

PHOTOMOSAICING AND AUTOMATIC TOPOGRAPHY GENERATION FROM STEREO  
AERIAL PHOTOGRAPHY

by

CRAIG BUCKLEY

B.S., Kansas State University, 2006

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering  
College of Engineering

KANSAS STATE UNIVERSITY  
Manhattan, Kansas

2008

Approved by:

Co-Major Professor  
Dr. Dale Schinstock

Co-Major Professor  
Dr. Chris Lewis

## **Copyright**

Copyright (c) 2008 Craig Buckley

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

## **Abstract**

The Autonomous Vehicle Systems Lab specializes in using autonomous planes for remote sensing applications. By developing an inexpensive image acquisition platform and the algorithms to post process the data, remote sensing can be performed at a lower monetary cost with shorter lead times. This thesis presents one algorithm that has shown to be an effective alternative to the traditional Bundle Adjustment (BA) algorithm used for making composite images from many individual overlapping images. BA simultaneously estimates camera poses and visible feature locations from blocks of overlapping imagery, but is computationally expensive. The alternate algorithm (ABA) uses a cost function that does not explicitly include the feature locations. For photographic sets covering large areas, but having overlap only between adjacent photos, the search space and consequently the computational cost is significantly reduced when compared to typical BA. The usefulness of the algorithm is demonstrated by comparing a digital elevation model created through the ABA with LIDAR data.

# Table of Contents

List of Figures .....	vi
List of Tables .....	vii
List of Abbreviations .....	viii
Acknowledgements .....	x
Notation .....	xi
CHAPTER 1 - Introduction and History .....	1
Description of the desired mapping products .....	1
Review of the history of photogrammetry .....	4
Literature Review .....	5
Mosaicing with well known pose and topography .....	6
Intensity based mosaicing .....	6
Feature based mosaicing .....	7
CHAPTER 2 - Description of Image to Mosaic Process and Workflow .....	9
Flight Path Generation .....	10
Content extraction and correlation .....	11
Triangulation .....	12
Terrain extraction .....	13
Mapping .....	13
CHAPTER 3 - Discussion of Hardware .....	14
Airframe .....	14
Piccolo II Autopilot .....	15
Cameras .....	16
Effects of the Rolling Shutter .....	16
CHAPTER 4 - Math and Associated Algorithms .....	18
Scale Invariant Feature Transform .....	18
Levenburg-Marquardt Minimization .....	20
Creation of a Scene .....	25
Camera Model and Projective Geometry .....	26

Coordinate Definitions and Transformations .....	27
Camera Calibration .....	32
CHAPTER 5 - Cost Function Definition for Bundle Adjustment and Alternative Bundle	
Adjustment.....	37
Bundle Adjustment Cost function.....	37
Bundle Adjustment Jacobian .....	39
Advantages and Disadvantages of Bundle Adjustment .....	39
Extensions of Bundle Adjustment.....	40
Examples of BA .....	40
Alternative Bundle Adjustment Cost Function.....	40
Advantages and Disadvantages of Alternative Bundle Adjustment .....	42
CHAPTER 6 - Artifacts of ABA .....	
Reduction of Scale .....	44
Layering.....	45
Conclusions.....	48
CHAPTER 7 - Experimental Results.....	
Layering Experiment.....	49
ABA with Aerial Photography .....	53
Mosaic Creation .....	58
CHAPTER 8 - Future Research and Conclusions .....	
Future Research .....	60
References .....	62
Appendix A - GNU Free Documentation License.....	63

## List of Figures

Figure 1.1 Example of a mosaic based upon initial pose estimates projected onto a plane	3
Figure 1.2 Mosaic based upon refined poses with images projected onto topography	4
Figure 2.1 A flight path with wind and coverage taken into consideration	11
Figure 2.2 Key points have been circled with lines connecting correlated pairs.	12
Figure 3.1 ECat UAV	14
Figure 3.2: Piccolo II Autopilot Hardware	15
Figure 4.1 Pseudo Code of the LM Algorithm	24
Figure 4.2 Estimating a feature location based upon pose estimates from 3 images	25
Figure 4.3 A pin-hole camera model projection with the image plane behind the CoP	26
Figure 1.4 Single dimension view of the projection onto a defined image plane	30
Figure 4.5 Formation of a vector based upon an extracted feature	31
Figure 4.6 Example of significant lens distortion	33
Figure 4.7 Corrected version of figure 4.4 using a calibration model	34
Figure 4.8 Mapping of radial distortion for Canon Digital Rebel XT camera	35
Figure 4.9 Mapping of tangential distortion components for Canon Digital Rebel XT	36
Figure 5.1 Calculation of the cost function for ABA in world space for two perspectives	41
Figure 6.1 An example of a sub-optimal zero cost solution with three estimates in a plane	46
Figure 6.2 A zero cost solution with variable scale	47
Figure 7.1 Initial scene with features and CoPs visible	50
Figure 7.2 Final solution with features and CoPs visible	51
Figure 7.3 6 selected features from figure 7.2	52
Figure 7.4 Extracted topography from 11 overlapping images	53
Figure 7.5 The extracted topography overlaid on a LIDAR scan	55
Figure 7.6 The extracted topography overlaid on a subset of the LIDAR scan	56
Figure 7.7 Side perspective of figure 7.4 to show a cross-section view	57
Figure 7.8 The difference between the topography and the LIDAR scan.	58

## List of Tables

Table 7.1 6 selected features from figure 7.2 showing the range of values associated with each feature	52
--	----

## **List of Abbreviations**

BA - Bundle Adjustment

ABA - Alternative Bundle Adjustment

pose - position and orientation

CoP - Center of Projection

pose - position and orientation

DSLR - Digital Single Lense Reflex

GIS - Geographical Information System

DTM - Digital Terrain Model

OI - Operator Interface

SIFT - Scale Invariant Feature Transform

SGD - Steepest Gradient Descent

GN - Gauss Newton

NED - North East Down Coordinate System

CCD - Charge Coupled Device

LM - Levenberg-Marquardt algorithm

TIN - Triangulated Irregular Network

CCT - Cloud Cap Technologies

IMU- Inertial Measurement Unit

TAS - Total Air Speed

PWM - Pulse Width Modulation

CAN - Controller Area Networking

KPBRS - Kansas Prairie Biological Research Station

GPS - Global Positioning System

DGPS - Differential Global Positioning System

GPIO - General Purpose Input Output

## **Acknowledgements**

I would like to thank Dustin Denault for his work with the autopilot system that made the experimental photos possible. I would also like to thank Andrea Noonan for her work on the flight path generation software. I have used figures and information from both theses while discussing each section in this thesis. Jeff Finley's photos of the camera calibration board made the calibration discussion more clear.

## Notation

Definitions and notations that are introduced throughout the thesis are documented here on one page for easy reference.

Each coordinate space variable has a subscript describing the coordinate system in which it resides. For example a variable such as  $p_w$  denotes a point in world space.

$$\mathbf{p}_f^w = \begin{bmatrix} x_f^w \\ y_f^w \\ z_f^w \end{bmatrix} - \text{world NED coordinate space feature}$$

$$\mathbf{p}_f^i = \begin{bmatrix} u \\ v \end{bmatrix} - 2 \text{ dimensional image space feature transformed from } p_w$$

$$\mathbf{p}_f^c = \begin{bmatrix} x_f^c \\ y_f^c \\ z_f^c \end{bmatrix} - \text{point in the body/camera frame used}$$

$$\mathbf{p}_f^{i'} = \begin{bmatrix} u \\ v \end{bmatrix} - 2 \text{ dimensional image space feature detected by the feature detection algorithm}$$

$\mathbf{u}$  - unit vector in body coordinates denoting the direction from the CoP towards the detected feature

$u, v$  - scalar coordinates in image space

$f$  - focal length of the camera and lens

$\mathbf{K}_k$  - pose vector for camera  $k$

$\phi$  - roll

$\theta$  - pitch

$\psi$  - yaw

$\mathbf{c}$  - cost vector

$J$  - partial of  $\mathbf{c}$  with respect to  $\mathbf{K}$ ,  $\frac{\partial \mathbf{c}}{\partial \mathbf{K}}$

$H$  - second partial of  $\check{c}$  with respect to  $\check{K}$ ,  $\frac{\partial^2 \check{c}}{\partial \check{K}^2}$

$\lambda$  - Damping term of Levenberg-Marquardt

$R_{c/w}$  - Rotation vector that will transform inertial coordinates to body coordinates

$\rho$  - distance from the principle point of an image

$k_1 L k_4$  - calibration coefficients

$k$  - a reference to a specific camera

$l$  - a reference to a specific feature

$n$  - number of features in a scene

$m$  - number of perspectives of each feature

$o$  - number of cameras

feature – object in 3d space that can be identified in multiple images with key points

key point – a perspective of a feature in image space

scene parameters – all variables that can be changed within the search space (pose, topography, and intrinsic camera parameters)

DTM – set of grid points with associated elevations for surface modeling

TIN – set of irregularly space points with associated elevations for surface modeling

## **CHAPTER 1 - Introduction and History**

The AVS lab specializes in remote sensing with inexpensive autonomous planes. Composite images (mosaics) comprised of many-high resolution images are one of the products offered by the post processing of images taken with the onboard camera. Though the avionic sensors monitor the aircraft state during a flight, the resulting data is not accurate enough to generate high quality mosaics. Through matching of overlapping content in the images, updates on the states of the aircraft can be made that give superior mosaics when compared with the original data. Bundle Adjustment (BA) has become the industry standard for making these state updates based upon multiple input images despite being computational expensive and non-robust. This thesis presents an Alternative Bundle Adjustment (ABA) algorithm to complement the standard BA algorithm used for making mosaics. ABA has the advantage of being computationally less expensive in certain situations and more robust when working with poor initial estimates of camera position and orientation (pose). This thesis discusses the algorithms associated with the process and how ABA compares with traditional BA. It also presents two experimental processes and a comparison to LIDAR topography.

### **Description of the desired mapping products**

The AVS Lab's ECat unmanned aerial vehicle (UAV) enables a commercial off the shelf digital single lens reflex (DSLR) camera to take high resolution, low altitude aerial images over any area accessible by short range aircraft. When the images are combined to form a mosaic, a significantly higher resolution image can be created than with standard high altitude aerial imagery. This also has the added benefit of combining large, hard to visualize data sets by eliminating the redundant overlapping sections. Solving for the necessary parameters and

applying them towards the final mosaic process is a non-trivial challenge. Such an image needs to take into account topography, radiation models, and geo-referencing in order to be a useful Geographical Information System (GIS) product. The completed mosaic not only represents a view from above, but represents distances correctly without distortion due to perspective. This is beneficial from a biological research perspective since data from multiple perspectives can be combined and compared in a consistent manner. It also allows a unified scale to be applied to all photos for relative size comparisons such as those necessary in estimating areas affected by burning fields. Additional sensor information can be integrated into the mosaic to correlate separate sensors since the aircraft state estimates have been improved through the mosaicing process. The ideal mosaic has the following qualities:

- Perspective due to non-nadir facing aircraft pose and the perspective transformation is eliminated
- Radiation models have taken into account the difference in light reflection due to topography
- Each unique object only appears once in its entirety in the final image
- Tall objects appear as though the photo was taken from above without significant distortion due to the height of the object



**Figure 1.1 Example of a mosaic based upon initial pose estimates projected onto a plane**

Figure 1.1 shows a sub-optimal image without many of the properties listed above. It was formed by projecting each image onto a plane calculated with the initial pose estimates. Objects in the left side of the image appear multiple times while the road in the center appears discontinuous and not in its entirety. Though this presents a rough estimate of the scene, it is not correct and can be improved to fit many of the ideals presented above through the use of photogrammetry. Fig 1.2 shows this corrected photo as an example of what it should look like.



**Figure 1.2 Mosaic based upon refined poses with images projected onto extracted topography**

### **Review of the history of photogrammetry**

Soon after photography was developed, it was understood that scientific measurements could be taken with a reasonable degree of accuracy. Surveying for urban planning, military intelligence, and measurement for architectural purposes were the first to enjoy success in the field [16]. The United States military and Geological Survey were among those who first used aerial film photography methods for land management and planning purposes. Large mosaics could be assembled by slicing and assembling the high-resolution photographs onto a single high altitude photo for global reference [16]. Though this created large high-resolution mosaics, many of the individual processes for assembling the photos were crude. Seams between images were distinct slices with very little blending or radiometric correction. Orthorectification, the process of removing distortion due to perspective and topography, was attempted by tilting the paper rather than taking into account topography [33][34][35]. Geo-referencing, associating geographic coordinates for an image location and image splicing, were completed manually which led to

accumulating error if accurate geo-referencing was not available. While many novel solutions addressed these problems, digital image processing has allowed many of the same operations to be performed with higher quality and much shorter processing times. Due to the flexibility allowed by image processing and software, many new or previously unmanageable methods have been implemented.

## **Literature Review**

The AVS Lab strives to create high resolution, low altitude mosaics from multiple smaller overlapping images. Unless both pose and topography are accurately known, the estimates of both must be improved before the images can be assembled into the desired mosaics. In the case that the pose is accurately known over an area with reliable topographical maps, the desired mosaic can be generated by projecting the images onto the topography. In the instance where pose and topography are unknown, this process must make assumptions about the environment or extract the necessary information from the photos alone.

As a first attempt at combining multiple photos into a mosaic, photos were projected onto a common plane using pose estimates without otherwise modifying the images. Figure 1.1 shows this result and some of the artifacts associated with it. As a second attempt, photos were projected onto that same plane, but given the freedom to translate to match intensity correlations with neighboring images. This improved the image, but errors due to orientation and scale were not eliminated. For a third and final approach, pairs of key points (features in image space) were detected in adjacent images to form a network of relationships between all overlapping images. By optimizing the pose parameters, better estimates of both orientation and position for each camera could be obtained in addition to the extraction of topography. The following sections

discuss the implementation details and characteristics of each method along with examples of published research.

### ***Mosaicing with well known pose and topography***

Mosaicing based upon the initial pose estimates is the most intuitive method for creating composite images and performs well when topography and pose are well known. When coupled with a metric camera, a sequence of images can be projected onto a topography with impressive results [36]. When pose is not well known, the same method will only give a crude mosaic regardless of the certainty of the topography. Due to the algorithm's simplicity, it has shown some success for near real-time mosaic creation over flat topography such as [23]. [24] has shown some success when a larger reference image and topography is already available.

