Video Mosaicking Using Ancillary Data to Facilitate Size Estimation

Eric R. Kee

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
In partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

A. Lynn Abbott, Chair
J. Mac Baker
Walling R. Cyre

January 17th, 2003
Blacksburg, VA

KEYWORDS: Computer vision, image processing, mosaicking, super resolution, object measurement.

Video Mosaicking Using Ancillary Data to Facilitate Size Estimation

Eric R. Kee

ABSTRACT

This thesis describes a mosaicking system designed to generate image mosaics that facilitate size estimation of 3-dimensional objects by improving data obtained with a multi-sensor video camera. The multi-sensor camera is equipped with a pulse laser-rangefinder and internally mounted inclinometers that measure instrument orientation about three axes. Using orientation data and video data, mosaics are constructed to reduce orientation data errors by augmenting orientation data with image information. Mosaicking is modeled as a 7-step refinement process: 1) an initial mosaic is constructed using orientation information obtained from the camera's inclinometers; 2) mosaics are refined by using coarse-to-fine processing to minimize an energy metric and, consequently, align overlapping video frames; 3) pair-wise mosaicking errors are detected, and removed, using a energy-based confidence metric; 4) mosaic accuracy is refined via color analysis; 5) mosaic accuracy is refined by estimating an affine transformation to align overlapping frames; 6) affine transformation approximations between overlapping video frames are used to reduce image noise through super-resolution; 7) original orientation data are corrected given the refined orientations of images within the mosaic. The mosaicking system has been tested using objects of known size and orientation accuracy has been improved by 86% for these cases.

ACKNOWLEDGEMENT

I would like to express my sincere thanks to my advisor, Dr. A. Lynn Abbott for his guidance, patience, and motivation in this thesis. I would also like to thank my committee members, Dr. Baker and Dr. Cyre for their time and patience serving on my committee and reviewing this thesis. Additionally, special thanks goes to the folks at The Brooks Forest Products Center: Phil Araman and Neil Clark. Thank you, Phil, for the opportunity to work on this project. Thank you, Neil, for countless brainstorming sessions, data collection trips, and general banter over the past year.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

TABLE OF SYMBOLS

Symbol		Definition	Page
F_v	–	A *video frame set*; a set of video frames taken of an image scene.	12
F_o	–	A *RO frame set*; a set of RO frames describing the orientation of video frames in a video frame set, F_v.	12
Φ	–	A *RO frame*; a set of range-orientation parameters α, β, ρ, and δ.	12
α	–	Inclination, measured from horizontal (degrees).	12
β	–	Azimuth, measured relative to magnetic north (degrees).	12
ρ	–	Camera roll (degrees).	12
δ	–	Range to laser-targeted object (feet).	12
Φ'	–	A *specialized RO frame*; a set of range-orientation parameters Δa, Δb, and $\Delta \Pi$ in units of pixels.	16
Δa	–	Vertical translation for a mosaic pair in a video frame set, in pixels.	16
Δb	–	Horizontal translation for a mosaic pair in a video frame set, in pixels.	16
$\Delta \Pi$	–	An affine warping matrix that specifies the affine motion field for a mosaic pair.	16
f_i	–	A video frame from a video frame set, F_v ($f_i \in F_v$).	19
f_l	–	The focal length of the multi-sensor camera.	19
η	–	Camera constant conversion ratio in pixels/mm; page	19
$E(i, j)$	–	The energy space function for a mosaic pair.	21
C_r	–	The confusion region between two video frames in a mosaic pair.	21
O_r	–	The overlap region between two video frames a mosaic pair.	22
μ_f	–	The mean intensity value in a video frame, f.	22
P_n	–	An image at level n of a Gaussian image pyramid.	24
PC_r	–	A confusion region size in a Gaussian image pyramid (a constant).	24
σ_{xy}	–	The covariance between two variables, x, y.	26
$C(x,y)$	–	The ternary m-point classification function.	27
$\mathbf{D}$	–	A deformation matrix, used in conjunction with $\mathbf{d}$ to specify an affine motion field.	31
$\mathbf{d}$	–	A displacement vector used in conjunction with $\mathbf{D}$ to specify an affine motion field.	31
$\mathbf{A}$	–	The 2x2 identity matrix added to $\mathbf{D}$ to make the affine warping function elegant.	32
$I(\boldsymbol{x})$	–	Image functions.	32
$J(\boldsymbol{x})$			
e	–	The 2-dimensional sum of squared differences error between two image functions.	32
$\mathbf{g}$	–	A 2-D gradient vector.	32
$F_i(x,y)$	–	A 2-D continuous image function.	37
$h_i(x,y)$	–	A 2-D filter function.	37
$I_s(u,v)$	–	A 2-D discrete image function generated from the continuous image $F_i(x,y)$.	37
S	–	A set of images that represent the same *image scene*.	38
$Q(x)$	–	A 1-D continuous image function specified between discrete pixel values.	39
E_i	–	A high-resolution pixel value between two pixels in a discrete image.	39

V_i	–	A pixel value in a discrete low-resolution image.	39
A	–	A free parameter used to compute E_i.	39
κ	–	The *angular bias*; used to compute the orientation of a video frame relative to horizontal.	41
λ	–	The optical-axis-relative calibration angle, see page 42 for geometric definition.	41
φ	–	The horizontal-relative calibration angle, see page 42 for geometric definition.	41
γ_i	–	The distance from the image center to the calibration point location in an image frame; see page 42 for geometric definition.	41

CHAPTER 1. Introduction

1.1 Introduction to Mosaicking

Traditional camera systems have inherent limitations on their field of view (FOV). Imaging hardware configurations, primarily focal length and lens arrangement, limit FOV to angular regions far less than a complete sphere. Typical FOV constraints vary widely depending on lens configuration but may be 22° (horizontal) by 22° (vertical). Unfortunately, many imaging applications necessitate FOV areas far beyond a few degrees.

Many techniques, dating back to the 1800s, exist to reduce camera FOV limitations. The first attempts to extend camera FOV were segmented panoramas—sequences of photographs placed in a semi-overlapping fashion to produce a panoramic effect. Although artistically effective, segmented panoramas of photographs provide only crude representations of the original scene. In 1845, Joseph Puchberger patented a swing lens camera that could panoramic images with a 150° horizontal FOV. A swing lens camera creates panoramic images by rotating the camera lens about its focal point so that light is mapped onto a section of film mounted on a curved plate. Since Puchberger's invention, many panoramic cameras have been created that have achieved a full 360° horizontal FOV [1]. Unfortunately, panoramic cameras do not fully resolve FOV limitations and require strictly controlled imaging conditions, even in their modern digital formats.

Another approach to extending FOV is to use special lens and mirror schemes to. Fisheye lenses

provide one possible solution to FOV limitations. Figure 1.1 illustrates an image taken with a fisheye lens [2]. Special mirrors, such as spherical and hyperbolic mirrors, also enable greater FOVs. Figures 1.2 and 1.3 illustrate images captured using specialized mirrors [3]. Special lenses and mirrors offer elegant solutions to the problems inherent in the first panoramic cameras. While the first panoramic cameras required special film and strictly controlled camera motion, lenses and mirrors allow cameras with normal FOVs to capture images of greater vertical FOV without imaging errors induced by camera motion. Lenses and mirrors, however, necessitate reduced imaging resolution by mapping large FOVs, onto the image plane of a traditional, limited-FOV, camera. The effects of reduced resolution are evident in the bottom portions of Figure 1.2b and in the top portion of Figure 1.3b.

Figure 1.1: An image taken with a fisheye lens, courtesy of [2].

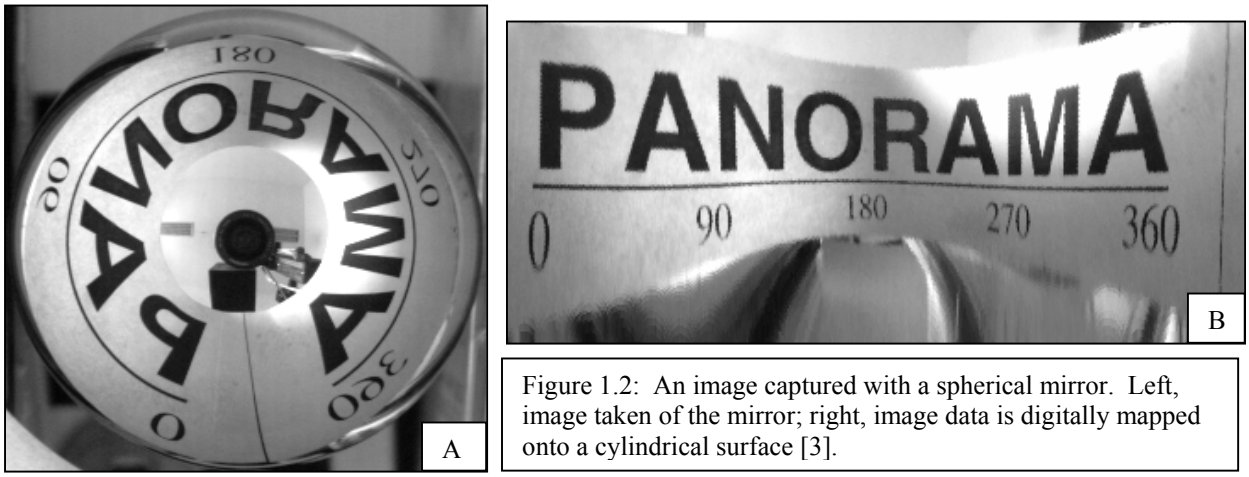

Figure 1.2: An image captured with a spherical mirror. Left, image taken of the mirror; right, image data is digitally mapped onto a cylindrical surface [3].

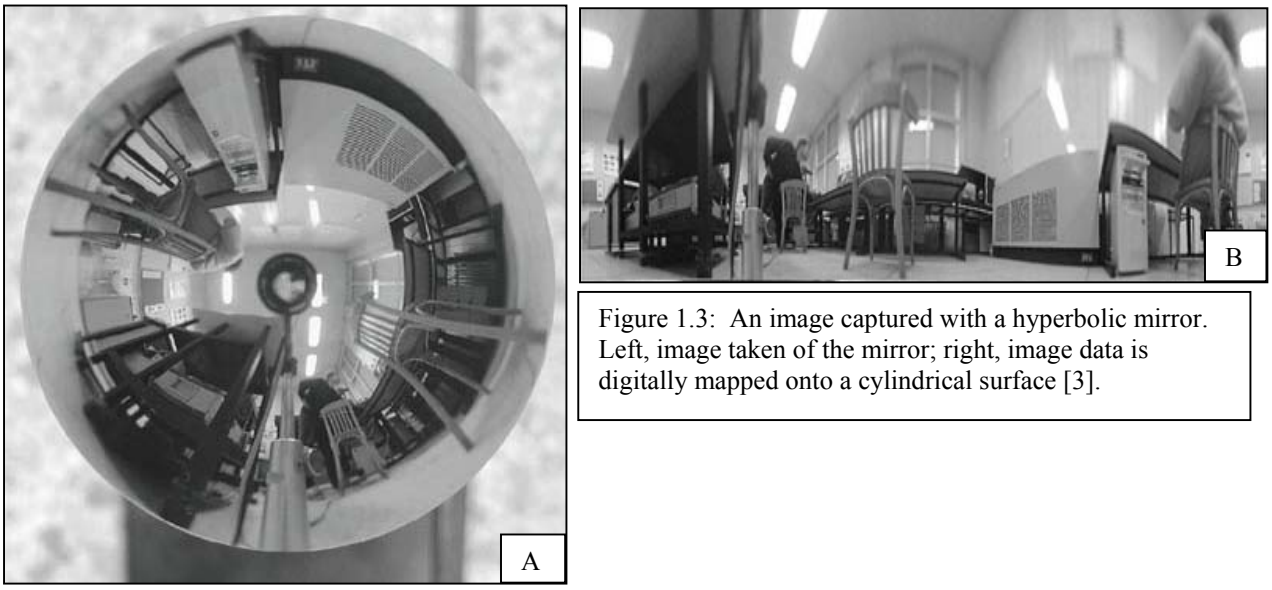

Figure 1.3: An image captured with a hyperbolic mirror. Left, image taken of the mirror; right, image data is digitally mapped onto a cylindrical surface [3].

Efforts to increase camera FOV have, recently, come full circle. The segmented images of the past have acquired a new name: mosaic images. Using modern digital imaging technology, automated techniques have been developed to generate mosaic images in a process now known as *mosaicking*. Mosaicking is the process of stitching multiple images together to form a single meta-image that represents the information content of its constituent images. The mosaicking of today, however, extends far beyond the capabilities of segmented images. Modern digital

mosaicking systems are able to seamlessly stitch mosaics from images recorded under a wide range of conditions.

Random camera pans, zooms, rotations, and object parallax are among the problems that obfuscate mosaicking. Figure 1.4 illustrates some mosaicking difficulties. The component images of Figure 1.4 were taken manually by visually aligning images through a camera viewfinder for a 180° horizontal FOV and placing overlapping images together. The red ellipses in Figure 1.4 mark imaging artifacts. Artifact 1 is created by changes in light sensitivity levels of the imaging device. Artifact 2 is caused by perspective effects at the edge of images—this creates a discontinuous horizon in Figure 1.4. Artifact 3 is caused by uncontrolled camera rotations—the light-colored road is incorrectly cutoff by an overlapping frame. The overlapping frame cannot be aligned to fit within the mosaic on both its left and right sides.

Figure 1.4: A panoramic mosaic image assembled as a simple segmented panorama. Red ellipses are drawn around imaging artifacts.

Perspective distortions—see ellipse 2 in Figure 1.40—become more significant as scene objects become closer to the camera. Figure 1.5 demonstrates an 80° horizontal pan of a room. Although the pan angle in Figure 1.5 is far less than the 180° pan found in Figure 1.4, perspective distortions are much greater because imaged objects are closer.

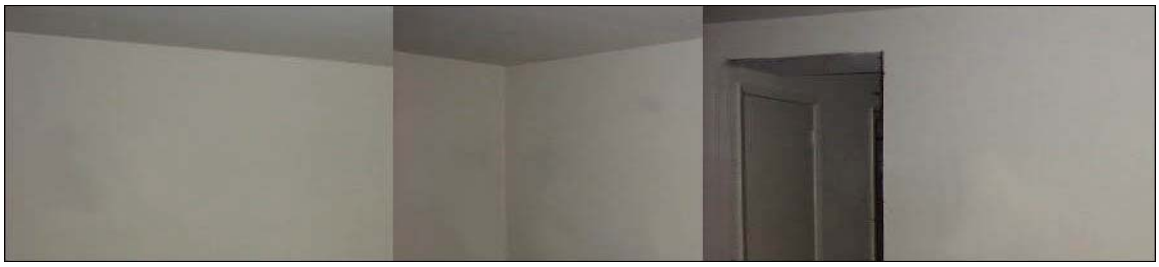

Figure 1.5: A segmented image of a room. The camera was panned approximately 80°. Perspective distortions are apparent along the ceiling.

Mosaicking difficulties resulting from perspective distortions, as demonstrated in Figures 1.4 and 1.5 above, limited initial mosaicking applications. Aerial and satellite photography were first to benefit from image mosaicking because the large distances involved impose few parallax effects and perspective distortions in image sequences. Satellite and aerial image mosaics enjoy the luxury of image data that is almost purely translational. For example, [4] present a mosaicking system that operates on aerial imagery that contains only translation and rotation. Unfortunately, most mosaicking situations involve data with significant angular displacements, camera translations, and diverse object depths.

Coordinate projective methods that assume cylindrical or spherical mosaic models can resolve perspective distortions present in Figures 1.4 and 1.5. Krishnan and Ahuja [5] present a mosaicking system capable of mosaicking regions similar to Figure 1.5 by rotating a camera about a point f_l units in front of the camera, where f_l is the camera's focal length (f_l is held constant while collecting mosaic data). Rotation about a point f_l units in front of the image plane allows low image redundancy data collection of cylindrical regions. McMillan and Bishop [6] use constant-focal-length settings to obtain mosaic data by rotating a camera mounted on a tripod. Data collected from the rotated camera is then mapped onto a cylindrical region.

In [6], McMillan and Bishop handle errors due to camera rotations about points other than the focal point by constraining imaged scenes to distant objects (where "distant" is defined relative to the focal length of the camera). Gumustekin and Hall [7] propose mosaicking by image projection onto a gaussian sphere centered at the camera's focal point. Once mapped onto a sphere, mosaics are generated by flattening portions of the sphere onto a tangent plane.

Unfortunately, techniques that assume cylindrical or spherical regions impose constraints on camera motion. Rotations about the focal point—approximate or exact—inherently suggest projection onto cylindrical regions. Most imaging schemes, however, do not ensure pure rotation about the optical center. Much research has been done to create mosaicking systems that can handle arbitrary motions [8, 9, 10, 11, 12, 13]. In [12] semi-spherical mosaic images are presented that suggest a FOV of 360° x 90°, the upper hemisphere of a sphere.

The mosaicking work of the past decade has solved many of the problems that complicate mosaicking. Today, mosaics can be generated in a wide variety of imaging situations. Through their availability, mosaics have become better known and applied to many different applications. The mosaic's use as a visualization device remains one of the most prevalent applications of mosaics. Visualization using mosaics has extended far beyond its initial conception, however. QuickTime VR (Virtual Reality) [14] is one of the best known applications of mosaicking . QuickTime VR uses multiple panoramic images to give the viewer the impression of navigating a virtual environment. Virtual environments are simulated by using multiple 360° panoramic mosaics and "hopping" between panoramic images. QuickTime VR and similar visualization-based mosaicking applications are the most direct extensions of mosaicking. However,

mosaicking is useful in a wide range of applications. Mosaics have applications in video indexing [15] [16], robot localization [17] [18], surveillance [19], 3D visualization [14] [20], video compression, video visualization, video enhancement, and video editing [15].

This thesis uses mosaicking to correct azimuth and inclination data obtained from a multi-sensor video camera (MSC) and facilitate object size estimation from video data. Few mosaicking systems have been created that utilize mosaicking as a method to refine ancillary sensor data and to provide measurement capabilities. However, ancillary orientation sensors have been utilized to reduce mosaicking complexity. In [21], a multi-sensor camera that measures its three dimensional orientation is used to help remove image distortions resulting from perspective distortions and rotations; however, muti-sensor data is electronically correlated with video data and is assumed to be accurate. Additionally, [21] mosaics data similar to that obtained in satellite or aerial mosaicking—the imaged regions are, or are assumed to be, nearly planar and camera motions, purely translational. This thesis presents a system designed to create mosaics of non-planar data for object measurement.

1.2 Motivation

This work was funded by the US Forest Service (USFS) as a new method to collect individual tree stem metrics. Volume, height, and biomass information are inventory metrics kept by the USFS to provide quantitative data about the forests in the US. Unfortunately, current data collection techniques are either very approximate or very time consuming.

There are two techniques currently used to obtain volume, height, and biomass metrics of US forests. The first technique requires one field worker to stand at the base of the tree and measure its breadth at chest height. Tree breadth at chest height and rough height information are used as inputs into known tree models to estimate tree metrics. Technique 1 is highly approximate and is considered secondary to technique 2. Technique 2 involves three to four field workers. Tree height, volume, and biomass are measured by constructing a segmented aluminum pole that rises the entire height of the tree's trunk. The segmented pole is constructed from a series of pole components that are pieced together as the segmented pole is raised along the trunk. Each pole component has a marking at its ends. Once the segmented pole has been constructed, the width of the tree is measured at the marked portions of each pole component using a *pentaprism caliper*. A pentaprism caliper determines distances of distant objects using a system of mirrors that are manually aligned when the instrument is pointed at its target. By interpolating between segments, the tree's volume is estimated; by counting the number of segments, the tree's height is known; and, with volume information, the tree's biomass is estimated.

