in the MST. To compute the transformation matrix, we first find features in each image using a SURF detector/descriptor [18]. The features in the first image are then placed into a K-d tree and associated with features in the second image using a best-bin-first search algorithm [19]. Once feature correspondences are found, they are refined using the RANSAC algorithm [15]. After removing outlier correspondences, the transformation matrix is computed using the singular value decomposition (SVD) as described in [14]. The resulting composite image from each pair of two images is then generated using the newly calculated homographies. If a composite image is blurred beyond a predetermined threshold then it is removed. At the end of this segment, we now have a set of estimated perspective transformations for each image pair in the subset of images that will be used in the BA method. This process is shown in orange in Figure 1.4.

1.3.3 Iterative Refinement

In the iterative refinement segment of our process, we find the optimal MAV pose estimates that align the captured images to create a high-quality mosaic while minimizing the difference in the pose estimates to that of the measured poses. The main algorithm used to find the optimal pose estimates is a constrained optimization process using bundle adjustment. The constrained optimization procedure treats the estimated feature-based frame-to-frame perspective transformations, homographies, as constraints to ensure that the final mosaic is an image of high quality. The homographies are determined during run time using the homographies found in the previous section and homographies generated using a reduced topology map (graph).

The segment starts by using the homographies found in the previous section as inputs into the BA method. Once the BA method is completed, we create a reduced topology map using the new pose estimates. The reduced topology map is a mixture of the topology graph and minimum spanning tree explained in the previous section. This map is then used to create a different set of homographies to be used in the in the BA method. This process is repeated until a maximum number of iterations are reached.

1.3.4 Post Processing

Upon completing the iterative refinement of the mosaic as described in Subsection 1.3.3, a single, large, integrated geo-referenced mosaic of the region of interest is created using the optimized pose estimates. This is done via the use of a virtual camera centered one meter above the desired area. Using the virtual camera and optimized poses, we project the MAV imagery onto a global coordinate system. We then use a multi-resolution blending algorithm to combine images and place them on the reference frame.

1.4 Thesis Organization

The remainder of the thesis is organized as follows. In Chapter 2, we describe the initial mosaic creation segment in more detail with the exception of the graph and MST units. In Chapter 3, we describe the iterative refinement segment in more detail with a focus on our BA method. This chapter also covers the creation of the minimum spanning trees (MST) and topology graph. In Chapter 4, we present the results of our BA process. In Chapter 5, we conclude the thesis and give comments on future work to extend and improve the work presented in thesis.

Chapter 2

Initial Mosaic Creation

In this chapter we describe the initial mosaic creation segment introduced in Section 1.3 in more detail with the exception of the graph and MST units (described in Section 3.1). These units are temporarily replaced by assuming that consecutive images captured by the MAV have sufficient overlap to enable creation of a mosaic. This segment generates the subset of MAV imagery and frame-to-frame perspective transformations that can be used by later steps to create a high quality geo-referenced mosaic. The segment is shown in Figure 2.1 and can be broken into six sub-steps which are described in the following sections.

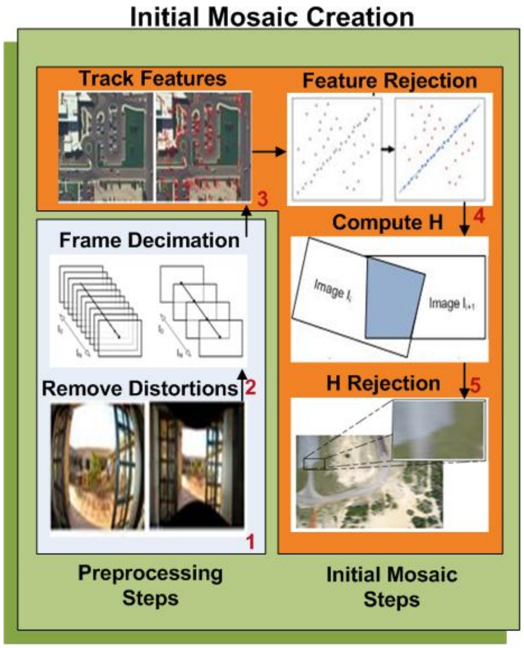
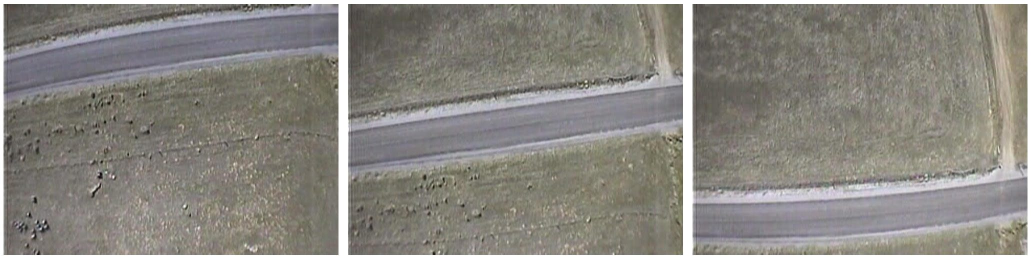


Figure 2.1: This figure show the initial mosaic creation segement with the exception of the graph and MST units.

2.1 Lens Distortion Correction

The first step in the mosaic creation process is the removal of the lens distortions that deteriorate the positional accuracy of points on the image plane. Two types of perspective distortions exist: radial and tangential. Radial distortions are deformations of the image along radial lines around the principal point, and tangential distortions are deformations perpendicular to radially distorted points. This is illustrated in Figure 2.3(d). Radial and tangential distortions occur when light rays passing through a lens are bent, thereby changing their directions, and intersecting the image plane at positions that deviate from the true rectilinear projection.

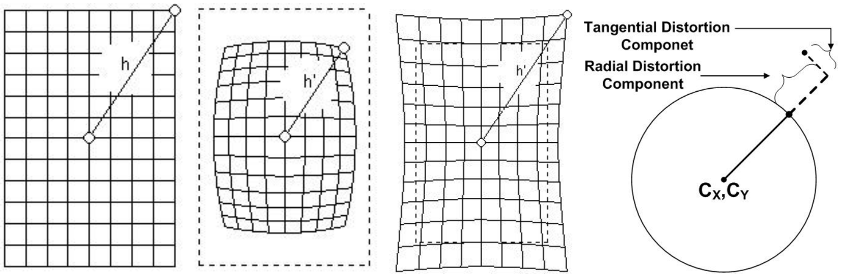


(a) Frame 2620 (b) Frame 2650 (c) Frame 2700

Figure 2.2: This figure shows the effects of lens distortion on the UAV images. All images show the same road segment from different viewpoints in the video.

As can be seen in Figure 2.2, the distortions in the image are easily recognized because they affect the way in which lines are perceived in the entire image. The manifestations of lens distortions fall into two main categories: barrel and pincushion. In the case of barrel distortions, straight lines appear to curve around the principal point giving the impression that they were imaged on the surface of a sphere instead of a plane. In the case of pincushion distortions, straight lines appear to curve away from the principal point. This is illustrated in Figure 2.3. These distortions cause the estimation of a perspective transformation to fail.

The effects of radial lens distortion throughout an image, I_d, can be approximated using a polynomial[20]. The distortion model used to determine coefficients



(a) (b) (c) (d)

Figure 2.3: This figure is an Illustration of the lens distortion categories. (a) Normal Image coordinates (b) Barrel Image coordinates (c) Pincushion Image coordinates (c) Radial and tangential distortions

associated with radial and tangential lens distortion is

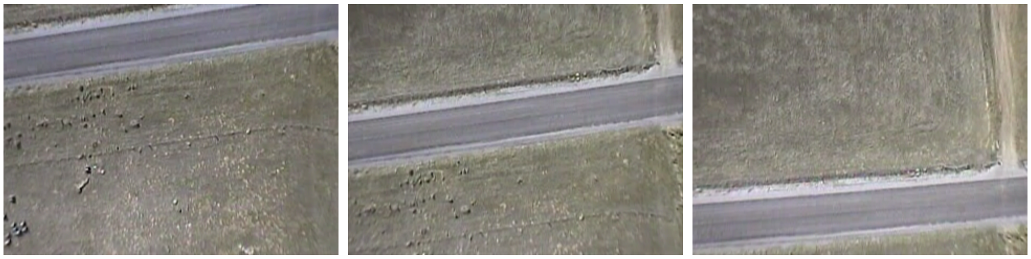
$$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} x_d\left(1 + k_1 r^2 + k_2 r^4\right) + 2p_1 x_d y_d + p_2\left(r^2 + 2x_d^2\right) \\ y_d\left(1 + k_1 r^2 + k_2 r^4\right) + 2p_2 x_d y_d + p_1\left(r^2 + 2y_d^2\right) \end{bmatrix}, \tag{2.1}$$

where $l_u = [x_u, y_u]^T$ is the location of a undistorted image pixel, $l_d = [x_d,\, y_d]^T$ is the location of a distorted image pixel, $C_R = [k_1, k_2]$ is the radial distortion coefficients, $C_T = [p_1, p_2]$is the tangential coefficients, and $r = x_d^2 + y_d^2$. The distorted image pixel locations x_d and y_d are calculated as

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} \frac{j-c_x}{f_x} \\ \frac{i-c_y}{f_y} \end{bmatrix}, \tag{2.2}$$

using the intrinsic camera parameters that define the internal geometry of a camera. The i and j in equation 2.2 are the row and column numbers of the pixel. The c_x and the c_y represent the center of the imaging area. The f_x and f_y represent the focal length in x and y direction. In this thesis, the intrinsic camera parameters are calculated using the calibration functions in OpenCV, a open source computer vision libraryintel06open.

An undistorted image can be created by creating a blank image I_u with the same proportions as I_d. Using Equation (2.1) we can transform distorted image location l_d to undistorted locations l_u. Using bi-linear interpolation we can compute



(a) Frame 2620 (b) Frame 2650 (c) Frame 2700

Figure 2.4: This figure shows the results of removing the effects of lens distortion on the MAV images. All images show the same road segment from different viewpoints in the video.

the intensity values of l_u and map them to the blank image I_u. Example results of this process can be seen in Figure 2.4.

2.2 Frame Decimation

It is relatively easy to acquire a large sequence of images for a desired region that in turn can be used to construct a mosaic. For example, a ten-minute flight with imagery collected thirty times every second will produce 18,000 images. However, these larger image sequences needlessly complicate the mosaic creation process. Larger input sequences acquired at higher frame rates typically have imagery that provides little to no new information about the environment. Therefore it is desirable to use a preprocessing step, called frame decimation [21], to produce a sparse but sufficient set of views for the creation of the mosaic which keeps the size of the problem manageable, regardless of the input frame rate. Incorporating such a preprocessing mechanism has a couple of advantages. The most important advantage is that the computational complexity associated with creating the mosaic is reduced (e.g. the complexity of the normal equation is reduced). Another benefit is that problems caused by bad focus values or transmission noise can sometimes be avoided.

There are a variety of ways to design this frame decimation unit. The easiest way is to simply reduce the frame rate by selecting every i^{th} image in the image sequence. However, this approach is undesirable because it can lead to the selection of blurred images and requires the frame rate to be entered by the user or preset by the software. In [21], an alternate approach is presented where the preprocessing unit automatically selects a subset of images with sufficient image quality and which provide new information so that the algorithmic results are improved. Their algorithm evaluates images based on a focus (sharpness) metric [22, 13, 23, 21] and incorporates a boundary detection mechanism used to determine overlap in images.

The implementation used in this thesis is a slight variation of [21] and is described in the following subsections. In Subsection 2.2.1, the frame decimation process is described. In Subsection 2.2.2, we describe the process of detecting image boundaries. In Subsection 2.2.3, we describe the focus metrics used in this process and others.

2.2.1 Decimation Process

The frame decimation process starts by measuring the quality of each MAV image using the Energy of Laplacian metric described in Section 2.2.3. This metric gives a quantitative value that describes the focus (sharpness) attribute of an image. Once all quality values are computed, we then iteratively detect image regions in the MAV image sequence and select an image in each region to represent it. These regions are found as described in Section 2.2.2.

The iterative process is performed using the temporal sequence of the MAV imagery and begins by finding the image region defined by the first image. Once the image region is found, we then select the image with the highest quality measure within a threshold range to represent that image region. This process is then repeated using the selected image from the previous iteration as the new center image in the region detection process. The iterative process is finished when all the MAV images have been processed.

This process is also shown in algorithmic form above to avoid confusion. The original MAV image sequence is represented by the set $\mathcal{I}$, the resulting decimated image sequence is represented by $\mathcal{J}$, and the image regions are represented by R. Lower case letters represent an image in a given set.

Algorithm 1 Frame Decimation Process

```
  Input: $\mathcal{I} := \{i | i = 0 \cdots n-1\}$
Initilize: $\mathcal{J} := \emptyset$

1. for each $(i \in \mathcal{I})$ compute quality measure $q$.
3. Set $i$ equal to the first image in $\mathcal{I}$ using temporal ordering.
4. Find the first image region, $R \subset \mathcal{I}$, centered at $i$.
5. Set $i$ equal to the image with the higest $q$ in $R$.
5. Add $i$ to $\mathcal{J}$, $i \subset \mathcal{J}$.
6. If $R$ contains the last image in the temporal order continue else goto step 4

Return: $\mathcal{J} \subset \mathcal{I}$
```

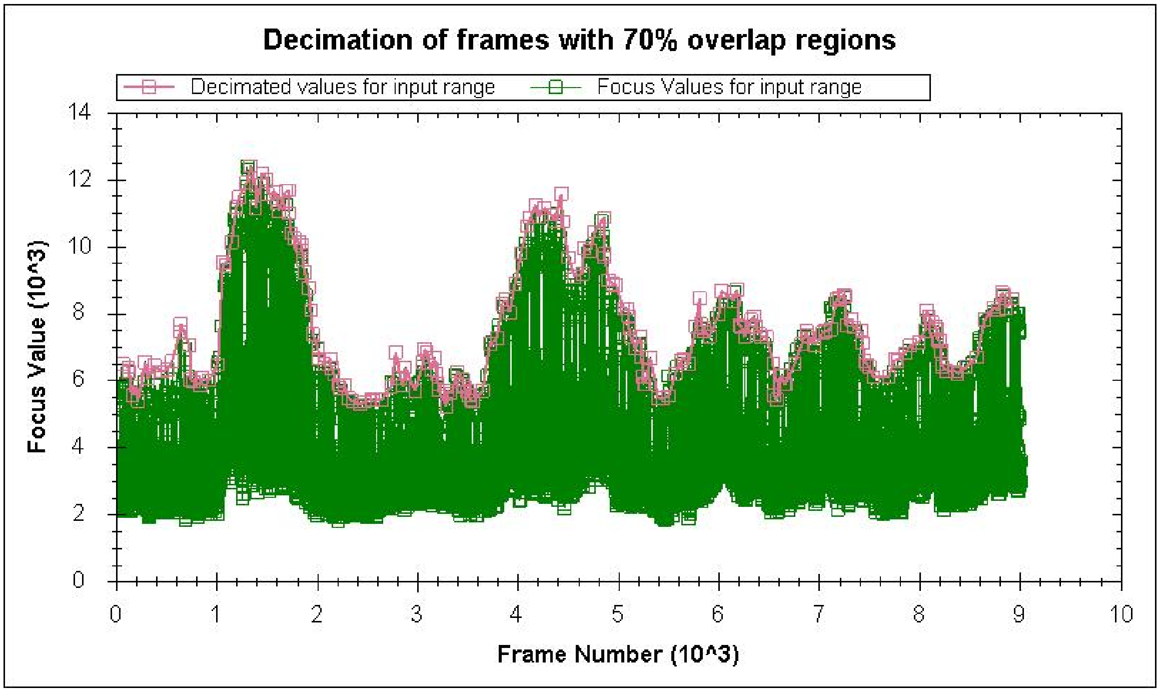


Figure 2.5: This figure shows the results of running the frame decimation process on a sequence of 9,000 images. The resulting sparse set of views contains over 200 images.