### ***Intensity based mosaicing***

Mosaics based upon projections onto a plane from the initial pose estimates can be improved by allowing the projections to translate to match intensity correlations between images. Intensity correlation is calculated by overlapping areas of an image and subtracting the values of one image from another. The overlapping image can then be translated to reduce the intensity difference between overlapping sections. Though this cannot improve errors in orientation and scale, it can improve the mosaic if there are errors in camera positions such as [9][17][21][22][31][32]. In the instance that a false match or an improper relationship has been found, this error will accumulate and can damage the rest of the mosaic. This method begins to fail when changes in lighting, orientation, or scale are necessary. For optimal results, each photo needs to be taken from a nadir perspective with consistent lighting at the same altitude.

Variations in any of these key elements will cause a poor mosaic. For these reasons, most aerial

mosaicing has moved away from this method despite it being the original process with film as described in [16].

### *Feature based mosaicing*

Feature based mosaicing develops relationships between photos before any projection happens by identifying common elements in overlapping images. By abstracting the data within an image to a set of correlated key points between images, the algorithm can be resistant to changes and errors in lighting, scale, and pose. The mosaicing algorithm is then free to adjust all parameters (pose, topography, and internal camera characteristics) within the system to find a solution that yields better estimates of all parameters to form a mosaic. Due to the flexibility allowed by abstracting the problem with features, many new algorithms have been presented to generate topography and mosaics simultaneously from sets of correlated key points.

Within feature based mosaicing, two groups have emerged. The first group of literature uses one or more homographies, projections onto planes, to account for perspective [11][12][13][14][15][26][27][28][29]. This method performs well for planar scenes or for scenes with few overlapping images. As images are added, error can accumulate when a non-planar environment is being represented with planar constructs. The second group, known as BA, uses an estimate of the position of each feature in world space to determine a surface onto which images can be projected. This method has shown the greatest promise since it is actively approximating the topography while improving the estimates for pose and camera characteristics. Examples of BA along with variations and optimizations can be found in [3][10][26]. Despite the superior final product offered by BA algorithms, they are computationally complex and non-robust. This thesis offers an alternative, ABA, which significantly reduces the computational complexity while improving the robustness at the cost of the quality of the final product. While

this is not a new concept and is published in [39], it is not a well-known algorithm and was developed independently.

## **CHAPTER 2 - Description of Image to Mosaic Process and Workflow**

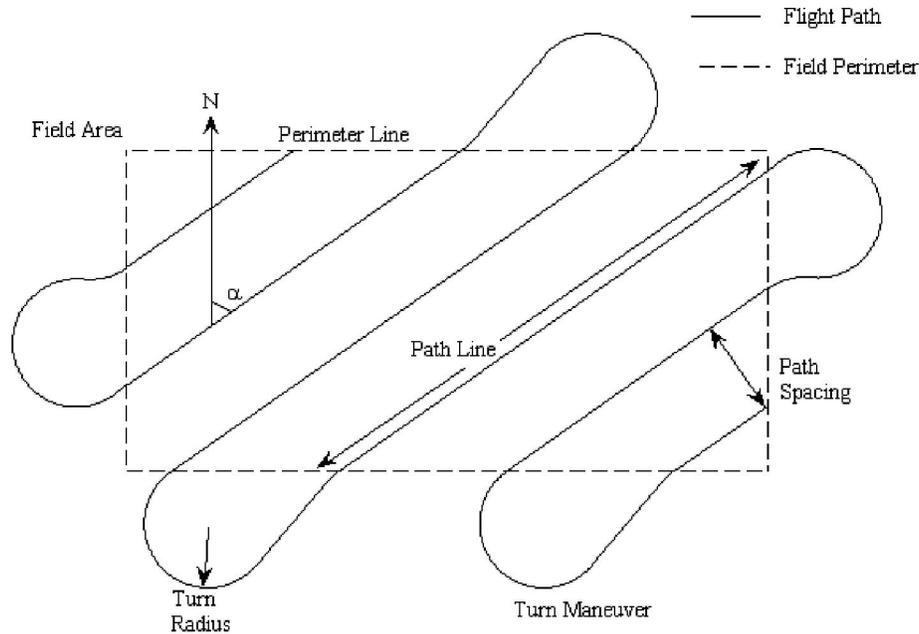
ABA and BA only address one step in the mosaicing process. This chapter gives a brief overview of each stage in the sequence of steps from planning the aerial survey to combining images into a single mosaic. The process proceeds as follows:

1. Given an area to survey, a flight plan is generated that covers the area with a sequence of passes optimized for wind direction, flight time, the UAV's maneuverability, and image overlap.
2. The UAV flies the generated path taking images with the onboard camera at specified locations.
3. The imagery is processed to extract key points from the overlapping regions of image pairs.
4. Using the key points as landmarks and the initial pose estimates from the autopilot, either BA or ABA is used to both improve the pose estimates and to estimate the location of each feature.
5. A digital terrain model (DTM) is created by interpolating elevations for a grid based upon the feature estimates from the previous step.
6. The surveyed images are then reprojected onto the DTM to create a correlation between the image content and the topography extracted previously.

The remainder of the chapter reviews each of these steps in more detail.

## **Flight Path Generation**

The selection of a flight path can have a large impact on the suitability of the aerial survey for the mosaicing process. Because cross winds induce the aircraft to crab in order to follow the prescribed path, flight paths are selected to be into and with the direction of the wind. Crabbing significantly diminishes the region of overlap between adjacent images and should be avoided. In addition to wind direction, the flight path manages the order in which a sequence of images is taken. If the path is designed to minimize hard turns, the UAV is more likely to be nadir facing when the image is taken. Along with orienting the camera in a more nadir pose, smoother turning prevents image smear due to a quickly rotating camera while the shutter is exposing the charge coupled device (CCD). The flight path also affects how the overlapping areas interact. The placement of the overlapping areas can have some implications on how the mosaicing process combines the aerial images. If careful attention is not paid to the layout, camera poses can be arranged in such a way that singularities in the algorithm greatly reduce the quality of the mosaic. The math and software used to generate the flight path is covered in greater detail in Andrea Noonan's thesis [30].



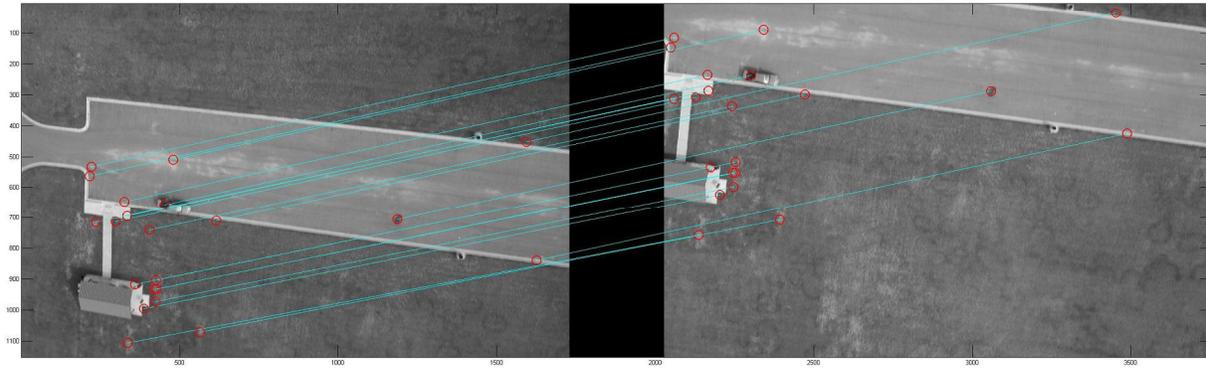
**Figure 2.1 A flight path with wind and coverage taken into consideration**

Figure 2.1 shows a typical flight path automatically generated to cover an area of interest denoted by the dashed box. The orientation defined by  $\alpha$  allows the UAV to spend most of the flight time into or with the wind. By ensuring that photos are taken when the UAV is flying into or with the wind, the survey images are likely to have greater overlap with adjacent images. This directly affects the mosaicing process since the layout of overlap forms the basis for the extraction of topography and pose updates.

### **Content extraction and correlation**

Content extraction and correlation algorithms operate on images to abstract common elements into a form that only contains the information useful to the algorithm operating on that data. In mosaicing, this process finds common objects or landmarks of opportunity known as features that are identified in multiple images. A set of key points and images are then associated with that feature. Each key point can then be used as a perspective of that feature when estimating changes in the scene parameters. Figure 2.2 shows an example of this for two overlapping

images. The key points are circled in each image and connected with lines that show the correlations. A similar process is done for areas with overlap from multiple images.



**Figure 2.2 Key points have been circled with lines connecting correlated pairs.**

## **Triangulation**

Triangulation refers to the simultaneous estimation of pose and topography. The key points from the previous step yield a bearing (direction only) from the camera that can be used with other perspectives of the same feature to estimate a position in world space for the associated feature. Updates for the pose and feature location can be calculated by assigning a cost to errors with the current state. If all the cameras can agree more by moving a feature in world space or adjusting the pose, then an update will be applied and the process will iterate until an improvement can no longer be found. A successful triangulation yields a state vector with a low cost for which all the cameras agree on locations for each feature and an updated set of poses for the cameras. The updated poses are more accurate than the original measurements relative to the overall scale and the accuracy of the optics that took the images. The feature locations form a triangulated irregular network (TIN) that describes the topography of the scene.

## **Terrain extraction**

TIN models may or may not cover the entire area, and the points are arranged in an irregular, unevenly spaced grid. As a result, some areas have clumps of data while others are sparsely populated. Luckily, the triangulation step also produced improved pose estimates. This allows further correlation searches using less distinct features to fill in some of the sparse regions producing a denser representation of the scene. Furthermore, densely populated TIN models are easily converted through interpolation into digital terrain models (DTMs) that have points on a regular grid.

## **Mapping**

The final step is to create the mosaic image by combining the imagery, the improved pose estimates, and the DTM. The images are projected onto the DTM to determine the color at each location. Most locations have color projected from two or more images due to overlap. Either the most nadir image is used, or a weighted blending determines the final projected image. This process creates an accurate visual and spatial representation of the surveyed scene. Ortho-rectified maps can then be created from this representation. In addition, viewers may be used to provide an interactive perspective of the scene from any location and orientation. Both of these methods make use of perspective transformations discussed later in chapter 4. Ortho-rectified maps are then created using oblique or orthographic projections.

## CHAPTER 3 - Discussion of Hardware

### Airframe

A SIG Kadet Senior ARF hobbyist kit is used for the ECat UAV airframe. The high wing design and approximate 2m wingspan provide stability and adequate lift. Thrust is generated by a Hacker C50 brushless motor with a 16x10" CAM carbon composite folding propeller. Hitec HS-81 servos actuate the elevator, rudder, aileron, and flap control surfaces. The ECat airframe is shown in figure 3.1.



**Figure 3.1 ECat UAV**

A few modifications to the basic airframe are necessary for hardware accommodation and mission specific requirements. A larger lightweight carbon composite payload bay was designed and inserted in place of the original balsa wood compartment, which relied on several cross-members for structural support. Another noticeable change includes the addition of skid style landing gear, allowing grass prairie landings. Finally, a balsa wood mount was fabricated for the autopilot and resides in the fore compartment between the firewall and the payload bay bulkhead.

## Piccolo II Autopilot

The Piccolo II autopilot, purchased from CCT, provides the functionality required for the Ecat UAV navigation and control applications. With a mass of 233 grams and a 12.2cm x 6.1cm x 3.8cm size, the autopilot avionics unit, pictured in figure 3.2, readily mounts within most UAV airframes.



**Figure 3.2: Piccolo II Autopilot Hardware**

A Motorola MPC555 processor, executing at 40 MHz, provides computation and communication with five RS232 payload ports, up to 10 PWM servo channels, 2 CAN ports, 6 GPIO pins, and 4 analog input pins. These peripheral devices are interfaced via a 44 pin D sub connector and a high density 25 pin microdot connector. The main cpu board has several daughter boards including a MHX-910 Datalink Radio chipset, a Motorola M12 GPS module, IMU, and dual ported mpxv50045 dynamic pressure and mpx4115a static pressure sensors. The radio link allows the streaming of data to and from a ground station unit that provides a networking interface between multiple avionics units and CCT's operator interface (OI) software. The OI, which executes on a PC, displays telemetry updates and also enables the dynamic changing of commands, gains, and flight plans. The IMU, pre-calibrated by CCT, delivers three axis gyro and accelerometer readings to the processor over a serial port. The gyros measure angular rates up to 5.2 radians per second, and the accelerometers record up to 10G

accelerations. The Motorola M12 GPS unit generates an estimate of the position and velocity of the Synergy Systems AR-05 antenna. The Piccolo also supports DGPS corrections received from the ground station. The Pitot and static ports provide air data information vital for TAS estimation.

## **Cameras**

A DSLR Canon Rebel XT was used as the onboard camera. It was selected for its light weight, high resolution (8 megapixel effective), high quality lenses, ability to manually control all settings, and the ability to remote trigger it. The camera has been mounted in a foam block that fills the payload bay to reduce vibration and movement during flight. Despite the advantages of using a consumer camera, numerous problems are introduced by using consumer camera. Though the available lenses are high quality, they have not been designed for minimal distortion. This causes a need for a frequent camera calibration procedure as outlined in chapter 4. In addition, the camera contains a rolling shutter that does not expose all elements of the CCD at the same time.

### ***Effects of the Rolling Shutter***

DSLR cameras control the exposure time with two opaque curtains that slide across the CCD at different times to expose it for the difference between the two trigger times. Exposure of the CCD occurs in the following order:

1. Curtain 1 moves to expose the CCD to light from the lens.
2. After a specified exposure time, curtain 2 follows curtain 1 to cover the CCD.
3. After all elements of the CCD are read, both curtains reset for the next image.

It takes 5ms for one curtain to travel across the CCD. For any exposure time less than 5ms, both curtains are moving at the same time. This creates a thin band of light that exposes the CCD. If

the camera is in motion at that time, the resulting image is not a frozen snapshot of the scene but a scan of a dynamic scene. This affects the image by moving pixels from where they would be if all elements of the CCD were exposed at the same time. The amount of warping present in an image depends on the rate of translation and rotation at the time the image was taken. A simple calculation compares the contribution due to translation and rotation as follows:

At 100m of elevation and 1rad/s, an object moves 100m/s relative to the camera. A pixel represents 2.6cm on the ground at 100m. For a certain translation time of 5ms, a shift of 19 pixels may be seen. This can be compared with the average air speed of 15m/s that would produce fewer than 3 pixels of shift. The ground speed depends on the wind. When comparing the two, it is important to note that small but quick rotations can easily generate more distortion than the typical translational velocity.