Although the second measurement technique is much more accurate than the first, the second technique is time consuming and can be dangerous. Despite the difficulties involved in the USFS data collection techniques, forest inventory data are updated annually. These data collection difficulties call for better measurement methods to improve estimation accuracy, reduce labor costs, and save data collection time. To expedite data collection, the USFS has created a special-purpose multi-sensor video camera (MSC) [22]. The camera is equipped with a laser range finder and three internally mounted inclinometers that measure MSC inclination, azimuth, and roll. This thesis describes a system that refines MSC image and orientation data so

that tree measurement using video data is possible.

1.3 Contributions of the Research

This thesis presents new ideas about applications and methods of mosaicking. Mosaicking is proposed as a measurement device for non-translational image data containing perspective distortions, contrast changes, and occlusions. Mosaicking is used for a novel purpose; mosaicking is used to correct multi-sensor data by augmenting sensor data with image information. Mosaicking is presented as a process constrained to preserve the geometry of constituent images. Affine matching is suggested as a method solely to improve mosaic image quality rather than a warping technique to project images to a common surface. Within the mosaicking process, an energy metric is defined that allows for automatic detection and resolution of mosaicking errors.

1.4 Organization of Thesis

This thesis is organized as follows: chapter 2 discusses the motivation behind this work, chapter 3 describes the mosaicking system implemented, chapter 4 presents the results generated by the mosaicking system, and chapter 5 presents conclusions drawn from the results and suggests areas for future work. Additionally, Appendix A illustrates the data sets used to test the mosaicking system.

CHAPTER 2. Data Collection and the Multi-Sensor Camera

2.1 The Multi-sensor Camera

This thesis uses a custom multi-sensor camera (MSC) designed by the USFS to provide an alternative to current methods of tree measurement. The camera and frame extraction system described in this chapter are the work of the USFS. For details of the system's development beyond those discussed in Chapter 2, see [22].

The MSC was created by mounting a video camera (Panasonic GP-CX161) into a multi-sensor instrument (the "Advantage Laser Rangefinder") made by Laser Atlanta Optics Inc. [23]. The Advantage Laser Rangefinder contains a pulse laser-rangefinder and internally mounted inclinometers that measure the instrument's orientation about three axes. The MSC's video camera has a 720 x 480 pixel CCD array with a custom lens arrangement and a 250 mm focal length that creates a field of view of approximately 5.5 degrees high by 8.24 degrees wide. Video and range-orientation (RO) data are asynchronously captured into two separate sequential data streams onto a videocassette and memory card (respectively). The video and RO data streams are synchronized offline. Video data are collected at 30 frames per second (interlaced) and RO data are subsampled to 10 readings per second from 238 raw measurements per second. The MSC is mounted on a monopod for ease of carrying and stability during data collection. The monopod also partially restricts instrument rotation about the camera axis.

Because the MSC records RO and video data asynchronously and at different frequencies, RO and video data are not perfectly correlated on the MSC. The video and RO data streams must be correlated offline to determine orientation measurements for each video frame. RO-video correlation is achieved by sorting through the RO data and computing the times that each RO entry was recorded. Video frames are then extracted from the MSC at the time of each RO data entry. The frame times are then used to trigger a frame capturing function to extract corresponding frames from the video sequence. Through sequential correlation, data sets of correlated RO-video frames can be created at various frequencies (limited by the RO sensor sampling rate, 10Hz).

2.2 The Data Collection Methodology

To collect data, the MSC should be positioned with little or no visual obstruction at approximately the same distance from the tree stem as the greatest height being collected. Data collection begins with the MSC aimed at a point on the stem where the range is not obstructed, as this range will be used to validate subsequent range measurements. The MSC is then aligned with the bottom point of the stem to create a base height for referencing subsequent height measurements. The MSC is then slowly rotated up the stem until the highest desired point is reached. This data collection process is repeated for any number of trees. The RO and video data stored on videocassette and memory card for one or more trees constitutes a *data set*.

Once the RO and video data have been recorded, *frame sets* are extracted from MSC camera data sets. We define a frame set to be a set of information for one object (i.e. multiple trees do not

occur within a single frame set). We define two types of frame sets, *video frame sets* (F_v) and *RO frame sets* (F_o). Let $F_v = \{f_0, f_1, \ldots f_n\}$ represent an ordered set of video frames and $F_o = \{\Phi_0, \Phi_1, \ldots \Phi_n\}$ represent a corresponding set of RO frames. An RO frame Φ is defined as $\Phi = \{\alpha, \beta, \rho, \delta\}$ where α, β, ρ, and δ are inclination, azimuth, roll, and range (respectively). The RO frame sets are built by parsing the RO data to generate frame sets monotonically increasing by approximate 1° changes in inclination angle. Given an RO frame set, F_o, we create F_v such that for all elements Φ_i in F_o, frame f_i in F_v corresponds to the visual data at orientation Φ_i. Frame sets F_o and F_v are used as the input to the mosaicking system used to generate mosaics and refine orientation data.

2.3 System and Multi-sensor Camera Error

The video-correlated RO data obtained from the MSC will contain errors. According to specifications, the MSC's RO data is accurate to ±0.4° in inclination, ±1.5° in azimuth, ±0.4° in roll, and ±0.5 feet in range. Using a focal length of 250 mm, the azimuth and inclination tolerances imply ±35 pixels (vertical) by ±131 pixels (horizontal) of error in image positioning. (Derivation of the pixel error, ±35 by ±131 pixels, is discussed in Section 3.3). Unfortunately, the data collection process introduces additional error beyond RO sensor tolerances: 1) asynchronous RO and video recording allow RO frames to lead or lag video data by small amounts; 2) quick MSC motions cause additional correlation error because orientation sensors are inertial; 3) 10 Hz orientation sampling produces data correlation ambiguities because video is sampled at 30 Hz (interlaced).

The MSC error, augmented by system error, creates orientation data that is only approximate, and increases RO-video correlation errors unpredictably. The amount of correlation error depends upon the amount of time lag between RO and video sensors and MSC motion. Data is collected using slow and steady hand-controlled camera motions to reduce the amount of RO-video correlation error in the system. Figure 2.1 illustrates typical RO data obtained from a MSC scan of a tree. Azimuth and roll data changes little while scanning a tree; inclination increases steadily. The range data in Figure 2.1 contains a large amount of noise from intermittent occlusions between the camera and the targeted object—this is typical of field data.

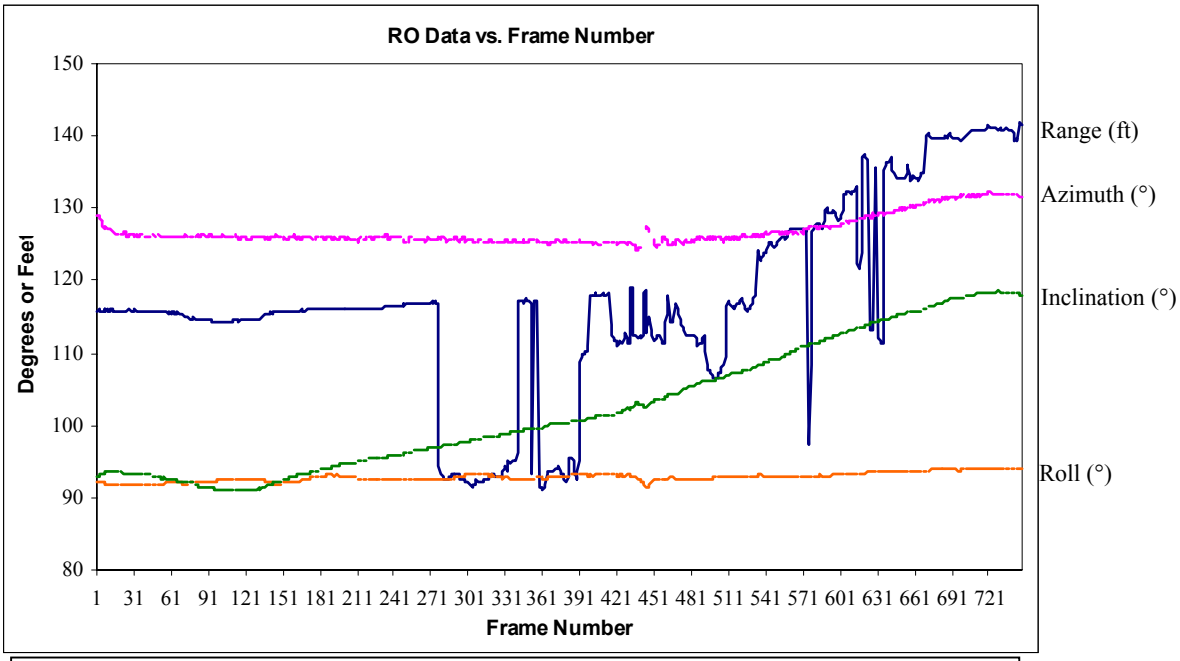

Figure 2.1: Plot of range, azimuth, inclination, and roll data for a RO-video dataset. Range data contains noise from occluding objects; azimuth is generally constant; inclination slowly increases toward the top of the object; roll remains relatively constant.

Table 2.1 illustrates the data contained in a correlated video and orientation frame set that is used to generate mosaics. Each orientation video frame contains a frame identification number, range, azimuth, inclination, and roll measurement; each video frame contains an image reference.

Table 2.1: Sample data returned from the frame extraction system.

Frame #	Range	Azimuth	Inclination	Roll	File Name
437	34.1	304.2	8.9	90.5	sm11-20055.dib
456	22.9	304.7	10.3	90.6	sm11-20056.dib
476	44.7	304.3	11.8	90.5	sm11-20057.dib
489	42.4	304.3	13.2	90.5	sm11-20058.dib
501	10.9	304.2	14.3	90.6	sm11-20059.dib
514	26.9	304.1	15.9	90.6	sm11-20060.dib
528	27	303.7	17.4	90.7	sm11-20061.dib
544	45.7	303.5	18.7	90.6	sm11-20062.dib
559	46	303.4	20.2	90.9	sm11-20063.dib
572	12.1	303.4	21.6	90.9	sm11-20064.dib
585	47	302.6	23.2	91	sm11-20065.dib
595	47.9	302.4	24.8	90.9	sm11-20066.dib

Figure 2.2 illustrates some data taken with the MSC. Each image is 740 x 480 pixels and contains two red overlays that are produced by the camera. The first overlay, a cross hair or "+" sign, marks the calibration point for the RO sensors. The azimuth, inclination, roll, and range measurements correlated with each frame are taken at the cross hair location in the images. The second overlay, marked by a symbol and a number, displays RO data about the current target. The MSC iteratively displays each RO data type in sequence.

Before mosaicking, video frame sets are preprocessed to remove image artifacts created by the overlay data and by interlacing effects of the camera. To remove overlay information, images are cropped to include the top 704 x 288 pixels. After cropping, video frame sets are de-interlaced by removing every other row and averaging pairwise columns. Preprocessing generates 352 x 144 pixel images without overlay data or interlacing errors.

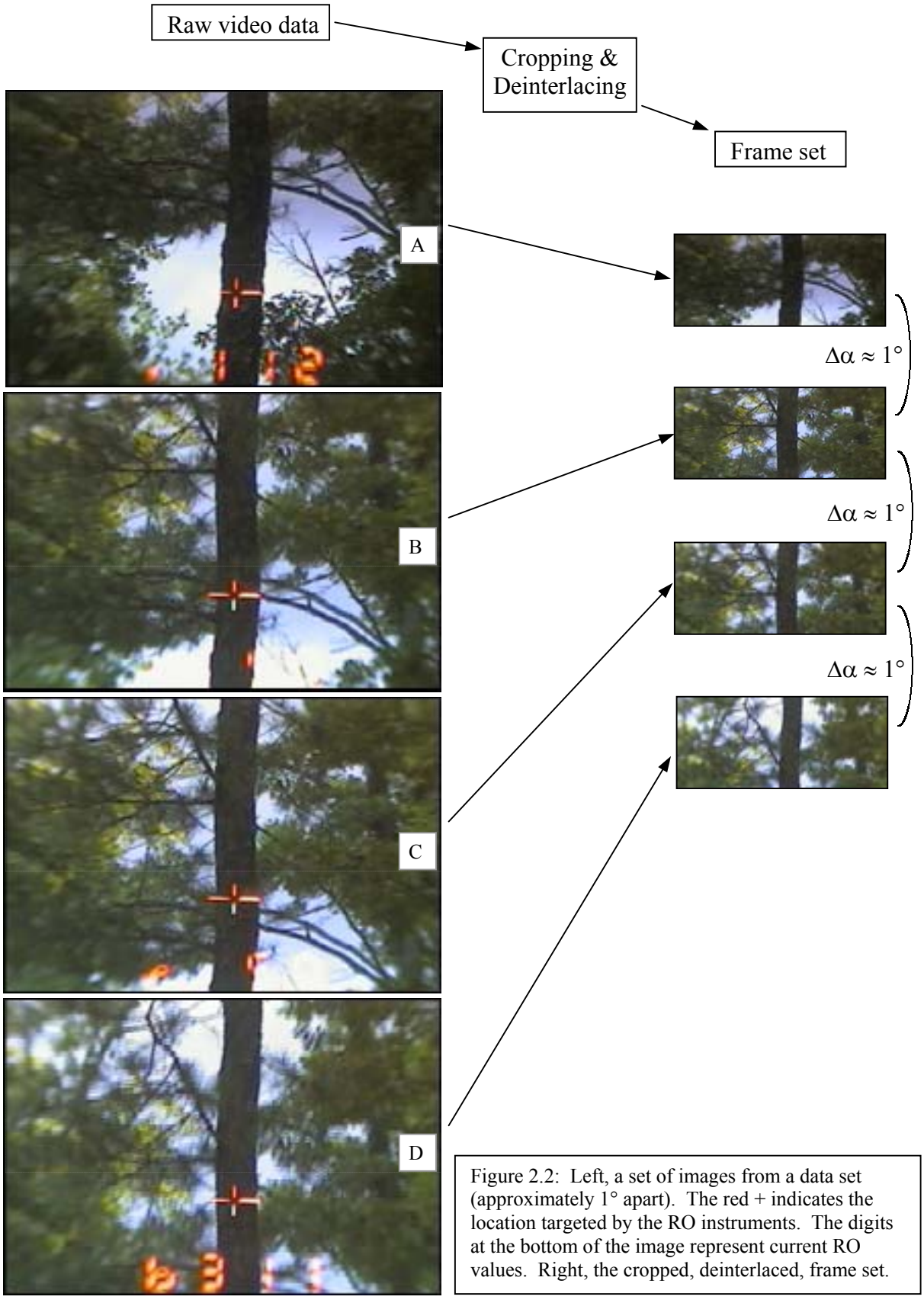

Figure 2.2: Left, a set of images from a data set (approximately 1° apart). The red + indicates the location targeted by the RO instruments. The digits at the bottom of the image represent current RO values. Right, the cropped, deinterlaced, frame set.

CHAPTER 3. The Mosaicking System

3.1 The 7-Step Angle Refinement Process

This thesis seeks to create a system that reduces errors in RO data—azimuth and inclination angles—and generate mosaics that preserve the original geometric relationships in their constituent images. The mosaicking process is designed as a seven-step sequence of mosaic refinements to create a mosaic image and improve the accuracy of RO data. The system accepts frame sets F_o and F_v as input and returns two results, a super-resolution-refined mosaic of connected video frames and a *raw mosaic*. We define a raw mosaic as a set of n video frames $\{f_i, \ldots f_{i+n}\}$ and n-1 *specialized orientation frames* $\{\Phi'_{i+1}, \ldots \Phi'_{i+n}\}$ that describe the relationship between images in a mosaic. Specialized orientation frames describe orientation relationships in units of pixels, rather than degrees. A specialized orientation frame $\Phi'_i = \{\Delta a_i, \Delta b_i, \Pi_i\}$ specifies a vertical translation, horizontal translation, and warping matrix (respectively) for each *mosaic pair* in a raw mosaic. We define mosaic pairs as video frame pairs $\{f_i, f_{i+n}\}$ such that f_{i+n} overlaps f_i when f_{i+n} is translated by some horizontal and vertical displacement Δb_i and Δa_i. Figure 3.1 illustrates a mosaic pair and the Φ'_i metrics used to describe the relationship between the two video frames. Together, the mosaic image and raw mosaic provide the data necessary for measurement of objects within each frame and for measurement of objects that exceed the camera's field-of-view.

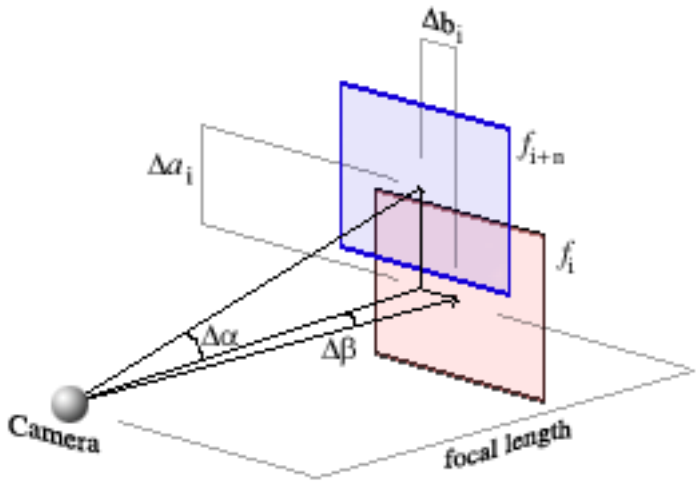

Figure 3.1: A mosaic pair. Frames f_i and f_{i+n} are separated by $\Delta\alpha$ degrees of inclination and $\Delta\beta$ degrees of azimuth. Measures Δa_i and Δb_i are specified in pixels. (f_i and f_{i+n} are drawn in the virtual image plane for clarity).

The seven-step refinement process is designed as follows:

1) An initial mosaic is generated directly from RO data;

2) The mosaic is refined by estimating pairwise translation using coarse-to-fine analysis for energy minimization;

3) Mosaicking errors are detected and removed using a mosaicking accuracy metric;

4) Translation amounts are refined using color analysis;

5) The mosaic is refined by estimating an affine warping of adjacent frames;

6) Video frame data is refined using super-resolution techniques;

7) The final mosaic is assembled; the mosaic image and raw mosaic are returned.

Our design is in the spirit of Peleg and Herman's work in [10, 11], which also preserves geometric relationships of constituent frames. In [11], Peleg views a video sequence as a series of 1-dimensional strips to be aligned into a mosaic by determining camera motion. Peleg and

Herman use coarse-to-fine analysis for energy minimization to determine translation parameters real time. Additional refinement is next performed offline to determine refine translations and rotations. This thesis uses data that is inherently offline and is coarsely sampled from the original video data. By using coarsely sampled video data, we reduce the amount of data processing needed to generate a mosaic and refine frame orientation data in F_o. Where [11] focuses on improving mosaic quality, we focus on improving orientation data and video frame accuracy.

3.2 Step 1 – Initial Raw Mosaic Construction

Before refinements to the mosaic can be made, an initial raw mosaic must be created. The initial mosaic is constructed by extracting mosaic pairs $\{f_i, f_{i+2}\}$ from a frame set that are separated by approximately 2° of inclination ($\Delta\alpha \approx 2°$). Because frame sets are obtained from the MSC at approximate 1° inclination increments, the initial raw mosaic does not utilize all data extracted from the MSC—this additional data is used for further refinement in Step 3. Two-degree (approximate) inclination increments provide a balance between the number of frames that must be processed to create a mosaic and the visual artifacts present in the final image. When angular change in a mosaic pair is small, contrast and perspective differences within mosaic pairs are typically small; when angular change is high, less data must be processed to generate a mosaic but greater contrast and perspective differences complicate mosaicking.