In Figure 2.5, we show the results of running the frame decimation process on an image set of 9,000 images. The green in this figure represent the focus values for each image in the set while the red represents the selected images for each boundary in the set. The x-axis spans the frame numbers found in the video, while the y-axis plots the focus values, with low focus numbers representing ”blurry” images. As can be seen, the frame decimation process significantly reduces the number of images to be processed. Furthermore, the process is successful in only selecting images that are clear. This increases the probability of finding a good perspective transformation between image pairs, explained in Section 2.3.2.

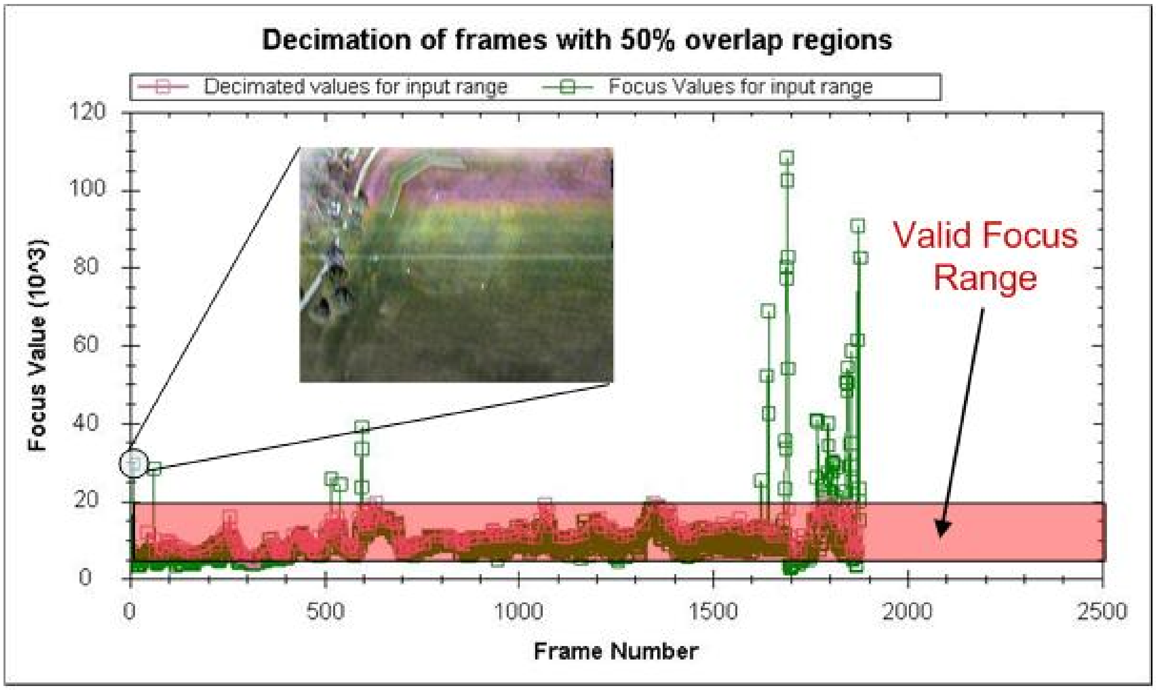


Figure 2.6: This figure shows the results of running the frame decimation process on a sequence of over 2,000 images. As can be seen the frame decimation can also be used to reject images that have significant noise due to transmission.

As mention above, this process can also be used to remove images that have been corrupted by transmission noise. This is because transmission noise in an image increases its focus measure. This is shown in Figure 2.6. In this figure, we show the results of the frame decimation process on a sequence of over 2000 images, with some images containing transmission noise. The highlighted red region in this figure represents images that are free of transmission noise and significant blurring. As can be seen, the frame decimation process can remove images that contain transmission noise by applying an upper limit on accepted focus values.

2.2.2 Image Region Detection

Image region detection is crucial to the success of the frame decimation process. A image region is defined as a temporal grouping of images with an overlap percentage exceeding a minimum threshold. This overlap is calculated in the image coordinate frame. To compute the overlap area, we first project the four corners of each image onto a center image using their pose information and a perspective transformation, explained in Section 2.3.1. The resulting projections (polygons) are then superimposed onto each other to determine the overlap percentage of image pairs. Images with overlap percentages above a user set threshold (typically above 50%) are grouped together to form a image region.

2.2.3 Focus Metric

The quality evaluation of an image is usually subjective and based on its perceived blurring. However, such approaches are laborious and lack robustness. In [22, 13, 23] a number of sharpness metrics are described that objectively measure the amount of blurr present in an image. The sharpness metric used in this thesis is the Energy of Laplacian metric because it is smooth, has a sharp maximum, and little to no local maxima. This is illustrated in Figure 2.7(c). The figure shows the focus values of a MAV image as it is blurred using a Gaussian function. As can be seen, the values quickly decrease as the image is blurred.

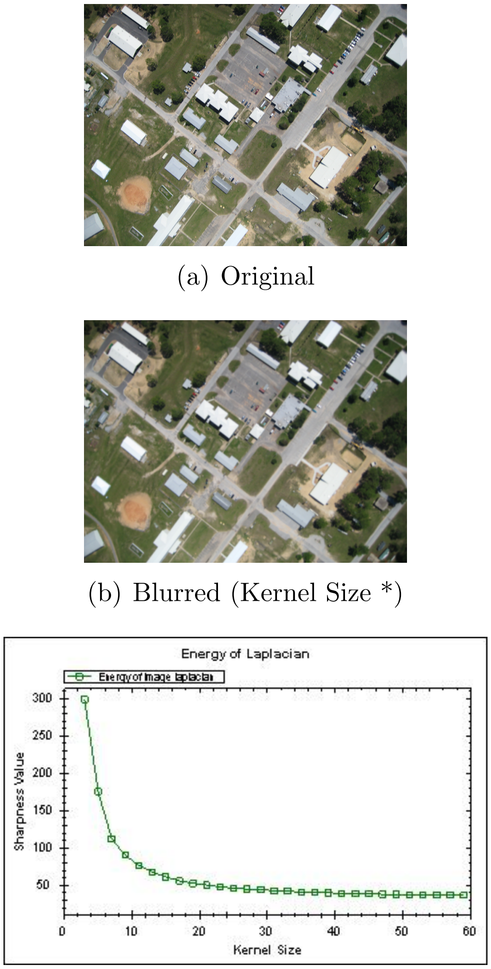
The Energy of Laplacian focus metric is defined as

$$EL\left(I\right) = \frac{1}{n} \sum_{x} \sum_{y} \left(g_{xx} + g_{yy}\right)^2, \tag{2.3}$$

where

$$\begin{aligned} g_{xx} + g_{yy} \quad = \quad & -I\left(x-1, y-1\right) - 4I\left(x-1, y\right) - I\left(x-1, y+1\right) \\ & -4I\left(x, y-1\right) - 20I\left(x, y\right) - I\left(x, y+1\right) \\ & -I\left(x+1, y-1\right) - 4I\left(x+1, y\right) - I\left(x+1, y+1\right), \end{aligned}$$

I is the whole image domain except for the image boundaries, and n is the number of pixels in an image.



(a) Original

(b) Blurred (Kernel Size *)

(c)

Figure 2.7: This figure shows the effectiveness of the Energy of Laplacian focus metric.

2.3 Image Alignment

This section describes the two methods we use for computing a perspective transformation (alignment) between two images. The first method explained is derived using a rigid body transformation and pose information. The second method explained uses feature correspondences and a full projective transformation model (8 free parameters). Furthermore, this section also explains a homography rejection mechanism which uses focus metrics.

2.3.1 Pose Based Alignment

Coordinate Frames

In this section we describe the various coordinate frames used to define the MAVs pose and rigid body transformations. The coordinate frames are discussed in a sequential order going from the world frame, to the body frame, to the camera frame, and to the lastly the image frame as illustrated in Figure 2.8. Each of these coordinate frames are combined to construct the rotation and translation used in the next section.

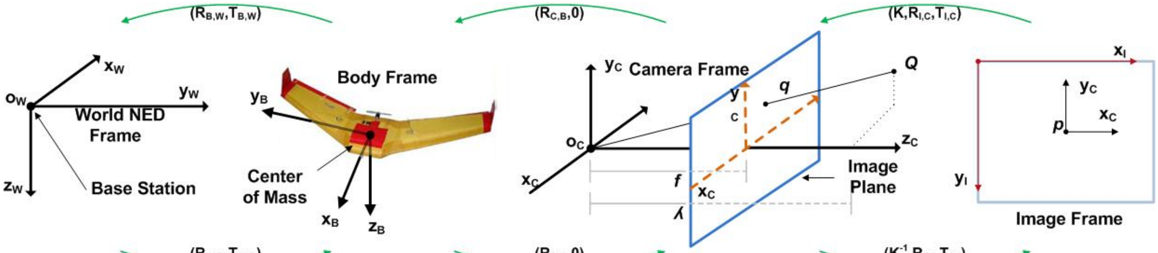


Figure 2.8: This figure illustrates the rigid body transformations between the world frame and the image frame.

World Frame - The world coordinate system used to encode the MAVs positional information is a north, east, down (NED) local Cartesian coordinate system. The NED coordinate system is setup such that the north axis is labeled x, the east axis y, and the down axis z. In this local coordinate system positive altitudes translate into a negative z. The unit of measure for the NED coordinate system is meters and has an origin defined by the base station of the autopilot.

Body Frame - The MAVs coordinate system, known as the MAVs body frame, originates at its center of mass with the x-axis pointing out of the nose, the y-axis pointing out of the right wing, and the z-axis pointing out of the belly of the aircraft. This is shown in Figure 2.9. The unit of measure for this system is also in meters. The transformation between the world frame and body frame is composed of a translation and rotation. The rotational information encodes a series of coordinate rotations and can be represented in several different ways, including Euler angles(ψRoll,θPitch, ψYaw), Direction Cosine Matrices (3×3matrix), or quaternion (4 element column vector).

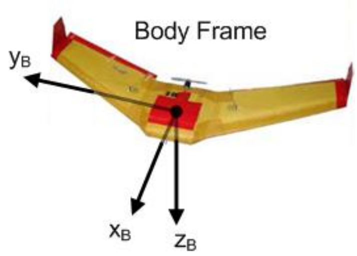
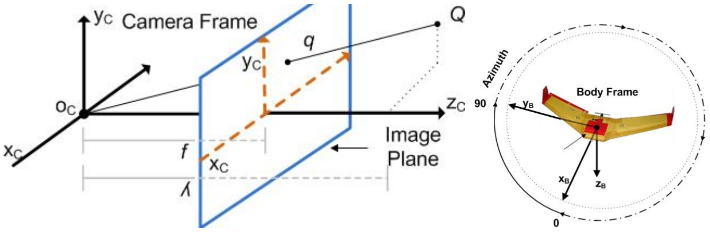


Figure 2.9: This figure shows the body frame of the MAV.

In our implementation, we use the quaternion representation because of its simplicity, mathematical ease, and lack of any singularities. A quaternion representation provides a simple way to compute the distance between two rotations. Furthermore, it is easy to meet the unity norm constraint when applied to optimization algorithms by simply re-normalizing the quaternion after each iteration [24, 25] (i.e. iterations of the BA process described in Chapter 3). The conversion between different representations can be found in [24].

Camera Frame - The rigid body transformation from any arbitrary coordinate frame to another is typically defined by a rotation and translation. However, the transformation of the body frame to the camera frame only requires a rotation. This is because the cameras location is assumed to be located at the MAVs center of mass (i.e. $\vec{0}$). The displacement from the center of mass is negligible compared to distances in the cameras field-of-view. The cameras x-axis goes to the right and the y-axis goes towards the top of the camera. This is illustrated in Figure 2.10.



(a) (b)

Figure 2.10: This figure illustrates the cameras coordinate frame and perspective model.

In the case of this thesis, we are using a fixed camera location which is rotated 90 degress in the azimuth (az) direction. This allows us to define the rotation from the body frame to the camera frame as a single rotation which is calculated as

$$R_{az} = \begin{bmatrix} cos\,(az) & -sin\,(az) & 0 \\ sin\,(az) & cos\,(az) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.4}$$

Image Frame - The most convenient method for representing image locations is pixel coordinates. In this coordinate system the x-axis goes through the right side of the image and represents the columns, the y-axis goes towards the bottom of the image and represents the rows, and the origin is located at the top left corner of the image. This is show in Figure 2.11. The unit of measure for this system is pixels while the word frame unit of measure is in meters. Therefore we need a transformation that takes the world 3D points and maps them to the image 2D points.

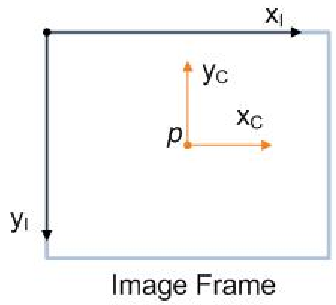


Figure 2.11: This figure shows the image frame in reference to the camera frame.

This is accomplished using a camera calibration matrix which is defined as

$$K = \begin{bmatrix} fs_x & fs_\theta & c_x \\ 0 & fs_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.5}$$

where f is the focal length, s_x is the scaling in the x direction, s_y is the scaling in the y direction, s_θ is the pixel skew, c_x is the x translation, and y_x is the y translation. Using this matrix we can define the mapping from world 3D points to image 2D points as

$$x_I = Kx_c, \tag{2.6}$$

where x_I is the point in the image and x_c is the point in the world.

Computation of H from the MAV Pose

As discussed in the Introduction, it is possible to derive a function that maps pose estimates into a homography matrix. While this function can be derived using pre-existing techniques, we describe it in more detail here to provide a complete description of our system.

To derive the matrix $\mathbf{H}_{i \leftarrow j}$ from world poses, we require the pose estimates $\vec{P}_i$ and $\vec{P}_j$. Because we are also estimating a bias quaternion, the first step of computing $\mathbf{H}$ is to compose the quaternion in the pose estimates with the bias quaternion (i.e.

constant error in rotation). These modified poses are used throughout the rest of this section without explicitly stating that the attitude estimates have been biased.

From rigid body motion, we can define the relation between the camera coordinate frame f_c and the world frame f_w as

$$\vec{X}_c = \mathbf{R}_{w\rightarrow c}(\vec{q})\vec{X}_w + T_c, \tag{2.7}$$

where $\vec{X}_c = [x, y, z]^T$ is a point in f_c, $R_{w\rightarrow c}$ is a rotation matrix (defined as a function of $\vec{q}$ in the pose estimate), and T_c is the location of the origin of f_c in f_w. Solving for $\vec{X}_w$ in terms of f_{cj} and substituting it into equation 2.7, we obtain

$$\vec{X}_{ci} \quad = \quad \mathbf{R}_{w\rightarrow ci}\left(\mathbf{R}_{w\leftarrow cj}\vec{X}_{cj} + T_{cj} - T_{ci}\right), \tag{2.8}$$

which is the rigid body transformation from f_{cj} to f_{ci}. Note that this equation is general for all points in f_{cj}. We will now add the constraint that all points we are interested on lie on a plane.