## **CHAPTER 4 - Math and Associated Algorithms**

Before the triangulation step can be discussed in detail, the supporting math and associated algorithms must be explained. These topics include the Scale Invariant Feature Transform, the Levenberg-Marquardt Minimization, Projective Geometry, and Camera Calibration. Chapter 4 gives an overview of each topic along with references that contain greater detail for the interested reader. The cost function is introduced in concept in the minimization section, but not defined for both algorithms until chapter 5.

### **Scale Invariant Feature Transform**

The Scale Invariant Feature Transform (SIFT) completes the feature extraction and correlation step described in figure 2.2. By locating areas of high contrast, SIFT detects key points that are likely to be identified in other overlapping images. Key points that appear to be similar, but in different images, represent possible matches for a single feature. If two images overlap significantly, a large portion of each image should have correlating matches. Ideally, SIFT would identify and correlate a dense set of key points evenly distributed throughout the overlapping regions. In practice, correlations tend to cluster in areas containing high contrast objects such as trees and rocks while few key points are identified in bland areas such as grassy plains. The pairwise correlations form a mesh of relationships between image pairs. For images with large neighboring overlap, individual features may appear, be identified, and correlated in several image pairs. Each pair of perspectives of a single feature provides an estimate of the location of that feature through the projections of that feature from the respective camera poses in which it appeared.

SIFT was designed to identify common objects in overlapping images despite significant changes in scale and viewing perspective. This is important to understand since a change in scale and perspective makes a numerical comparison of two images difficult. SIFT overcomes the scale difficulty by sub-sampling the image multiple times to create a sequence of reduced scale layers that contain increasingly sub-sampled copies of the original image. Key points are then selected when the intensity of one pixel is higher or lower than all surrounding pixels in that layer and the next sub-sampled layer. If two overlapping images have different scales, then the same feature should be detected in both images but in different layers. To identify a key point as unique regardless of orientation, a descriptor is created based upon the intensity changes within a predefined distance. Since the descriptor uses gradient vectors from an area around the key point, there will be many of them, each with a different magnitude and direction. All gradient vectors are then given a common origin and rotated to align the largest gradient at 0 degrees. The largest 64 vectors are then described with two-vectors and concatenated giving a large descriptor that identifies the key point in question. The descriptor can then be compared with other key points in overlapping images through use of the dot product operator. Similar descriptors will have a value close to unity when normalized by the product of the two vector magnitudes. A match is considered successful when the normalized dot product is significantly larger than all other comparisons in the image. This guarantees that a comparison is not only the most likely candidate, but also much better than any other key point.

Scale space elegantly handles many of the challenges that have prevented effective image correlation from earlier algorithms. Since SIFT takes place in scale space, the descriptors are rotated relative to the largest vector, and works with key points instead of large areas, it is ideal for finding overlapping sections despite changes in scale and perspective.

Though these elements work well to eliminate false positives, repetitive objects and objects that change in appearance as the perspective or the illumination changes will give incorrect results. A common example of this is shown when three images of overlapping content are analyzed. An object may be detected as common in the first two images and the last two images but not the first and last. This makes for a challenging problem when trying to identify a relationship between the first and last images. A similar problem arises with photos of repetitive objects. If an area has many overlapping images inconsistent correlation is more likely to happen but the calculated data is also more valuable since the feature in question appears in more images. The process of identifying features based upon pairs of correlations has been completed using an original feature manager algorithm that organizes the pairs to make a complete set of features and the associated key points. The feature manager makes use of the widely known SIFT algorithm developed by David Lowe [1]. David Lowe's implementation has been used for all function calls within the feature management software. More information can be found by reviewing [19][20] and the widely available literature associated with it.

### **Levenburg-Marquardt Minimization**

Both ABA and BA rely on a minimization algorithm to find a low value for a sum-of-squares cost function. The cost functions associated with BA and ABA are nonlinear with respect to the parameters varied in the minimization. The Levenberg-Marquardt (LM) algorithm, a hybrid approach using both Steepest Gradient Descent (SGD) and the Gauss-Newton (GN) algorithm, has been shown to work well on non-linear problems and is commonly used in BA. Both BA and ABA define a vector of cost elements  $\check{c}$ . The inner product of this vector is the scalar cost function minimized over the state vector  $\check{K}$ . Here the state vector  $\check{K}$  is a concatenation of parameters describing camera poses, and in some cases other variables describing the feature

locations and camera parameters. The vector of parameters describing a camera pose is as follows:

$$\mathbf{v}_{K_k} = \begin{bmatrix} \phi_k \\ \vartheta_k \\ \psi_k \\ x_k^w \\ y_k^w \\ z_k^w \end{bmatrix} \quad 4.1$$

Here  $\phi / \vartheta / \psi$  are defined as roll/pitch/yaw using the rotation matrix defined in equation 4.9 and  $x_k^w y_k^w z_k^w$  are defined as the CoP for camera  $k$  in world coordinates. The parameters describing the location of a feature  $l$  is defined as

$$\mathbf{v}_{F_l} = \begin{bmatrix} x_l^w \\ y_l^w \\ z_l^w \end{bmatrix} \quad 4.2$$

where each component describes the position of the feature in world space.

An example of the complete state vector  $\mathbf{v}_K$  for a BA process concatenates all  $\mathbf{v}_{K_k}$  with the feature locations.

$$\mathbf{v}_K = \begin{bmatrix} \mathbf{v}_{K_1} \\ \mathbf{v}_{K_2} \\ \mathbf{M} \\ \mathbf{v}_{K^o} \\ F_1 \\ F_2 \\ \mathbf{M} \\ F_m \end{bmatrix} \quad 4.3$$

The optimization process attempts to find a value for  $\mathbf{v}_K$  that minimizes the overall cost,  $C = \mathbf{v}_c^T \mathbf{v}_c$ .

To aid in this minimization, the Jacobian  $J = \frac{\partial \mathbf{v}_c}{\partial \mathbf{v}_K}$ , is defined. The Levenburg-Marquardt

minimization process iteratively updates the state vector  $\mathbf{v}_K$  using both the Jacobian and the

gradient of the cost. The gradient of the vector cost with respect to  $\check{K}$  is calculated by  $2J^T \check{C}$ . It describes the direction of steepest cost increase for the current  $\check{K}$ . The Hessian matrix  $H$  is defined as  $\frac{\partial^2 \check{C}}{\partial \check{K}^2}$ , but is approximated by  $J^T J$ . This is important in the minimization process since  $H$  describes the curvature of the cost around the current state  $\check{K}$ .  $H$  assists in determining the update that LM calculates with

$$\delta \check{K} = (J^T J + \lambda I)^{-1} J^T \check{C} \quad 4.4$$

instead of relying solely on the gradient. The terms  $J^T J$  and  $\lambda I$  within the inverse represent the approximation of the Hessian and the gradient respectively, and  $\lambda$  adjusts the tradeoff between SGD and the GN algorithm. The minimization starts with a large initial value for  $\lambda$  since SGD should deliver the fastest convergence when far from a solution.  $\lambda$  is then reduced by 10 when the estimated update  $\delta \check{K}$  yields a lower cost. The reduction of  $\lambda$  places more weight on the GN algorithm until  $\delta \check{K}$  fails to yield a lower cost. At this point, less weight is put on the GN algorithm by increasing  $\lambda$  by 10. The tradeoff between the two algorithms is novel since it shifts between two algorithms to exploit the strengths of both methods. SGD converges faster when far from a minimum but slows when it encounters the curvature typically seen near a minimum. GN uses the curvature from the approximation of  $H$  to calculate an update that will converge on the minimum faster than SGD when near it. At no time does LM use exclusively one algorithm but instead weights the SGD component of the update more or less depending on the success of previous values of  $\lambda$ .

The pseudo code shown in figure 4.1 outlines the LM algorithm used in the triangulation step.

1.  $J$ ,  $\check{K}$ , and  $C$  are saved before any elements of  $\check{K}$  are changed.
2. An update for the state vector is calculated.

3.  $J$  and  $C$  are corrected to reflect the new choice of  $\check{K} + \delta\check{K}$ .
4. If  $C$  is lower than  $C_{last}$ ,  $\check{K} + \delta\check{K}$  is kept. Otherwise  $\delta\check{K}$  is discarded. In both cases,  $\lambda$  is updated.

```

for i = 1 to maxIterations
     $J = J_{last}$  (1)
     $\mathbf{K} = \mathbf{K}_{last}$ 
     $C = C_{last}$ 
     $\delta\mathbf{K} = (J^T J + \lambda I)^{-1} J^T \mathbf{c}$  (2)
    calculate  $J$  and  $C$  for  $\mathbf{K} + \delta\mathbf{K}$  (3)
    if (  $C < C_{last}$  ) { (4)
         $\lambda = \lambda / 10$ 
         $\mathbf{K} = \mathbf{K} + \delta\mathbf{K}$ 
    }
    else{
         $\lambda = \lambda * 10$ 
         $J_{last} = J$ 
         $C_{last} = C$ 
         $\mathbf{K}_{last} = \mathbf{K}$ 
    }
}

```

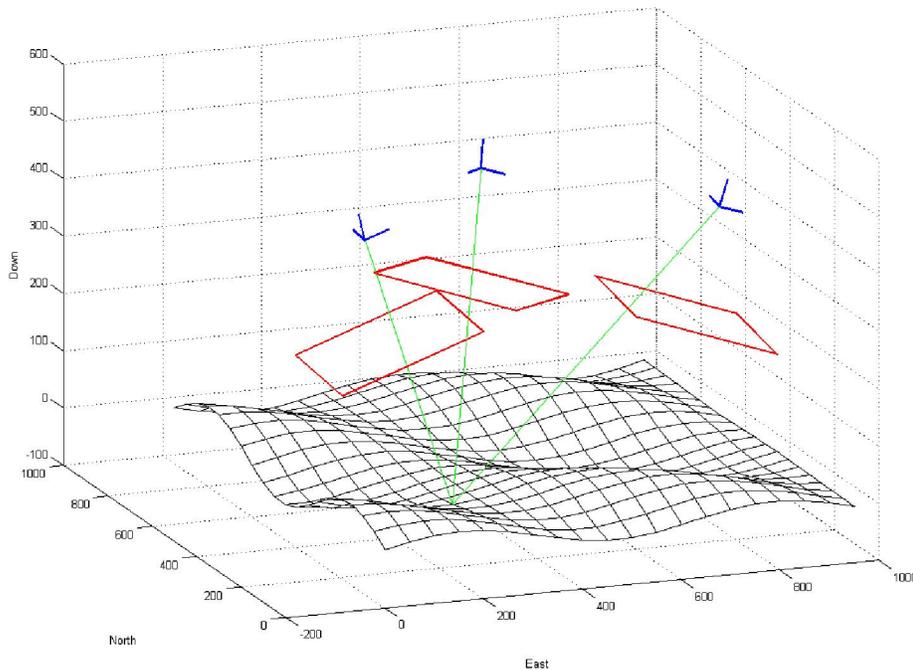
**Figure 4.1 Pseudo Code of the LM Algorithm**

It is important for the cost function to be analytic, and for the Jacobian to be computationally tractable. The LM algorithm requires computation of the Jacobian with each iteration in the minimization process. The minimization is terminated when a maximum number of iterations is reached, the norm of the gradient  $\|2J^T \mathbf{c}\|$  falls below a threshold,  $C$  falls below a threshold, or  $\lambda$  reaches a maximum value indicating that a better solution cannot be found. Marquardt recommended replacing the  $\lambda I$  term with  $\lambda * \text{diag}(J^T \mathbf{c})$ . This replaces the diagonal of the identity matrix with the half of the gradient and gives more weight to the terms with a higher gradient.

This update did not appear to improve performance in ABA and is therefore not used in the minimization routine. A more detailed description and derivation of the LM algorithm can be found in [2] [5][18].

### Creation of a Scene

A scene is comprised of the camera poses and the set of vectors that point towards the key points detected in the images. This creates the framework upon which the minimization and terrain can be computed. Figure 4.2 shows the ideal solution for one feature and three images. The green lines represent the vectors that project through the key point and the feature. Unfortunately, due to errors, the initial pose estimates are not likely to yield a solution like figure 4.2, rather, the green lines will be skew.



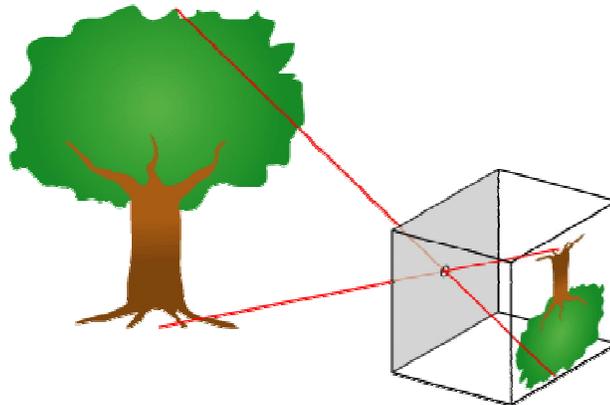
**Figure 4.2 Estimating a feature location based upon pose estimates and vectors created from the detected key points in 3 overlapping images.**

## Camera Model and Projective Geometry

Mathematics and projective geometry are used in photogrammetry to describe the projection of 3-dimensional points onto a 2-dimensional image or the reverse projection. Through a nonlinear projection, any point in a 3-dimensional space can be projected onto an image plane using a pinhole camera model. The same process can also be reversed to calculate an infinite number of points on a line that correspond with a point in image space. Figure 4.3 is a graphical representation of the pinhole camera model and illustrates three notable components.

- The object in 3-D space to project
- The Center of Projection (CoP) that all rays pass through
- The image plane on which images are projected

Models such as the one shown in figure 4.3 use an image plane that is one focal length behind the CoP. This creates an image that has been flipped vertically and horizontally. All references in this thesis refer to an image plane that is one focal length in front of the CoP where images are not flipped.



**Figure 4.3 A pin-hole camera model projection with the image plane behind the CoP [40]**

The key points to understand about a pinhole camera model are as follows:

- All light rays pass through a single point, the CoP

- Since the CoP has no area, the projected image is always in focus
- The image plane has a finite size
- The field of view for the camera can be adjusted based upon the image plane size and distance from the CoP

More information about the properties and assumptions of the pinhole camera model can be found in [26].