For each mosaic pair in the initial raw mosaic, azimuth and inclination data from F_o is converted

into horizontal and vertical pixel displacements Δb and Δa in Φ'. Pixel displacements, Δb and Δa, are calculated for each mosaic pair using the associated change in azimuth and inclination $\Delta\beta$ and $\Delta\alpha$. Figure 3.2 illustrates the geometric model used to determine the pixel offsets $\Delta\alpha$ and $\Delta\beta$ for each mosaic pair. ($\Delta\beta$ is computed using the same model as $\Delta\alpha$).

In Figure 3.1, $\Delta\alpha$ represents the change in inclination between two adjacent frames, f_i and f_{i+n}. Camera motion is assumed to be approximately about the camera's focal point. (Frames are drawn in the virtual image plane, for clarity). The displacement between frames f_i and f_{i+n} is computed using the focal length, f_l = 250 mm, and the tangent function.

$$\Delta a = f_l \, \eta \, \mathrm{Tan}\,(\Delta\alpha)$$

Where η is a camera constant in pixels/mm. Equation 3.1 computes Δa_i as the distance from the optical center of f_{i+n} to the point in f_{i+n} that represents the information imaged at the optical center of f_i. Because successive image planes are not parallel, translation does not fully model the relationship between adjacent frames; however, inclination changes of 2° impose only subtle image distortions in the system.

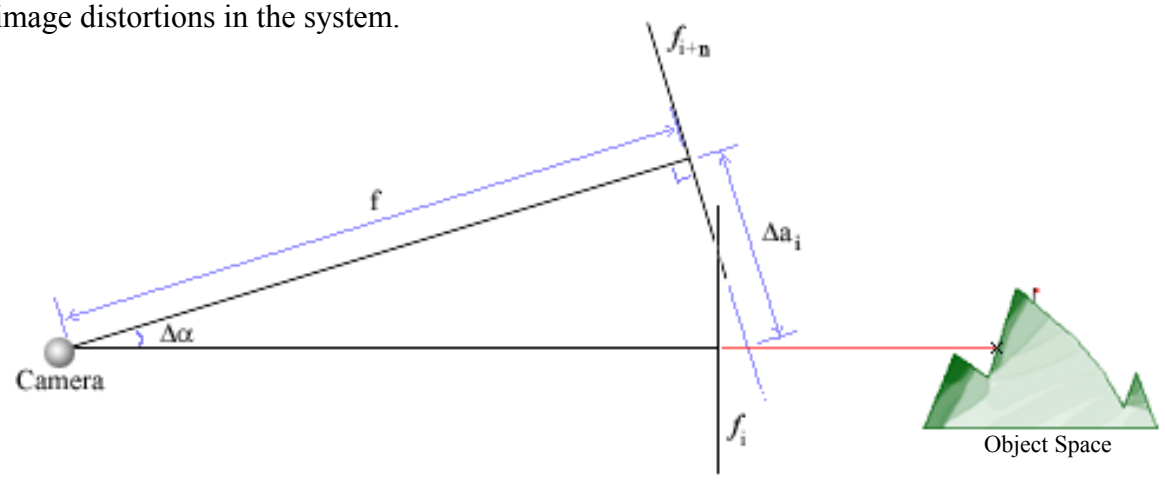

Figure 3.2: The geometric arrangement used to derive the relationship between inclination angle and pixel disparity between successive frames f_{i+n} and f_i. Light from the imaged object strikes the imaging plane f_i directly at the optical center. The same light strikes image plane f_{i+n} Δa_i pixels below its optical center (see intersection of blue and red lines). Δa_i is computed using the depicted geometric assumptions.

The pairwise pixel displacements in each mosaic pair are referred to as mosaic points, or *m-points*. If f_i and f_{i+n}, are two video frames in a video frame set $F_v = \{f_1, f_2, \ldots, f_n\}$, an m-point (x, y) is a coordinate in f_i that achieves an approximate frame correspondence when the lower left corner of frame f_{i+n} is positioned at location (x, y) in frame f_i. Figure 3.3 demonstrates a possible m-point for a sample mosaic pair. Note that the m-point in Figure 3.3 does not properly align the images—we use the term m-point to refer to any possible alignment of a mosaic pair. Given a set of m-points for a mosaic frame set, the initial raw mosaic is constructed. Each mosaic pair in the initial raw mosaic contains orientation data (the computed m-point), image data, and warping data. Note that the initial raw mosaic specifies only translation to match each mosaic pair.

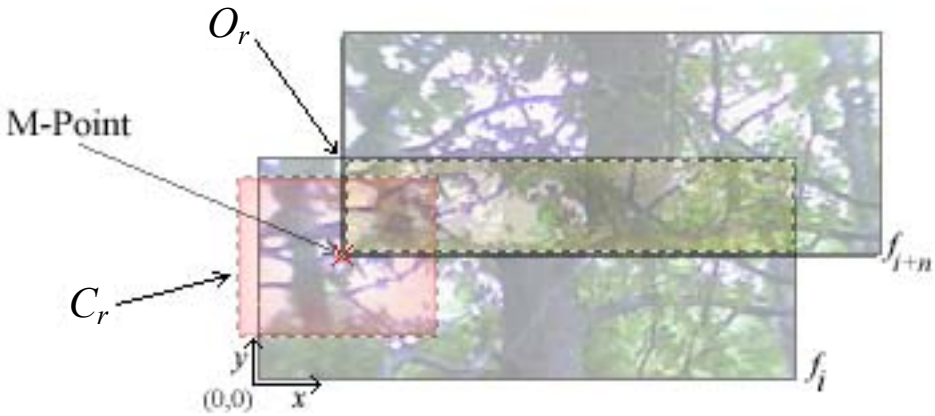

Figure 3.3: A possible m-point location within a mosaic pair. The red "x" marks a possible m-point. An m-point is determined by the location of the lower left corner of f_{i+n} in frame f_i. Region O_r (the overlap region) is shown in blue; region C_r (the confusion region) is shown in red.

3.3 Step 2 – Mosaic Refinement Using Coarse-to-fine Energy Analysis

The m-points specified in the initial raw mosaic depend only on initial RO data and need refinement through image matching. The system uses these estimated m-points (*guess m-points*) to determine the *actual m-point*—the m-point that creates minimum visual discontinuities when

two successive frames are mosaiced by translation. In Step 2, the initial raw mosaic is refined by minimizing an energy metric in coarse-to-fine processing.

Let [$e_{\Delta a}$, $e_{\Delta b}$] represent the inclination and azimuth error (in pixels) inherent in the camera and frame extraction system. The actual m-point lies within [$\pm e_{\Delta a}$, $\pm e_{\Delta b}$] pixels of the guess m-point—we refer to the 2[$e_{\Delta a}$, $e_{\Delta b}$] pixel window that surrounds the guess m-point as the guess m-point's *confusion region*, C_r (see Figure 3.2). Assuming no external errors are imposed upon the orientation data, the actual m-point must lie within the confusion region. The mosaicking system uses [$e_{\Delta a}$, $e_{\Delta b}$] = [±100, ±100] pixels to provide fault tolerance; although, calculations using the manufacturer reported inclination and azimuth accuracies (±1.5° and ±0.4°, respectively) suggest that the confusion region should be [±34, ±131] pixels. Extra vertical fault tolerance is needed to compensate for RO-video frame correlation errors and little horizontal fault tolerance is needed because the intended application involves little change in azimuth.

Once the confusion region has been computed, the system must determine the location of the actual m-point within the confusion region. To find the actual m-point, we define an energy function, *E(i, j)*, as a metric for the "goodness" a particular m-point. *E*(*i, j*) defines the *energy space* of a mosaic pair—the space of all possible m-points and their energy values. If *E*(*i, j*) is a perfect metric, the extreme value in the energy space corresponds to the actual m-point. In this work, we experiment with two energy functions, Equation 3.1 and 3.2:

$$E(i,j) = \frac{\sum_{\forall (i,j) \in Or} \left| f_n(i,j) - f_{n+m}(i - \Delta b_n, j - \Delta a_n) \right|}{\left\| O_r \right\|} \tag{3.1}$$

$$E(i,j) = \frac{\sum_{\forall (i,j) \in Or} \left[f_n(i,j) - \mu_{f_n} \right] \left[f_{n+m}(i - \Delta b_n, j - \Delta a_n) - \mu_{f_{n+1}} \right]}{\sqrt{\sum_{\forall (i,j) \in Or} \left[f_n(i,j) - \mu_{f_1} \right]^2} \sqrt{\sum_{\forall (i,j) \in Or} \left[f_{n+m}(i - \Delta b_n, j - \Delta a_n) - \mu_{f_{n+1}} \right]^2}} \quad (3.2)$$

where

$$\mu_{f_n} = \frac{\sum_{\forall (i,j) \in O_r} f_n(i,j)}{\| O_r \|} \quad (3.3)$$

Equation 3.1 is similar to the normalized sum of squared differences but uses absolute difference rather than squared difference to reduce computation time. Equation 3.1 and 3.2 are defined within the *overlap region* (O_r) of a mosaic pair. In Figure 3.2, the overlap region is shaded in yellow. For all points (*i, j*) in the overlap region (O_r) we sum the absolute difference between image f_i at point (*i, j*) and image f_{i+n} at point (i-Δb_{i+n}, j-Δa_{i+n}). (Because Δb_{i+n} and Δa_{i+n} specify the pixel displacement between f_i and f_{i+n}, they allow us to transform coordinates in frame f_i to coordinates in frame f_{i+n}). We normalize the sum of absolute differences by the number of pixels in the overlap region (O_r) for a given m-point (Figure 3.2 illustrates an overlap region). Without normalization, m-points with large O_r inherently return higher energy values than m-points with small O_r. The normalized sum of absolute differences (NSAD) is theoretically zero for an optimum m-point; therefore, we minimize NSAD energy.

Equation 3.2 is the normalized cross-covariance (NCCV) of all pixels in the overlap region. The mean, μ_f, is calculated using the pixels in O_r using Equation 3.3. The extreme point in NCCV energy space corresponds to the maximum value of the NCCV; therefore, we maximize NCCV energy.

Restricting computation of Equation 3.1 and 3.2 within the confusion region reduces data processing and eliminates erroneous values energy values. Erroneous values of $E(i,j)$ occur when information content is low between a mosaic pair at a given m-point. Low information content for an m-point occurs when a mosaic pair does not contain significant image features or when the overlap-region between frames is small. Cases of small overlap necessitate the use of orientation data to formulate a guess m-point. Without a guess m-point and confusion region, energy minimization functions must be sensitive to the size of the corresponding overlap region.

Energy computation is used in a 4-level coarse-to-fine refinement scheme, using a Gaussian image pyramid. Coarse-to-fine refinement provides two advantages: first, computation complexity is reduced with smaller data sets; second, coarse-to-fine refinement allows the mosaicking system to discover actual m-points that may lie just beyond the confusion region. At the smallest level of a 4-level Gaussian image pyramid, the confusion region is reduced by a factor of 2^4, or [±3,±3] for our confusion region ([±50,±50]).

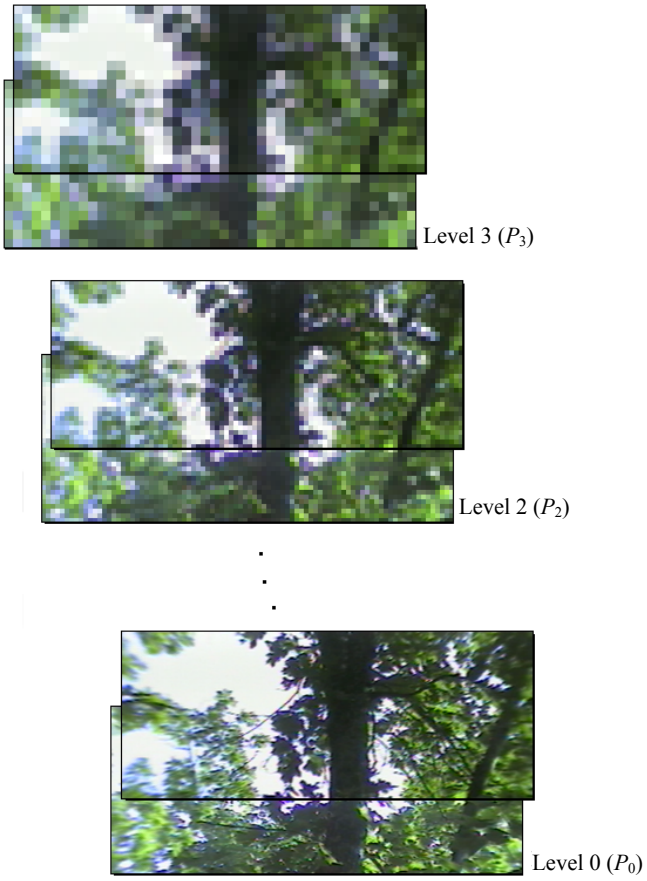

Figure 3.4: Coarse-to-fine mosaicking using a Gaussian image pyramid.

Figure 3.4 illustrates coarse-to-fine mosaicking. At each successive level of the

pyramid, image resolution is one quarter the resolution of the preceding level (image width and height are halved as the pyramid level increases). The coarse-to-fine process begins at level 3 (P_3) of the Gaussian pyramid and ends at level 0 (P_0), the full resolution image. Once an m-point $m = (x,y)$ has been computed for some P_n, m is used to specify the guess m-point in P_{n-1}. Additionally, computation of m reduces the size of the confusion region in P_{n-1}. Because each pixel in P_n represents four pixels in P_{n-1}, the pyramid confusion region, PC_r, is ideally [±1, ±1] pixels in P_{n-1}. However, mosaicking errors in P_n necessitate fault tolerance—we use PC_r = [±4, ±4] for P_{n-1} in coarse-to-fine mosaicking refinement. Experimentation with different size confusion regions suggests that C_r = [±50, ±50] and PC_r = [±4, ±4] offers the greatest reliability.

As coarse-to-fine mosaicking processes each P_i, m-point error is reduced. At each successive level of the pyramid, m-point precision increases by a factor of two. Precision optimally reaches ±1 pixel after coarse-to-fine refinement.

3.4 Step 3 – RO Data Error Detection and Removal Using Mosaicking Accuracy Metric

Step 3 attempts to resolve mosaicking errors that occur during Step 2. There are two causes of mosaicking error: first, RO data may be erroneously correlated with video data; second, mosaic pairs may contain insufficient, or misleading, information content. If RO data is erroneously correlated for a mosaic pair, the actual m-point may not lie within the confusion region. When the actual m-point is not present within the confusion region, the minimum energy value tends to occur near the edges of the confusion region—this causes Step 2 to produce m-points that are

brashly incorrect. If information content is low within a mosaic pair, there may be ambiguous or multiple m-points within a confusion region—this also causes coarse-to-fine analysis to return incorrect m-points.

To provide fault tolerance, we use the shape of the energy space at an m-point as a classifier that may be used to distinguish erroneous m-points. Correct m-points usually have extrema of high curvature in energy space; incorrect matches usually have extrema of low curvature. Empirical data were used to determine characteristic extrema shapes that are present in the energy space when Step 2 selects incorrect m-points. Figures 3.5-3.6 illustrate typical energy space plots; Figure 3.7 illustrates an energy space that contains an erroneous extrema. In the energy space plots below, bright blue indicates areas of high energy; boxed regions are inserted as brightened magnifications of the extreme point in the energy space. Figures 3.4-3.7 were generated using Equation 3.1 (NSAD energy); therefore, the minimum value (darkest blue) in Figures 3.4-3.7 corresponds to the m-point as measured by NSAD energy.

Figures 3.5-3.7 illustrate the topographical distinctions between accurate and inaccurate energy spaces. Accurate energy spaces—energy spaces whose extreme point closely corresponds to the actual m-point—possess a concavity/convexity of high curvature. Energy spaces with erroneous m-points have broad features and, often, non-circular contour lines around the extreme point. The magnified extreme point in Figure 3.7 illustrates the shape of an erroneous extreme point.

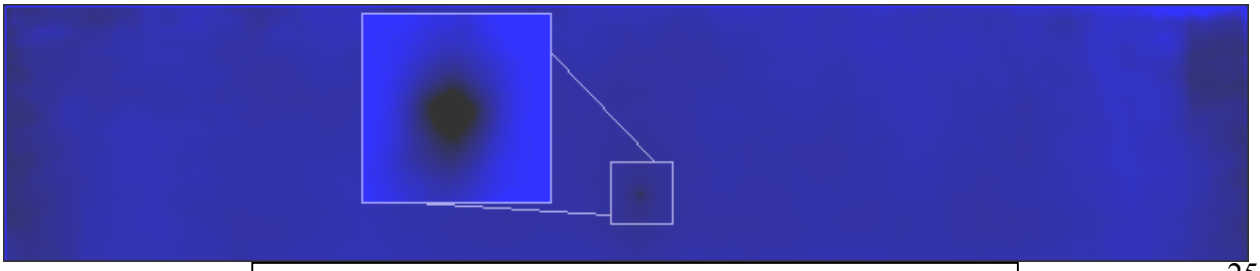

Figure 3.5: Energy space with a correct actual m-point location.

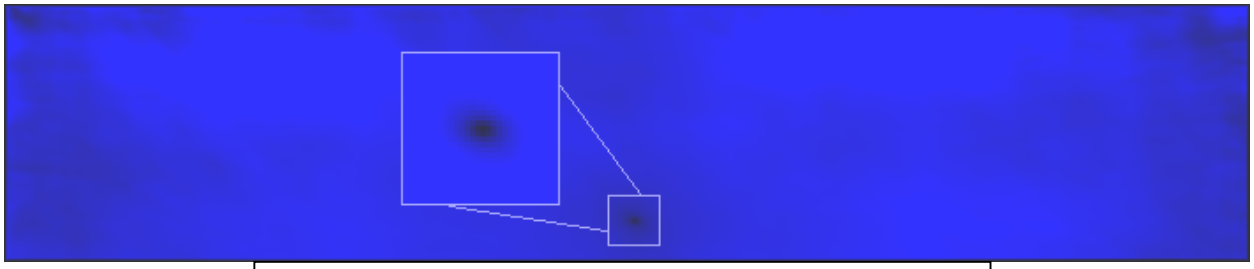

Figure 3.6: Energy space with a correct actual m-point location.

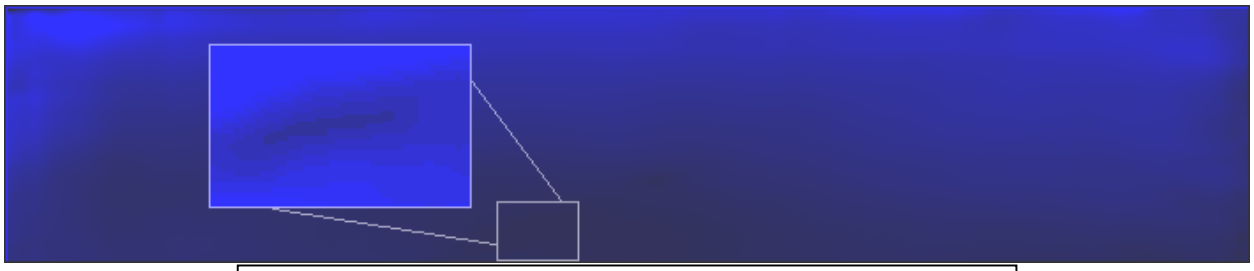

Figure 3.7: Energy space with an incorrect actual m-point location.