Let π be a planar surface where all $\vec{X} \in \pi$ and $\vec{n}$ is a normalized vector orthogonal to π. The existence of π implies that

$$\left\langle \mathbf{R}_{w\rightarrow c}\vec{n}, \vec{X}_c \right\rangle \; = d \;\; , \quad \frac{1}{d}\left(\mathbf{R}_{w\rightarrow c}\vec{n}\right)^T \vec{X}_c = 1, \tag{2.9}$$

for all $\vec{X} \in \pi$, where d is the minimal distance of the camera from the plane. We can now substitute equation 2.9 into equation 2.8 yielding

$$\vec{X}_{ci} = \mathbf{R}_{w\rightarrow ci}\left(\mathbf{R}_{w\leftarrow cj}\vec{X}_{cj} + T_{ji}\frac{1}{d}\left(\mathbf{R}_{w\rightarrow c}\vec{n}\right)^T \vec{X}_{cj}\right), \tag{2.10}$$

where $T_{ji} = T_{cj} - T_{ci}$. Simplifying and adding in the calibration matrices K_{ci} and K_{cj} for the two cameras, we can now rewrite the equation for the perspective transformation matrix $H_{i\leftarrow j}$ as

$$\mathbf{H}_{i\leftarrow j} \quad = \quad \mathbf{K}_{ci}\mathbf{R}_{w\rightarrow ci}\left(I + T_{ji}\frac{1}{d}\vec{n}^T\right)\mathbf{R}_{w\leftarrow cj}\mathbf{K}_{cj}^{-1}. \tag{2.11}$$

This process is illustrated in Figure 2.12.

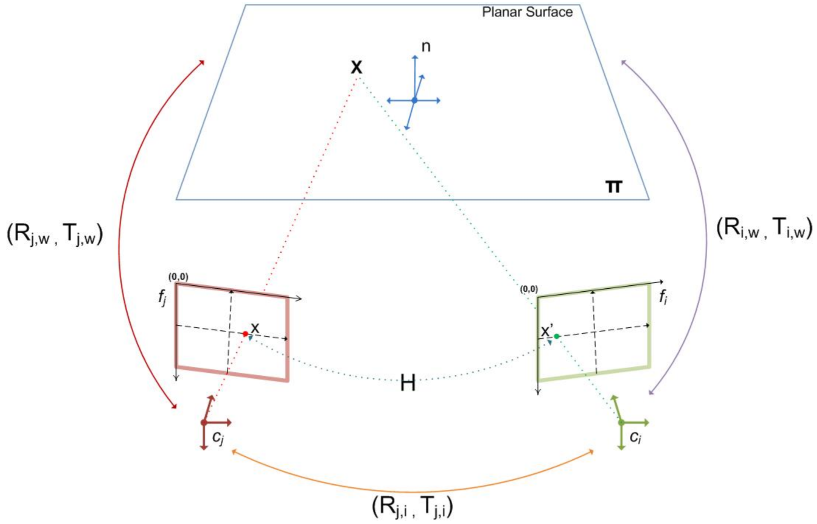


Figure 2.12: This figure illustrates the process discussed in Section 2.3.1.

2.3.2 Feature Based Alignment

The accurate alignment of adjacent images is a crucial step in obtaining geometrically correct mosaics. The best way to ensure that an accurate alignment is obtained would be to align each pixel location in the source image to that of the destination image. However, as the resolution of the imagery increases this method becomes impracticable because of its computational cost. A popular alternative to direct registration is feature-based approaches, which align images using a subset of the avaliable features. Such approaches are fast in comparison to direct registration and have a much lower computational cost.

However, there are a variety of feature representations to choose from, each of which has its own pros and cons. The selection of which depends on the application needs. In context of this thesis, we use the SURF(Speeded-Up Robust Features) representation [18] because of its robustness to changes in scale, rotation, noise, illumination, and perspective [26]. Furthermore, this feature representation performs well in wide baseline situations and provides a descriptor of the feature.

The following subsections describe SURF in more detail and how it is used to compute a perspective transformation.

SURF Fast-Hessian Detector

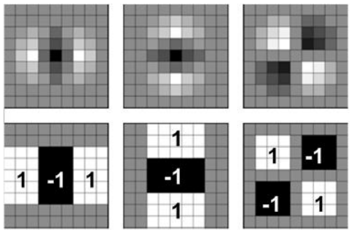


Figure 2.13: This figure shows the Laplacian of Gaussian Approximations using boxfilters. The top row represents the actual $\mathcal{L}_{xx}$, $\mathcal{L}_{yy}$, and $\mathcal{L}_{xy}$. The bottom row represents is the box filter versions of $\mathcal{L}_{xx}$, $\mathcal{L}_{yy}$, and $\mathcal{L}_{xy}$.

The SURF detector is based on the determinant of the Hessian matrix. The Hessian matrix as a function of space X and scale σ is calculated as

$$\mathcal{H}\left(X,\sigma\right) = \begin{bmatrix} \mathcal{L}_{xx}\left(X,\sigma\right) & \mathcal{L}_{xy}\left(X,\sigma\right) \\ \mathcal{L}_{yx}\left(X,\sigma\right) & \mathcal{L}_{yy}\left(X,\sigma\right) \end{bmatrix}, \tag{2.12}$$

where $\mathcal{L}_{xx}\left(x,\sigma\right)$ refers to the convolution of the second order Gaussian derivative $\frac{\partial^2 g(\sigma)}{\partial x^2}$, similarly for $\mathcal{L}_{xy}$ and $\mathcal{L}_{yy}$, and $X = [x, y]$ is a point in the image I. These deriviates are know as the Laplacian of Gaussians (LoG) and are aproximated using box filter representations of the respective kernels. These box filters are shown in Figure 2.13.

The location and scale of interest points are selected by relying on the determinant of the Hessian. The determinant is the the blob response at location $X = [x, y, \sigma]$ and is calculated as

$$det\left(\mathcal{H}_{approx}\right)) = \mathcal{D}_{xx}\mathcal{D}_{yy} - \left(0.9\mathcal{D}_{xy}\right)^2, \tag{2.13}$$

where $\mathcal{D}_{xx}$, $\mathcal{D}_{yy}$, and $\mathcal{D}_{xy}$ are the box filters shown in Figure 2.13.

Interest points are localized in the scale and image space by applying a non-maximum suppression in a 3x3x3 neighbourhood. Finally, the local maxima found using the approximate determinant are interpolated in scale and image space. This process is explained in detail in [18, 27]. Figure 2.14 shows the feature points found in two MAV images using the SURF detector.

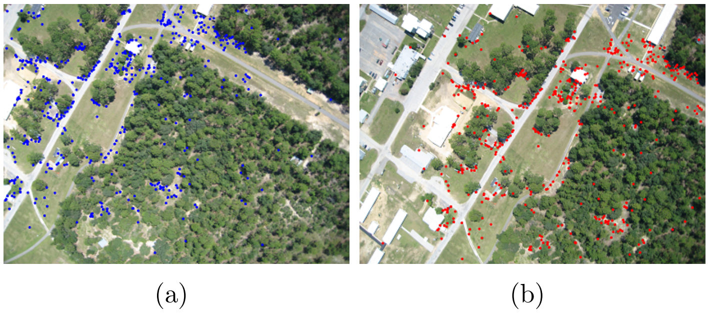

(a) (b)

Figure 2.14: This figure shows the results of the SURF feature detector when applied to two MAV images.

SURF Descriptor

The SURF descriptor describes how pixel intensities are distributed within a scale dependent neighborhood of each interest point detected by the Fast-Hessian. The process for extracting the descriptor is as follows. First, each detected interest point is assigned a unique orientation. The orientation is computed using Haar wavelet responses in both x and y directions. This allows the feature to be invariant to image rotations. Second, a square window around the interest point is constructed which contains the pixels that will form entries in the descriptor vector. The window size is 20σ and is oriented along the direction defined by the found orientation. Third, the window is then divided into 4x4 sub regions. A Haar wavelet of size 2σ is calculated for each subregion using 25 distributed sample points. This produces 4 values for each subregion with 64 values total to be used in the descriptor. The generated descriptor is now invariant to scale, rotation, noise, illumination, and perspective.

Feature Correspondences

The correspondence of SURF features is a relativly simple process made possible by its discriptor. Feature matching is done through a Euclidean-distance based nearest neighbor search. Given a feature point $X = [x, y]$ in the source image, the search finds the nearest neighbour of X in the set of destination points. Robustness to noise is added by rejecting points for which a ratio of the nearest neighbour distance to the second nearest neighbour distance is greater than some threshold. However, the task of finding the nearest Euclidean-distance neighbour can become costly as the number of features increase. This cost can be minimized using a K-d tree and a Best Bin First(BBF) search [19], which is the implementation used in this thesis. Figure 2.16 shows the correlation of two MAV images. As can be seen, the correlation process for the most part correctly matches the feature locations.

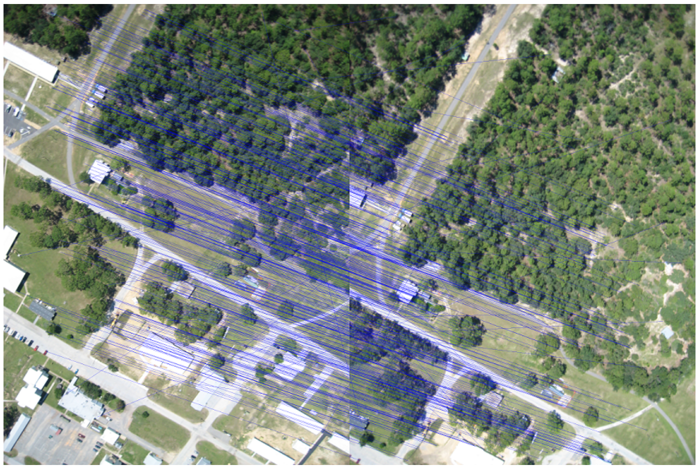


Figure 2.15: This figure shows the correlation of SURF feature detectors using the same two MAV images used in Figure 2.14. This figure has been rotated 90 degrees for display purposes.

Estimation of Homography

Once a set of $2D$ to $2D$ point correspondence, $\mathbf{x_i} = [\mathbf{x_i}, \mathbf{y_i}]^{\mathbf{T}} \leftrightarrow \mathbf{x_i'} \left[\mathbf{x_i'}, \mathbf{y_i'}\right]^{\mathbf{T}}$, is computed, it is possible to compute the perspective transformation, H, such that

$\mathbf{x_i^{'}} = \mathbf{Hx_i}$. A common method used for computing this perspective transformation is the Direct Linear Transformation(DLT) algorithm as described in [14]. This method is repeated here to provide a complete description of our system.

The vectors $\mathbf{x_i^{'}}$ and $H\mathbf{x_i}$ have the same direction but vary in magnitude by a nonzero scale factor because they are homogeneous vectors. Therefore, the equation $\mathbf{x_i^{'}} = \mathbf{Hx_i}$ may also be expressed in terms of the vector cross product, $\mathbf{x_i^{'}} \times \mathbf{Hx_i} = \mathbf{0}$. The explicit form of this cross product is given as

$$\mathbf{x_i^{'}} \times \mathbf{Hx_i} = \begin{pmatrix} y_i^{'}\mathbf{h_3^T x_i} - \mathbf{z_i^{'}h_2^T x_i} \\ z_i^{'}\mathbf{h_1^T x_i} - \mathbf{x_i^{'}h_3^T x_i} \\ x_i^{'}\mathbf{h_2^T x_i} - \mathbf{y_i^{'}h_1^T x_i} \end{pmatrix}, \tag{2.14}$$

where $\mathbf{h_1}$, $\mathbf{h_2}$, and $\mathbf{h_3}$ represent the rows of H. This equation can be rewritten in matrix form as

$$\mathbf{x_i^{'}} \times \mathbf{Hx_i} = \begin{bmatrix} \mathbf{0^T} & -z_i^{'}\mathbf{x_i^T} & y_i^{'}\mathbf{x_i^T} \\ z_i^{'}\mathbf{x_i^T} & \mathbf{0^T} & -x_i^{'}\mathbf{x_i^T} \end{bmatrix} \begin{pmatrix} \mathbf{h_1} \\ \mathbf{h_2} \\ \mathbf{h_3} \end{pmatrix} = \mathbf{0}, \tag{2.15}$$

with the third row of the matrix omitted because it is linearly dependent on the first two.

This leaves us with an equation of the form $A_i h = 0$, where A_i is 2×9 matrix. The A matrix can then be constructed by stacking each A_i matrix computed from feature correspondence. Given this matrix and a minimun of 4 feature corespondences, we can now solve for the elements of H via an SVD, with the unit singular vector corresponding to the smallest singular value being the solution.

The DLT algorithm works well when the only source of error in the estimation comes from the measurement of feature locations. However, some feature locations will inevitably be mismatched and introduce other sources of errors, known as outliers. These outliers can severely disturb the estimated homography and consequently should be removed. This is accomplished using a ***ran***dom ***sa***mple ***c***onsensus

(RANSAC) algorithm [15] to identify and eliminate outliers from the homography estimation.

The RANSAC algorithm partitions the feature correspondences into inliers (largest consensus set) and outliers (remaining feature correspondences). To compute the inliers, the algorithm first selects a random sample of 4 correspondences and computes the homography H using the DLT algorithm. Once H is computed, it is used to map feature locations in the source image to the destination image. A distance $d_{\perp}$ between the mapped and calculated correspondence locations is then computed and used to determine inliers for which $d_{\perp} < t = \sqrt{5.99}\sigma$ pixels. This process is repeated for N samples, where N is determined adaptively. The set with the largest number of inliers is then used to compute the final H. Figure 2.16 shows an estimated homography using RANSAC and DLT algorithms. The red and blue dots in the image are outlier feature locations while the purple dots are the inliers.

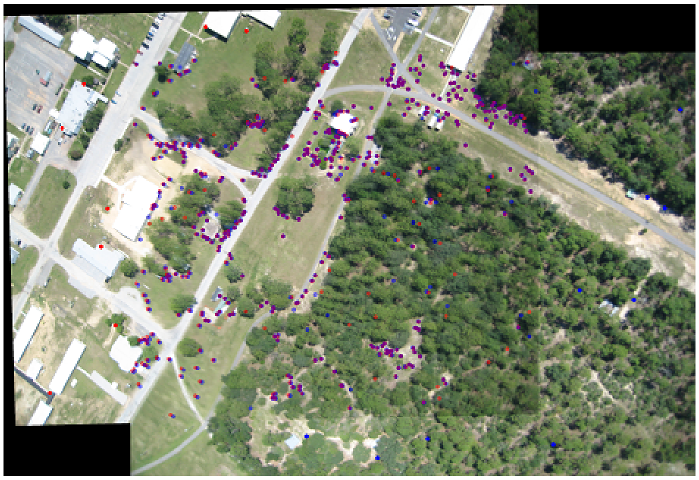
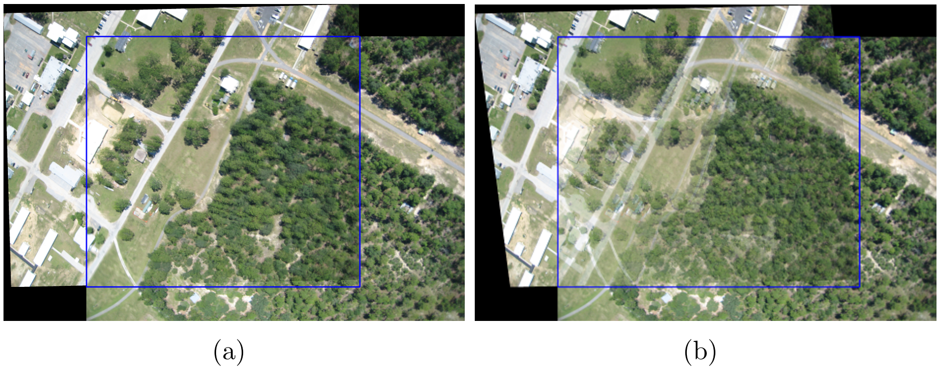


Figure 2.16: This figure shows the estimation of a perspective transformation using the MAV images shown in Figure 2.14 and Figure 2.16.