## Coordinate Definitions and Transformations

Several coordinate frames have been defined in 4.5-8 to assist in the visualization and creation of vectors in remaining discussions. Each vector can be expressed in another coordinate system through the use of transformations defined in 4.9-13. Every vector is denoted with a superscript declaring the coordinate frame it is relative to and a subscript for reference. A brief overview of the coordinate systems follows:

### Image space coordinates

$$\mathbf{p}_f^i = \begin{bmatrix} u \\ v \end{bmatrix} - \text{2D image space feature} \quad 4.5$$

$$\mathbf{p}_f^{i'} = \begin{bmatrix} u \\ v \end{bmatrix} - \text{2D image space feature detected by the feature detection algorithm} \quad 4.6$$

- Forms the 2 dimensional coordinate system in which images are captured.
- The units are decimal pixels.
- The origin is placed at the center of the image.

### Camera space coordinates

$$\mathbf{p}_f^c = \begin{bmatrix} x_f^c \\ y_f^c \\ z_f^c \end{bmatrix} \quad 4.7$$

- Adds an additional dimension to image space to form a 3D coordinate system aligned with the body of the camera.
- One separate coordinate frame for the pose of each image.
- The origin is defined as the center of projection of the camera.
- Units are meters.

### World space coordinates

$$\mathbf{p}_f^w = \begin{bmatrix} x_f^w \\ y_f^w \\ z_f^w \end{bmatrix} \quad 4.8$$

- Is defined using the North East Down (NED) convention.
- Has an origin defined by the base station of the autopilot.
- Units are meters.

Images are initially processed using the image coordinate system discussed in equation 4.5. Key points are then identified in the coordinate system defined in equation 4.6. The image coordinate system can be extended to create the camera space coordinates by adding a third dimension. This yields a coordinate system that can express vectors relative to the image plane at the time a specific image was taken. Lastly, vectors in camera space can be expressed in the world space coordinates by taking into consideration the pose of the plane at the time the image was exposed. The coordinate spaces defined in 4.5-8 set a framework for vectors defined in each coordinate to be expressed in other coordinates. This allows vectors such as key points that have been defined in image space to be observed in camera space or compared with correlated key points in world space. Likewise, an estimate of a feature in world space can be projected into camera space and expressed in image space to be compared with the original detected key point. The ability to move between the many camera and image space coordinates to world space provides the ability

to make comparisons between overlapping images. The ability to express vectors in a common space provides the fundamental basis for BA and ABA. The transformations to express vectors from one space into another will follow in the paragraphs surrounding equations 4.9-15.

Transformations from world space to camera space are accomplished with a rotation and a translation. Equation 4.1 defines the Euler angles used for the rotation and the position necessary for the translation for a specific camera denoted by the subscript  $k$ . Equation 4.9 can be used to transform vectors from world space into camera space and back again. This transformation is detailed in equations 4.11 and 4.12.

$$R_{c/w} = \begin{bmatrix} \cos(\phi)\cos(\psi) & \cos(\phi)\sin(\psi) & -\sin(\psi) \\ -\cos(\phi)\cos(\psi) + \sin(\phi)\sin(\phi)\cos(\psi) & \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\phi)\sin(\psi) & \sin(\phi)\cos(\phi) \\ \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\phi)\cos(\psi) & -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\phi)\sin(\psi) & \cos(\phi)\cos(\phi) \end{bmatrix} 4.9$$

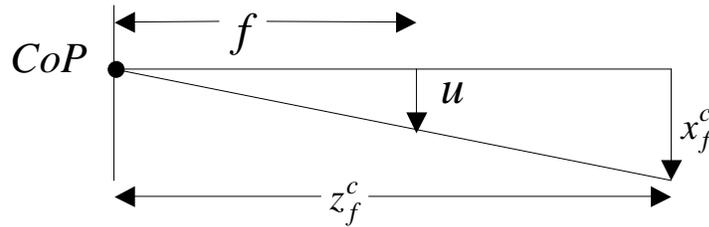
To move from camera space to world space as shown in 4.11, a feature location estimate  $\mathbf{p}_f^w$  is pre-multiplied by the rotation matrix defined in 4.8 and added to the camera position vector defined in equation 4.1 and 4.10. The additional translation is necessary since  $R_{c/w}$  can only perform rotations around the origin. The opposite transformation is described in equation 4.11. The translation must be performed in the opposite order since the rotation is defined around the origin.

$$\mathbf{p}_{cam}^w = \begin{bmatrix} x_k^w \\ y_k^w \\ z_k^w \end{bmatrix} 4.10$$

$$\mathbf{p}_f^w = \begin{bmatrix} x_f^w \\ y_f^w \\ z_f^w \end{bmatrix} = R_{c/w}^T \mathbf{p}_f^c + \mathbf{p}_{cam}^w 4.11$$

$$\mathbf{p}_f^c = \begin{bmatrix} x_f^c \\ y_f^c \\ z_f^c \end{bmatrix} = R_{c/w} \cdot (\mathbf{p}_f^w - \mathbf{p}_{cam}^w) 4.12$$

Equations 4.10-12 show how vectors and positions can be expressed in either world space or camera space. The transformation from world space can be taken a step further into image space with a projection based upon the pinhole camera model. Given a vector in camera space  $\mathbf{v}_{p_f}^c$  a corresponding vector  $\mathbf{v}_{p_f}^i$  in image space can be calculated using equation 4.13. Equation 4.13 performs a projection onto a defined image plane by multiplying by a ratio of similar triangles. Figure 4.4 shows this projection in a single dimension for  $u$  and  $x_f^c$ . The point  $x_f^c$  represents an object in camera coordinates to be projected on to an image plane that is  $f$  away from the CoP.  $u$  represents the projection of  $x_f^c$  onto the image plane. The two similar triangles are formed from  $u$ ,  $x_f^c$ , and the CoP.



**Figure 4.4 Single dimension view of the projection of a point in camera coordinates onto a defined image plane.**

$\mathbf{v}_{p_f}^i$  can then be found by multiplying by the ratio of  $f$  and  $z_f^c$ .

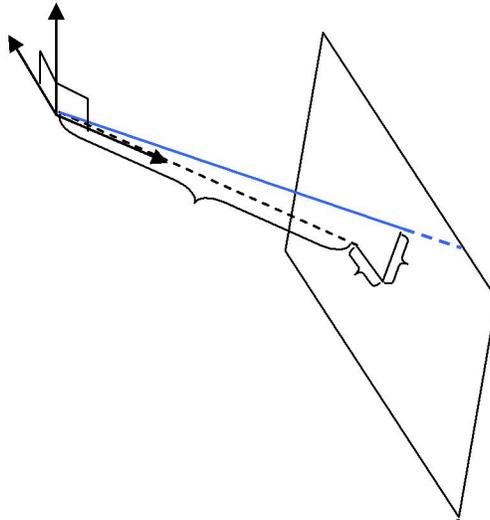
$$\mathbf{v}_{p_f}^i = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{f}{z_f^c} \cdot \begin{bmatrix} x_f^c \\ y_f^c \end{bmatrix} = \frac{f}{[0 \ 0 \ 1] \cdot \mathbf{v}_{p_f}^c} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \mathbf{v}_{p_f}^c \quad 4.13$$

Given the transformation shown in equation 4.13, it is important to realize two key points. The transformation is non-linear since it is divided by  $z_f^c$ . It then follows that any cost functions based upon projections of world space points onto an image plane will be non-linear as well. This has computational implications since a derivative of that cost function will be taken with respect to the state vector  $\mathbf{K}$ . The second notable point is that the projection from world

coordinates to image coordinates is not reversible. Since a degree of freedom is lost when converting from camera coordinates to image coordinates ( $\mathfrak{R}^3 \Rightarrow \mathfrak{R}^2$ ), there exist an infinite number of points along a line that that would yield the same  $\mathbf{p}_f^i$ . This implies that a feature must have multiple perspectives to estimate a location in world space as described in figure 4.2.

A set of infinite points can be calculated that contains the original  $\mathbf{p}_f^c$  discussed in equation 4.13 by adding a dimension to the detected key point  $\mathbf{p}_f^i$ .

$$\mathbf{p}_f^{c'} = \begin{bmatrix} u \\ v \\ f \end{bmatrix} \quad 4.14$$



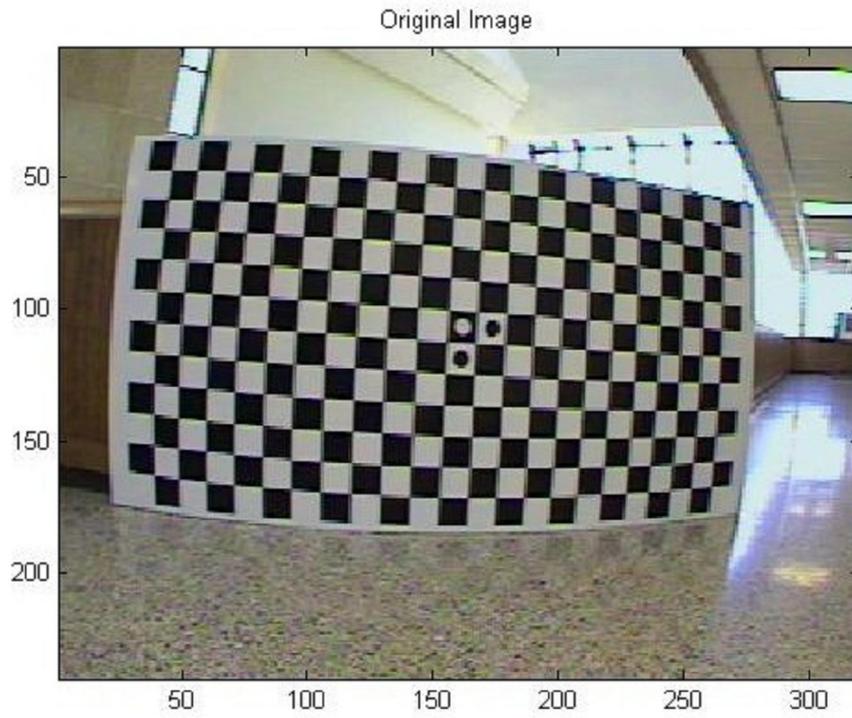
**Figure 4.5 Formation of a vector based upon an extracted feature**

Figure 4.5 depicts a single camera frame from figure 4.2 to graphically show the definition in equation 4.14. Although a single perspective of a feature does not define the location for  $\mathbf{p}_f^w$ , the feature's location  $\mathbf{p}_f^w$  can be estimated with multiple perspectives. Each perspective defines a vector along which the feature should lie. This forms the basis of the triangulation step and will be discussed in detail in chapter 5.

## Camera Calibration

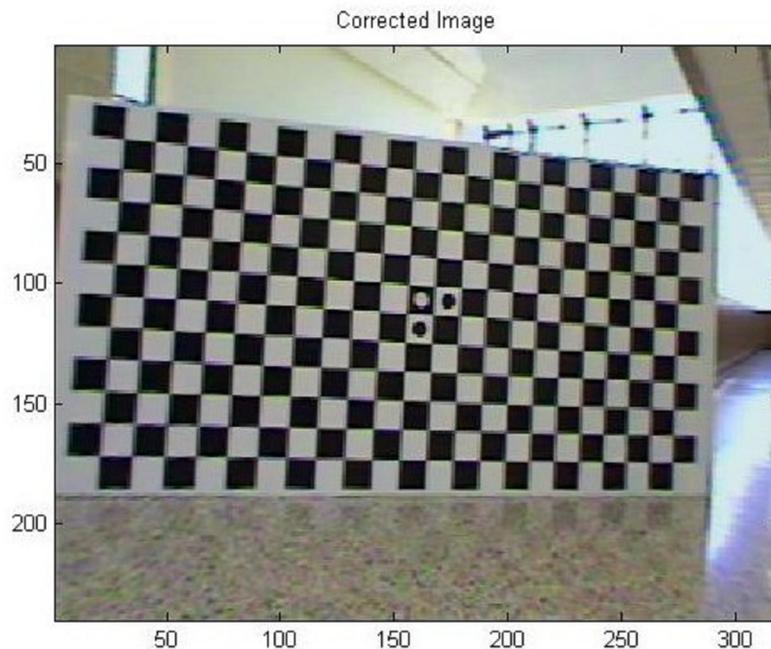
The pinhole camera model is an idealization of what happens inside a camera lens. Real lenses have a systematic bias associated with the way the glass distorts light as it passes through. The distortion of the light path creates an image where straight lines no longer appear straight and geometric relationships between objects are not preserved. An empirical model has been developed that uses multiple distorted images of a precisely known target to characterize a lens. By altering the coefficients  $k_1, k_2, p_1,$  and  $p_2$  as defined in equations 4.16 and 4.17, a set of coefficients can be found that describe and correct the observed lens distortions.

The process for calculating the coefficients of the distortion model is referred to as camera calibration. By using a target with known feature locations such as a checkerboard, the features detected in the image can be assumed to be evenly spaced and orthogonal in world space. The use of the checkerboard therefore eliminates the need to estimate the locations of the features and allows the calibration algorithm to instead solve for the pose of each camera and the distortion coefficients. Figure 4.6 shows an example of a checkerboard image used in the calibration process. It is important to note how the straight edges of the checkerboard appear curved in the distorted image. Lines intersecting near the center of the image are not as visibly affected since distortion is exaggerated near the edge of the image. Figures 4.6 and 4.7 were both taken from Jeff Finley [37].



**Figure 4.6 Example of significant lens distortion**

The model discussed in equations 4.16 and 4.17 shows the calculated displacement of each pixel as a combination of radial and tangential distortions. Radial distortions make the largest contribution to the total distortion and are the most visible when comparing an undistorted image with the original. This can be confirmed by observing the graphical lens distortion shown in figures 4.8 and 4.9. The lens distorts light towards the center of the image. To correct the distortion that took place in the lens, pixels are moved outward according to the vector calculated by the distortion model to create an undistorted image. The difference between a distorted image and a corrected image can be seen in figure 4.7. Figure 4.7 contains significantly less content since pixels have been moved outside the bounds of the image during the correction process. In photogrammetry, the corrected location for a detected key point is computed and assumed to be part of the pinhole camera model for all future references.