To measure the shape of the energy space at an extrema, we calculate the covariances, σ_{xx}, σ_{yy}, and σ_{xy} at the m-point (x, y) in energy space. Let $E_{i=x}(j) = E(i=x, j)$ and $E_{j=y}(i) = E(i, j=y)$ for some m-point (x, y) in an energy space, $E(i, j)$. We define

$$\sigma_{xx} = \mathrm{Cov}(E_{j=y}(n), E_{j=y}(n)), \quad \sigma_{yy} = \mathrm{Cov}(E_{i=x}(n), E_{i=x}(n)), \quad \sigma_{xy} = \mathrm{Cov}(E_{j=y}(n), E_{i=x}(n))$$

and define the mean of $E_{j=y}(n)$ and $E_{i=x}(n)$ as

$$\mu_{j=y} = \frac{\sum_{n=-t}^{t} E_{j=y}(x+n)}{2t} \qquad \mu_{i=x} = \frac{\sum_{n=-t}^{t} E_{i=x}(x+n)}{2t}$$

where t specifies an interval about (x, y) for which $\mu_{j=y}$ and $\mu_{i=x}$ is defined. We calculate t as the minimum distance from (x, y) to the edge of $E(i, j)$; however, we restrict t such that $0 \le t \le 12$ pixels. Given $0 \le t \le 12$, $\mu_{j=y}$ and $\mu_{i=x}$ measure the mean energy value $\pm t$ about pixels (x,y) along $j=y$ and $i=x$ in $E(i, j)$.

Given t, $\mu_{j=y}$, and $\mu_{i=x}$, we define

$$\sigma_{xx} = \mathrm{Cov}\left[E_{j=y}(x), E_{j=y}(x)\right] = \frac{\sum_{i=-t}^{t}\left(E_{j=y}(x+t) - \mu_{j=y}\right)^2}{2t} \tag{3.4}$$

$$\sigma_{yy} = \mathrm{Cov}\left[E_{i=x}(y), E_{i=x}(y)\right] = \frac{\sum_{i=-t}^{t}\left(E_{i=x}(y+t) - \mu_{i=x}\right)^2}{2t} \tag{3.5}$$

$$\sigma_{xy} = \mathrm{Cov}\left[E_{j=y}(x), E_{i=x}(y)\right] = \frac{\sum_{i=-t}^{t}\left[E_{j=y}(x+t) - \mu_{j=y}\right]\left[E_{i=x}(y+t) - \mu_{i=x}\right]}{2t} \tag{3.6}$$

We calculate Equations 3.4-3.6 at the m-point (*x, y*) in *E*. Restricting *t* to $0 \le t \le 12$ pixels restricts σ_{xx}, σ_{yy}, and σ_{xy} computations to small regions about the m-point—this allows us to obtain local information about the shape of the energy space.

Using σ_{xx}, σ_{yy}, σ_{xy}, we classify each m-point as correct, incorrect, or indeterminate. Let

$$C(x, y) = \min\left[\, |\sigma_{xx}|, |\sigma_{yy}|, |\sigma_{xy}| \,\right], \tag{3.7}$$

We define three classes of m-points, using $C(x,y)$: correct, incorrect, or indeterminate. Threshold values on $C(x, y)$ were determined experimentally. Figure 3.8 illustrates a subset of the data used to determine the threshold values; the full data set contains 558 m-points with mean $C(x,y)$ of 150 (maximum, 850). All incorrect m-points conditions are depicted in Figure 3.8. M-points in Figure 3.8 are labeled as correct or incorrect, blue or red (respectively). M-point correctness was determined by visually inspecting mosaic pairs for errors. Given Figure 3.8, data can be classified as correct, incorrect, or indeterminate. We select two threshold values such that for $0 \le C(x,y) \le 3$ an m-point is considered incorrect, for $3 < C(x,y) \le 11$ an m-point is considered indeterminate, and for $11 < C(x,y)$ an m-point is considered correct. Figure 3.8 depicts the threshold locations in the scatter plot. Table 3.1 shows a subset of the empirical data used to determine the threshold values, sorted by the minimum covariance.

Table 3.1: Subset of covariance values used to determine m-point classifier thresholds. M-points were manually verified for correctness. All incorrect m-points occur at the top of the table after sorting by minimum absolute covariance.

M-point	σ_{xx}	σ_{yy}	σ_{xy}	Minimu Covariance	Correct Match?
1	0.02	0.07	0.017	0.017	Not Correct
2	0.029	0.29	0.08	0.029	Not Correct
3	0.15	0.34	0.20	0.15	Not Correct
4	0.72	0.71	0.19	0.19	Correct
5	0.43	0.26	0.20	0.20	Not Correct
6	4.50	0.28	-0.22	0.28	Not Correct
7	1.40	0.62	0.76	0.62	Not Correct
8	0.85	3.80	1.50	0.85	Not Correct
9	0.96	10.98	2.87	0.96	Not Correct
10	1.06	33.67	-1.34	1.06	Not Correct
11	1.38	4.02	2.27	1.38	Not Correct
12	9.43	1.72	2.23	1.72	Not Correct
13	32.68	1.73	-4.57	1.73	Not Correct
14	2.72	2.27	1.75	1.75	Not Correct
15	8.31	17.13	3.47	3.47	Correct
16	4.66	7.62	4.69	4.66	Correct
17	8.48	17.31	10.44	8.48	Correct
18	12.98	8.55	9.65	8.55	Correct
19	17.68	14.63	9.57	9.57	Correct
20	10.43	20.69	11.23	10.43	Correct
21	19.85	61.10	12.73	12.73	Correct
22	15.73	30.72	20.20	15.73	Correct
23	17.9	17.89	17.80	17.80	Correct
24	18.52	25.00	20.81	18.52	Correct
25	18.76	29.82	23.04	18.76	Correct

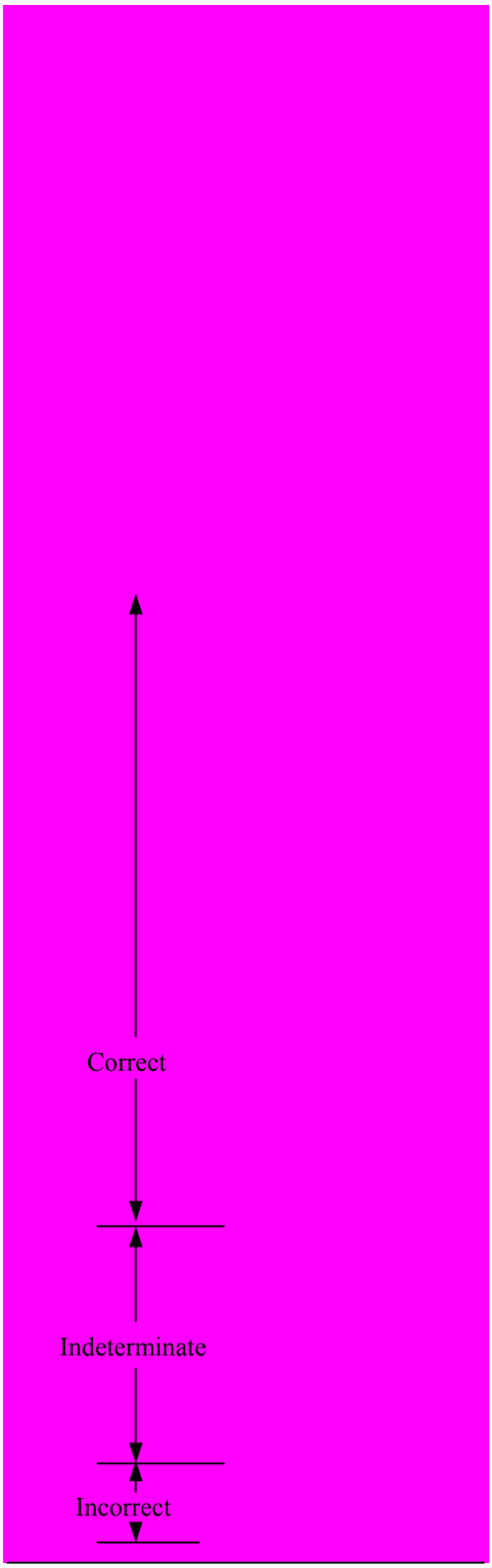

Figure 3.8: 1-Dimensional plot of $C(x,y)$ values along the y-axis. Correct cases, (blue) are well distinguished from the incorrect cases (red) in $C(x,y)$ space. Red points are subtly offset from the y-axis to ensure no blue points are hidden.

M-point error conditions, once identified, are resolved by *adding* RO-video frames to the raw mosaic. Because F_o—and therefore, F_v— are sampled at approximate 1° angle increments and the raw mosaic initially contains frames approximately 2° apart, erroneous m-point conditions can be handled by adding remaining intermediate frames. For some mosaic pair $\{f_i, f_{i+2}\}$ with an incorrect m-point, we add an additional frame, f_{i+1}, to create two mosaic pairs, $\{f_i, f_{i+1}\}$ and $\{f_{i+1}, f_{i+2}\}$. Figure 3.9 illustrates the location of f_{i+1} relative to f_i and f_{i+2}. Mosaic pairs $\{f_i, f_{i+1}\}$ and $\{f_{i+1}, f_{i+2}\}$ typically contain fewer perspective and contrast distortions relative $\{f_i, f_{i+2}\}$ because inclination change is more subtle. RO-video frame addition into the raw mosaic using an m-point classifier allows the system to process less data without sacrificing mosaic quality through incorrect m-points. After f_{i+1} has been added to the raw mosaic, mosaic pairs $\{f_i, f_{i+1}\}$ and $\{f_{i+1}, f_{i+2}\}$ are mosaiced using coarse-to-fine analysis in Step 2, as described in Section 3.3.

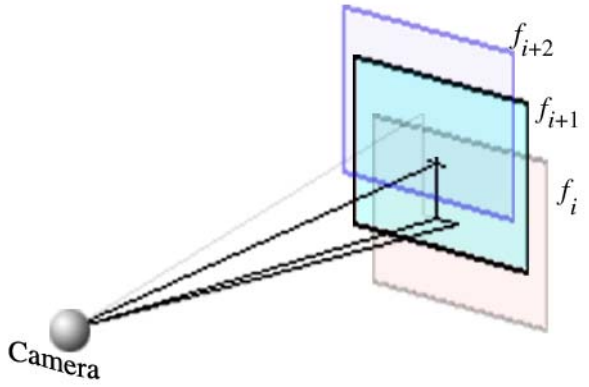

Figure 3.9: Addition of an intermediate frame, F_{i+1}, to the raw mosaic. The new frame is shown in light blue; the old frames are shown lightly above and below the new frame.

3.5 Step 4 – RO Data Refinement Using Color Analysis

Steps 1-3 generate relatively accurate raw mosaics. Step 4 increases m-point precision by using color data to enhance the m-point energy metrics (Equations 3.1 and 3.2) used in initial coarse-to-fine energy analysis. Color data is not used in initial coarse-to-fine energy analysis to reduce computational complexity. However, the results of coarse-to-fine energy analysis and m-point error detection in Steps 2 and 3 allow highly constrained energy-space searches for additional mosaic refinement. Step 4 refines mosaic accuracy using NSAD and NCCV energy functions summed across all color planes. Equations 3.8 and 3.9 are modified versions of the energy functions used in initial coarse-to-fine analysis.

$$E(i,j) = \sum_{\forall ColorPlanes} \frac{\sum_{\forall (i,j) \in Or} \left| F_n(i',j') - F_{n+1}(i'',j'') \right|}{\left\| O_r \right\|} \tag{3.8}$$

$$E(i,j) = \sum_{\forall ColorPlanes} \frac{\sum_{\forall (i,j) \in Or} \left[F_n(i',j') - \mu_{F_n} \right] \left[F_{n+1}(i'',j'') - \mu_{F_{n+1}} \right]}{\sqrt{\sum_{\forall (i,j) \in Or} \left[F_n(i',j') - \mu_{F_1} \right]^2} \sqrt{\sum_{\forall (i,j) \in Or} \left[F_{n+1}(i'',j'') - \mu_{F_{n+1}} \right]^2}} \tag{3.9}$$

The energy metrics are again refined using coarse-to-fine analysis; however, PC_r—the confusion region used as processing moves down the pyramid—no longer need be [±4, ±4]. Step 4 uses PC_r = [±2, ±2].

3.6 Step 5 – RO Data Refinement by Affine Transformation Estimation

The previous steps produce mosaics using pure translation. The MSC, although mounted on a monopod, allows rotations and other image distortions between mosaic pairs. Step 5 refines mosaic accuracy by estimating pairwise image rotations and distortions as affine transformations. Using affine data, mosaic are refined, both in translation accuracy and in image quality through super-resolution based noise removal (Step 6 discusses super-resolution). Affine transformation parameters are estimated, as described in [24], using a Newton-Raphson style minimization procedure similar to [25].

Given two image frames, I and J, we view the change between the frames as image motion—I and J contain similar information, but oriented differently. Using an initial estimate of the motion field, determined as pure translation during preceding steps, we compute the motion field within the overlap region, O_r, as an affine transformation or affine motion field. Because image pairs are extracted at approximate 2° increments, perspective distortions are small for the intended application—affine motion is a reasonable approximation to image motions within a mosaic pair. And, because we do not intend to warp image frames to a common manifold, affine motion may be assumed without consideration of global perspective distortions.

Let the affine motion field be specified by **D**—a deformation matrix—and **d** a displacement vector.

$$\mathbf{D} = \begin{bmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{bmatrix} \qquad \mathbf{d} = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

Given **D** and **d**, we can express the motion of a single point, $\mathbf{x} = [x, y]^T$, in image J as $\mathbf{Ax} + \mathbf{d}$

where

$$\mathbf{A} = \begin{bmatrix} 1 + d_{xx} & d_{xy} \\ d_{yx} & 1 + d_{yy} \end{bmatrix}$$

Image *I* is, then, approximately equivalent to *J* when points in *J* are remapped by the motion field, **Ax + d.** Therefore,

$$I(\mathbf{x}) \approx J(\mathbf{Ax} + \mathbf{d}) \tag{3.10}$$

Given a motion field that maps *J* onto *I*, we define an error term, e, as the sum of squared differences between corresponding pixels in images *I* and *J*. Equation 3.12 is evaluated within the overlap region, O_r, of images *I* and *J* (when *J* is warped by **Ax + d**).

$$e = \iint_{O_r} \left[J(\mathbf{Ax} + \mathbf{d}) - I(\mathbf{x}) \right]^2 d\mathbf{x} \tag{3.11}$$

To minimize this error metric, we differentiate with respect to the unknown transformation parameters, d_{xx}, d_{xy}, d_{yx}, d_{yy}, d_x, and d_y. In matrix form:

$$\frac{1}{2}\frac{\partial e}{\partial \mathbf{D}} = \iint_{O_r} \left[J(\mathbf{Ax} + \mathbf{d}) - I(\mathbf{x}) \right] \mathbf{g}\ \mathbf{x}^T d\mathbf{x} = 0 \tag{3.12}$$

$$\frac{1}{2}\frac{\partial e}{\partial \mathbf{d}} = \iint_{O_r} \left[J(\mathbf{Ax} + \mathbf{d}) - I(\mathbf{x}) \right] \mathbf{g}\ d\mathbf{x} = 0 \tag{3.13}$$

where

$$\mathbf{g} = \begin{bmatrix} g_x & g_y \end{bmatrix}^T = \begin{bmatrix} \frac{\partial J}{\partial x} & \frac{\partial J}{\partial y} \end{bmatrix}^T$$

To solve this system, we linearize the system using the truncated Taylor expansion.

$$J(\mathbf{Ax}+\mathbf{d}) \approx\ J(\mathbf{x}+\mathbf{u}) \approx J(\mathbf{x}) + \mathbf{g}^T \mathbf{u} \tag{3.14}$$

By linearizing equations 3.12 and 3.13, we obtain the following system (details can be found in [26]):

$$\mathbf{Tz} = \mathbf{a} \tag{3.15}$$

where

$$\mathbf{T} = \iint_w \begin{bmatrix} x^2 g_x^2 & x^2 g_x g_y & xy g_x^2 & xy g_x g_y & x g_x^2 & x g_x g_y \\ x^2 g_x g_y & x^2 g_y^2 & xy g_x g_y & xy g_y^2 & x g_x g_y & x g_y^2 \\ xy g_x^2 & xy g_x g_y & y^2 g_x^2 & y^2 g_x g_y & y g_x^2 & y g_x g_y \\ xy g_x g_y & xy g_y^2 & y^2 g_x g_y & y^2 g_y^2 & y g_x g_y & y g_y^2 \\ x g_x^2 & x g_x g_y & y g_x^2 & y g_x g_y & g_x^2 & g_x g_y \\ x g_x g_y & x g_y^2 & y g_x g_y & y g_y^2 & g_x g_y & g_y^2 \end{bmatrix} d\mathbf{x} \tag{3.16}$$

and

$$\mathbf{a} = \iint_w \left[I(\mathbf{x}) - J(\mathbf{x}) \right] \begin{bmatrix} x g_x \\ x g_y \\ y g_x \\ y g_y \\ g_x \\ g_y \end{bmatrix} d\mathbf{x} \tag{3.17}$$

Solving Equation 3.15 for **z** gives us.

$$\mathbf{z} = \begin{bmatrix} d_{xx} & d_{yx} & d_{xy} & d_{yy} & d_x & d_y \end{bmatrix}^T \tag{3.18}$$

Coordinates, x and y, are measured relative to the center of the window area being matched. Iterative computation of **z** in a Newton-Raphson minimization scheme allows us to determine the proper values of the deformation matrix and the displacement vector by incrementally updating **D** and **d** with the values estimated in **z**. Henceforth, we will refer to this system as the *gradient descent* system—we use gradient descent to estimate the affine transformation parameters.

Gradient descent search for affine motion parameters works well if an initial guess for **D** and **d** is available**.** Step 5 initially assumes that $\mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and **d** is defined by refined m-point data from Steps 1-4 ($\mathbf{d} = [\Delta b_i, \Delta a_i]^T$). Once **D** and **d** are initialized, the iterative search is conducted until the maximum change in d_{xx}, d_{yx}, d_{xy}, d_{yy}, d_x, or d_y exceeds a minimum threshold. The

threshold value, 0.01, was determined experimentally to balance estimation accuracy and computation time.

Because overlap regions can be as large as 350 x 90 pixels, iterative search can be time consuming. Overlap region thresholds were tested to limit gradient descent data to regions less than 350 x 90. However, experimental results suggest that gradient descent must operate on the entire overlap region to reliably estimate the affine motion field. Gradient descent converges to improper minima more frequently when overlap region thresholds impose tighter data constraints on the matching system.

Gradient computations were made using a 3x3 Sobel operator and a 5x5 Gaussian-based operator. The 3x3 Sobel operator proved more optimal; the 5x5 Gaussian-based gradient operator does not expedite gradient descent convergence and is more complex than the 3x3 Sobel operator. Additionally, we use bi-linear interpolation to determine grayscale values at real-valued image locations. Time constraints do not allow for more complex interpolation schemes.

Typically, the gradient descent system converges to the correct solution for the affine motion field within approximately 12 iterations. However, incorrect convergence occasionally occurs and is often signified by slow convergence (20 iterations, or more, may occur when the gradient descent converges improperly). We implement an iteration threshold to help reduce the effect of erroneous gradient descent convergence. If the iteration threshold is exceeded, iterative computation ends and the affine motion field is assumed to be pure translation ($\mathbf{d} = [\Delta b_i, \Delta a_i]^T$ and $\mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$), as calculated in preceding steps. Mosaic pairs that do not converge properly in

the gradient descent system are flagged. Flagged mosaic pairs do not use gradient descent results for m-point refinement and are not refined using super-resolution techniques because a suitable match was not found. When the affine motion field can be estimated, the translation values d_x and d_y are used to refine the m-point data.

3.7 Step 6 – Video Frame Data Refinement Using Super-resolution Techniques

After Steps 1-5, the raw mosaic describes the proper orientation of all RO-video frames in the mosaic. Next, we reduce imaging errors inherent in the MSC's CCD sensors and lens system by utilizing the data in the overlap region of mosaic pairs. Using overlap region data and an affine motion field for each mosaic pair, we enhance image quality using super-resolution based noise removal techniques.