2.3.3 Focus Based Rejection of Homographies

Using the process described above, we can estimate a perspective transformation $\hat{H}_{s \to d}$ that maps a source image I_s onto a destination image I_d using the DLT and RANSAC algorithms. While the RANSAC algorithm enables us to remove outliers, it does not guarantee that all outliers are removed. Therefore, we also perform outlier rejection on the estimated perspective transformation using a focus metric. The key insight in this process is that errors in estimated perspective transformation translate into a blurring affect in the overlap regions of the composite images. This is illustrated in Figure 2.17. As mention in Section 2.2.3, a focus metric objectively measures the amount of blurr present in an image and can therefore be used to measure the accuracy of the perspective transformations given its resulting composite image.


(a)

(b)

Figure 2.17: This figure shows how inaccuracies in an estimated perspective transformation affect the resulting composite image.

The process starts by finding the overlap region in each image. To compute the overlap regions, we first create a blank image, $I_{s \rightarrow d}$, to hold the pixel values of the source image mapped in the destination coordinate frame. Once this image is created, $\hat{H}_{s \rightarrow d}$ is used to map pixel locations in I_s into $I_{s \rightarrow d}$. The newly populated $I_{s \rightarrow d}$ and original I_d images are then combined using an alpha blend and the result is stored into new image I_c. We then find the overlapping region of $I_{s \rightarrow d}$, I_d, and I_c using the process described in Section 2.2.2.

Once the overlap regions are defined, we can compute the focus values of each region. The focus metric used is the variance of the image (image energy) which is less punative in comparision to the Energy of Laplacian metric discussed in Section 2.2.3. The discrete version of this metric is defined as

$$V\left(I\right)=\frac{1}{n}\sum_{x}\sum_{y}\left(I^{2}\left(x,y\right)-\mu^{2}\right), \tag{2.16}$$

where

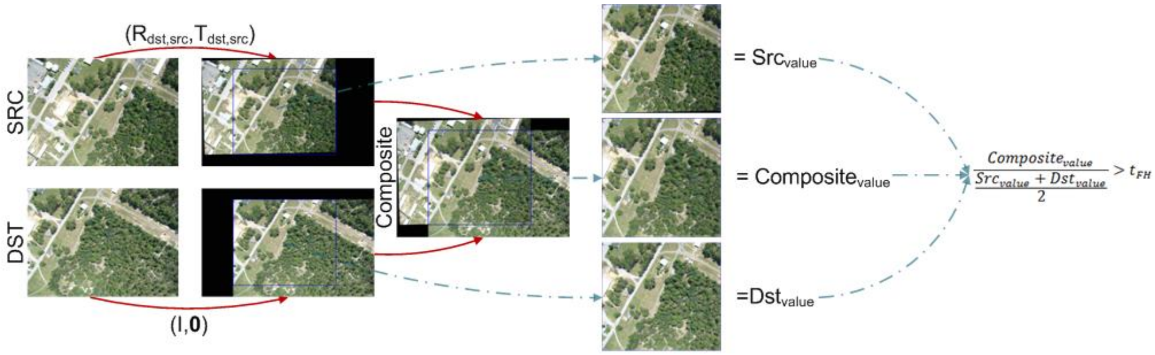
$$\mu \quad = \quad \frac{1}{N}\sum_{x}\sum_{y}I\left(x,y\right),$$

x and y are pixel coordinates, I is the image being evaluated, and N is the number of pixels being evaluated.

The computed focus values are then used to compute the ratio of the composite region to the average of the source and destination region values. If the ratio is above a user defined threshold, then the estimated projective transformation is accepted. This can be computed as

$$\frac{Composite_{value}}{\left(Source_{value}+Destination_{value}\right)*.5}\geq t_{f}. \tag{2.17}$$

This process is shown in Figure 2.18.



$(R_{dst,src}, T_{dst,src})$
SRC
DST
Composite
(I,0)
$= Src_{value}$
$= Composite_{value}$
$= Dst_{value}$
$\frac{Composite_{value}}{\frac{Src_{value}+Dst_{value}}{2}} > t_{FH}$

Figure 2.18: This figure shows the outlier rejection of estimated perspective transformations.

2.4 Chapter Summary

Once this process is complete, a sparse subset of the initial MAV imagery that most represent the desired area has been selected. In addition, a series of feature-based perspective transformations for this subset of MAV imagery has been created. The perspective transformations will be used as constraints in the initial iteration of the bundle adjustment method described in the next chapter.

Chapter 3

Iterative Refinement of Pose Estiamtes

In this chapter we describe the iterative refinement segment introduced in Section 1.3.3 in more detail. The primary purpose of this segment is to find the optimal MAV pose estimates that align the captured images to create a high-quality mosaic while minimizing the difference in the pose estimates to that of the measured poses. The main algorithm used to find the optimal pose estimates is a constrained optimization process using bundle adjustment. The constrained optimization procedure treats the estimated feature-based frame-to-frame perspective transformations, homographies, as constraints to ensure that the final mosaic is an image of high quality. However, which homographies are used as constraints is an input to that procedure and must be determined at run-time. Therefore, our algorithm consists of two main sub-groups: constrained optimization using bundle adjustment (BA) and image-to-image homography selection. Together these groups form a nested iterative process for refining the MAV pose estimates. This is illustrated in Figure 3.1.

This nested iterative process is constructed using three individual iterative processes. Going from the innermost to the outermost iteration loop, they are as follows: bundle adjustment (BA) method, constrained optimization using Lagrange multipliers, and topology enhancement.

The innermost iteration loop, bundle adjustment, attempts to find the point in the parameter space that most closely (in a weighted-least squares sense) maps to the measurements collected. This method iteratively estimates the parameter values that predict the measured values most correctly according to our cost function. The measurement space is defined by the MAV pose estimates $\vec{P}_i$, feature-based frame-to-frame perspective transformations $\hat{\mathbf{H}}_{i \leftarrow i+1}$, and prior knowledge of the world, such

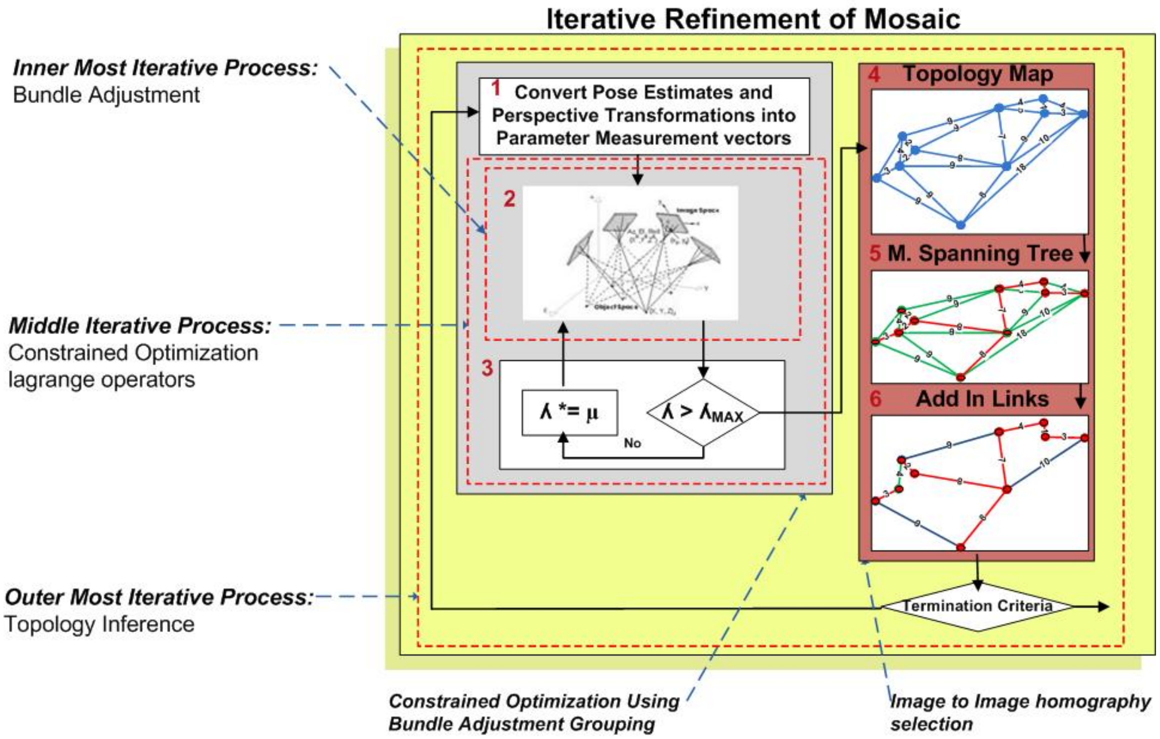

Iterative Refinement of Mosaic
Inner Most Iterative Process:
Bundle Adjustment
Middle Iterative Process:
Constrained Optimization
lagrange operators
Outer Most Iterative Process:
Topology Inference
1 Convert Pose Estimates and Perspective Transformations into Parameter Measurement vectors
2
Image Space
Object Space
3
ʎ *= μ
ʎ > ʎMAX
No
4 Topology Map
5 M. Spanning Tree
6 Add In Links
Termination Criteria
Constrained Optimization Using
Bundle Adjustment Grouping
Image to Image homography
selection

Figure 3.1: This figure show the iterative refinement segment of the mosaic creation process shown initially in Figure 1.4.

as the assumed normal vector $\vec{n}$ and rotational bias terms $\vec{q}$. The parameter space is defined by the parameters to be estimated which include the MAV pose estimates $\hat{P}_i$, normal vector $\hat{n}$, and rotational bias terms $\hat{q}$.

The pose estimates used to construct the parameter and measurement spaces contain the position T and attitude $\vec{q}$ information of the MAV body for a given image. Position information is encoded using a local North-East-Down (NED) Cartesian coordinate system as explained in Section 2.3.1. Attitude information is represented using a quaternion representation also explained in Section 2.3.1. Each pose can be represented as follows

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \\ \vec{q} \end{bmatrix}. \tag{3.1}$$

The BA method is completed when one of three termination criteria is met. First, the BA method stops when a local minimizer is reached. In other words, the process stops when the direction of steepest descent is approximately zero resulting in nearly identical parameter estimates in each iteration. Second, the BA method stops when the change in parameter estimates is small (i.e. below some threshold). This can occur when the steps in the steepest descent direction move us around the local minimizer. Finally, the BA method stops when a maximum number of iterations are completed. As in all iterative processes, this is a safeguard against an infinite loop.

The middle iteration loop, constrained optimization using Lagrange multipliers, evaluates the BA method at various Lagrange multipliers λ. These Lagrange multipliers provide us with a strategy for enforcing the constraints placed on homographies in our BA cost function and are explained in Section 3.2. The iteration loop starts by evaluating the BA method at a minimum Lagrange multiplier, $\lambda = \lambda_{min}$. Once completed, the BA method is revaluated using the last measurement and parameter values at a new λ increased by a scale factor σ_L. This is continued until a maximum Lagrange multiplier is reached, $\lambda = \lambda_{max}$.

Upon the completion of this middle iteration loop, we have an improved set of MAV pose estimates , $\hat{P}_i$, that were computed using the initial MAV pose estimates $\vec{P}_i$ and feature-based perspective transformations $\hat{\mathbf{H}}_{i \leftarrow i+1}$ between image pairs. At this point, these pose estimates are used to either create a geo-referenced mosaic or to determine a new set of image pairs using a reduced topology map [10, 13] generated from $\hat{P}_i$. The reduced topology map maximizes the overlap between image pairs thereby minimizing the possibility of errors in the estimation of feature-based perspective transformations. Each edge in this map represents a valid image pair. The image pairs are then used to compute a new set of feature-based perspective transformations that are used in the BA method to improve its accuracy.

The outermost iteration loop, topology enhancement, encompasses the creation of the reduced topology maps. Because image pairs chosen to constrain the optimization procedure can have a significant effect on the resulting pose estimates, the topology map created in the outermost iteration level alternates between different configurations. To compute this reduced topology map, we start by evaluating the middle level using the initial MAV pose estimates and feature-based perspective transformations output from the initial mosaic creation segment. Once completed, the middle iteration is revaluated using a new set of perspective transformations generated from the image pairs determined by the reduced topology map. This is repeated for a preset iteration count determined by the user.

The remainder of the chapter discusses the individual components in more detail and is organized as follows. In Section 3.1, we describe the creation of the topology map used in the outer most level. In Section 3.2, we describe our constrained optimization setup using our BA cost function and our implementation of the BA method.

3.1 Selection of Image Pairs for BA Process

As discussed in the introduction of this chapter, we use a reduced topology map (graph) to determine the image pairs and subsequently the feature-based perspective transformations to be used in the BA method. To compute the reduced topology graph, we first compute a full topology graph and then its minimum spanning tree. The reduced topology graph is a mixture of these two graphs.

To create the full topology graph, we first project the four corners of each image i in image set $I_{i \to n}$ onto the reference frame using MAV pose information and perspective transformations, explained in Section 2.3.1. Each projected center location is then used to represent a vertex location in the topology graph. The resulting projection of the four corners (polygons) are then superimposed onto each other to create new polygons that represent the overlap areas (intersections) of polygon pairs [28]. The overlap area is then converted into an overlap percentage and used to create the edges in the topology graph. The edges in the graph represent candidate frame-to-frame perspective transformations between image pairs and are directly determined via the overlap percentage. The edges are inversely weighted according to their overlap percentage (i.e. $w \propto \%^{-1}$). As the edge weights approach ∞ they are removed using a pruning process described below. The process of creating the topology graph is illustrated in Figure 3.2.