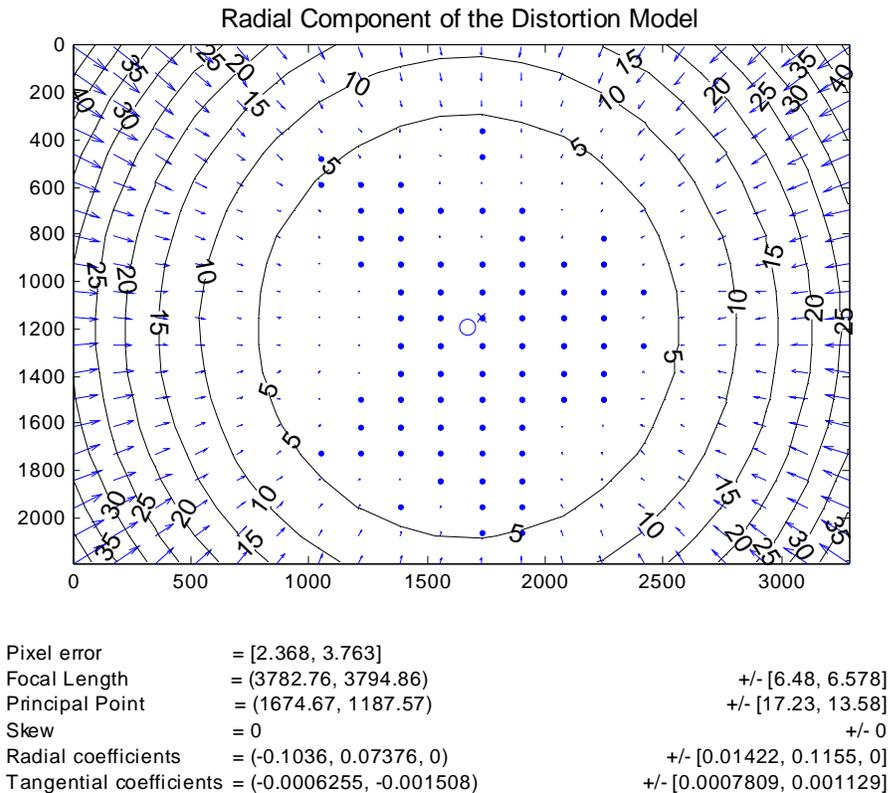


**Figure 4.7 Corrected version of figure 4.4 using a calibration model**

All distortion modeling has been done with a software package described in Heikilla [4] and implemented in The Camera Calibration Toolbox for MATLAB [25]. A map of the distortion model for the Rebel XT and the lens is shown in figures 4.8 and 4.9.

The radial error is shown to dominate the image distortion and is introduced in equation 4.16.  $\rho$  is defined as the distance from the center of the image. The equation and figure both demonstrate how radial distortion is greatly exaggerated near the edge of the image.

$$\begin{bmatrix} \delta u \\ \delta v \end{bmatrix} = \begin{bmatrix} u(k_1\rho^2 + k_2\rho^4 + \mathbf{K}) \\ v(k_1\rho^2 + k_2\rho^4 + \mathbf{K}) \end{bmatrix} \quad 4.16$$

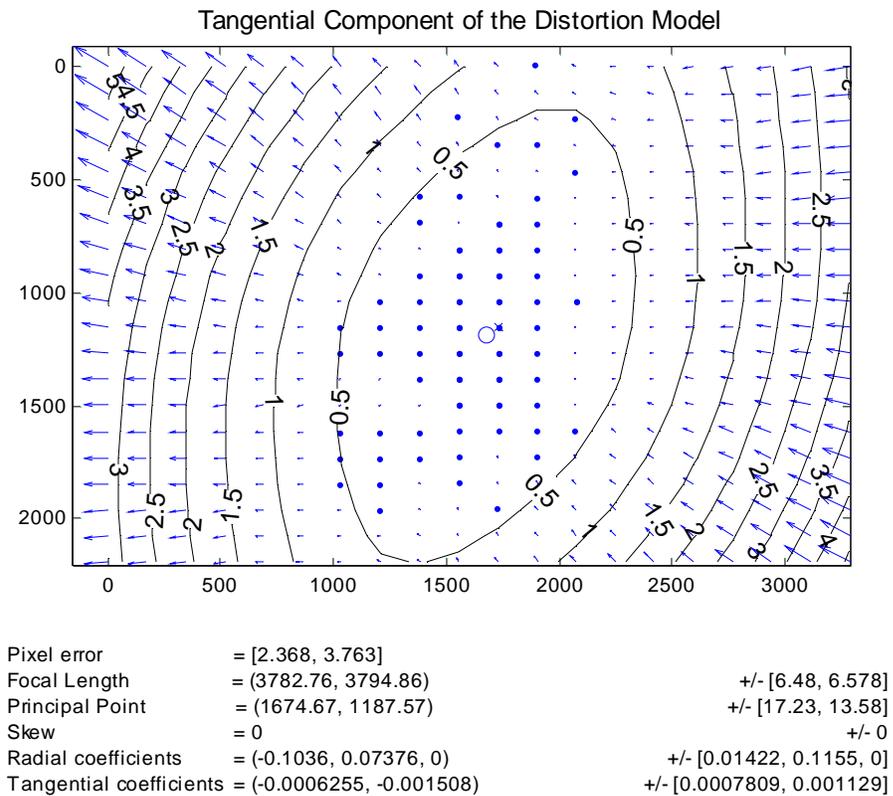


**Figure 4.8 Mapping of radial distortion for Canon Digital Rebel XT camera**

The tangential error is described by the following equation where  $\rho$  is again defined as the distance from the principle point of the image.

$$\begin{bmatrix} \delta u \\ \delta v \end{bmatrix} = \begin{bmatrix} 2p_1uv + p_2(\rho^2 + 2u^4) \\ p_1(\rho^2 + 2v^4) + 2p_2uv \end{bmatrix} \quad 4.17$$

Tangential distortion is a product of misalignment of the optical axis of the lens to create a lens configuration that is not orthogonal to the ccd and does not intersect the center of the ccd. This moves the principle point of the image and provides a shift in one direction to all pixels in the image. Due to advances in the machining mounts and the quality control of the placement of the ccd, tangential distortion is a small and usually negligible factor in the calibration model though the shift in principle point is significant. More detail on the camera calibration derivations can be found in [4][6][7][8].



**Figure 4.9 Mapping of tangential distortion components for Canon Digital Rebel XT camera**

## CHAPTER 5 - Cost Function Definition for Bundle Adjustment and Alternative Bundle Adjustment

The cost function is the most critical part of the design of the triangulation step. It has a significant effect on the overall performance of the algorithm, the computational complexity, and the flexibility for additional variations. It also affects the preprocessing stages and how the images should be taken for optimal coverage. Both BA and ABA fulfill the requirements of triangulation, but each with different advantages and disadvantages. Chapter 5 builds upon chapter 4 by defining a cost function for BA and ABA. A discussion then follows each cost function that outlines the implications on the minimization and solution. The BA algorithm is reviewed but not discussed in detail due to the literature that is widely available in [3][10][26].

### Bundle Adjustment Cost function

The cost function for BA measures the magnitude of the error between a detected feature in image space and the image location of a projected feature estimate from world space. More specifically, the element of the cost vector  $\mathbf{c}$  for one given feature is:

$$c = \left| \mathbf{p}_f^i - \mathbf{p}_f^{i'} \right| \quad 5.1$$

where  $\mathbf{p}_f^i$  is calculated through the transformations described in 4.12-13 from a point in world space  $\mathbf{p}_f^w$  using the current state vector.  $\mathbf{p}_f^{i'}$  is defined as the location of a detected feature in image space. Based upon the relationship of measurements associated with  $\mathbf{c}$ , there is one cost measurement for each detected feature in each image. Despite  $c$  being a scalar, it is calculated by calculating the magnitude of two additional scalars from image space,  $u$  and  $v$  from equations

4.5.  $c$  therefore represents two measurements since both components must be minimized for  $c$  to be minimized. Based upon the number of measurements from detected features and the number of unknown parameters, it is possible to determine if there are enough measurements to solve for the unknowns. For example, if there are  $o$  cameras with  $n$  features each with enough overlap so each feature is seen in  $m$  images, the combined cost vector  $\check{c}$  will be of size  $mn$  by 1. If the scene contains areas with large overlap, the cost vector can grow prohibitively large. Conversely it can be quite small if limited overlap is available. In the former case, the problem is greatly over-constrained and computationally expensive. The latter case may be computationally simple, but risks becoming under-constrained if there are fewer constraints than degrees of freedom. A more complete analysis of these requirements follows.

- Every perspective of a feature contributes 2 equations. (u and v in the image plane)
- Each feature has 3 unknowns for the position in world space.
- Each calibrated camera has 6 unknowns for the pose.

To create a constrained system of equations, the number of equations must be greater than or equal to the number of unknowns. Due to errors in detection and noise in the measurements, the system of equations should be over-constrained with more measurements than unknowns. At minimum the following inequality should be met:

$$2nm \geq 3n + 6o \quad [38] \quad 5.2$$

The key point of the BA cost function is the world space estimation for each feature that is included with the state vector  $\check{K}$  as described in equation 4.3. It is notably different from the state vector defined in ABA that does not contain feature location estimates. More differences will be discussed in detail after ABA is explained.

## Bundle Adjustment Jacobian

Since the cost function  $\chi$  is computed based upon the state vector  $\hat{K}$ , the detected feature locations, and a known camera projection model,  $J$  can be computed analytically. This reduces the computational complexity and increases the accuracy of the resulting  $J$  calculation when compared with a numerical estimate. Due to the large search space,  $J$  can grow to become prohibitively large. If there are  $o$  cameras with 6 unknown parameters and  $n$  features with enough overlap so each feature is seen in  $m$  images, then the resultant  $J$  is of size  $n*m$  by  $6*o+3*n$ . It is important to take special notice of this size since the inversion of such a large Hessian will take significant computation resources.

## Advantages and Disadvantages of Bundle Adjustment

The image space cost function has some great advantages due to the abstraction provided by the projection model.

- Each feature has a single estimated location in world space. The certainty of that estimation can be calculated by the costs associated with that feature. A cost for which there is a relatively large error means the certainty of that location is poor.
- There is no reward or punishment for changing the overall scale of the system.
- $J$  naturally segments itself out into blocks that can be manipulated using sparse matrix techniques.

Despite the many advantages, there is one major drawback. The search space for BA is very large. This presents a computational challenge that can be difficult to solve. To assist with this, a great deal of literature has documented the sparse nature of the BA cost function. Despite the large  $n*m$  by  $6*o+3*n$  size of  $J$ , it can be broken into smaller blocks on which it is significantly easier to perform complex operations. Since the inverse of a matrix is considered

$O(n^3)$  at worst using the Gaussian elimination method, a large search space can make the Hessian ( $6*o+3*n$  square) inversion problematic. Reference [10] discusses many of the improvements and solutions to the computational problem discussed here.

### **Extensions of Bundle Adjustment**

The flexibility of the BA cost function makes it ideal for adding optimizations such as camera calibration. The parameters defined in 4.16-17 can be added to the state vector  $\vec{K}$  and the cost function adjusted appropriately. References in [26] describe BA with camera calibration. Work on this thesis performed calibration before the triangulation step due to the static nature of the camera setup.

### **Examples of BA**

Bundle Block BA (the commercial term referring to sparse matrix enhanced BA) has been implemented in several commercially available products. Though most of these products could be used to solve the triangulation problem, very little modification of the algorithm is available due to the commercial aspects. Other products using BA:

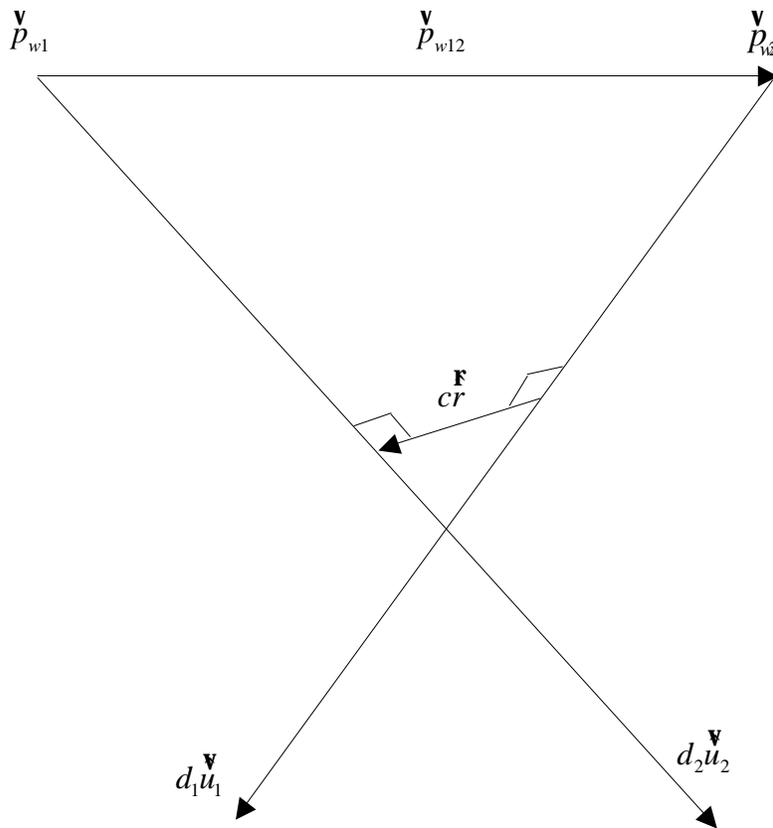
- Sparse Bundle Adjustment by M. Lourakis and A. Argyros
- SOCET SET from BAE Systems
- Leica Photogrammetry Suite for ERDAS Imagine

### **Alternative Bundle Adjustment Cost Function**

Unlike BA, ABA does not explicitly estimate the location of each feature. Instead it defines the minimum distance between a pair of rays emanating from two cameras in the direction of the observed feature. This yields a more intuitive cost function along with a much smaller state

vector  $\vec{k}$  for minimization. This greatly simplifies the computation of the update step, and in many cases  $J$ .

Figure 5.1 below shows the calculation for one element of the cost vector  $\vec{c}$ . Just as in BA, the cost function is an analytic calculation with an analytic  $J$  as well.  $\vec{p}_{w1}$  and  $\vec{p}_{w2}$  are the CoPs for an image.  $\vec{u}_1$  and  $\vec{u}_2$  are unit vectors that pass through the image points for a feature detected in both images.  $\vec{r}$  is a unit vector that is perpendicular to both  $\vec{u}_1$  and  $\vec{u}_2$ .  $\vec{P}_{12}$  is the line between the CoPs.