Image data inherently contains noise from errors in the MSC imaging system. As light moves through the elements of the camera's imaging system, it is affected by imperfections in the lens system and sensing elements. A camera lens focuses light onto the imaging plane, but also imposes unintended geometric modifications on the incoming data. Large-scale warping errors produce effects such as pincushion, barrel, and spherical distortions. Small lens aberrations also create more subtle image distortions like blurring. Light that has passed through the lens system is next digitized by a CCD array. The digitization process samples light at a discrete number of locations. As the image is sampled, it undergoes additional distortions. Finally, a digital representation of the corrupted original image is stored.

Super-resolution work seeks to reduce imaging errors that corrupt the data from the *image scene* and increase image resolution. We use the term image scene to refer to the physical object(s) being recorded by a camera. Images taken of an object can be viewed as a sampling of that object's visual information. Super-resolution combines multiple samples of an image scene to reduce image distortions and improve image resolution.

Super-resolution work dates back to the early 1980s ([27]). In the past decade, various approaches have been proposed to increase image resolution and quality using multiple image samples. In [28] and [29], a back-projection technique similar to back-projection methods used in computer aided tomography is used to enhance images given an imaging model for blurring, spatial quantization, and additive noise. In [31], Bascle, Blake, and Zisserman extend the ideas of [28] and [29] to include motion blur imposed by camera motions. In [30], super resolution is achieved using a sensor model presented in [32] and specialized image warping techniques. In [33], Chiang and Boult extend [30] to handle more general lighting conditions. In [35] Baker and Kanade develop a system that determines a recognition-based prior probability given a set of input images that contain specific classes of objects.

In this thesis we use the ideas presented in [32] and [34] to enhance image scene data within mosaic pairs. Image enhancement is performed using data in the overlap region of each mosaic pair. Because mosaic sequences are spaced at approximate 2° inclination increments, there are typically only two image samples for each point in object space. [36] suggests that super-resolution scaling requires more than two images. In addition, sub-pixel accuracy is unnecessary and time consuming in this application. Therefore, we do not magnify mosaic images via super-

resolution techniques. Instead, we use the ideas in [32] and modify the super-resolution algorithm in [30] to remove imaging errors without magnification.

Super-resolution and warping inherently require resampling of image data at non-integer locations. The work presented in [32] defines an imaging model used to derive continuous functional representations of images given their discrete samples. There are two types of continuous images discussed in [32], image reconstructions and image restorations. Image reconstruction seeks to derive a continuous image from an image's discrete samples. Image restoration extends reconstruction by assuming that the image has been degraded prior to sampling and attempts to estimate the original continuous image from erroneous samples. We use the ideas of restoration and reconstruction to achieve super-resolution based image refinement.

The full imaging model used in [32] clarifies the relationship between reconstructions and restorations. The imaging process is modeled as a series of filtering operations that modify and degrade the original image signal that enters the camera. Let $F_1(x,y)$ represent the original continuous image before it enters the camera. When $F_1(x,y)$ strikes the lens, it is distorted by a blurring factor $h_1(x,y)$, generating image $F_2(x,y)$. $F_2(x,y)$ is then distorted geometrically by $h_2(x,y)$ to form $F_3(x,y)$. For example, geometric distortions such as pincushion and barrel may be represented by $h_2(x,y)$. $F_3(x,y)$ then strikes the camera's imaging sensor where it undergoes further blurring by the camera's point spread function (PSF) $h_3(x,y)$ to generate image $F_4(x,y)$. Next, $F_4(x,y)$ undergoes spatial sampling at the discrete recording elements giving us $I_s(u, v)$ (where u and v are integers). Finally, an A/D converter quantizes the samples in $I_s(u,v)$ to obtain

the measured image, $I(u,v)$. Figure 3.10 illustrates the imaging model. Image restoration estimates F_3 from the recorded image, I, while image reconstruction estimates F_4. See [32] for further discussion of the constraints and assumptions of this model.

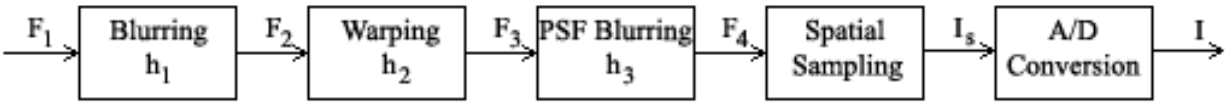

Figure 3.10: The imaging model presented in [Boult & Wolberg 1993].

Chiang and Boult use image restoration and reconstruction for super-resolution in a four-step process that operates on a set of images, S, that represent the same image scene [30]. We modify the four-step process by down sampling the fused image in Step 4 to obtain refined images at the same scale as the original data. To refine image data in this thesis, we use the following steps:

1) Choose an image I_r in S to be the reference image and compute the motion field between all $I \in \{S - I_r\}$ and I_r;
2) Scale I_r up to obtain an image restoration and warp all other images to the up-scaled I_r using the motion field computed in step one;
3) Create a super-resolution image by fusing all the images together;
4) Down sample the super-resolution image using the CCD PSF.

We apply this image refinement system to mosaic pairs. Let f_i and f_{i+n} constitute a mosaic pair where f_{i+n} is of greater inclination than f_i ($S = \{f_i, f_{i+n}\}$). The affine motion field, computed during preceding mosaic orientation refinement, is used to warp f_{i+n} into f_i within the overlap region. f_i and f_{i+n} are restored, fused, and then down sampled using image reconstruction techniques.

To perform restoration, [32] derives a quadratic restoration function assuming a simple Rect PSF that spans exactly one pixel. Treating image dimensions seperably, let $Q_i(x)$ represent the restored image at a single pixel location, where x ranges over the pixel's width ($0 \le x \le 1$). To ensure that the restored image is continuous, we define E_i and E_{i+1} as the values at pixel boundaries and set $Q(0) = E_i$ and $Q(1) = E_{i+1}$. Given values at the pixel boundaries, we define the pixel value $V_i = \int_0^1 Q(x)\,dx$. From these three constraints, we obtain a quadratic polynomial for $Q(x)$:

$$Q(x) = E_i + (6V_i - 2E_{i+1} - 4E_i)x + 3(E_{i+1} + E_i - 2V_i)x^2 \tag{3.19}$$

Figure 3.11 illustrates the relationships between Q, E, and V. The boundary pixel value, E, forces the restoration to be continuous, as implied in Figure 3.11.

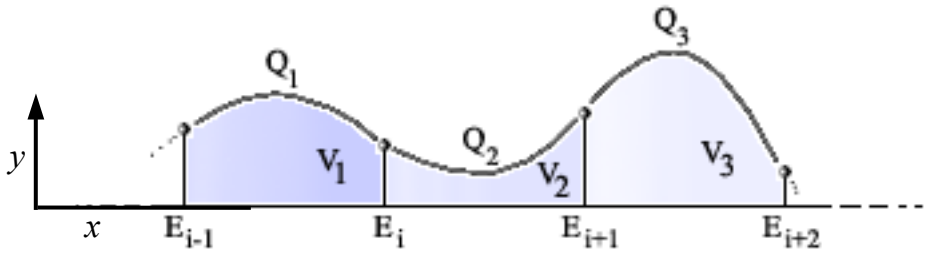

Figure 3.11: The relationships between Q, E, and V are illustrated above. The x axis represents the image plane; the y axis represents image intensity. Each pixel, V_i, is the integral of Q_i—the original continuous signal—between the pixel boundaries E_i and E_{i+1}.

And, using cubic convolution to obtain E_i and E_{i+1},

$$E_i = \frac{1}{8}\left(AV_{i-2} + (4-A)V_{i-1} + (4-A)V_i + AV_{i+1}\right) \tag{3.20}$$

$$E_{i+1} = \frac{1}{8}\left(AV_{i-1} + (4-A)V_i + (4-A)V_{i+1} + AV_{i+2}\right) \tag{3.21}$$

Where A is a free parameter in Equations 3.20 and 3.21. According to [32], A is generally used between [-3, 0]. However, we experiment with a wide range of values ($-20 \le A \le -0.5$).

Up sampled images in S are next fused together by averaging corresponding pixels. The resulting fused image is then down sampled using a Rect PSF. Down sampling is accomplished by applying the reconstruction equation derived in conjunction with Equation 3.19 (see [32]).

$$Q_r(w) = V_i + (E_{i+1} - E_i)w + (2E_{i+2} - E_{i+1} + 3(V_{i+1} - V_i))w^2 + (E_{i+2} - E_i - 2(V_{i+1} - V_i))w^3 \quad (3.22)$$

Where $w = x + 0.5$.

By applying this four-step process, data in mosaic pair overlap regions O_r is enhanced. Imaging artifacts are removed by restoring overlap image data according to the assumed PSF filter and averaging between two images f_i and f_{i+n}. Down sampling has the effect of sharpening the image—we do not have enough images to obtain a sharp SR image at higher resolutions.

3.8 Step 7 – Mosaic Stitching and Object Measurement

Given refined m-point and video data in a raw mosaic, mosaics are *stitched* together. We define stitching as the process of assembling a single mosaic image from a set of oriented frames. Mosaics are stitched by placing super-resolution refined video frame data from frames $f_i \in F_v$ into a single meta-image. Frames f_i are left unwarped so that the final mosaic image may be considered as a connected set of individual frames. Mosaic construction using unwarped video frames ensures that each pixel in the final mosaic can be associated with a single frame in F_v—this allows measurement of objects that exceed the MSC's FOV.

We use orientation data from each constituent frame in a mosaic in conjunction with camera

constants to measure objects depicted in stitched mosaics. In this section, we discuss computation of inclination and azimuth angles from refined m-point data for some frame $f_i \in F_v$. Using range data δ, for f_i, height and volume measurements follow directly from inclination and azimuth measurements of points within a stitched mosaic.

We expand the geometric model depicted in Figure 3.2 to calculate the inclination and azimuth of a point within a mosaic. Figure 3.11 illustrates the expanded model used to determine the inclination or azimuth angle to an arbitrary point within a mosaic. In this discussion, we will address computation of inclination angle because the azimuth computation is conducted in the same manner.

To determine the inclination at an arbitrary point within a mosaic, we first calculate the mosaic's *angular bias*, κ. The angular bias describes the mosaic's orientation relative to horizontal (for inclination) and compass north (for azimuth). We compute the angular bias of a mosaic using a calibration point, as depicted in Figure 3.11. The calibration point is determined by the MSC as specified in the MSC data collection process: before collecting data, the MSC is pointed at an unobstructed point along the tree to record an accurate RO-video frame. Given the geometric relationships in Figure 3.11,

$$\kappa = \varphi - \lambda$$

and

$$\lambda = Tan^{-1}\left[\frac{\gamma_1}{f\eta}\right]$$

where f_l is the focal length, η is a camera constant in pixels/mm, and γ_1 is the distance, in pixels, depicted in Figure 3.11.

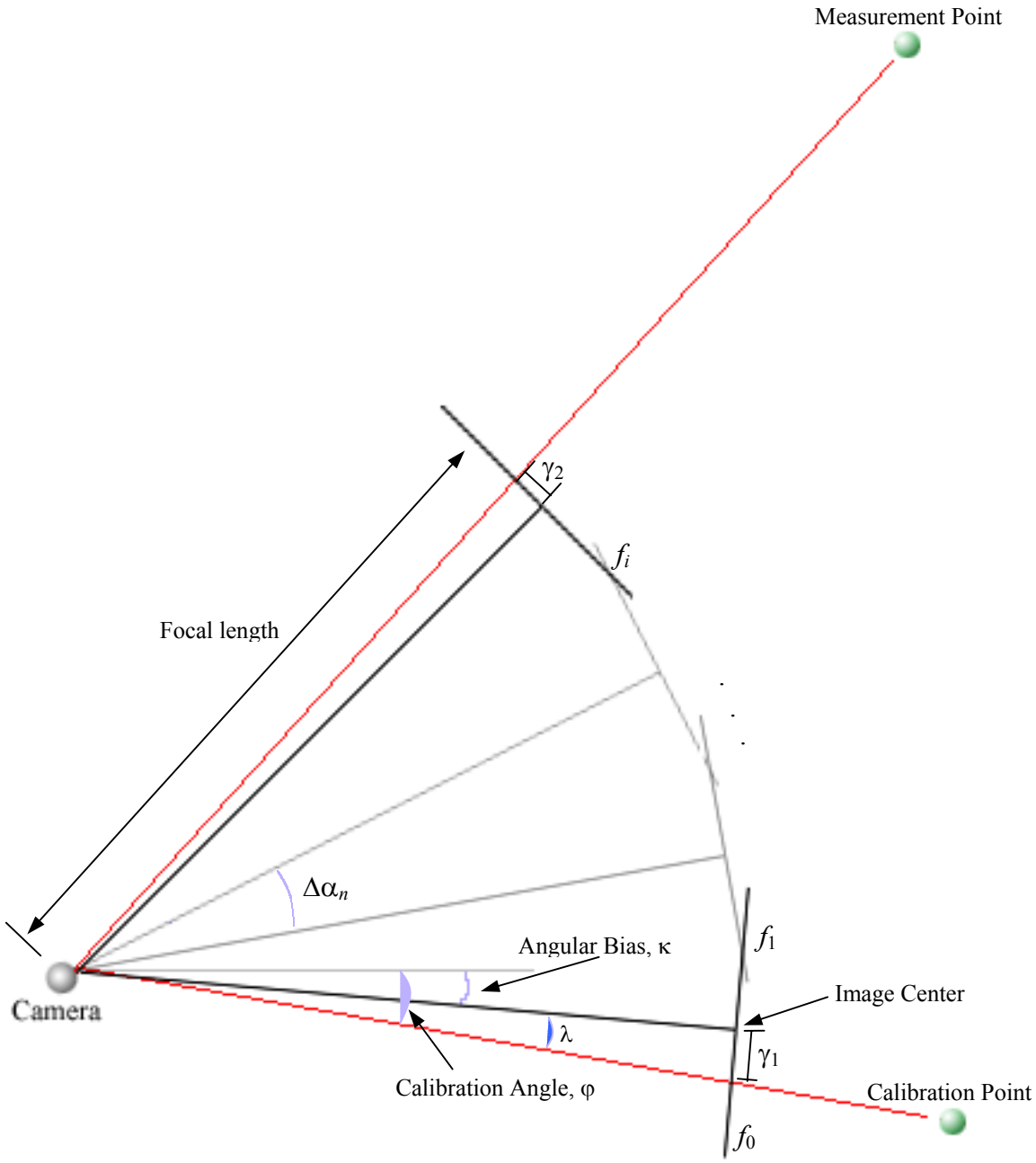

Figure 3.11: Geometric model used to compute azimuth or inclination angles in generated mosaics.

Given the mosaic's angular bias, we assume that the camera is approximately rotated about its focal point and compute the inclination at some measurement point in a mosaic as

$$\alpha_p = \sum_{n=0}^{i} \Delta\alpha_n + \mathrm{Tan}^{-1}\left[\frac{\gamma_2}{f\eta}\right] - \kappa \tag{3.23}$$

where

$$\Delta\alpha = \mathrm{Tan}^{-1}\left[\frac{\Delta a}{f\,\eta}\right] \tag{3.24}$$

where f_l is the focal length, η is a camera constant in pixels/mm, and Δa_i is the vertical component of the m-point for some mosaic pair, m_i.

CHAPTER 4 Results

4.1 Results Overview and Introduction of Terms

In this chapter, we present mosaicking results for each subset of the mosaicking process. In this section, we review the topics to be presented, introduce attributes common to mosaics generated by the system, and define terms used to judge mosaic visual quality. In Section 4.2, we present the final results of the mosaicking system (results after Steps 1-7 have been completed). In Sections 4.3 through 4.8, we present incremental results illustrating each individual step of the mosaicking system: Section 4.3 presents results after Step 1, initial raw mosaic creation; Section 4.4 presents results after Step 2, mosaic refinement through coarse-to-fine energy analysis; Section 4.5 presents results after Step 3, covariance based m-point error correction; Section 4.6 presents results after Step 4, color refinement; Section 4.7 analyses the gradient descent system of Step 5; Section 4.8 presents results after Step 6, super-resolution image refinement. Finally, we provide results that demonstrate measurement using mosaics in Section 4.9. In sections 4.3 through 4.8, we assess error using visual and numerical techniques in image dimensions; in Section 4.9, we provide quantitative analysis of mosaic precision improvement by measuring objects with refined mosaics.

Generated mosaics share common characteristics based upon the MSC and mosaicking system's design:

1) The stitching system assumes that mosaic frame sets are sequences of video frames

monotonically increasing by inclination angle. Therefore, the stitching system generates mosaics that are taller than they are wide, although the mosaicking system allows mosaics of any orientation.

2) Because images are not warped to account for scale changes and perspective distortions, the upper portions of the mosaics are slightly foreshortened.

3) Jagged edges on mosaic sides are indicative of the mosaic construction technique—unwarped image strips from video frames in F_v are de-noised using super-resolution techniques and copied into the final mosaic.

4) MSC overlay data may bleed into video frame regions that are used by the mosaicking system. Overlay data bleeding typically affects imaging sensors when MSC imaging sensitivity is high (and image contrast is low). Therefore, red hues from the cropped overlay are occasionally present in low-contrast video frames of mosaics.

4) Some mosaics, particularly those of regular structures (buildings), exhibit *mosaic drift*. Mosaic drift is a result of the MSC data collection process, not mosaicking: frame sets recorded with the MSC monopod tilted creates roll about the camera's optical axis. Global roll, as opposed to roll within mosaic pairs, causes mosaics to depict an image scene as if it were tilted. Mosaic drift is generally subtle and is reasonably avoidable through careful data collection.

Mosaics exhibit various artifacts imposed by the MSC and mosaicking system. We use these artifacts to judge mosaic visual quality:

1) *M-point artifacts.* Incorrect m-point estimation creates image discontinuities at the seams between mosaic pairs. We refer to discontinuities resulting from incorrect m-point estimation as m-point artifacts.

2) *Contrast artifacts*. Significant contrast changes within mosaic pairs may cause imaging artifacts when mosaics are created—horizontal lines are formed by contrast differences between adjacent frames in the mosaic. We refer to image artifacts caused by contrast changes as contrast artifacts.

3) *Lens artifacts*. The MSC lens system causes spherical image aberrations that create blurring and warping at the edges of video frames. Therefore, the edges of mosaics, although sharpened somewhat by super-resolution noise removal, are blurred. We refer to artifacts caused by lens aberrations—blurring and warping—as lens artifacts.

4) *CCD artifacts*. MSC imaging sensors are subject to various types of noise and corrupt pixels on an individual level during imaging. The type of CCD noise imposed on images particularly depends on MSC light sensitivity—at different levels sensitivity levels, CCD sensors create distorted pixel values. We refer to CCD errors as CCD artifacts.

5) *Ghosting artifacts.* Gradient descent estimation of affine motion may converge incorrectly. When incorrect convergence occurs, super-resolution techniques—which average mosaic pairs together within O_r—create "ghost" images. Ghost images appear as semi-transparent images overlaid onto the mosaic. We refer to artifacts generated by ghost images as ghosting artifacts.

6) *Parallax artifacts.* Mosaic frame sets contain image scenes of objects at many different depths. Objects close to the MSC can create significant parallax effects that cannot be approximated via translation or affine motion fields. We refer to image discontinuities that result from parallax effects as parallax artifacts.

4.2 Mosaicking System Results

The mosaicking system was tested on frame sets of both natural scenes and manmade structures. For each image scene, two unique frame sets were extracted from the original multisensor data set. We refer to the two extracted frame sets as the even and odd frame sets; consequently, we refer to the mosaics as even or odd, to distinguish the frame set used to generate a given mosaic. Even and odd frame sets allow us to extend the amount of data available for system testing. All data sets were processed using NSAD energy (Equation 3.1) because experimentation found NCCV energy (Equation 3.2) to be prohibitively time consuming and produce similar results to NSAD energy. Comparison between NCCV and NSAD energy analysis is presented in Section 4.4. All results presented are produced using NSAD energy unless otherwise noted.