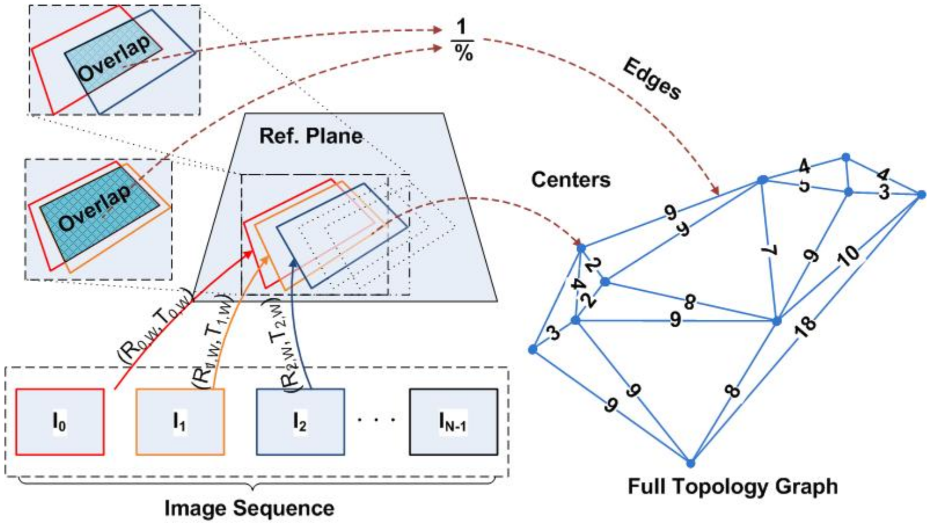
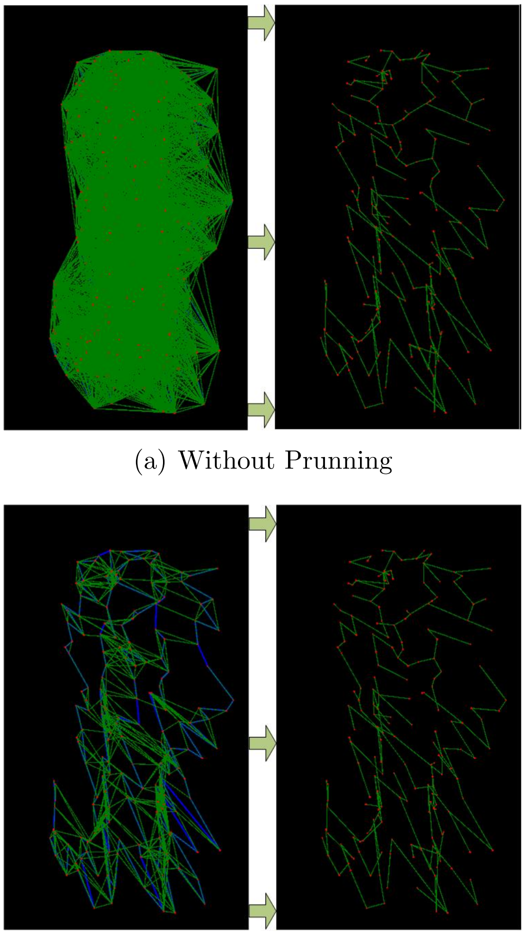


Figure 3.2: This figure illustrates the process of creating the pruned topology map using the MAV pose estimates and the fram-to-frame rigid body transformations.

Once the full topology graph is created, a minimum spanning tree (MST) that maximizes the overlap between candidate image pairs is created. This MST has a couple of advantages. First, it produces a fully connected topology graph with a minimum number of edges. Reducing the number of edges in the topology graph also reduces the BA measurement space size thereby also reducing computational cost associated with solving the normal equations (i.e. $\mathcal{A}\vec{x} = \vec{b}$). Second, by maximizing the overlap between image pairs we also reduce the possibility of error in the estimation of feature-based perspective transformations.

To create the MST of the topology graph, we first run a pre-pruning process to remove edges that are above a maximum threshold value t_b. As stated above, the edges of the full topology graph represent candidate image pairs. For each candidate image pair a feature-based perspective transformation is estimated, evaluated, and possibly rejected as described in Section 2.3.2. However, as the overlap between image pairs decrease the possibility of rejection increases. This results in wasted



(a) Without Prunning

(b) With Prunning

Figure 3.3: This figure shows the results of prunning the edges of the topology graph before the creation of the MST.

computations and an increased process time. Therefore, image pairs above t_b are removed from consideration before the creation of the MST.

Figure 3.3 shows the placement of approximately 150 images in the reference frame using the described process. The blue lines represent the temporal path of the MAV, green lines represent the candidate frame-to-frame perspective transformations (edges of graph), and the red dots represent the center of each projected image (vertices of graph). As can be seen, the pruning process significantly reduces the number of edges to be considered in the creation of the MST and results in the same MST.

After pruning the edges of the topology graph, we use Prim's algorithm [17] to find a MST for the given topology graph. This algorithm works as follows. First, select an arbitrary vertex, b, in the topology graph, T, and use it as the first vertex in the MST, M. Second, choose a vertex in T with a minimal connected edge weight to a vertex in M and add it to M. Third, repeat this process until all vertices in T are also in M. This process is illustrated in Figure 3.4.

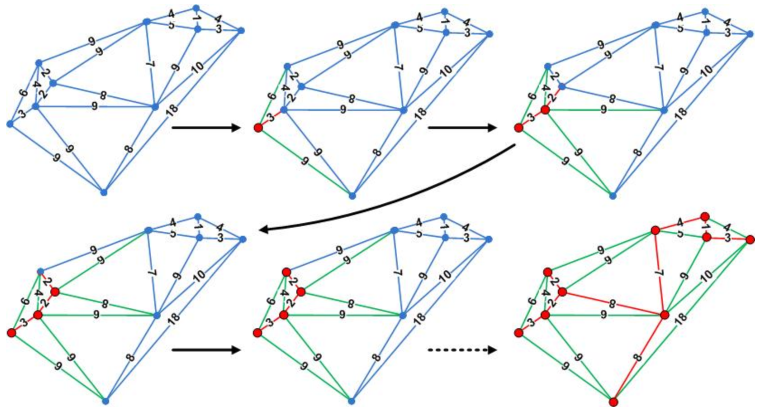


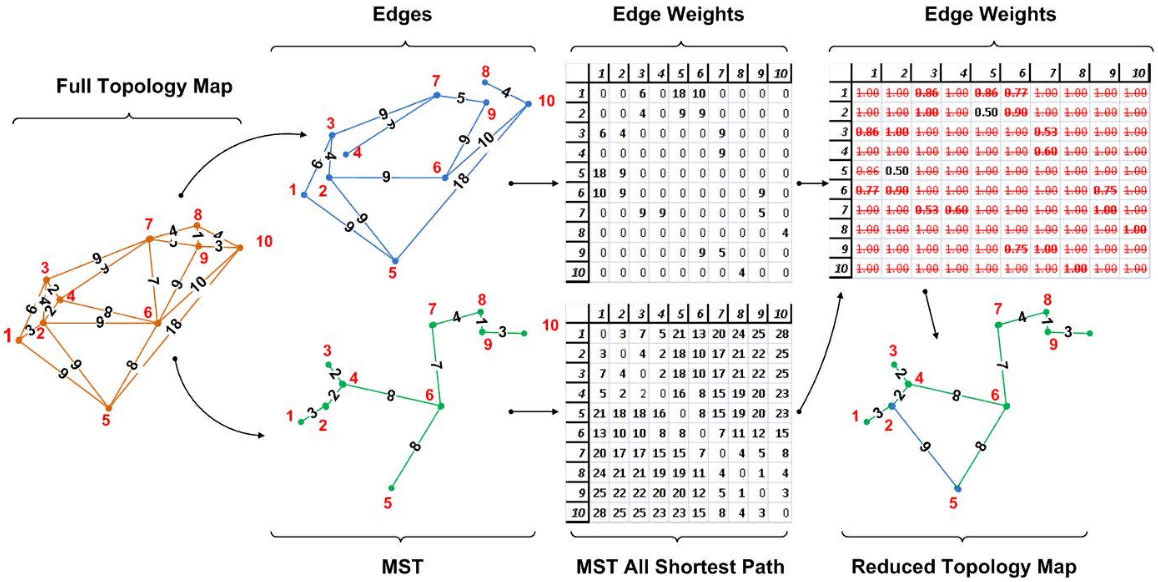
Figure 3.4: This figure shows the various stages of selecting image pairs using the initial MAV pose estimates.

Given that we now have a topology graph G_{full} and its MST G_{MST}, we can create the reduced topology graph mentioned in the introduction. To compute this graph, we start by initializing a new graph $G_{reduced}$ using the computed MST. Edges are then incrementally added to $G_{reduced}$ using the criteria described in [29, 13] which are as follow: (i) add edges that link images with significant overlap and (ii)create a "material shortcut" between vertices. Material shortcuts are defined as edges that have the greatest effect on the accuracy of the BA method, such as the first edge to close a loop or edge that ties together two swaths.

The first criterion has already been meet due to the pruning process used to create the topology map and subsequently the MST. The edges in these maps are directly computed from the overlap percentages of image pairs. Furthermore, only image pairs above a specified overlap percentage are included in the maps. The second criterion is determined via the ratio of the current path length that connects two vertices and the cost of the new edge. This is computed as

$$\gamma_{I,j} = \frac{w_e}{d_{sp}}, \tag{3.2}$$

where d_{sp} is the shortest path that connects two vertices and w_e is the weight associated with the new edge. The j and i represent the vertices being evaluated. This ratio goes from 0 to 1 with low values representing good candidate links.



Edge Weights

	1	2	3	4	5	6	7	8	9	10
1	0	0	6	0	18	10	0	0	0	0
2	0	0	4	0	9	9	0	0	0	0
3	6	4	0	0	0	0	9	0	0	0
4	0	0	0	0	0	0	9	0	0	0
5	18	9	0	0	0	0	0	0	0	0
6	10	9	0	0	0	0	0	0	9	0
7	0	0	9	9	0	0	0	0	5	0
8	0	0	0	0	0	0	0	0	0	4
9	0	0	0	0	0	9	5	0	0	0
10	0	0	0	0	0	0	0	4	0	0

Edge Weights

	1	2	3	4	5	6	7	8	9	10
1	~~1.00~~	~~1.00~~	~~0.86~~	~~1.00~~	~~0.86~~	~~0.77~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~
2	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	0.50	~~0.90~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~
3	~~0.86~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~0.53~~	~~1.00~~	~~1.00~~	~~1.00~~
4	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~0.60~~	~~1.00~~	~~1.00~~	~~1.00~~
5	~~0.86~~	0.50	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~
6	~~0.77~~	~~0.90~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~0.75~~	~~1.00~~
7	~~1.00~~	~~1.00~~	~~0.53~~	~~0.60~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~
8	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~
9	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~0.75~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~
10	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~	~~1.00~~

MST All Shortest Path

	1	2	3	4	5	6	7	8	9	10
1	0	3	7	5	21	13	20	24	25	28
2	3	0	4	2	18	10	17	21	22	25
3	7	4	0	2	18	10	17	21	22	25
4	5	2	2	0	16	8	15	19	20	23
5	21	18	18	16	0	8	15	19	20	23
6	13	10	10	8	8	0	7	11	12	15
7	20	17	17	15	15	7	0	4	5	8
8	24	21	21	19	19	11	4	0	1	4
9	25	22	22	20	20	12	5	1	0	3
10	28	25	25	23	23	15	8	4	3	0

Figure 3.5: This figures illustrates an iteration of the gready algorithm used to add links back into the MST.

Using the ratio described above and greedy algorithm similar to the one presented in [13], we can now choose which edges have to be added to the reduced topology graph. The greedy algorithm starts by removing all edges in G_{full} that also exist in G_{MST}. Once all duplicate edges are removed, we compute allshortest paths between vertex pairs using Dijkstra's algorithm and store the results in a matrix $\mathcal{M}_{asp}$. Each entry of the matrix represents the shortest path from a source vertex (row index) to a destination vertex (column index). Likewise, we create a matrix, $\mathcal{M}_w$, that holds the weights associated with the candidate edges in G_{full}. The two matrixes are then divided to give us the needed ratio. Using the computed ratio, we then remove all edges in G_{full} that have a ratio above a user defined threshold t_{ratio}. Out of the remaining edges in G_{full}, we select the edge with lowest ratio and add it to $G_{reduced}$. This process is repeated until all edges are removed from G_{full}. An iteration of the greedy algorithm is illustrated in Figure 3.5.

3.2 Constrained Optimization Using Bundle Adjustment

3.2.1 Constrained Optimization Setup

As discussed in Section 1.2, the fundamental approach used to create a geo-referenced mosaic is to perform a constrained optimization problem using bundle adjustment. The optimization problem we are trying to solve is:

$$\begin{gathered}\min_{\vec{P}_i,\vec{n},\vec{q}_b} \sum_{i=1}^{N} (\vec{P}_i - \hat{P}_i)^2 \quad s.t. \\ \vec{h}(\vec{P}_i, \vec{P}_{i+1}, \vec{n}, \vec{q}_b) = \hat{h}_i, \\ ||\vec{n}||_2 = 1, \\ ||\vec{q}_{pi}||_2 = 1, \text{and } ||\vec{q}_b||_2 = 1,\end{gathered} \tag{3.3}$$

where $\vec{n}$ is the normal vector (in world coordinates) of the plane being imaged, $\vec{q}_b$ is a "bias" quaternion representing constant errors in the pose estimates from the IMU/GPS unit on-board the MAV, $\vec{h}(\vec{P}_i, \vec{P}_{i+1})$ is a function, described in Section 2.3.1, for computing the homography given two poses and the normal vector, and $\hat{h}_i$ is derived from the matrix $\hat{\mathbf{H}}_{i \leftarrow i+1}$, which is the homographies generated from feature correspondences. Because the matrix $\hat{\mathbf{H}}$ is defined up to a scale factor, the bottom right element of this matrix is set to 1. $\hat{h}_i$ is then set equal to

$$\hat{h}_i = \begin{bmatrix} \hat{\mathbf{H}}_{1,1} \\ \hat{\mathbf{H}}_{1,2} \\ \hat{\mathbf{H}}_{1,3} \\ \hat{\mathbf{H}}_{2,1} \\ \hat{\mathbf{H}}_{2,2} \\ \hat{\mathbf{H}}_{2,3} \\ \hat{\mathbf{H}}_{3,1} \\ \hat{\mathbf{H}}_{3,2} \end{bmatrix}, \tag{3.4}$$

where the subindices #,# represent the row and column, respectively, of $\hat{\mathbf{H}}$.

We handle the constraints imposed on this optimization in two ways. The constraints on $\vec{n}$, $\vec{q}_b$, and $\vec{q}_{pi}$ are imposed by re-normalizing these vectors to one after each iteration of the bundle adjustment [24]. The constraint on $\vec{h}(\vec{P}_i, \vec{P}_{i+1}, \vec{n}, \vec{q}_b)$ is handled using a Lagrange multiplier λ, leading to

$$\min_{\vec{P}_i, \vec{n}, \vec{q}_b} \sum_{i=1}^{N} (\vec{P}_i - \hat{P}_i)^2 + \lambda \left(\vec{h}(\vec{P}_i, \vec{P}_{i+1}) - \hat{h}_i \right)^2, ||\vec{q}_b||_2 = 1, ||\vec{q}_{pi}||_2 = 1, \text{ and } ||\vec{n}||_2 = 1.$$

For a given value of λ, this minimization function can be solved using bundle adjustment. Generally we start with a λ of .001, increasing by factors of 10 until $\lambda = 100,000$. For non-final values of λ, we do not enforce as strict of a convergence criteria for bundle adjustment in order to speed up the optimization process.

For initial conditions on our constrained optimization problem, we initialize the estimated poses to the measured poses returned by the IMU/GPS system on-board the MAV. The normal vector $\vec{n}$ is initialized to a vertical vector ($\begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$), and the bias quaternion to the identity quaternion ($\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$).

3.2.2 Bundle Adjustment Implementation

Bundle adjustment is a method for finding the closest point on a manifold defined by a parameter vector $\vec{p}$ to a measurement vector $\vec{x}$. In our case, the measurement vector can be written as:

$$\hat{x} = [\hat{P}_0, \cdots, \hat{P}_N, \hat{h}_0, \cdots, \hat{h}_{n-1}, \hat{n}_v, \hat{q}_0]^T, \tag{3.5}$$

where $\hat{n}_v = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$ and $\hat{q}_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$ are added to the measurement vector to represent prior knowledge of typical values for $\vec{n}$ and $\vec{q_b}$ (the ground should have a normal that is close to vertical, and the biases should be small.)