**Figure 5.1 Calculation of the cost function for ABA in world space for two perspectives of one feature**

A single element of the cost vector is defined by equations 5.3 and 5.4

$$\mathbf{r} = \frac{R_w^{c1T} p_f^{c1} \times R_w^{c2T} p_f^{c2}}{\left| R_w^{c1T} p_f^{c1} \times R_w^{c2T} p_f^{c2} \right|} \quad 5.3$$

$$c = \mathbf{r} \bullet \mathbf{p}_{w12} \quad 5.4$$

Here,  $p_f^{ci}$  is the detected feature vector defined in equation 4.14 and  $R_w^{ci}$  is the rotation matrix in equation 4.9 for camera each of two cameras. The magnitude of the minimum distance between the two rays can be found by projecting the vector between and two points on the rays onto the unit vector defining the mutually perpendicular direction, as in equation 5.4. A feature appearing in  $n$  images will have a total of  $(n-1)*n$  elements in the cost vector associated with it.

### **Advantages and Disadvantages of Alternative Bundle Adjustment**

Moving the cost function into world space changes the algorithm in several ways. The simpler cost function is the first obvious advantage and can lead to a simpler  $J$ . The simpler cost function also reduces the search space by only estimating the parameters associated with the camera poses. ABA adjusts  $J$  to  $n*m*(m-1)$  by  $6*o$  as opposed to  $n*m$  by  $6*o+3*n$  for BA. This reduction in search space reduces the size of  $H$  to  $6*o$  square rather than  $6*o+3*m$  square for BA. This makes for a simpler computation step when  $H^{-1}$  is calculated in the update step of LMA. The requirements for a constrained system of equations remain similar to the requirements for BA outlined in equation 5.2. However the number of required observations is smaller since there are no feature locations to solve for. The updated requirements are shown in equation 5.5.

$$2nm(m-1) \geq 6o \quad 5.5$$

- Each feature has multiple estimated locations in world space.
- Moving the cameras closer together reduces the overall scale and cost

- $J$  is less sparse than BA, but still sparse.

Reducing the search space provides the biggest advantage over traditional BA. It also allows for changes in necessary overlap and camera models that do not conform to the pinhole camera model. The change in cost also introduces several artifacts that will be discussed in chapter 6.

## CHAPTER 6 - Artifacts of ABA

Despite the advantages of ABA's cost function, reducing the search space has some negative effects on the final  $\check{K}$ . The lack of explicit estimates for the feature locations allows the minimization algorithm to move in directions that yield a low cost with multiple estimates for each feature location. This results in a layering artifact that allows for a discontinuous topography surface. In addition to the layering artifact, the overall scale of the camera positions naturally shrinks since that will minimize the overall cost. Both of these artifacts and some remedies are explored in this chapter.

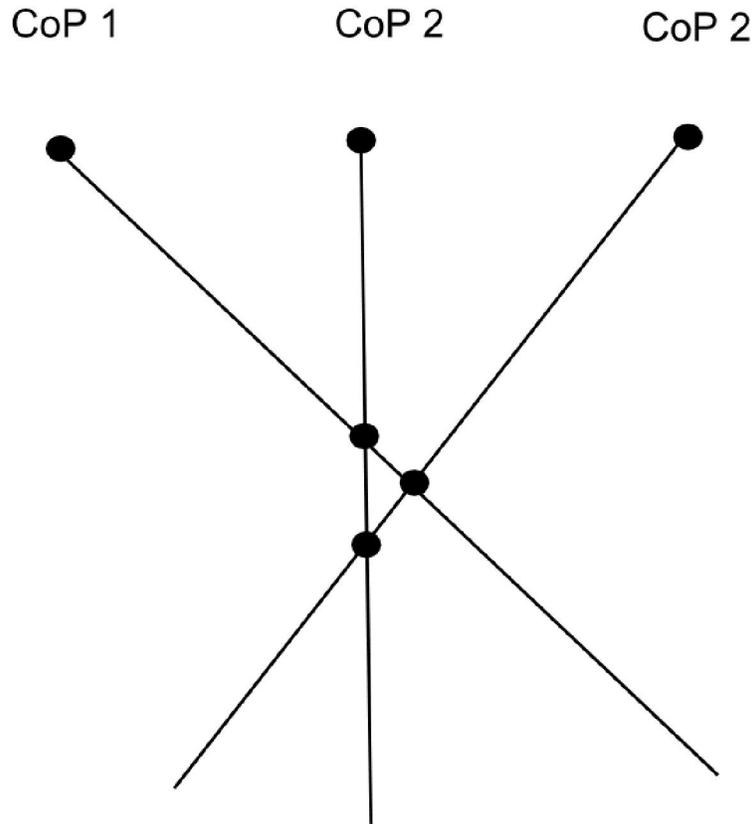
### Reduction of Scale

The LM algorithm works to find a minimum solution by adjusting  $\check{K}$  in any direction that will yield a smaller cost. By reducing the distance between cameras, the entire scene is scaled down. This scaling effect also reduces the cost since it is based upon measurements in world space. This consistently rewards solutions with cameras closer together regardless of how well the extracted topography matches the true terrain. To retain an approximate scale, the differences between the initial guesses at the camera locations and those in  $\check{K}$  are added to the cost vector  $\check{c}$ . This additional cost penalizes the cameras that begin to drift from the initial GPS location, but also allows them to adjust for inaccuracies as is often necessary. A weighting factor can also be added to this cost vector element to encourage adjustments in orientation, the weakest estimate, but not position at the beginning of the search. The weighting can be reduced once the orientations are more certain. It should never be eliminated since the positions in  $\check{K}$  will reduce to a single point and become singular if left un-weighted. It is important to note that despite the

weighting, the set of final positions is always slightly smaller than the initial estimates from GPS. Since ABA rewards solutions with smaller scale, the cameras will move towards a central point until the weighting penalty adds more cost than is saved by moving the cameras. Since a higher weighting will generate a higher cost for the same distance from the original GPS, the amount by which the scale is reduced is inversely proportional with the weighting.

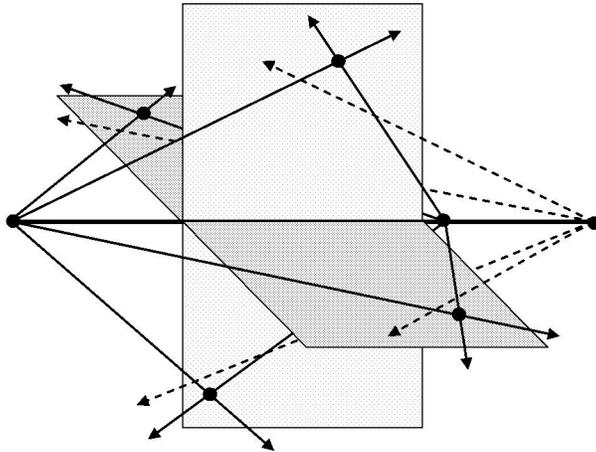
### **Layering**

Since ABA does not keep an explicit estimate of the location of each feature, a cloud of estimated points exists for any one feature. This cloud should be reasonably small, but in some instances, singularities exist that make it large. Adding to this effect, a large spread for one feature is often closely correlated with a large spread in neighboring features. Neighboring features then share a similar range in elevation estimates and appear to form multiple layers. Since each location estimate is created from the intersections of unit vectors from two images, one layer appears for every pair of overlapping images. If an area has many perspectives with a wide range of elevations, a simple averaging technique would create discontinuities on the final surface where a layer ends. Fig 6.1 shows how this effect can be created despite a cost near zero.



**Figure 6.1** An example of a sub-optimal zero cost solution with three estimates in a plane. Much like figure 5.1, each line in figure 6.1 represents a scaled unit vector from a separate CoP intersecting the neighboring vectors. In this instance, a minimal cost solution has been found with multiple location estimates for a single feature. The intuitive solution to fix this artifact would be to add perspectives to increase the number of measurements. This does not reduce the uncertainty in this situation if the new perspectives are collinear with the original set of perspectives. If new perspectives were instead added such that they offer a view from the side of the same feature, the triangle in figure 6.1 would be forced to collapse due to the introduction of a new cost that cannot minimize the cost without minimizing the cloud. A perspective that yields a unit vector perpendicular to the paper in figure 6.1 would provide the optimal placement. This would greatly increase the cost for that feature while allowing an update that would eventually reduce the range of estimates and yield a better solution. Since the placement of CoPs is so

important, flight path planning and overlap is critical. If the overlap is strongly supported in one direction, and weakly supported in another, it will generate situations such as the one shown in figures 6.1-2. This has been tested in simulation, shown in figures 7.2-3, and witnessed with real data from photographs.



**Figure 6.2 A zero cost solution with variable scale**

Figure 6.2 demonstrates the scaling issue applied to the layering effect that occurs with the ABA when all of the true camera positions are collinear or nearly collinear. In figure 6.2, A, B and C are three camera locations and a, b, c and d are feature locations. It is assumed that the scale between A and B is fixed and that the ray intersections from A and B occur at the true feature locations. It is also assumed that the true position for C is along the line of A and B. If the camera pose for C is oriented correctly, then it may slide along the line of the cameras while maintaining zero cost. The rays from A and B for a particular feature form a plane. If C is collinear with A and B, then its ray will also lie in the same plane. The rays from C will therefore intersect the rays from A and B and provide zero cost irrespective of its position along the true line of the cameras. BA does not have this issue since the feature locations are solved for explicitly and scale from one pair of images propagates throughout the entire set block.

## **Conclusions**

Despite the advantages of a greatly reduced state vector, the final minimized state vector and topography can have some limitations if the above artifacts are not considered. There are ways to avoid the pitfalls through proper image layout and minimization techniques. However, both solutions limit the final product and do not address the root cause of fewer defined states in the state vector. The disadvantages of the limited search space could be greater than the advantages of traditional BA when used in conjunction with other information for validity checks. When coupled with BA or another triangulation algorithm that does not suffer from the same artifacts, a final combined algorithm better than either could be possible.

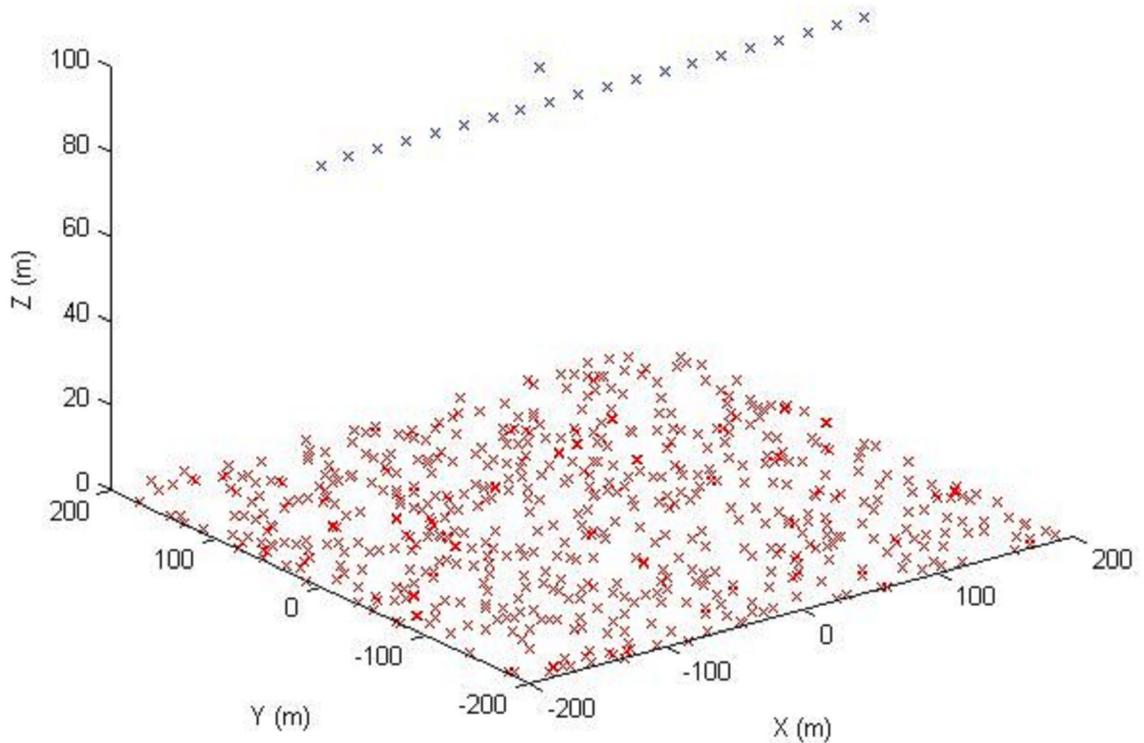
## CHAPTER 7 - Experimental Results

Chapter 7 discusses experimental results that helped discover and demonstrate the artifacts presented in chapter 6. Two experiments were designed to show those artifacts and to prove the practicality with a final mosaic. The first experiment shows a simulated environment in which cameras and features can be arbitrarily placed and solved for using either algorithm.  $\hat{K}$  was found using ABA and the intersections points were calculated as shown previously. The intersection points calculated from  $\hat{K}$  shown in figure 7.2 show the areas that had multiple overlapping perspectives that were collinear and those that were not. By using both collinear and non-collinear CoPs in the same minimization, it is clear that a low cost state vector can yield good or poor results depending on the camera placements. The second experiment shows the results from a set of images taken from 200m above the prairie and the comparisons to LIDAR scans of the same area. This experiment shows the practicality of ABA to produce the desired mosaics with experimental images as shown in figure 1.2.