Figures 4.1 through 4.6 are a subset of the twenty-two mosaics generated by the mosaicking system for testing. Figures 4.1 through 4.6 are typical results from the test data set. Occlusion is common in field data. Figures 4.1 and 4.3 are the upper and lower halves of mosaic 091evn (mosaic 091evn is divided into two figures to show detail). The original 091even mosaic is contiguous and formed from approximately 40 mosaic pairs. The base of mosaic 091evn is approximately 27 feet away from the MSC. Lens and CCD artifacts can be seen in mosaic 091evn. Lens artifacts create noticeable blurring along the edges of the mosaic but do not significantly impact the interior of the mosaic, even in image locations where original video frames contain lens aberrations. Blurring from lens aberrations in the interior of the mosaic can be reduced using super-resolution based image enhancement techniques as described in Section 3.7. Super-resolution image enhancement results are discussed in Section 4.8. In addition to lens

artifacts, CCD artifacts are also illustrated. CCD artifacts are particularly visible at high-contrast edges—Figure 4.2 is an enlarged portion of Figure 4.1 and illustrates CCD artifacts. We refer to the CCD artifact in region B of Figure 4.2 as *pixel bleeding*. Pixel bleeding in mosaic 091evn is visible as a blue halo along the dark region formed by the tree's branches.

Figures 4.3 and 4.4 provide illustration of the mosaics generated by the system under different lighting conditions. Figure 4.4 illustrates image quality in the center of the image—bark texture is clear. Overall, mosaic 091evn contains few artifacts.

Figure 4.1: A) Top portion of mosaic 091evn, see next page for bottom portion. The original image size is, 620x2370 pixels; recorded tree height 51 ft; location, southwest Virginia; species, Northern Red Oak. B) The area depicted in Figure 4.2.

Figure 4.2: A) Magnification of area B in Figure 4.1, to illustrate image detail. Magnification factor is 2x. B) Region depicting pixel bleeding.

Figure 4.4: Magnification of B in Figure 4.3 to show image detail. Magnification factor is 2.

Figure 4.3: A) The bottom portion of mosaic 091evn is 27 feet away from the MSC. See preceding page for top portion of mosaic 091. Original mosaic size: 620x2370 pixels. B) The magnified data illustrated in Figure 4.4.

Figures 4.5 and 4.6 illustrate more mosaics generated by the mosaicking system. Figure 4.5, mosaic 252evn, taken from approximately 50 ft away, is made from a diverse range of contrast levels. Although contrast artifacts in Figure 4.5 are reasonably subtle when compared to Figure 4.6, contrast levels vary greatly in mosaic 252evn. Contrast artifacts are clearly demonstrated by both Figure 4.5 and 4.6. In Figure 4.6, contrast artifacts are particularly significant at the top portion of the tree. *Feathering algorithms* exist which can smooth transitions between images of differing contrast; however, this thesis aims to preserve the integrity of individual video frames at the expense of contrast artifacts.

Figure 4.5 demonstrates the mosaicking system's performance on a large tree with relatively few obstructions. However, the tree in Figure 4.6 (mosaic 590evn) is heavily occluded by leaves and branches. Figure 4.6 illustrates the mosaicking system's capability—through covariance based error detection—to mosaic data with low or ambiguous information content. Portions of mosaic 590evn are completely occluded by leaves and branches; however, close inspection reveals that the tree is properly mosaiced.

Figure 4.5: Mosaic 252evn. The tree is approximately 50 ft from MSC.

Figure 4.6: Mosaic 590evn. The tree is approximately 30 feet away from the MSC.

Next, we test the mosaicking system on data collected beyond the intended application. To test the mosaicking system's ability to maintain geometric relationships (visually), we mosaic image scenes of buildings. Building frame sets pose a unique challenge, as regular structures with repeating patterns create information ambiguities that complicate mosaicking. In the most extreme cases, even human intervention cannot properly mosaic images with regular patterns—in these cases, mosaicking is an under constrained problem. Information ambiguity is discussed further in Section 4.4.

Figures 4.7-4.9 are buildings on the Virginia Tech campus. Figure 4.7—Burruss Hall—was recorded approximately 160 feet away from the building. Detail elements B, C, and D illustrate the image resolution and quality of the mosaic. Detail E illustrates a subtle ghosting artifact. When gradient descent estimation of the affine motion field is close, but not correct, ghosting artifacts create blurry regions rather than semi-transparent objects. In Figure 4.8 a moving person appears to be a ghosting artifact—a person walking into the building during data collection appears in two locations. The second location, marked B, is a product of the person's motion within the image scene during data collection. Figure 4.8 also provides visual illustration of data collection speeds—the person travels approximately 10 feet within the image scene before the camera images the data surrounding region B.

Figures 4.8 and 4.9 illustrate further mosaicking results on regular structures.

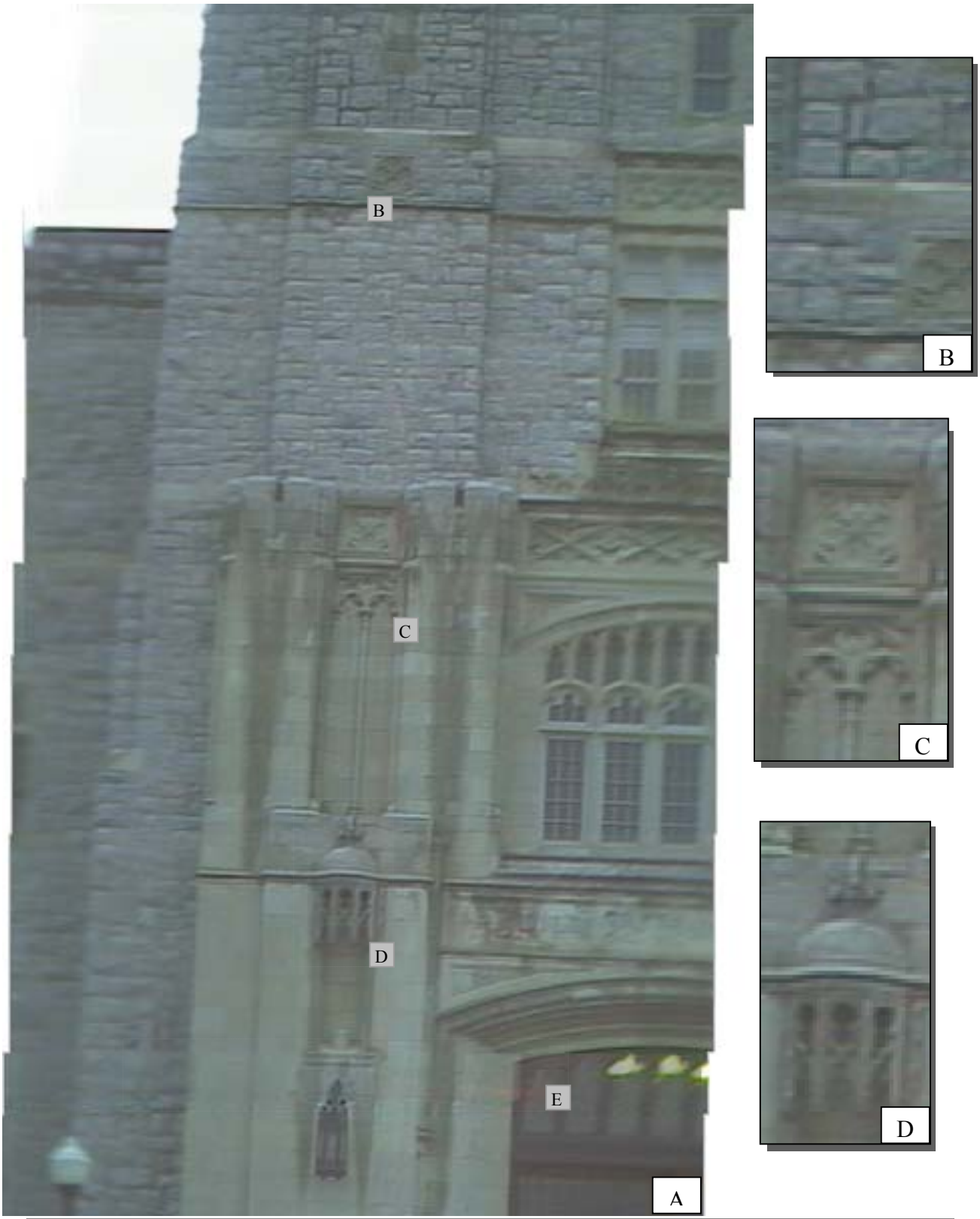

Figure 4.7: A) Mosaic 005evn from data set burr02. Slight camera roll while recording causes mosaic to be oriented slightly diagonally. Portions B, C, and D are magnified sections of A, to illustrate detail. E) Subtle ghosting artifacts create a blurring effect.

Figure 4.8: A) Mosaic 038evn from data set bur02—Burruss Hall, Virginia Tech (approximately 100 ft tall). Camera roll during data collection causes mosaic drift. B) A person walking into the building during data collection *simulates* a ghosting artifact.

Figure 4.9: Mosaic 4005evn from data set sm3—Lane Stadium, Virginia Tech, approximately 100 feet tall.

4.3 Step 1 Results – Initial Mosaic Creation

Sections 4.3 through 4.8 illustrate the mosaic refinement process at each step. We use frame sets 590evn, 091evn, 328evn, and 172evn to track mosaic improvements through the system and illustrate system functionality in Sections 4.3-4.8. See Appendix A for final results of all mosaics used in testing.

The mosaic refinement process begins by constructing a raw mosaic from frame set data. Figures 4.10-4.13 exhibit characteristics of typical mosaics generated after Step 1. Figures 4.10 and 4.11 are cropped so they may be displayed at higher resolution and provide close inspection of errors present after Step 1. Figures 4.12 and 4.13 provide examples of entire mosaic sequences. Guess m-points generated during Step 1 are subject to all system error—note that, in Figure 4.12, a guess m-point specifies zero overlap (see region A).

Figure 4.10: Mosaic 590evn, a Northern Red Oak, after initial construction of raw mosaic (Step 1). Poorly mosaiced frames are the result of error in the MSC data collection system.

Figure 4.11: Bottom portion of mosaic 091evn after initial construction of raw mosaic (Step 1). Poorly mosaicked frames are the result of error in the MSC data collection system.

Figure 4.12: Mosaic 172evn is typical output from Step 1. At region A, a guess m-point specifies a negative overlap—an erroneous value.

To provide quantitative analysis of mosaicking error, Table 4.1 lists actual mosaic points (in pixels) for each mosaic pair in Figure 4.13. Actual m-points in Figure 4.13 were determined by manually mosaicking frame set 328evn. Actual m-point coordinates x and y are listed in columns 2 and 3 of Table 4.1. The guess m-points computed in Step 1 are listed in columns 4 and 5. And, the x and y distance from the actual m-point and the guess m-point is listed in columns 6 and 7.

Although some guess m-points are more than 50 pixels from the actual m-point, all actual m-points, $m_i = (x_i, y_i)$ in Table 4.1, lie within the confusion region of mosaic pair i. Because we use $PC_r = [4,4]$ at P_3, $C_r = [4*2^4, 4*2^4] = [64,64]$. Therefore, $PC_r = [4, 4]$ at P_3 allows additional fault tolerance needed to properly mosaic 328evn.

Figure 4.13: Mosaic 328evn is typical output from Step 1 and illustrates mosaic drift.

Table 4.1: Comparison between actual m-points and computed m-points in mosaic 328evn after Step 1. Actual m-points were determined by mosaicking the frame set manually. All units are in pixels.

Mosaic 328evn	Actual M-Point		Step 1 M-Point		Step 1 Absolute Error	
Pair #	*x*	*y*	*x*	*y*	*x*	*y*
328-330	12	56	-17	56	29	0
330-332	3	76	-8	74	11	2
332-334	15	126	-17	82	32	44
334-336	19	92	21	78	2	14
336-338	13	88	13	87	0	1
338-340	1	72	17	78	16	6
340-342	9	106	-17	78	26	28
342-344	14	112	17	78	3	34
344-346	12	75	13	87	1	12
346-348	25	72	-30	74	55	2
348-350	12	83	13	82	1	1
350-352	2	121	8	82	6	39
352-354	14	55	4	87	10	32
354-356	9	91	21	74	12	17
356-358	1	109	-4	87	5	22
358-360	12	93	-8	78	20	15
360-362	20	81	8	82	12	1
362-364	31	38	34	69	3	31
364-366	2	112	-4	78	6	34
366-368	2	99	-4	87	6	12
368-370	10	71	-26	74	36	3
370-372	5	83	61	91	56	8
372-374	11	113	13	82	2	31
374-376	-2	76	0	78	2	2
376-378	30	45	39	74	9	29
378-380	13	103	21	78	8	25
380-382	9	90	39	87	30	3
382-384	7	81	4	87	3	6
384-386	6	101	21	78	15	23
386-388	20	73	13	87	7	14
388-390	10	74	4	74	6	0
390-392	10	85	26	69	16	16
392-394	7	89	0	78	7	11
394-396	7	78	26	78	19	0
396-398	24	90	8	69	16	21
Average					13.9	15.4
Variance					200.3	172.3

4.4 Step 2 Results – Coarse-to-fine Energy Analysis

In Step 2, mosaic accuracy is refined using coarse-to-fine energy minimization. We experimented with two energy metrics, NSAD and NCCV energy. NCCV energy was found to be prohibitively time consuming and produce similar results to NSAD energy. Testing indicates that NCCV mosaicking is approximately 8.3 times slower than NSAD mosaicking: for a 35 frame frame set, NSAD mosaicking took 37 seconds on a Pentium-III 450 MHz machine and NCCV mosaicking took 308 seconds (5 min, 8 sec). Because NSAD energy produced highly accurate results across the test data sets, we compared NSAD and NCCV energy to determine which metric is most reliable.

Figure 4.14 illustrates mosaicking results produced with NSAD and NCCV energy, respectively. NSAD mosaicking errors are present in mosaic A of Figure 4.14; similar NCCV mosaicking errors are present in mosaic B. NCCV mosaics were compared against NSAD mosaics for a subset of the full test data set. Comparison suggests that NCCV and NSAD similarly reliable. Figure 4.14 is representative of the similarity between NCCV and NSAD mosaicking results. Regions C, D, E, and F mark similar errors similar in NSAD and NCCV mosaicking. Because NCCV and NSAD mosaicking were similarly accurate and similarly reliable, NSAD mosaicking is considered optimal for its speed.

Henceforth, all presented mosaics were produced using NSAD energy. Figures 4.15-4.18 illustrate NSAD mosaicking results produced by Step 1 of the mosaicking process. We use mosaics 590evn, 091evn, 172evn, and 328evn as presented in Section 4.3 to illustrate the mosaic

refinement process. Despite sporadic mosaicking errors (i.e. errors in mosaic 590evn, Figure 15), mosaic visual accuracy has been visibly improved in all cases.

Figure 4.14: NSAD-NCCV performance comparison using data set 172evn. NSAD analysis was found to produce results similar to NCCV analysis. Mosaics A and B are representative of NSAD and NCCV performance in error conditions. Regions C, D, E, and F mark common error regions between both NSAD

Figure 4.15: A) Mosaic 590evn after Step 2, coarse-to-fine NSAD refinement. B) Mosaicking error 1—tree stem does not line up within the mosaic pair. C) Mosaicking error 2—tree stems are clearly discontinuous.

Figure 4.16: A) Bottom portion of mosaic 091evn after coarse to fine NSAD refinement. B) A subtle mosaicking error.

Table 4.2 allows quantitative analysis of m-point improvement for mosaic 328evn. After Step 2, all m-points are within 2 pixels of the actual m-point determined by hand mosaicking 328evn. When rotations and distortions are present within mosaic pairs, m-point accuracy cannot be analyzed by hand mosaicking images; therefore, computed m-points that lie within 2 pixels of the hand-computed actual m-points in Table 4.2 may be argued as correct, if mosaic pairs contain rotations and distortions. M-point error is significantly improved through coarse-to-fine NSAD energy mosaicking. Mosaicking results from Step 2 are typically accurate within a few pixels of the actual m-point. When errors occur, m-points are typically highly inaccurate and cause noticeable visual discontinuities. Mosaicking error within Step 1 is primarily a result of poor RO-video correlation, although information ambiguities, discussed in Section 4.5, cause error as well.

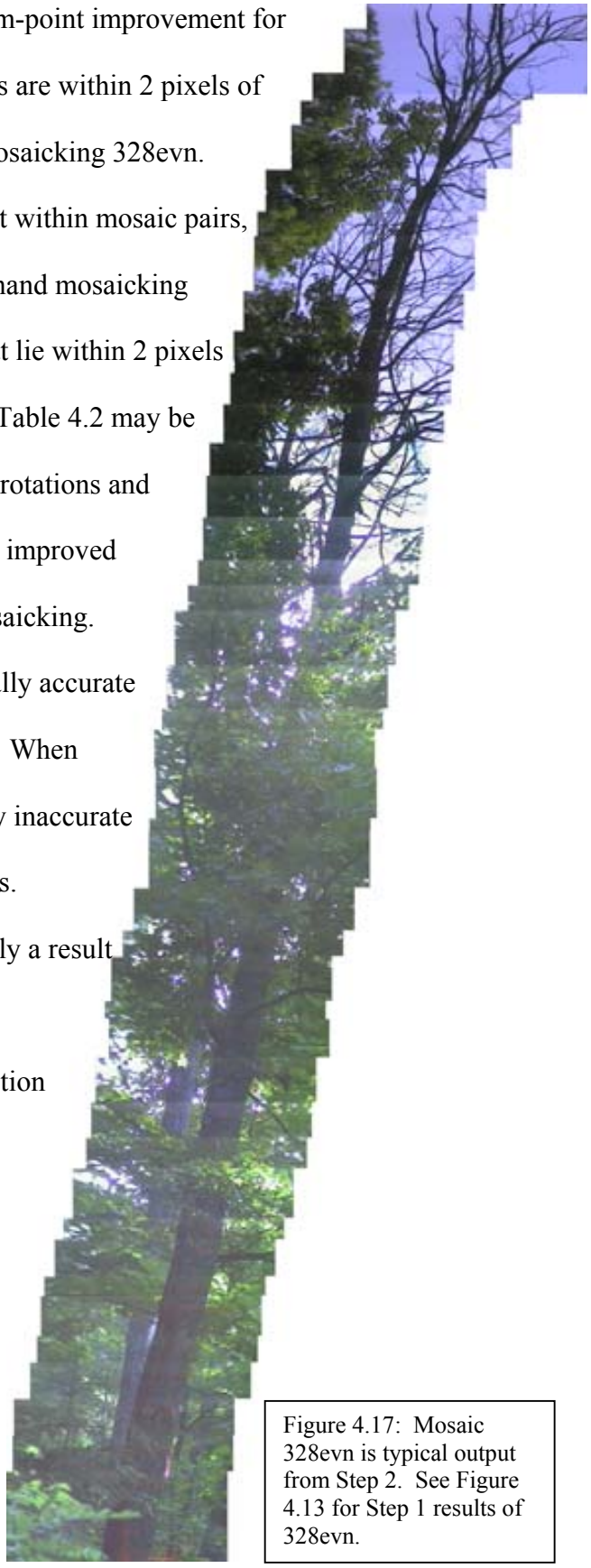

Figure 4.17: Mosaic 328evn is typical output from Step 2. See Figure 4.13 for Step 1 results of 328evn.

Figure 4.18: A) Mosaic 172evn—typical output from Step 2. B) Mosaicking error #1 (see Figure 4.14 for detail). C) Mosaicking error #2. See Figure 4.12 for Step 1 results of 172evn.

Table 4.2: Quantitative analysis of mosaicking accuracy in mosaic 328evn after NSAD mosaicking in Step 1 and Step 2. Actual m-points were derived by hand-mosaicking frame set 328evn before computer mosaicking with Step 1 and Step 2. M-point accuracy is significantly increased.