The parameter space over which bundle adjustment will iterate is defined by the parameters which are being optimized in the constrained optimization problem (P_i, $\vec{n}$, and $\vec{q}_{bias}$). The total parameter vector is defined as:

$$\vec{p} = [\vec{P}_0, \cdots, \vec{P}_N, \vec{n}, \vec{q}_{bias}]^T. \tag{3.6}$$

To perform this optimization we use a weighted Gauss-Newton method. This method iteratively linearizes the function to be minimized in the neighborhood of the current estimate, by solving linear systems known as *normal equations.* The normal equations are defined as

$$\begin{aligned} \mathcal{A} &= \mathbf{J}^T \Sigma^{-1} \mathbf{J}, & (3.7) \\ \vec{g} &= \mathbf{J}^T \Sigma^{-1} \vec{\epsilon_p}, & (3.8) \\ \vec{x}_E^+ &= \mathcal{A}^{-1} \vec{g}, & (3.9) \end{aligned}$$

where $\mathbf{J}$ is an $m \times n$ matrix containing the partial derivatives of the cost function (i.e. a Jacobian matrix), Σ is the covariance matrix which represents the distance metrics which are defined below, $\mathcal{A}$ is an $m \times m$ matrix that contains the approximated second derivatives of the cost function, $\vec{\epsilon}_P$ is the residual error in pose estimates and homography matrices, $\vec{g}$ is the gradient, and change $\vec{x}_E^+$ in the parameters for the next iteration. The Jacobian Matrix is formulated as

$$\mathbf{J} = \begin{bmatrix} \mathcal{P} \\ \mathcal{H} \end{bmatrix}, \tag{3.10}$$

where

$$\mathcal{P} = diag\left(\frac{\delta\vec{P}_1}{\delta\vec{P}_1}, \frac{\delta\vec{P}_2}{\delta\vec{P}_2}, \cdots, \frac{\delta\vec{P}_n}{\delta\vec{P}_n}, \frac{\delta\vec{n}}{\delta\vec{n}}, \frac{\delta\vec{q}_{bias}}{\delta\vec{q}_{bias}}\right) = I_{n+7}, \tag{3.11}$$

and

$$\mathcal{H} = diag\left(\begin{bmatrix} \frac{\delta\mathbf{H}_1}{\delta P_1} & \frac{\delta\mathbf{H}_1}{\delta P_2} \end{bmatrix}, \cdots, \begin{bmatrix} \frac{\delta\mathbf{H}_{n-1}}{\delta P_{n-1}} & \frac{\delta\mathbf{H}_{n-1}}{\delta P_n} \end{bmatrix}\right). \tag{3.12}$$

Due to the block diagonal nature of both $\mathcal{P}$ and $\mathcal{H}$, efficient bundle adjustment can be used to solve the normal equations.

The covariance matrix Σ is assumed to be diagonal with weighting for each element as defined in Table 3.1. The weighting on the x, y, z, and $\vec{q}$ parameters represent the assumed error present in IMU/GPS estimates. The covariance on the attitude parameter q is relatively large due to a variety of issues associated with the acquisition of this measurement. First, the small size of an MAV airframe limits the weight, size, and power available for payloads, necessitating the use of low-quality sensors for the IMU, leading to noisy pose estimates. Second, the delays in the system are not accurately measured. This can lead to inaccuracies in the alignment of MAV pose estimates and imagery. Third, attitude information may not be available. The weighting of $\vec{n}$ is chosen such that it only allows minimal deviations from the initial estimate of $\hat{n}_v$. The covariance for $\vec{q}_b$ is also chosen to be quite large to allow for a large range of bias values (more motivation for these values will be found in the results section).

While the covariance values described above all have fairly straight-forward physical meanings, the covariance on the homography values is a bit more complicated. The goal of the constraint in the constrained optimization is to ensure that the poses chosen for the MAV camera cause the images to be aligned as well as a free mosaic created from only feature based methods. Therefore, we derive the covariances to represent movement in pixel locations due to changes in the homography

Table 3.1: Covariance Matrix Weights

Parameters	Weighting
x, y, z	1
$\vec{q}$	.1
$\vec{n}$	.0001
$\vec{q}_b$	1
$\mathbf{H}_{1,1}\mathbf{H}_{1,2}\mathbf{H}_{2,1}\mathbf{H}_{2,2}$	$\frac{1}{\lambda\ max\left(I_{height},I_{width}\right)} * \frac{1}{o_p}$
$\mathbf{H}_{1,3}\mathbf{H}_{2,3}$	$\frac{1}{\lambda} * \frac{1}{o_p}$
$\mathbf{H}_{3,1}\mathbf{H}_{3,2}$	$\frac{1}{\lambda\ max\left(I_{height},I_{width}\right)^2} * \frac{1}{o_p}$

parameters. Because different elements of the homography matrix have different effects on the pixels, we have covariances that vary as shown in Table 3.1. Homography covariances are also scaled by one over the overlap percentages, $\frac{1}{o_p}$, to give presidence to homgraphies with higher degrees of overlap. To utilize bundle adjustment within a constrained optimization framework, the homography covariances are also scaled by $\frac{1}{\lambda}$.

3.3 Chapter Summary

Upon the completion of the pose optimization process as described above, we have an optimized set of MAV pose estimates, $\hat{P}_i$. At this point, these pose estimates are used to create a single, large, integrated geo-referenced mosaic. This is done via the use of a virtual camera centered one meter above the desired area. Using the virtual camera and optimized poses, we project the MAV imagery onto a global coordinate system. We then use a multi-resolution blending algorithm to combine images and place them on the reference frame. The results of this process are shown in the next chapter.

Chapter 4

Results

In this chapter, we evaluate the geo-referenced mosaics created using the constrained optimization framework with bundle adjustment and topography refinement. Two main attributes are essential in a geo-referenced mosaic: (i) the mosaic needs to be "visually appealing" and (ii) the mosaic needs to be accurately geo-located in the world. The results presented in this chapter will show that the proposed mosaicing framework generates mosaics with these qualities. Furthermore, we will show that the geo-referenced mosaics created using this framework are comparable to prior work.

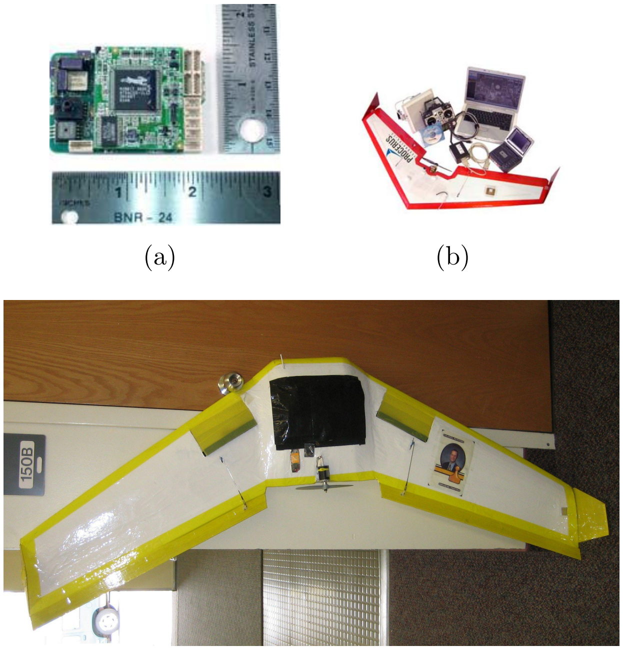
The chapter is organized as follows. In Section 4.1, we describe the aerial platforms used to collect MAV flight video and the generation of synthetic video. In Section 4.2, we evaluate the visual quality of the mosaic created using the proposed framework. In Section 4.3, we evaluate the geo-referencing accuracy of the generated mosaics.

4.1 Experimental Setup

In order to evaluate our system, we used imagery collected from two different MAV platforms and synthetically generated video. This section describes these items in detail and is organized as follows. In Subsection 4.1.1, we describe the BYU MAGICC LAB Test Platform used in the collection of MAV video. In Subsection 4.1.2, we describe the alternate MAV Test Platform used in the collection of MAV video. In Subsection 4.1.3, we describe the synthetically generated video sequence used in the evaluation of the proposed process.

4.1.1 BYU MAGICC LAB Test Platform

The MAV used in the BYU MAGICC Lab is a hand launchable Delta-wing test platform constructed from EPP foam. Figure 4.1, shows the key items in this system. This test platform uses a Procerus Kestrel^{TM} autopilot that is equipped with three-axis accelerometers and gyros, two pressure sensors, and a μ-blox AG GPS receiver. These sensors are used to obtain position and altitude measurements (i.e. pose estimates). The pose estimates computed by the Kestrel autopilot are transmitted to the ground at 25Hz over a 115.2 kBaud radio modem. This unit is shown in Subfigure (a).



(a) (b)

(c)

Figure 4.1: This figure shows the BYU MAGICC lab test platform used in the collection of MAV imagery. (a) Procerus Kestrel^{TM} Autopilot (originally developed at BYU). (b) Procerus Ground Station setup and MAV. (c) Actual airframe used in the collection of MAV image sequences.

The airframe used to collect the MAV imagery is shown in Subfigure (c). This airframe has an expanded payload bay, wingspan of $\sim 1.5m$, an empty weight of 3.2lbs, and a payload capacity of 1.8lbs. The airframe uses a brushless electric motor for propulsion, an electronic speed controller, and is fueled by lithium polymer batteries. It can be equipped with either a 640 x 480 SONY camera or a Canon SD1100 Elph 8 Mpixel commercial camera. The SD1100 can be set to take one high-resolution image per second and stores still imagery onboard the MAV. The video collected from these two cameras is sent to the ground station using a 2.4GHz NTSC video transmitter (30 frames a second).

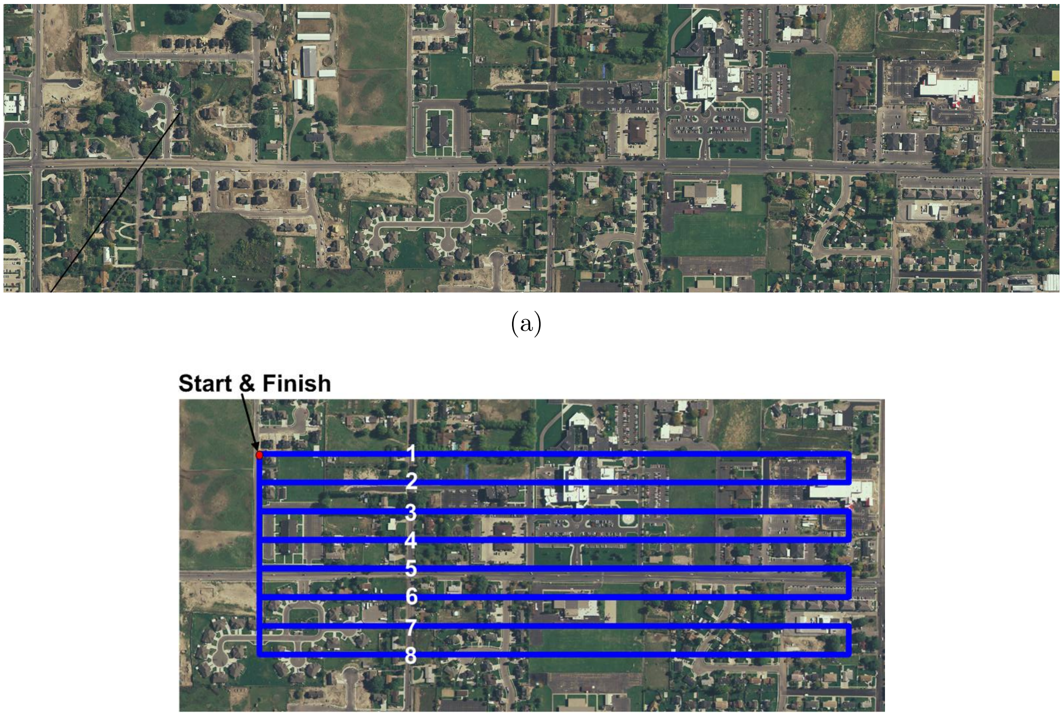
This platform is advantageous because it uses the Procerus ground station, which enables frame-level synchronization of pose information (including attitude) from the autopilot with video collected by the MAV. The Procerus ground station is shown in Subfigure (b). The items used in the ground station include a laptop to run the Procerus ground station software, a communication box for receiving and transmitting MAV packets, Garmin GPS unit for video synchronization, and a RC transmitter for operational control (i.e. user fail-safe).

4.1.2 Alternate Test Platform

In addition to the MAGICC lab test platform, we also collected imagery using a second test platform. This test platform is an MAV used by the US Armed Forces and has a wingspan of 1.3 m and weighs about 4.2 lbs. The platform was equipped with a Canon SD1000 Elph 7 Mpixel commercial camera, which was programmed to take one high-resolution image per second. Imagery and pose estimates collected from the MAV are synchronized using the EXIF time stamps of the imagery and GPS time stamps of pose estimates. Note that tight synchronization between this airframe's autopilot data and imagery was *not* achieved, leading to erroneous attitude estimates (i.e. no roll and pitch information). GPS, however, with its slower update rate, was effectively synchronized with the image data.

4.1.3 Generation of Synthetic Video

Due to limitations placed on acceptable flight areas and the availability of reference imagery, we also verify the proposed system using synthetically generated video. The synthetic video is used as a ground truth to verify the visual appearance and geo-locations of the mosaic. Furthermore, by using synthetic video we are able to test the proposed system using large data sets.



(a)

(b)

Figure 4.2: This figure shows the large scale geo-referenced image used in the creation of synthetic video and the lawnmower test pattern that was used. (a) shows the high-resolution aerial reference image downloaded from agrc.utah.gov (i.e. 40.382777 Latitude -111.767322 longitude). (b) shows the lawnmower pattern used for generation of synthetic video.

The synthetic video is created using a large area geo-referenced image and rigid body transformations as described in Section 2.3.1. During the creation of the synthetic video, we create a synthetic MAV that flies at at fixed altitude over the reference imagery, with random roll and pitch angles. The yaw angle is fixed in a direction determined by the lawnmower patter displayed in Figure 4.2(b). Once the "true" poses of the MAV have been computed and the images generated for those poses, the pose estimates are corrupted with noise. In addition to the added noise, the generated images are also blurred using a Gaussian function. All angles are generated using a uniform distribution. The reference image used to create the synthetic video along with the lawn mower flight pattern, as shown in Figure 4.2(a), were downloaded from agrc.utah.gov and are high-resolution aerial images taken of the state of Utah (i.e. 40.382777N -111.767322E).

4.2 Evaluation of Visual Quality

As stated in the introduction of this chapter, it is essential for a mosaic to be "visually appealing". In other words, the mosaic needs to accurately represent the area being observed as perceived by an end user. In this section, we will show that the generated geo-referenced mosaics have this attribute. This is shown using synthetically generated data and actual MAV flight data collected using the aerial platforms described in Section 4.1.

This section is subdivided into three main subsections. In Section 4.2.1, we describe the quality metrics used to quantify percevied differences in mosaics. In Subsection 4.2.2, we compare the mosaic generated from optimized pose estimates to mosaics created using the actual poses on synthetic video. In Subsection 4.2.3, we evaluate the image quality of the mosaics generated using actual MAV video.