### Layering Experiment

A series of collinear CoPs will create a singularity as discussed in figure 6.1 and 6.2. To demonstrate this point, and allow for experimentation, a simulator was created that would allow for arbitrary placement of CoPs and features. Images were taken from the perspective of each CoP and stored along with a state vector and set of feature locations. Error was then added to the state vector and ABA was given the opportunity to correct the error using the generated images. Since the true feature locations and state vector are known, a quantifiable comparison between ABA and the truth is known. Figure 7.1 shows a scene with 20 collinear perspectives and one

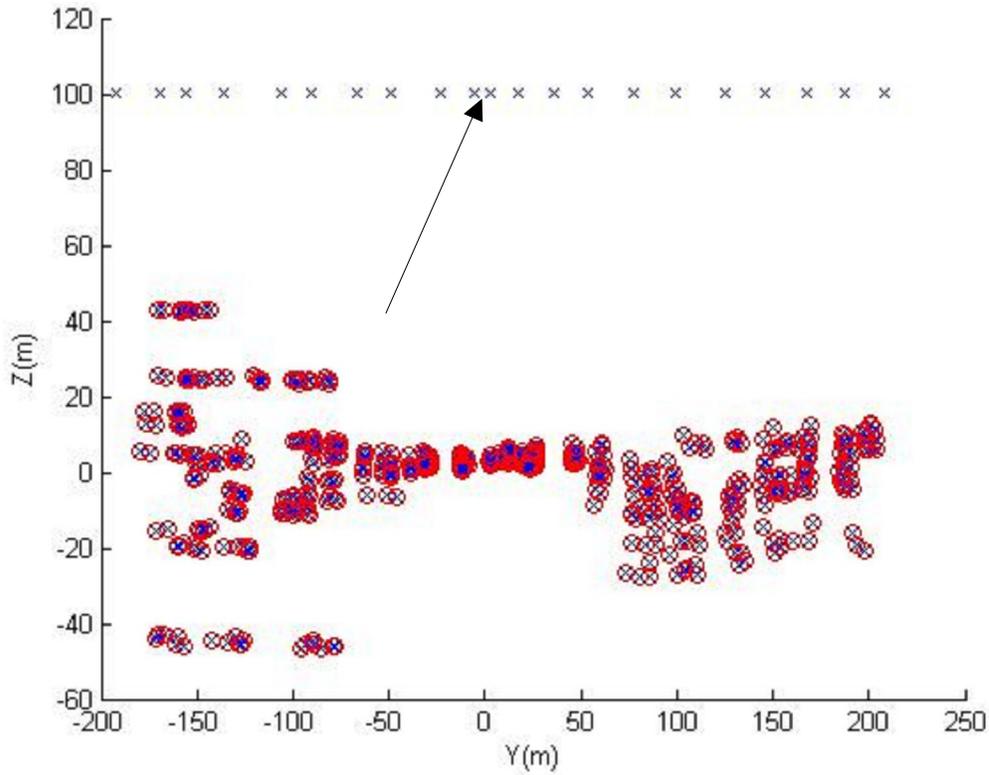
additional offset perspective. Areas that fall within the collinear perspective will separate into a set of layers as discussed in chapter 6. The area within view of the offset perspective will settle into a single surface. The x's in the xy plane in figure 7.1 represent features visible to the cameras denoted by x's above the xy plane.



**Figure 7.1 Initial scene with features and CoPs visible**

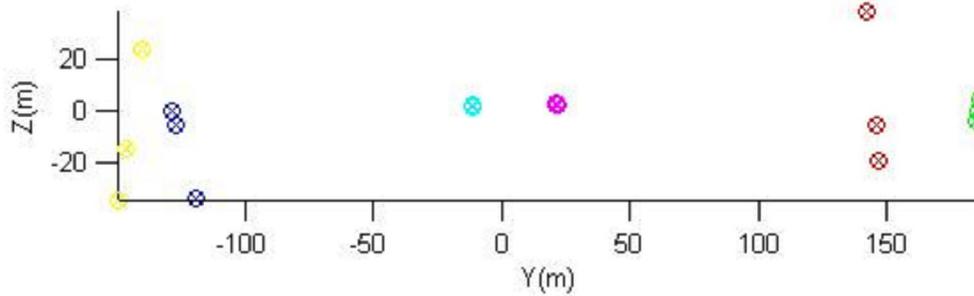
Figure 7.1 from the simulation provides a scenario in which the layering artifact can be directly observed and tested. For this experiment, a randomly generated error is added to the initial pose estimates to simulate uncertainty from the autopilot sensors. (a range of  $\pm 10m$  for position and  $\pm 11.5^\circ$  for orientation) The minimization algorithm is then allowed to achieve a low cost by adjusting the state vector. However, the intersection points begin to degrade long before a minimum is discovered. As ABA continues to find a lower cost solution, the layers begin to

separate more. This effect is significantly less pronounced in the area observed by an additional non-collinear perspective. This area can be seen in the middle of the extracted intersection points shown in figure 7.2.



**Figure 7.2 Final solution with features and CoPs visible**

Figure 7.2 shows the calculated intersection points from the state vector found by ABA from the CoPs with error created by the simulation shown in figure 7.1. The sections on the left and right have separated into distinct layers due to collinear CoPs without a side perspective. The section in the middle has a significantly smaller range for each feature due to the non-collinear CoP indicated with the arrow. To show this effect more clearly, the intersection points of figure 7.2 have been redrawn in figure 7.3 with fewer features.



**Figure 7.3 6 selected features from figure 7.2 showing the range of values associated with each feature**

Figure 7.3 shows 6 of the same features shown in figure 7.2. Each feature has its own color to show the range of position estimates. The table below shows the range for each dimension for each feature.

	Range in X (m)	Range in Y (m)	Range in Z (m)
Yellow	5.37	2.39	35.80
Blue	5.23	5.53	18.80
Cyan	.24	.18	.64
Magenta	.17	.21	.62
Red	1.88	6.9	30.2
Green	.79	.90	3.8

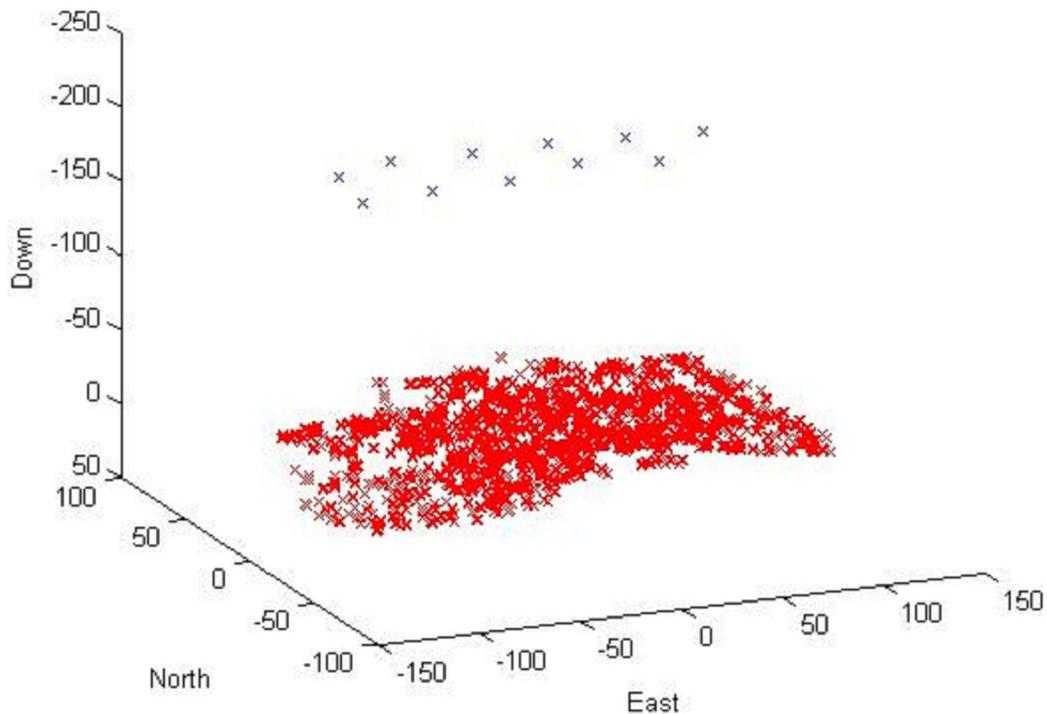
**Table 7.1 6 selected features from figure 7.2 showing the range of values associated with each feature**

The cyan and magenta features are within the perspective of the offset camera and have a small range for the position estimate. This differs significantly when compared with the other 4 features. It is important to note that all features have a significantly larger range in the Z direction than the X and Y due to the arrangement of the cameras and the relatively small field of

view. It is also important to notice that each feature has a vertical range of position estimates. A non-collinear CoP does not eliminate the position uncertainty, but it does reduce it.

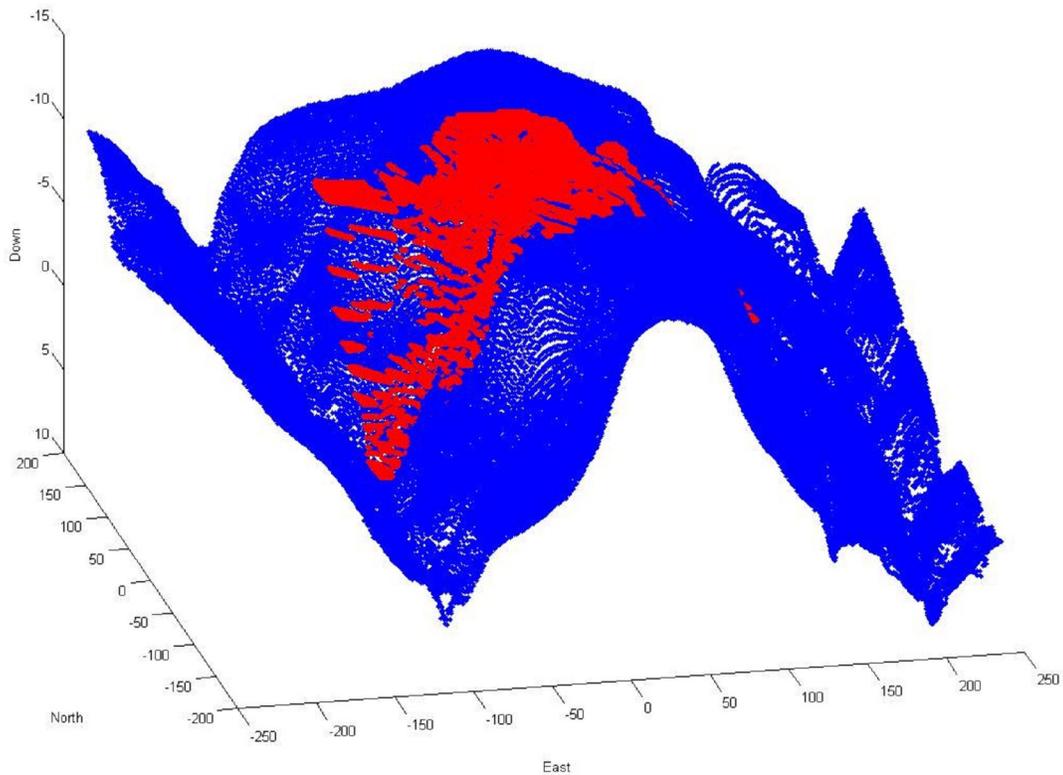
### **ABA with Aerial Photography**

The second experiment shows the extracted topography from a set of overlapping images and compares the results to a LIDAR scan of the same area. The extracted topography can then be combined with the aerial imagery to create a composite mosaic with geo-referenced content. The example shown in figures 7.4-8 has been generated with two parallel overlapping strips of images, one with of 5 images and the other with 6. The final arrangement of cameras is shown with the CoPs represented as blue x's above a surface of red x's representing the extracted topography.



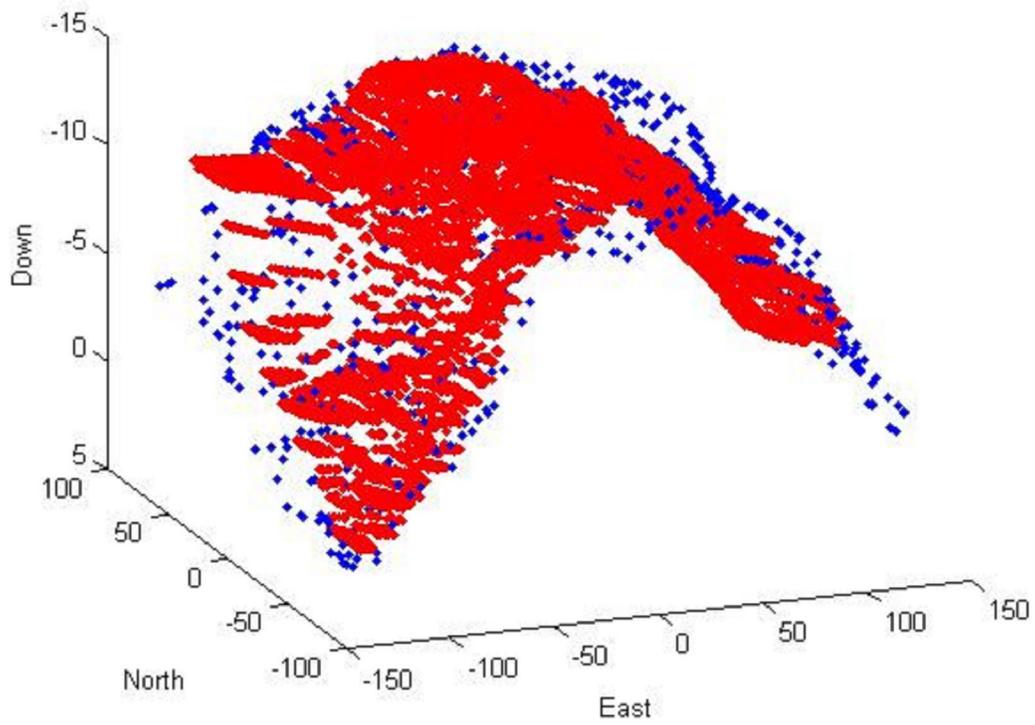
**Figure 7.4** Extracted topography from 11 overlapping images

The calculated intersection points can be plotted along with the LIDAR scan of the same area for an accuracy comparison. The poses naturally wander in scale, position, and orientation as the minimization algorithm searches for a low cost solution. This yields a final surface that is rotated, scaled, and translated away from the LIDAR scan. To make a valid comparison, the DTM must be operated on by scale, rotation, and translation operations to align it with the LIDAR data. To provide reference for the scaling/rotation/translation operation, locations on the DTM were identified in the field and assigned geo-references with a GPS unit. The selected DTM points were then transformed for a best fit to the LIDAR scan with a least squares solution using the previously mentioned geo-references. The final model was then modified by manual adjustment. The final operation was a combination of rotation, translation and scaling. This allows for the correction of the wandering and adjustment of scale.