Mosaic 328evn	Actual M-Point		Step 1 M-Point		Step 1 Error		Step 2 M-point		Step 2 Error	
Pair #	x	y	x	y	x	y	x	y	x	y
328-330	12	56	-17	56	29	0	12	56	0	0
330-332	3	76	-8	74	11	2	3	77	0	1
332-334	15	126	-17	82	32	44	14	126	1	0
334-336	19	92	21	78	2	14	19	92	0	0
336-338	13	88	13	87	0	1	13	89	0	1
338-340	1	72	17	78	16	6	1	72	0	0
340-342	9	106	-17	78	26	28	10	106	1	0
342-344	14	112	17	78	3	34	15	112	1	0
344-346	12	75	13	87	1	12	12	76	0	1
346-348	25	72	-30	74	55	2	27	72	2	0
348-350	12	83	13	82	1	1	12	83	0	0
350-352	2	121	8	82	6	39	2	121	0	0
352-354	14	55	4	87	10	32	14	56	0	1
354-356	9	91	21	74	12	17	9	91	0	0
356-358	1	109	-4	87	5	22	1	110	0	1
358-360	12	93	-8	78	20	15	13	91	1	2
360-362	20	81	8	82	12	1	19	81	1	0
362-364	31	38	34	69	3	31	31	38	0	0
364-366	2	112	-4	78	6	34	3	112	1	0
366-368	2	99	-4	87	6	12	2	98	0	1
368-370	10	71	-26	74	36	3	10	71	0	0
370-372	5	83	61	91	56	8	6	84	1	1
372-374	11	113	13	82	2	31	11	113	0	0
374-376	-2	76	0	78	2	2	-2	75	0	1
376-378	30	45	39	74	9	29	30	45	0	0
378-380	13	103	21	78	8	25	14	104	1	1
380-382	9	90	39	87	30	3	8	90	1	0
382-384	7	81	4	87	3	6	8	81	1	0
384-386	6	101	21	78	15	23	7	101	1	0
386-388	20	73	13	87	7	14	20	73	0	0
388-390	10	74	4	74	6	0	10	73	0	1
390-392	10	85	26	69	16	16	11	86	1	1
392-394	7	89	0	78	7	11	7	88	0	1
394-396	7	78	26	78	19	0	7	78	0	0
396-398	24	90	8	69	16	21	24	89	0	1
Average					13.9	15.4			0.4	0.4
Variance					200	172			0.3	0.3

4.5 Step 3 Results – Covariance-based Error Detection and Removal

Mosaicking errors in Step 2 result from RO-video correlation errors or information ambiguities within a mosaic pair. If RO-video correlation errors are present, actual mosaic points can lie outside of the confusion region and will not be examined as possible solutions. When actual m-points lie outside of the confusion region, m-points are computed as the best value within the confusion region—this value is often far from the actual m-point. If information ambiguities exist in a mosaic pair, the information contained in the two images does not provide an unambiguous actual m-point. Some mosaic pairs are ambiguous even when inspected by hand—these cases will generate errors in any computer vision system.

Covariance analysis of energy space geometry identifies mosaicking errors that occur when orientation data is incorrect or when information ambiguities exist. Figure 4.19 shows two energy spaces: the first energy space contains a well-defined minimum that specifies an accurate actual m-point; the second energy space contains an ambiguous trough that is associated with diagonal ambiguity found in mosaic pair 86-87 in mosaic 001odd. Bright blue indicates areas of high NSAD energy; dark blue indicates areas of low NSAD energy.

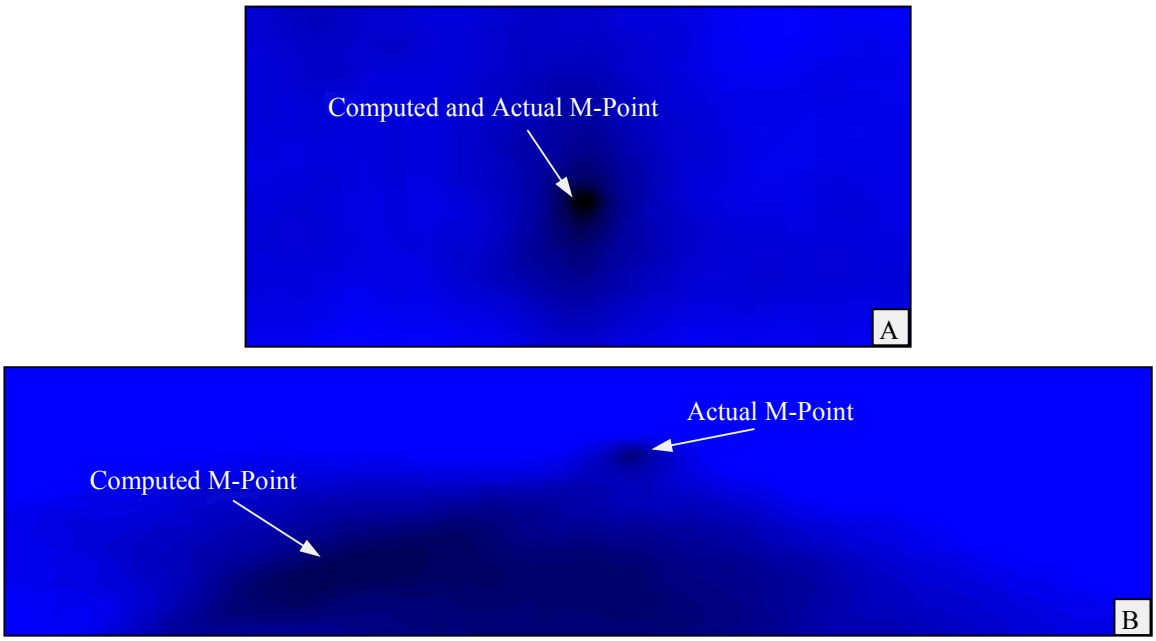

Figure 4.19: Two energy spaces generated from mosaic pairs from mosaic 001evn. Dark blue indicates areas of low NSAD energy. A) An energy space with an accurate computed m-point. B) An energy space from a mosaic pair containing information ambiguity—two local minima exist. The computed m-point and actual m-points do not correspond.

Figure 4.20 shows mosaic pair 086-087 from mosaic 001evn and the source of diagonal ambiguity in energy space B of Figure 4.19. A diagonal branch is the only source of mosaicking information in mosaic pair $\{f_{86}, f_{87}\}$ and is dominated, in NSAD computation, by an ambiguous leaf canopy with few features. Figure 4.21 illustrates the effect of information ambiguity on the shape of the energy space for $\{f_{86}, f_{87}\}$. In Figure 4.21, the shape of the energy space is illustrated by white contour lines. The computed m-point is in the center of a large, diagonal, depression. The shape of the energy space around the computed m-point can be related to the information available in f_{86} and f_{87}: the diagonal shape of the energy space corresponds with the m-points that align the thickest branch in Figure 4.20.

Finally, Figure 4.22 shows the alignment of mosaic pair 086-087 at the computed mosaic point.

The large diagonal branch in $\{f_{86}, f_{87}\}$ appears to be properly aligned; however, closer inspection of Figure 4.22 reveals subtle clues that the computed mosaic point is incorrect. The features that suggest that f_{86} and f_{87} have been improperly mosaiced are not strong enough to overcome the noise generated by the ambiguous leaf canopy in NSAD computation. Fortunately, cases of information ambiguity are rare within data sets of natural scenes. Most significant mosaicking errors are a result of poorly correlated orientation and frame data.

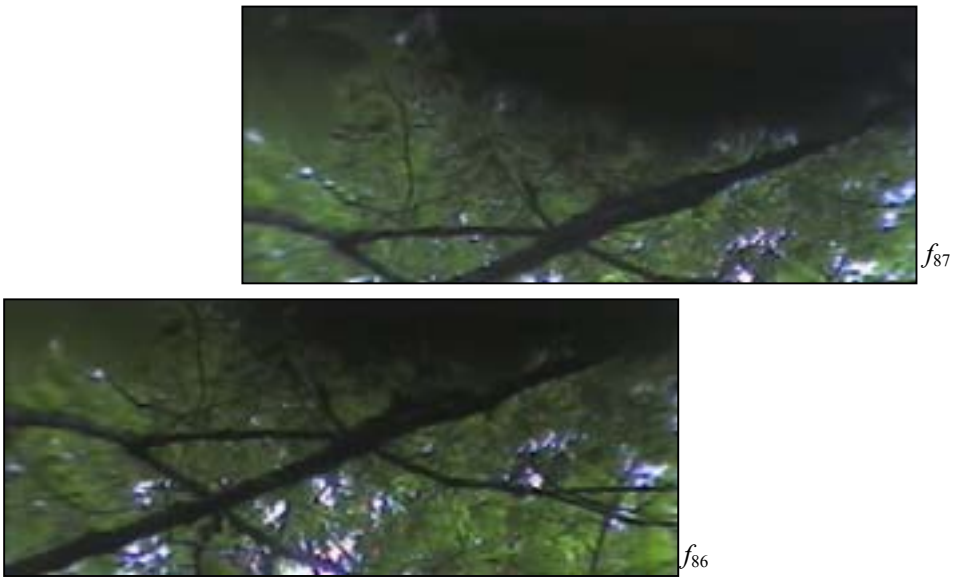

Figure 4.20: Frames f_{86} and f_{87} exhibit information ambiguity. Information ambiguity occurs when there is little visual information per pixel. A low frequency canopy of leaves dominates frames f_{86} and f_{87}; a small branch contributes little mosaicking information to the NSAD sum, by percent.

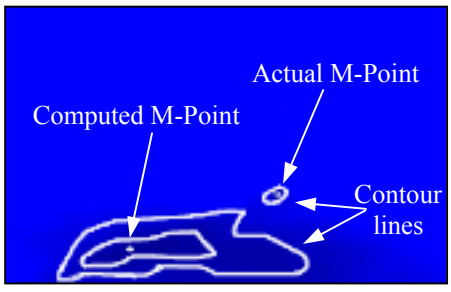

Figure 4.21: The confusion region associated with mosaic pair $\{f_{86}, f_{87}\}$. White contour lines are drawn to illustrate the shape of the energy space around the computed and actual m-points.

Figure 4.22: Frame f_{87} is positioned at the computed m-point in the energy space, mosaicking frames f_{86} and f_{87} together.

Figures 4.23-4.26 present results of covariance based mosaic error detection and removal. We use mosaics 590evn, 091evn, 328evn, and 172evn to illustrate the mosaic refinement process. Mosaics 590evn, 091evn, and 172evn contained mosaicking errors prior to covariance based error detection; however, all mosaicking errors were removed by adding intermediate frames. In Figures 4.23-4.26, boxed regions mark the location of errors present after Step 2, but corrected in Step 3. Figure 4.25 illustrates the effect of covariance based error detection on mosaics without errors—Step 3 does not modify mosaics that do not contain error conditions.

Figure 4.23: A) results of mosaicking 590evn after Step 3, covariance based error detection and removal. The errors present at locations B and C after Step 2 have been removed. See Figure 4.15 for Step 2 output.

Figure 4.24: A) results of mosaicking 091evn after Step 3, covariance based error detection and removal. The error present at location B has been removed after Step 3. See Figure 4.16 for Step 2 output.

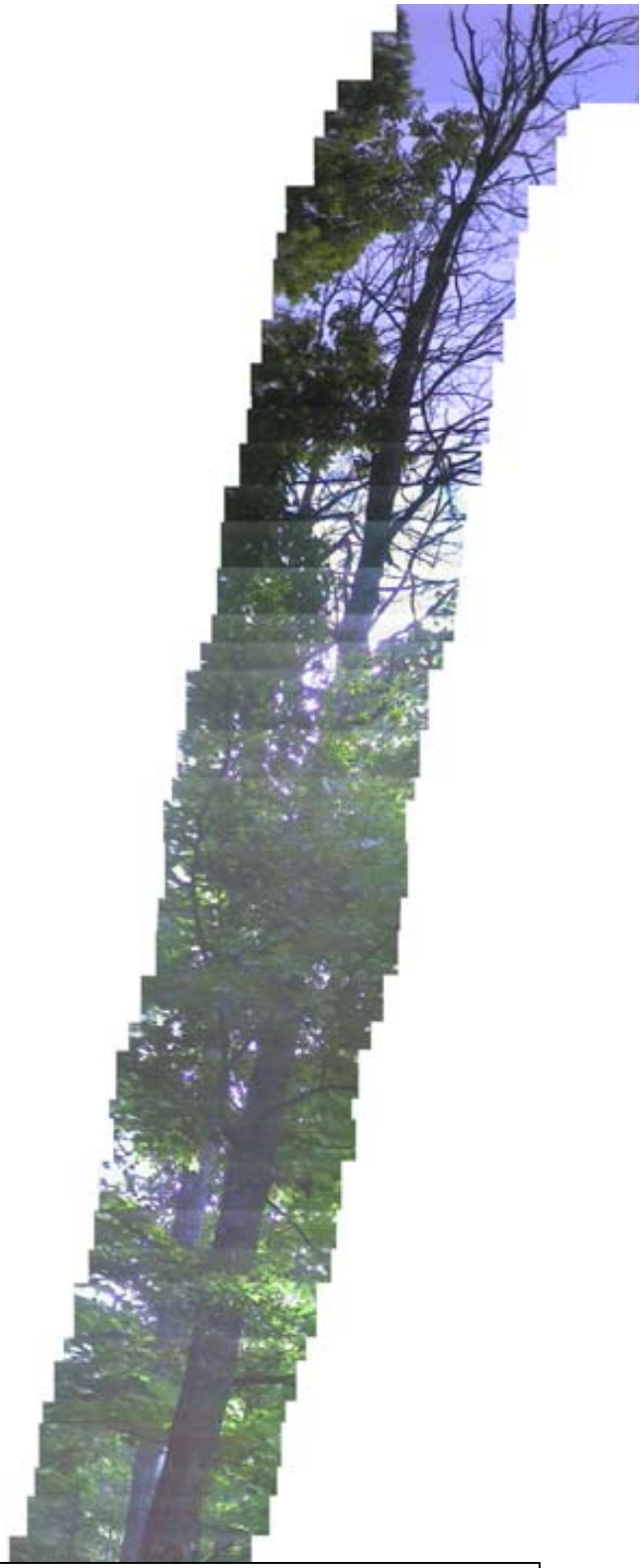

Figure 4.25: Mosaic 328evn after m-point error detection and removal. No visible errors were present in 328evn after Step 2 and covariance analysis does not impose new errors. See Figure 4.17 for Step 2 mosaicking results on 328evn.

Figure 4.26: A) mosaic 172evn after Step 3 error detection. Errors once at B and C have been detected and removed using covariance analysis. See Figure 4.18 for Step 2 mosaicking results.

4.6 Step 4 Results – Color Analysis

Color analysis was found to have subtle effects on individual m-point locations that create global mosaicking effects—subtle m-point refinements throughout the mosaic cause global mosaic dimensions to vary. Table 4.3 compares mosaicking accuracy after Step 3 with mosaicking accuracy after

Step 4. Columns 6 and 7 in Table 4.3 list mosaicking error after coarse-to-fine mosaicking and columns 10 and 11 list mosaicking error after color analysis in Step 4—these results are representative of mosaic improvements typically made during Step 4.

Subtle m-point modifications made by color analysis are difficult to test for correctness. Because actual m-points (listed in column 2 of Table 4.3) are hand derived, it may be argued that m-point locations 1 to 2 pixels away are correct and that the hand-generated m-points are erroneous. Alternative mosaic evaluation methods are needed to determine if the global mosaicking effects of color analysis are beneficial to mosaic accuracy. Global mosaic accuracy testing requires the use of mosaics to measure objects in object dimensions, not pixels; however, current camera calibration data for the MSC is not accurate enough to evaluate Step 4 results in object dimensions. Section 4.9 discusses mosaicking measurement accuracy and camera calibration issues.

Table 4.3: Quantitative comparison of mosaicking accuracy after Step 3 and Step 4 in mosaic 328evn. Actual m-points were derived by hand-mosaicking frame set 328evn before computer mosaicking with Step 3 and Step 4. Step 4 illustrates subtle mosaic changes after Step 4. Mosaic changes are highlighted in red. See Table 4.2 for results from Step 1 and Step 2.

Mosaic 328evn	Actual M-Point		Step 3 M-Point		Step 3 Absolute Error		Step 4 M-point		Step 4 Absolute Error	
Pair #	x	y	x	y	x	y	x	y	x	y
328-330	12	56	12	56	0	0	12	56	0	0
330-332	3	76	3	77	0	1	3	77	0	1
332-334	15	126	14	126	1	0	15	126	0	0
334-336	19	92	19	92	0	0	19	92	0	0
336-338	13	88	13	89	0	1	13	89	0	1
338-340	1	72	1	72	0	0	1	72	0	0
340-342	9	106	10	106	1	0	10	106	1	0
342-344	14	112	15	112	1	0	15	112	1	0
344-346	12	75	12	76	0	1	12	72	0	3
346-348	25	72	27	72	2	0	27	72	2	0
348-350	12	83	12	83	0	0	12	83	0	0
350-352	2	121	2	121	0	0	2	121	0	0
352-354	14	55	14	56	0	1	14	56	0	1
354-356	9	91	9	91	0	0	9	91	0	0
356-358	1	109	1	110	0	1	1	110	0	1
358-360	12	93	13	91	1	2	13	91	1	2
360-362	20	81	19	81	1	0	19	81	1	0
362-364	31	38	31	38	0	0	31	39	0	1
364-366	2	112	3	112	1	0	3	112	1	0
366-368	2	99	2	98	0	1	2	99	0	0
368-370	10	71	10	71	0	0	10	71	0	0
370-372	5	83	6	84	1	1	6	84	1	1
372-374	11	113	11	113	0	0	11	113	0	0
374-376	-2	76	-2	75	0	1	-2	75	0	1
376-378	30	45	30	45	0	0	30	45	0	0
378-380	13	103	14	104	1	1	14	104	1	1
380-382	9	90	8	90	1	0	8	90	1	0
382-384	7	81	8	81	1	0	8	82	1	1
384-386	6	101	7	101	1	0	7	101	1	0
386-388	20	73	20	73	0	0	20	73	0	0
388-390	10	74	10	73	0	1	10	73	0	1
390-392	10	85	11	86	1	1	11	86	1	1
392-394	7	89	7	88	0	1	7	88	0	1
394-396	7	78	7	78	0	0	7	78	0	0
396-398	24	90	24	89	0	1	24	89	0	1
Average					0.40	0.43			0.37	0.5
Variance					0.30	0.31			0.29	0.49

4.7 Step 5 Results –Affine Motion Estimation

Affine motion estimation in Step 5 allows the mosaicking system to determine m-point locations when mosaic pairs contain distortions such as perspective distortions and rotations. Rotations within a mosaic pair, for example, are common near the tops of trees where steady data collection is difficult using a monopod. M-point refinement was found to resolve m-points to sub-pixel locations. Although sub-pixel accuracy expedites super-resolution based image refinement (see Section 4.8), careful camera calibration—beyond the scope of this thesis—is needed to rigorously evaluate mosaic accuracy improvements resulting from affine motion estimation. Therefore, we discuss error conditions that may result during affine motion estimation in this section.

Affine motion estimation errors occur when the gradient matching system converges to an erroneous local minimum within the evaluation function (Equation 3.11). Erroneous convergence typically makes subtle modifications to m-point data; therefore, m-point errors resulting from errors in the affine motion field are subtle. Unfortunately, errors in the affine motion field have a significant impact during Step 6, super-resolution based image refinement. In this section, we use super-resolution results to illustrate errors that can occur during affine motion estimation.