4.2.1 Quality Evaluation Measures

In this thesis, we use the peak signal-to-noise (PSNR) measure and the structural similarity index measure (SSIM)[30] to evaluate the quality of the mosaics generated from synthetic imagery. The PSNR ratio is used because it is simple to calculate and it is a commonly used metric when comparing imagery (i.e. used to evaluate compression quality). The SSIM, while harder to compute, is normalized for luminance and contrast making it a more robust. Both measures are metrics (i.e. assume original imagery exists) and only evaluate the image quality based on its luminance (i.e. gray scale).

The PSNR is given in decibel units (dB), which measure the ratio of the peak signal and the difference between two images. The PSNR ratio is computed as

$$PSNR = 20 * \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right), \tag{4.1}$$

where MAX_I is the maximum possible pixel value of the image and MSE is the mean squared error of two $m \times n$ monochrome images, I_{orig} and $I_{modified}$. The MSE is calculated as

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{i=0}^{n-1}\left\| I_{orig}\left(i,j\right) - I_{modified}\left(i,j\right)\right\|^2. \tag{4.2}$$

The result of the SSIM measurement is a decimal value between -1 and 1. A value of 1 means both images are identical. The SSIM metric is calculated as

$$SSIM\left(x,y\right) = \frac{\left(2\mu_x\mu_y\right)\left(2cov_{xy}+c2\right)}{\left(\mu_x^2+\mu_y^2+c_1\right)\left(\sigma_x^2+\sigma_y^2+c_2\right)}, \tag{4.3}$$

where μ_x and μ_y are the average x and y values, σ_x^2 and σ_y^2 are the variance of x, and y, cov_{xy} (scalar) is the the covariance of x and y, c_1 and c_2 are stabilizer constants.

4.2.2 Results Using Synthetic Imagery

The synthetic video enables us to compare a mosaic generated from optimized pose estimates to mosaics created using the actual poses estimates. The synthetic video used containes 5,000 images (before frame decimation) and was created as described in Section 4.1.3.

In Figure 4.3, we show three mosaics created during various stages of the mosaic creation process and a mosaic created using the actual poses. In Subfigures (a) and (b), we show the mosaics created using the actual and initial pose estimates. It is immediately apparent that the mosaic created from the initial pose estimates is of far lesser quality than that of the true pose information. In Subfigures (c) and (d), we show mosaics created using the optimized pose estimates of the first and last iteration of the constrained optimization process using bundle adjustment and topology refinement. By inspection, we can see that both mosaics significantly improve the initial pose estimates and resemble the mosaic created from the true pose information. However, it is not immediately apparent as to how accurately these mosaics represent the optimal mosaic, Subfigure (a).

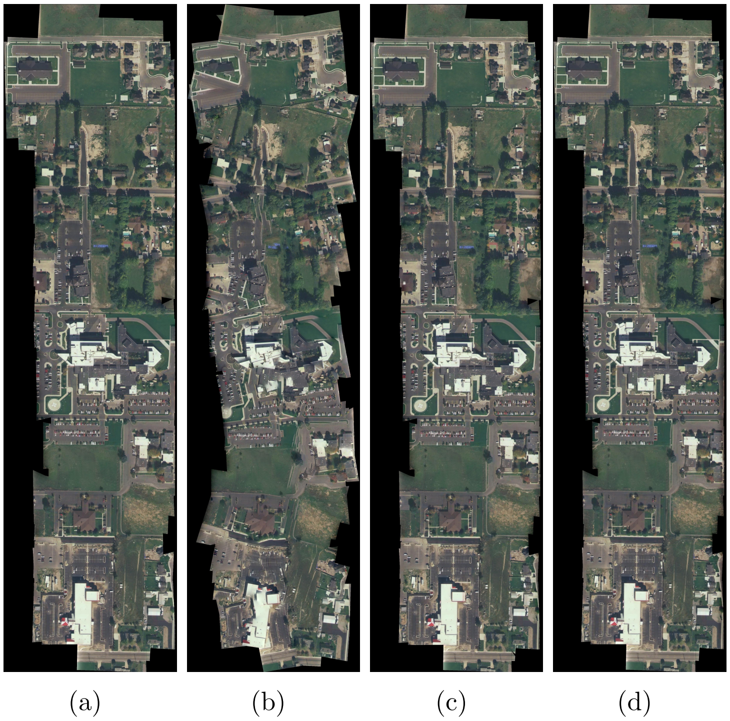
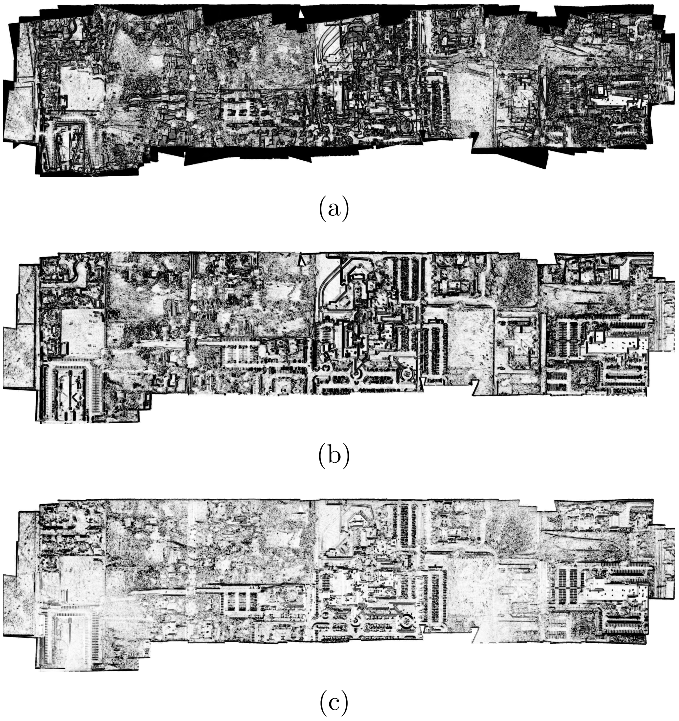


Figure 4.3: These figures show the mosaics created using true, noisy, and refined pose estimates. Each image is rotated by 90 degrees. (a) Mosaic created using true pose estimates. (b) Mosaic created using noisy pose estimates. (c) Mosaic created using after one complete iteration of the full nested BA process. (d) Mosaic created using after four complete iteration of the full nested BA process.

In order to evaluate the accuracy of each mosaic, we use the PNSR and SSIM metrics described in the Subsection 4.2.1. Table 4.1 shows the quality measures of each mosaic. This table demonstrates the significant improvements achieved by using a constrained optimization framework with bundle adjustment and revising the topography graph.

Table 4.1: PSNR and SSIM Quality Measures

Metric	Initial	First Iteration	Fourth Iteration
PSNR(dB)	+29.29 dB	+31.74 dB	+33.45 dB
SSIM([-1,1])	0.4658	0.6002	0.7499



(a)

(b)

(c)

Figure 4.4: This figures shows the magnitude of the differences between the reference and computed mosaics, where white indicates no difference and black represents maximal difference. Each image is rotated by 90 degrees. (a) SSIM map comparing the true and initial pose estimates. (b) SSIM map comparing the true and initial optimized pose estimates. (c) SSIM map comparing the true and topology optimized pose estimates.

In Figure 4.4, we display the magnitude of the differences between the reference and computed mosaic, where white indicates no difference and black represents maximal difference. In Subfigure (b), we demonstrate the improvements achieved using bundle adjustment in a constrained optimization framework without any modification to the topography graph. By modifying the topography graph and re-running the constrained optimization, even more improvements are achieved as shown in Subfigure (c). As can be seen in Subfigures (b) and (c), there is virtually no black border around the images. This shows that we can quite accurately align the mosaics with the reference imagery. Note that the dark regions still present in Subfigure (c) represent differences of less than 1m, demonstrating that our algorithm is capable of producing mosaics that accurately represent the area being observed.

4.2.3 MAV Imagery

In addition to synthetic imagery, we have verified our approach on using imagery from real MAVs. In specific, we will show that the created mosaics are contiguous, crisp, clear, and virtually free of any distortions introduced by the mosaic creation process. The imagery was collected using aerial platforms described in Section 4.1 and constitutes two flights worth of data.

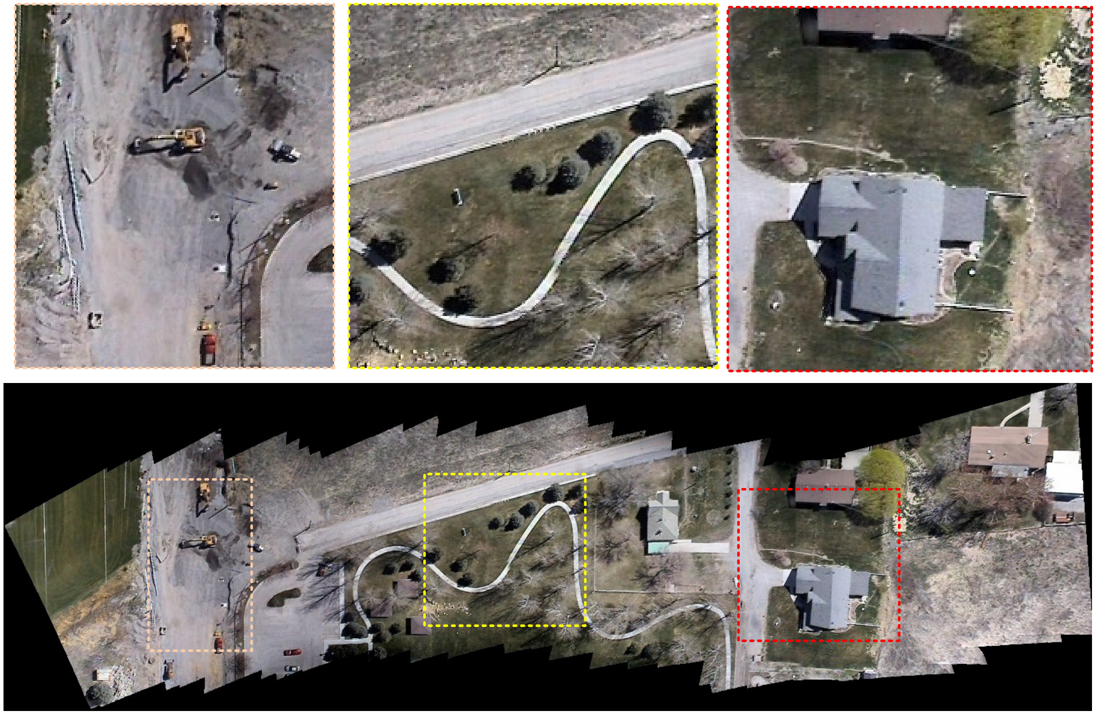


Figure 4.5: This figure shows the quality of an actual MAV mosaic created using 600 MAV images of Vineyard in Utah.

The first flight was conducted using the MAGICC lab test platform and collected 900 viable images (before frame decimation) spanning an area over 500 meters long. This imagery was collected using a 640 x 480 SONY camera. The mosaic created from this sequence can be seen in Figure 4.5. Notice how vehicles, curbs, roads, buildings, and even rocks in the mosaic are crisp and clear. In addition, notice how lines are well defined and straight. By visual inspection, we can see that the process creates a clear and consistent mosaic.

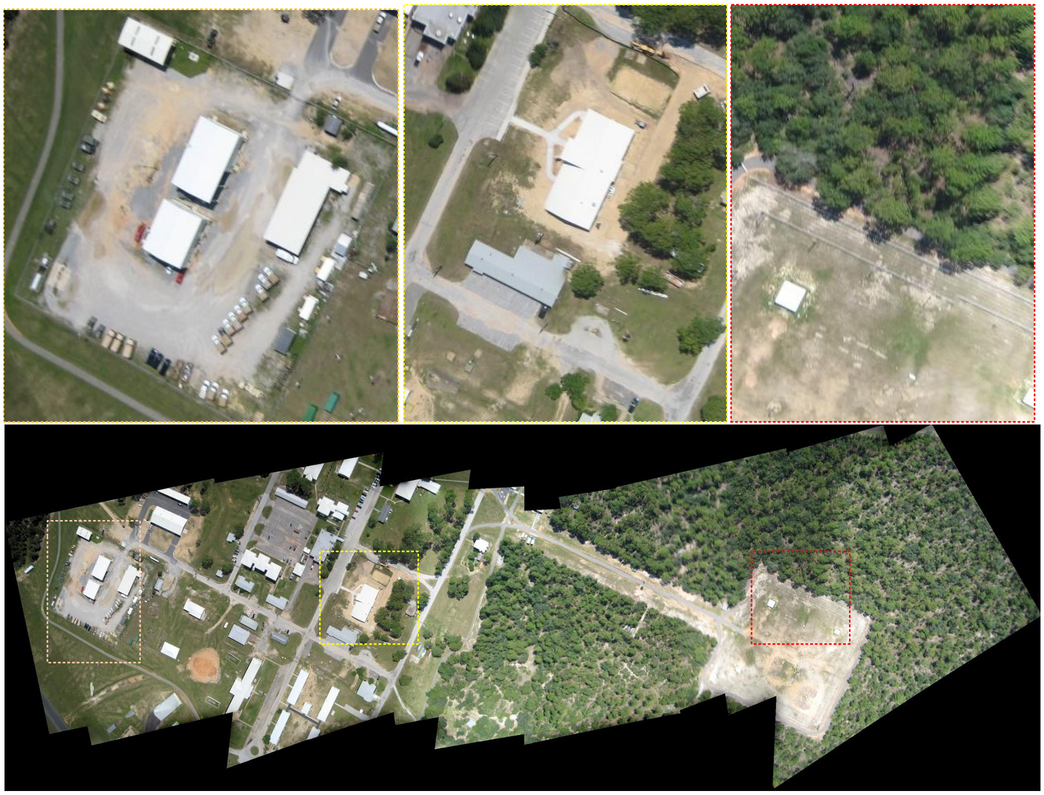


Figure 4.6: This figure shows the quality of an actual MAV mosaic created using 17 high-resolution images of Florida location.

We also tested the method using 13 high-resolution images collected by the alternate test platform. The imagery was captured using a Canon SD1000 Elph 7 Mpixel commercial camera. The resulting mosaic spans an area over 1000 meters long and can be seen in Figure 4.6. Note that tight synchronization between this airframe's autopilot data and imagery was *not* achieved, leading to erroneous attitude estimates (i.e. no roll and pitch information). Given the absence of attitude information, we are still able to create a consistent mosaic thereby showing that our process exhibits the first attribute of a geo-referenced mosaic.

4.3 Evaluation of Geo-location Accuracy

Now that we have shown that the proposed framework has the ability to create visually appealing mosaics, we show that it also has the second described attribute of a geo-referenced mosaic, accurately geo-located mosaics. Furthermore, we show that the proposed framework can produce comparable geo-located mosaics when compared to prior methods, such as [1] which is similar in design. In order to evaluate the geo-location accuracy of a mosaic, it is overlaid on imagery from Google Earth. This process assumes that the imagery in Google Earth is accurately geo-located.

In Figure 4.7, we show a geo-referenced mosaic created using optimized pose estimates and the data from the first flight. The geo-location errors in this mosaic were less than 7m across the entire surface. This was determined by measuring the displacement between the reported building, road, and pathway locations. Given a geo-location accuracy of 7m, we have demonstrated the second attribute needed to construct a geo-referenced mosaic.