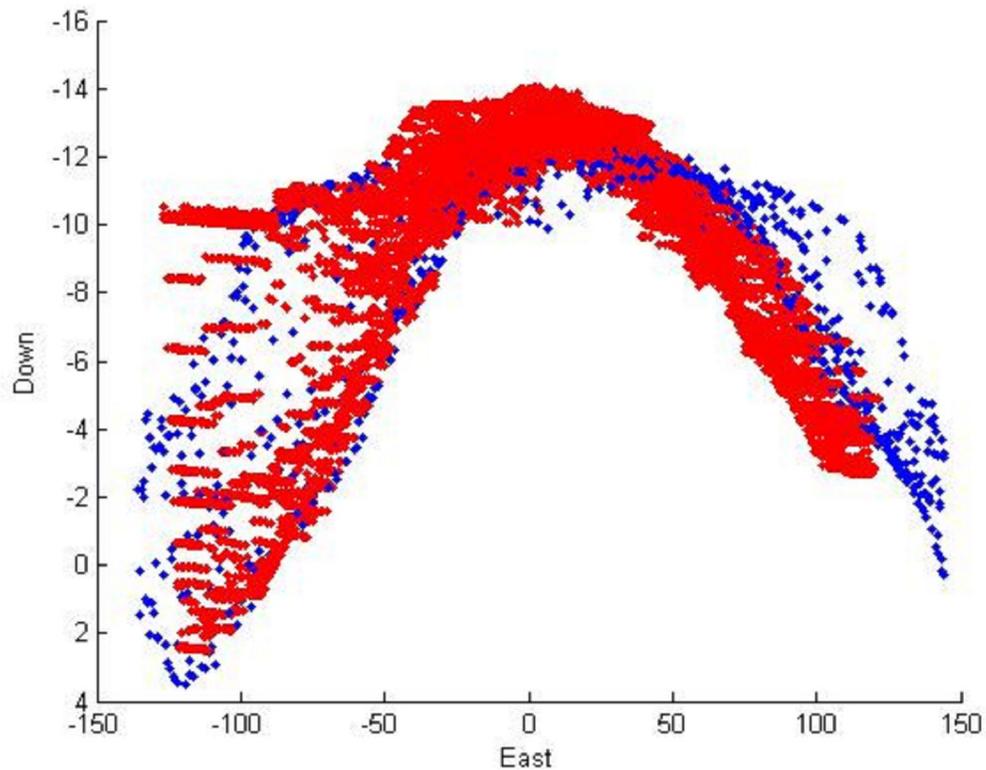


**Figure 7.5 The extracted topography overlaid on a LIDAR scan**

Figure 7.5 gives a broad view of how well the data fits the LIDAR scan for the same area. By sub-sampling just the LIDAR data surrounding the DTM, a more complete perspective of the data can be seen. Figure 7.6 shows a side perspective in which some of the data match can be visualized. A plot of the differences between the LIDAR scan and the DTM shows the accuracy of the middle section in figure 7.8.

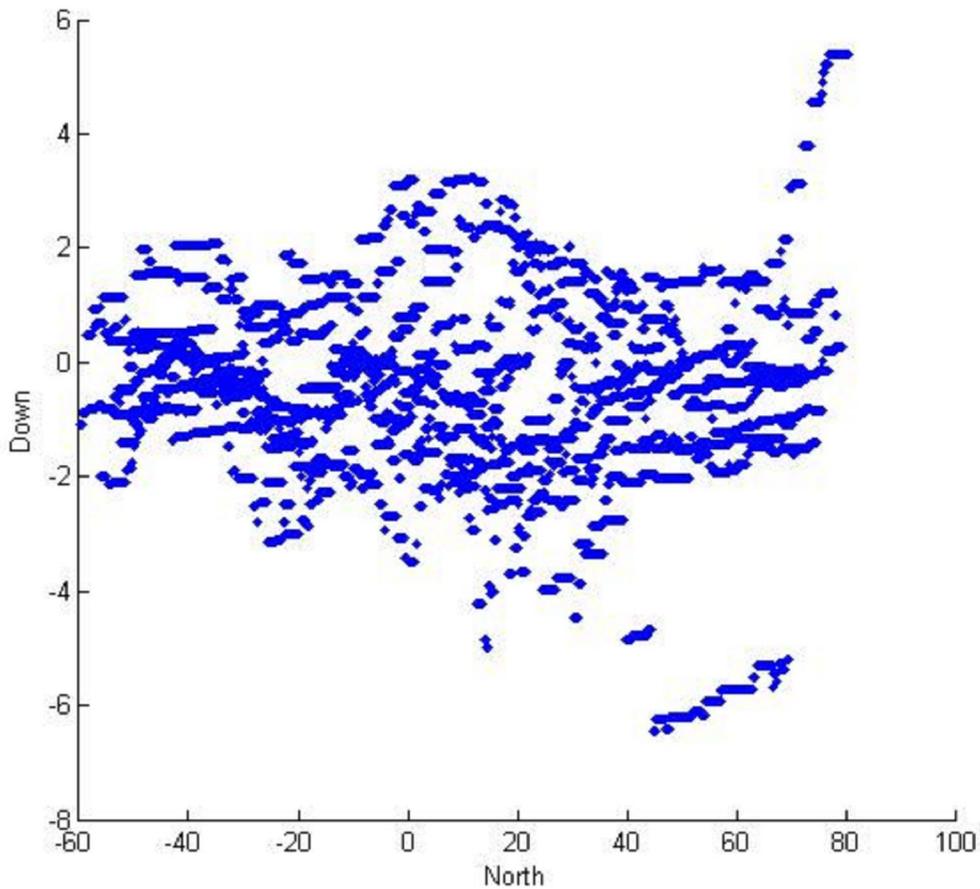


**Figure 7.6** The extracted topography overlaid on a subset of the LIDAR scan



**Figure 7.7 Side perspective of figure 7.6 to show a cross-section view**

If the calculated DTM is subtracted from the LIDAR surface, a surface can be calculated to show the difference between the two models. The center section has high-frequency content that stays within a 3m meter band and is a minor instance of the layering effect being averaged. The edges begin to deteriorate due to fewer tie-points from fewer overlapping images. If given more photos, the areas of great overlap would be significantly larger and less prone to distortion near the edge.



**Figure 7.8 The difference between the topography and the LIDAR scan. It is important to note the error range is mostly within 3m for data near the center of the DTM.**

### **Mosaic Creation**

The geo-referenced mosaicing process can be completed once the DTM has been extracted and geo-referenced. The resulting map corresponds the content from the images to geo-spatial references with the referenced DTM. This allows all images to be combined and eliminates the redundant information from multiple overlapping images. The mosaic shown in figure 1.2 has been developed by projecting each image onto the DTM discussed above. For any location on the DTM, the camera with the most direct perspective is chosen to assign the pixel value. This creates a composite image with visible seams, but with content that is more likely to be correct

than if it were developed using pixel values from the edge of the images. There is no blending or feathering between images though it might improve the aesthetic qualities. Averaging and blending alters the data in unknown ways and is to be avoided. Radiometric corrections could be made if the angle of the surface was known for each location on the DTM. This data has not been calculated and would be an ideal extension for future work.

## **CHAPTER 8 - Future Research and Conclusions**

ABA has been shown to work well in refining the estimates of both feature locations and camera poses. However the final estimates are not close enough to create DTM's and mosaics for applications requiring high precision. Despite the lack of accuracy the improvements are enough to make the problem tractable for BA to finish. This can be useful for finding solutions when previously BA would not be able to. In addition, the estimates are improved enough where the key point detection problem can be recalculated with a better estimation of where a detected key point might be able to find related matches. This can greatly reduce the search space for each possible key point while increasing the probability that the match is correct. When better correlation matches are coupled with better initial estimates, a good final product is more likely to be achieved.

- ABA is computationally more efficient than BA in some situations.
- ABA provides an improved set of estimates for BA to operate on.
- ABA has problems with scale and feature location estimation.

### **Future Research**

The work presented in this thesis could be expanded in several ways to allow for better state vector estimation and mosaic creation. A better key point detection algorithm would assist the algorithm with additional measurements between images. Areas that had sparse key point coverage such as the corners of the DTM would have much more data to extract the topography from. If an improved key point correlation was performed after the initial ABA step but before

the final BA process, a denser and more accurate set of correlations could be used. In addition, more work could be done on the visualization and output processes. BA could also be applied towards camera models without closed form projection models. This would eliminate the world space to image space transformation that is so important in the BA algorithm. Work is continuing in this area and appears to be the most logical application of ABA.

## REFERENCES

- [1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110
- [2] D. Marquardt "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." *SIAM J. Appl. Math.* 11, 431-441, 1963.
- [3] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment - a modern synthesis," Springer-Verlag, 2000.
- [4] J. Heikkilä and O. Silvén. "A Four-step Camera Calibration Procedure with Implicit Image Correction," University of Oulu, Finland.
- [5] K. Madsen, H. Nielsen, and O. Tingleff, "Methods for Non-Linear Least Squares Problems," Technical University of Denmark, April 2004.
- [6] J. Heikkilä. "Geometric Camera Calibration Using Circular Control Points," University of Oulu, Finland.
- [7] R Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE: Journal of Robotics and Automation*, Vol. RA-3, NO. 4, Aug. 1987
- [8] M. Heikkila, M. Pietikainen, "An Image Mosaicing Module for Wide-Area Surveillance," Machine Vision Group, Infotech Oulu and Department of Electrical and Information Engineering, University of Oulu, Finland
- [9] S. Hsu, H. Sawhney, R. Kumar, "Automated Mosaics via Topology Inference," *IEEE Computer Graphics and Applications*, March/April 2002
- [10] M. Lourakis and A. Argyros, "The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm," CVRL ICS, Heraklio, Crete, Tech. Report FORTH-ICS/TR-340, Aug. 2004.
- [11] E. Kang, I. Cohen and G. Medioni, "Fast and Robust 2D Parametric Image Registration," Institute for Robotics and Intelligent Systems School of Engineering, University of Southern California
- [12] S. Mann and R. Picard, "Virtual Bellows: Constructing High Quality Stills From Video," *IEEE 1994, MIT Media Laboratory*
- [13] S. Peleg and J. Herman, "Panoramic Mosaics by Manifold Projection," David Sarnoff Research Center and The Hebrew University
- [14] H. Sawhney R. Kumar, "True Multi-Image Alignment and its Application to Mosaicing and Lens Distortion Correction," Sarnoff Corporation, CN5300, Princeton, NJ 08530
- [15] N. Chiba and H. Kano, "Feature-Based Image Mosaicing."
- [16] W. Kilford "Elementary Air Survey," Pitman Publishing, Second Edition 1970
- [17] L. Brown, "A Survey of image registration Techniques," *ACM Computing Surveys*, Vol. 34, No. 4, December 1992
- [18] S. Roweis, "Levenberg-Marquardt Optimization,"
- [19] D. Lowe, "Local feature View Clustering for 3D Object Recognition," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, December 2001
- [20] D. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. of the Internation Conference on Comptuer Vision*, Corfu September 1999
- [21] H. Sawhney, S Hsu, and R. Kumar, "Robust Video Mosaicing through Topology Inference and Local to Global Alignment," *Proc. of the European Conf. on Computer Vision* 1998
- [22] N. Gracias, "Mosaic-based Visual navigation for Autonomous Underwater Vehicles," Dissertation from Universidade Tecnica de Lisboa Instituto Superior Tecnico, Lisbon Portugal, December 2002
- [23] B Ready, C. Taylor, and R. Beard, "A Kalman-Filtered Based Method for Creation of Super-resolved Mosaicks," *Electrical and Computer Engineering*, Brigham Young University.
- [24] R. Cannata, M. Shah, S. Blask, and J. Van Workum, "Autonomous Video Registration Using Sensor Model Parameter Adjustments," School of EECs, University of Central Florida Orlando and Harris Corporation
- [25] Jean-Yves Bouguet, "Camera Calibration Toolbox For Matlab," online 11.1.2007 [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)
- [26] T. Luhmann, S. Robson, S. Kyle, and I. Harley, "Close Range Photogrammetry," Whittles Publishing, 2006
- [27] S. Peleg, B. Rousso, A Rav-Acha, and A. Zomet, "Mosaicing on Adaptive Manifolds," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol 22 No. 10 October 2000
- [28] B. Rousso, S. Peleg, and D. Finci, "Mosaicing with Generalized Strips," School of Computer Science, The Hebrew University of Jerusalem, Israel
- [29] A. Zomet, S. Peleg, and C. Arora, "Rectified Mosaicing: Mosaics without the Curl," School of Computer Science, The Hebrew University of Jerusalem, Israel
- [30] A. Noonan, "Flight Plan Generation for Unmanned Aerial Vehicles," a thesis, Kansas State University, September 2007
- [31] H Sawhney, R. Kumar, G. Gendel, J. Bergen, D. Dixon, V. Paragano, "Videobrush: Experiences with Consumer Video Mosaicing," David Sarnoff Research Center, Princeton, NJ
- [32] S. Peleg, J. Herman, "Panoramic Mosaics with VideoBrush," David Sarnoff Research Center, Princeton, NJ
- [33] M. Kasser and Y. Egels, "Digital Photogrammetry," Taylor and Francis, 2002
- [34] "Manual of Photogrammetry," Fourth Edition, 1980 American Society of Photogrammetry
- [35] B. Hallert, "Photogrammetry," McGraw-Hill Book Company, 1960
- [36] D. Sullivan and A. Brown, "High Accuracy Autonomous Image Georeferencing Using a GPS/Inertial-Aided Digital Imaging System", *Proceedings of ION NTM 2002*, San Diego, CA, January 2002
- [37] Jeff Finley
- [38] B. Henderson, "A Novel Method for Extrinsic Self Calibration of Wide-Baseline Three-Dimensional Computer Vision Systems," a thesis, University New Mexico, July 2006
- [39] P. Dias, V Sequeira, F Vaz, J Goncalves, "Registration and Fusion of Inensity and range Data for 3D Modelling of Real World Scenes," *Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modleing*, 2003
- [40] B. Mellish "How a pin hole camera works" May 2008. Online. Under GNU License <http://en.wikipedia.org/wiki/Image:Pinhole-camera.png>

## **Appendix A - GNU Free Documentation License**

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### **0. PREAMBLE**

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

### **1. APPLICABILITY AND DEFINITIONS**

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque

formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

**A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

**B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

**C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.

**D.** Preserve all the copyright notices of the Document.

**E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

**F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

**G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

**H.** Include an unaltered copy of this License.

**I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

**J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

**K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

**L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

**M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

**N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

**O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in

the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic

equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the

Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.