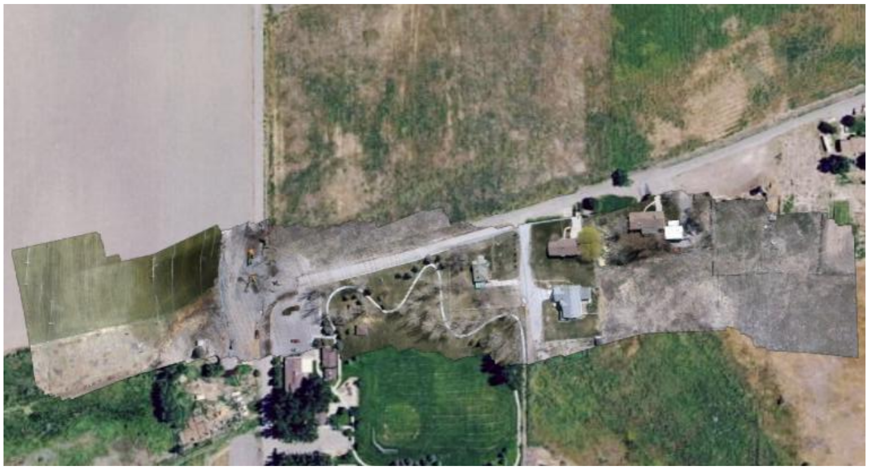


Figure 4.7: This figure shows the geo-referenced mosaic created using the work presented in this thesis.

In order to test the robustness of the process, a geo-referenced mosaic was created using optimized pose estimates and the data from the second flight. The resulting mosaic can be seen in Figure 4.6. By visual inspection of location at the ends and center of the mosaic, we found that geo-location errors for this mosaic ranged from 16 to 30 meters. This is a significant improvement compared with the raw telemetry information, which led to errors in the range of 9 to 320 meters. Having demonstrated the two attributes of a geo-referenced mosaic and shown a robustness to noisy pose estimates, we next compare our proposed framework to prior work.

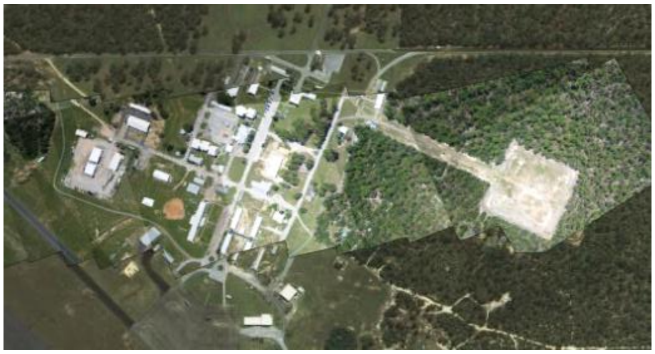
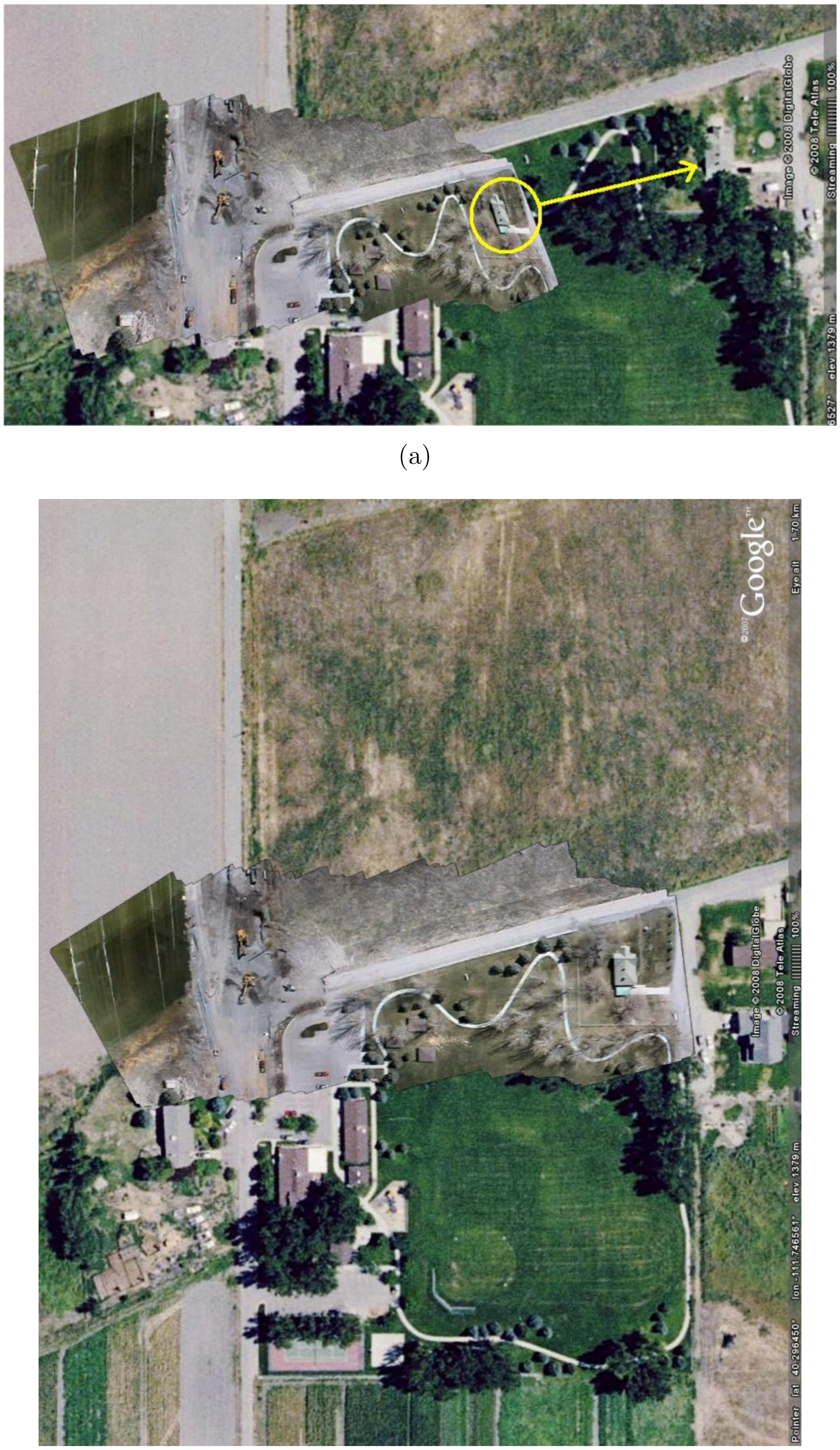


Figure 4.8: This figure shows the geo-location accuracy of a actual MAV mosaic created using 17 high-resolution images of Florida location.

In Figure 4.9, we show two mosaics created using different mosaic frameworks and the MAV data from the first flight as described above. In Subfigure (a), we show a mosaic that was created using a single pose estimate to orthorectify and geo-reference the first image of the mosaic. All subsequent images were then projected onto the mosaic using this image. Note how slight errors in the initial pose estimate deteriorate the geo-location accuracy of the mosaic. Measuring the displacement in the location of the house at the bottom of the image, [1] reported an error in geo-location of approximately 75m.

In Subfigure 4.9(b), we show a mosaic that was created using the UKF framework presented in [1]. The reported geo-location accuracy of this method was approximately 5m. This framework is similar to the work presented in this thesis because it also uses perspective transformations computed between images to improve the accuracy of MAV pose estimates coming from a GPS/IMU system. As can be seen, our proposed system is able to produce comparable results over longer sequences of imagery. The method in [1] is only able to construct mosaics consisting of less than 400 images while ours is only limited by the size of the video sequence. Our method has been shown to produce high quality geo-referenced mosaics and visually appealing mosaics over large data sets.



(a)

(b)

Figure 4.9: This figure shows two mosaics created using different frameworks and the same video sequence. Each image is rotated by 90 degrees. (a) Geo-referenced mosaic created using a single pose to geolocate the mosaic. (b) Geo-referenced mosaic created using the work presented in [1].

Chapter 5

Conclusion

This thesis has presented a novel modification to the traditional BA method enabling the creation of georeferenced mosaics that are comparable in quality to prior methods. This method makes direct use of pose information obtained from an IMU/GPS unit thereby allowing us to accurately geo-reference the mosaic to the world.

The key insight used to set up this bundle adjustment problem is that the frame-to-frame homography mapping is also a function of the poses of the camera at the time the images were taken. Because they define a "visually appealing" mosaic, the computed perspective mappings can be treated as constraints on the true pose estimates. A measurement of this pose is directly computed by the IMU/GPS system on-board the MAV. If we assume that for each frame an estimate of the camera location is returned by the IMU/GPS system, then a constrained optimization routine can be used to determine the set of camera poses that are closest to the measured pose values while meeting the constraints imposed by the frame-to-frame homographies.

Using our method, we have demonstrated mosaics created from over 900 images resulting in geo-location errors on average of less than 7m. We have also shown using synthetically generated video that our method can handle over 7,000 images. Furthermore, we have demonstrated geo-registration without any attitude information from the MAV. Visually appealing mosaics were achieved in both cases.

5.1 Recomendations for Future Work

Even though the proposed system has been shown to achieve its objectives, there are still key areas that could be improved. The first of these is the extension of the current algorithm to create 3D mosaics of desired areas. This would necessitate a fundamental change in the BA so that it works with non-planar terrain and the incorporation of digital elevation information. However, by making such changes to the design we can increase the contextual information of the mosaic and subsequently it usability.

Second, the algorithm can be extended to incorporate homographies generated using reference imagery (i.e. reference-to-image perspective transformation). Currently, when a map is being created, we assume there is no pre-existing geo-registered imagery available. However, when prior imagery is available, the accuracy and alignment of images to each other and to the geo-coordinate system can be improved. This change would only require minor modifications to the parameter and measurement spaces, such as the addition of a virtual camera pose and virtual camera calibration.

Third, the algorithm can be extended using existing superresolution techniques. Currently, when multiple images are being captured of a given area, a single image is chosen to represent the area. However, when multiple images are captured of an area, it is possible to achieve "super-resolution", or better resolution due to multiple images than can be achieved by any one image. This change would increase the visual information of the mosaic and subsequently it usability.

Lastly, the current implementation can be speed up by optimizing the current implementation. This can be done in a variety of ways. First, we can use optimized linear algebra libraries that take full advantage of the sparseness normal equations thereby speeding up the BA method. Second, the program would benefit from the explicit use of vectorization instruction sets, such as the SSE and AltiVec instruction sets. These instruction set would also speedup matrix operations. Third, the application would also benefit from explicit parallelization. This would allow the application to fully utilize the available resources.

Bibliography

[1] E. D. Andersen, “A surveillance system to create and distribute geo-referenced mosaics from suav video,” masters Thesis. xxiv, 3, 61, 62, 63, 64

[2] D. S. A. Brown, “High accuracy autonomous image georeferencing using a gps/inertial-aided digital imaging system,” *Institute of Navigation*, January 2002. 2

[3] S. Negahdaripour and X. Xu, “Mosaic-based positioning and improved motion-estimation methods for automatic navigation of submersible vehicles,” *IEEE Journal of Oceanic Engineering*, vol. 27, pp. 79–99, 2002. 3

[4] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, “Improving vision-based planar motion estimation for unammaned aerial vehicles through online mosaicing,” in *2006 IEEE International Conference on Robotics and Automation*, 2006. 3

[5] C. N. Taylor and E. D. Andersen, “An automatic system for creating geo-referenced mosaics from mav video,” in *IEEE International Conference on Intelligent Robots and Systems*, 2008. 3

[6] R. Kumar, H. Sawhney, J. C. Asmuth, A. Pope, and S. Hsu, “Registration of video to geo-referenced imagery,” in *Fourteenth International Conference on Pattern Recognition*, vol. 2, Aug 1998, pp. 1393–1400. 4

[7] R. Kumar, S. Samarasekera, S. Hsu, and K. Hanna, “Registration of highly-oblique and zoomed in aerial video to reference imagery,” in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 4, Sep 2000, pp. 303–307. 4

[8] Y. L. G. Medioni, “Map-enhanced uav image sequence registration and synchronization of multiple image sequences,” *Computer Vision and Pattern Recognition*, pp. 1–7, 2007. 4

[9] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, “Bundle adjustment–a modern synthesis,” *Vision Algorithms: Theory and Practice*, vol. 1883, pp. 298–372, 2000. 5

[10] H. S. S. Steve Hsu and R. Kumar, “Automated mosaics via topology inference,” *IEEE Computer Graphics and Applications*, vol. 22, pp. 44–54, Mar/Apr 2002. 6, 39

[11] O. P. H. Singh, "Toward large-area mosaicing for underwater scientific applications," *IEE Journal of Oceanic Engineering*, vol. 28, no. 4, oct 2003. 6

[12] C. BUCKLEY, "Photomosaicing and automatic topography generation from stereo aerial photography," pp. 1–83, 2006. 6, 7

[13] V. M. Roberto Marzotto, Andrea Fusiello, "High resolution video mosaicing with global alignment," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004. 6, 7, 17, 20, 39, 43, 44

[14] R. Hartley and A. Zisserman, *Multiple View Geometry.* Cambridge, UK: Cambridge University Press, 2003. 6, 10, 31

[15] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. 8, 10, 32

[16] *Intel Integrated Performance Primitives Reference Manual*, A70805-021us ed., Intel, 2007. 9

[17] R. L. R. Thomas H. Cormen, Charles E. Leiserson and C. Stein, *Introduction to Algorithms,*, 2nd ed. MIT Press and McGraw-Hill,, 2001. 9, 42

[18] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *9th European Conference on Computer Vision*, 2006. 10, 27, 29

[19] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 17–19 June 1997, pp. 1000–1006. 10, 30

[20] Intel Corporation, "Open source computer vision library," August 2006, http://www.intel.com/technology/computing/opencv. 14

[21] D. Nister, "Frame decimation for structure and motion," *SMILE*, pp. 17–34, 2000. 16, 17

[22] T. C. M. Subbarao and A. Nikzad, "Focusing techniques," *Optical Engineering*, p. 28242836, 1993. 17, 20

[23] N. Ng Kuang Chern, P. A. Neow, and J. Ang, M. H., "Practical issues in pixel-based autofocusing for machine vision," in *Proc. ICRA Robotics and Automation IEEE International Conference on*, vol. 3, 2001, pp. 2791–2796. 17, 20

[24] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," vol. 1-35, October 2006. 23, 46

[25] J. J. Kuffner, "Effective sampling and distance metrics for 3d rigid body path planning," in *Proc. IEEE International Conference on Robotics and Automation ICRA '04*, vol. 4, Apr 26–May 1, 2004, pp. 3993–3998. 23

[26] P. P. Johannes Bauer, Niko Sunderhauf, “Comparing several implementations of two recently published feature detectors,” *International Conference on Intelligent and Autonomous Systems*, 2007. 27

[27] C. Evans, “Notes on the opensurf library,” January 2009. 29

[28] K. Schutte, “An edge labeling approach to concave polygon clipping,” 1995. 40

[29] S. H. Harpreet S. Sawhney and R. Kumar, “Robust video mosaicing through topology inference and local to global alignment,” *Proc. of the European Conf. on Computer Vision,*, 1998. 43

[30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” vol. 13, no. 4, pp. 600–612, April 2004. 55