Figure 4.27 demonstrates an error in the gradient matching system. In Figure 4.27, the gradient matching system converged to an incorrect local minimum, thus failing to properly estimate the affine motion field. Because the computed affine motion field is incorrect, super-resolution

based image refinement produces blurry results when fusing images in a mosaic pair. Super-resolution based refinement is applied to image B in Figure 4.27. Image A in figure 4.27 shows video data prior to super-resolution effects. Poor affine motion estimation causes image blurring.

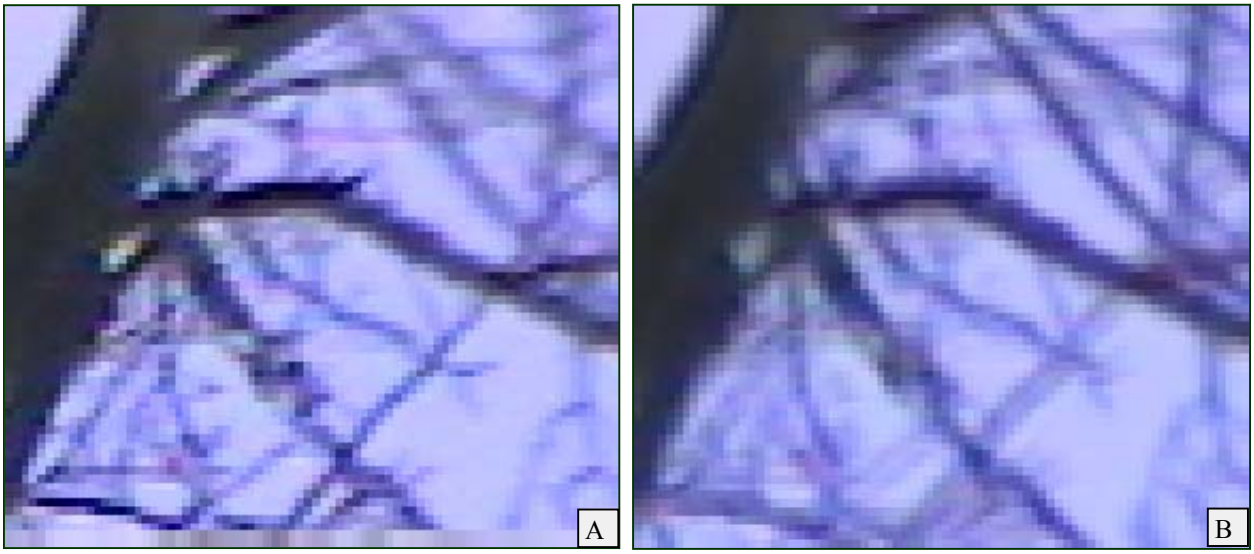

Figure 4.27: Images A and B illustrate a typical gradient descent convergence error from mosaic 328evn. A) Mosaic section after Step 5. B) Mosaic section after Step 6 (merging an incorrectly warped mosaic pair).

Ghosting artifacts are a direct result of poor affine motion estimation. Figure 4.28 illustrates ghosting effects that occur when the gradient matching system erroneously converges. Figures 4.29 and 4.30 provide additional detail of mosaics that contain errors caused by erroneous affine motion fields.

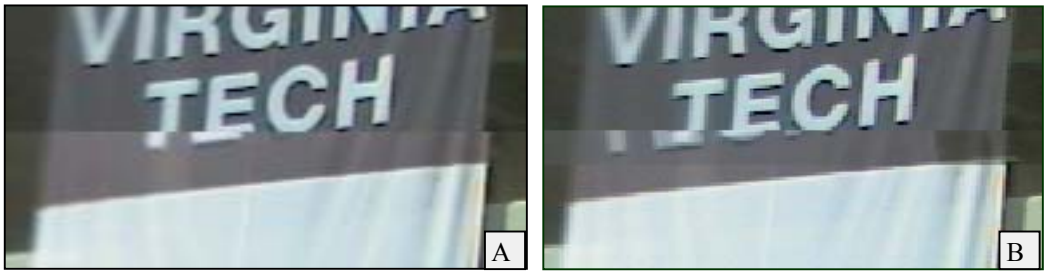

Figure 4.28: A) A section from mosaic 4039evn after Step 1 (no affine motion error has occurred). B) A section from mosaic 4039evn after Step 6 (incorrect gradient descent convergence). Merged frames after Step 6 do not truly overlap using the computed affine motion field and a ghosting error is created.

Figure 4.29: A) A portion of mosaic 4039 odd after Step 1 (no affine motion errors are present). B) A portion of mosaic 4039odd after Step 6. In B, blurring is evident within the overlap region. Blurring results from incorrect affine motion estimation.

Figure 4.30: A) A portion of mosaic 001evn after Step 1 (no affine motion estimation errors are present). B) A portion of mosaic 001evn after Step 6. in B, blurring is evident in the top half of the image where an error occured during affine motion estimation.

4.8 Step 6 Results – Super-Resolution Based Image Refinement

In this section, we discuss super-resolution results obtained from accurate affine motion fields. Given an accurate affine motion field for a mosaic pair, Step 6 removes or reduces common imaging artifacts such as contrast arifacts, lens artifacts, and CCD artifacts. Figure 4.31 illustrates a typical mosaic with contrast artifacts that were reduced after Step 6. Contrast artifacts can be reduced by Step 6 because mosaic pairs are fused together by averaging within the overlap region (See section 3.7 for detail). Because we do not implement image merging operations such as feathering, the severity of contrast artifacts depends greatly upon the amount of contrast change within a mosaic pair. Step 6 is able to significantly reduce or remove moderate contrast artifacts; however, sudden contrast changes cannot be smoothed by super-resolution image refinement. See Figure 4.23 for an example of an image sequence that contains severe contrast changes.

Figure 4.32 illustrates a typical mosaic with lens artifacts that were reduced after Step 6. Image A in Figure 4.32 illustrates spherical lens artifacts that are present at the edges of video frames recorded with the MSC. Because an image scene is recorded in different regions of f_i and f_{i+n} in a mosaic pair $m_i = \{f_i, f_{i+n}\}$, lens artifacts differ between f_i and f_{i+n}. Thus, if some regoin x of an image scene in f_i is corrupted by lens artifacts, x may not be corrupted in f_{i+n}. If x is not corrupted in f_{i+n}, Step 6 can reduce the impact of lens artifacts within m_i. In Figure 4.32, lens artifacts are removed because x is not corrupted in both f_i and f_{i+n}.

Figure 4.31: A) A portion of Mosaic 500evn prior to Step 6. B) Mosaic 500evn after Step 6. Contrast lines apparent in image A are significantly reduced in Image B. Data is shown magnified at 1.5x.

Figure 4.32: A) Mosaic 091evn contains significant lens artifacts (spherical blurring) at edges prior to Step 6. B) Mosaic 091evn after Step 6. Step 6 significantly reduces the lens artifacts present in image A.

Figure 4.33 illustrates a mosaic with CCD artifacts that were reduced during Step 6. Image A in Figure 4.33 contains both spectral effects and pixel bleeding. Because the errors at C and D in Figure 4.33 are not present in both f_i and f_{i+n} in the mosaic pair, m_i, CCD artifacts are removed by super-resolution based image refinement in Step 6.

Figure 4.33: A) Mosaic 252evn contains CCD artifacts at high-contrast edges prior to Step 6. B) Mosaic 251evn after Step 6. CCD artifacts, pixel bleeding and spectral effects are reduced in image B. Pixel bleeding at C and D is reduced after Step 6.

Results of Step 6 are highly dependant upon the selection of the free parameter A in Equation 3.20 and 3.21 used to compute the edge pixel values E_i and E_{i+1}. Selection of A determines the effect of cubic convolution on the up-sampled images. As A decreases, images become sharper. We experimented with $-20 \leq A \leq -0.5$ to determine what value of A produces best results in Step 6. Figure 4.34 illustrates a portion of a mosaic which has been refined using different values of A. As the trial choice in Step 6, we use $A = -11$. As A decreases, high frequency edges become more prevalent and create noise. At $A = -20$, image sharpening is detrimental—noise edges are enhanced, particularly in tree canopies.

Figure 4.34: Testing of *A*, a free parameter in cubic convolution. At *A* = -0.5, mosaic 001evn is much blurrier than at *A* = -9.

4.9 Measurement Results

In this section, we give quantitative analysis of mosaic accuracy improvement and demonstrate that generated mosaics facilitate size estimation. To test mosaic precision and accuracy, we use inclination data gathered with a Barr and Stroud (B&S) inclinometer, accurate to ±0.06°.. Four unique trees were recorded and mosaicked using the MSC and the mosaic refinement process. Next, the B&S inclinometer was taken into the field to record inclination angles of measurement points along the trunk of the actual tree. Thus, we have an independent metric for the correct dimension of the mosaiced trees. Using an independent metric of tree dimension, we compare mosaicking computations for accuracy.

In Tables 4.4 through 4.7, angles are computed using Equation 3.23, assuming the imaging model illustrated in Figure 3.11. In each table, column 1 lists the measurement point number within that mosaic, column 2 lists the inclination angle measured using the B&S inclinometer, column 3 lists the inclination computed from the mosaic after Step 1 of the refinement process, column 4 lists the amount of error between angles computed after Step 1 and angles measured with the B&S inclinometer, column 5 lists the angles computed after Step 7, column 6 lists the amount of error between angles computed after Step 7 and angles measured with the B&S inclinometer, and Column 7 lists the percentage change in mosaicking error between Step 1 and Step 7. We find the average accuracy improvement to be 86%.

Table 4.4: Numerical validation of the mosaicking refinement process for mosaic 8090. Computed mosaicking angles after Steps 1 and 7 are compared to B&S inclination measurements. The average mosaic accuracy improvement is 92.63%

Mosaic 8090 Point #	Correct Inclination (B&S)	Inclination After Step 1	Error After Step 1	Inclination After Step 7	Error After Step 7	Accuracy Improvement (%)
1	11.9°	15.2°	3.3°	12.7°	0.8°	75.56
2	14.4°	18.7°	4.3°	14.7°	0.3°	93.02
3	27.2°	32.7°	5.5°	27.3°	0.1°	98.18
4	41.3°	49.2°	7.9°	41.6°	0.3°	96.20
5	53.3°	61.6°	8.3°	53.3°	0.0°	100

Table 4.5: Numerical validation of the mosaicking refinement process for mosaic 8173. Computed mosaicking angles after Steps 1 and 7 are compared to B&S inclination measurements. The average mosaic accuracy improvement is 93.22%

Mosaic 8173 Point #	Correct Inclination (B&S)	Inclination After Step 1	Error After Step 1	Inclination After Step 7	Error After Step 7	Accuracy Improvement (%)
1	5.1°	7.4°	2.3°	5.0°	0.1°	95.65
2	15.1°	20.6°	5.5°	15.0°	0.1°	98.18
3	28.5°	35.7°	7.2°	27.6°	0.9°	87.5
4	39.1°	47.3°	8.3°	38.4°	0.7°	91.57

Table 4.6: Numerical validation of the mosaicking refinement process for mosaic 8000. Computed mosaicking angles after Steps 1 and 7 are compared to B&S inclination measurements. The average mosaic accuracy improvement is 75.41%

Mosaic 8000 Point #	Correct Inclination (B&S)	Inclination After Step 1	Error After Step 1	Inclination After Step 7	Error After Step 7	Accuracy Improvement (%)
1	14.8°	21.0°	6.2°	12.7°	2.1°	66.13
2	15.6°	22.2°	6.6°	13.8°	1.8°	72.73
3	31.9°	38.6°	6.6°	30.4°	1.6°	75.76
4	41.8°	53.3°	11.5°	40.3°	1.5°	86.96
5	50.4°	60.2°	9.8°	48.1°	2.4°	75.51

Table 4.7: Numerical validation of the mosaicking refinement process for mosaic 8207. Computed mosaicking angles after Steps 1 and 7 are compared to B&S inclination measurements. Points 5, 6, and 7 contain 1° of error because a mosaicking error occurs in the mosaic. The average mosaic accuracy improvement is 82.96%

Mosaic 8207 Point #	Correct Inclination (B&S)	Inclination After Step 1	Error After Step 1	Inclination After Step 7	Error After Step 7	Accuracy Improvement (%)
1	4.2°	8.2°	4.0°	4.8°	0.6°	85
2	15.8°	20.5°	4.6°	14.0°	1.8°	60.87
3	28.9°	36.7°	7.8°	28.5°	0.4°	94.87
4	29.6°	37.9°	8.3°	29.3°	0.3°	96.39
5	36.4°	45.2°	8.8°	34.7°	1.7°	80.68
6	40.8°	48.8°	8.0°	38.9°	1.9°	76.25
7	51.5°	59.0°	7.5°	50.5°	1.0°	86.67

Tables 4.4 - 4.7 provide indication of the measurement capabilities of generated mosaics; however, camera calibration issues obfuscate mosaic precision improvements. While the inclination measures produced by the B&S inclinometer are accurate to 0.06°, the current MSC data collection process and camera calibration constants restrict mosaic computations to much lower levels of precision. The precision after Step 6 may exceed the levels suggested by the errors in Tables 4.4 through 4.7—camera and system calibration impose error on all mosaic angle calculations.

Figures 4.35 - 4.38 illustrate the data used to derive the values in Tables 4.4 - 4.7. Red crosshairs have been placed at each measurement point in each mosaic. B&S inclinometer measurements were taken to the locations marked by the crosshairs and the mosaics were used to compute the inclination angle of the crosshairs. Additionally, crosshair locations provide visual interpretation of mosaic accuracy improvements.

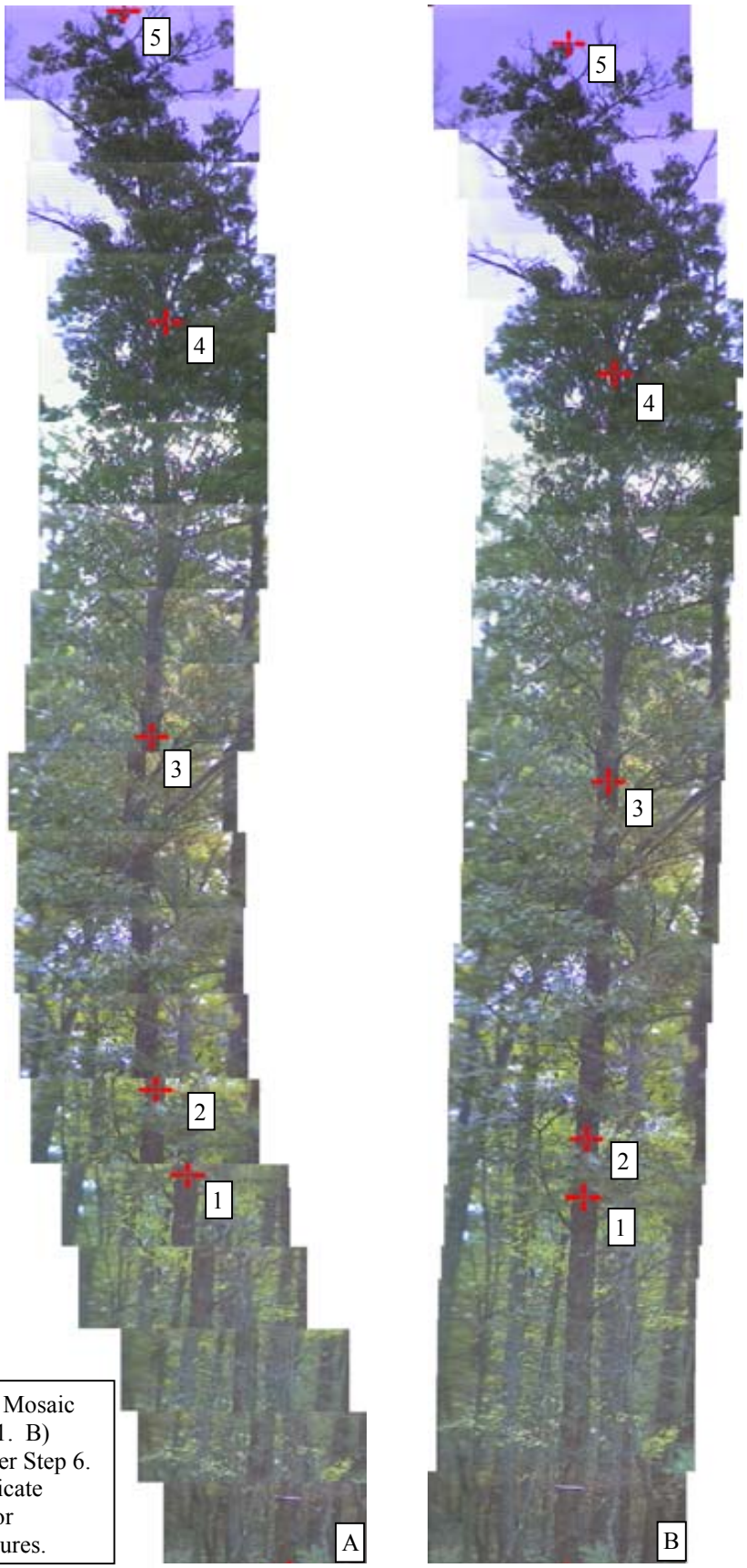

Figure 4.35: A) Mosaic 8090 after Step 1. B) Mosaic 8090 after Step 6. Markers 1-5 indicate locations used for inclination measures.

Figure 4.36: A) Mosaic 8173 after Step 1. B) Mosaic 8173 after Step 6. Markers 1-4 indicate the locations used for inclination measures.

Figure 4.37: A) Mosaic 8000 after Step 1. B) Mosaic 8000 after Step 6. Markers 1-5 indicate the locations used for inclination measures.

Figure 4.38: A) Mosaic 8027 after Step 1. B) Mosaic 8027 after Step 6. Markers 1-7 indicate the locations used for inclination measures.

CHAPTER 5 Conclusions and Suggestions for Future Work

5.1 Conclusions

An automated mosaicking system has been developed to generate image mosaics using data obtained with a multi-sensor video camera. By employing ancillary data, mosaics are generated which preserve the geometric information present in their constituent video frames and facilitate size estimation. The mosaicking system was designed as a seven-step process:

1. Initial raw mosaic construction
2. Mosaic refinement using coarse-to-fine energy analysis
3. Mosaic error detection and removal using covariance analysis
4. Mosaic refinement using color analysis
5. Mosaic refinement by affine motion estimation
6. Super-resolution based noise removal
7. Mosaic stitching from raw mosaics and refinement of multi-sensor orientation data

In Step 1, guess m-points are derived from MSC azimuth and inclination data. RO data are shown to contain error levels that prohibit size estimation without additional refinement techniques. RO errors result from MSC errors and data collection errors. Mosaicking results from Step 1 illustrate the need for mosaicking refinement.

In Step 2, coarse-to-fine analysis and energy space constraints are used to optimize an energy

metric and, therefore, compute an m-point location for mosaic pairs. NSAD energy is shown to be superior to NCCV energy: NSAD energy provides comparable accuracy and reliability with NCCV energy but is less computationally complex. Mosaics refined with NSAD energy in Step 2 are shown to be highly accurate when compared to manually generated mosaic points.

In Step 3, covariance based m-point error detection and removal is used to correct errors that occur during Step 2. Step 3 is shown to detect and remove mosaicking errors that result from ambiguous NSAD energy spaces and RO-video correlation errors. It is rare that inter-frame differences—resulting from perspective changes, histogram changes, and object motion—cause false positive m-point classifications in NSAD space that cannot be resolved by covariance analysis.

In Step 4, m-point precision is further refined by using color data to augment mosaicking information in NSAD energy space. Color refinements were shown to offer subtle m-point improvements over initial mosaicking results. M-point data determined by manually mosaicking a frame set indicates that color analysis typically modifies m-point locations by only 2 or 3 pixels.

In Step 5, gradient descent is used to estimate affine motion parameters between images in a mosaic pair and determine exact m-point locations when mosaic pairs contain rotations and distortions. Affine motion estimation is found to have subtle sub-pixel effects on m-point locations. Evaluation of mosaic refinement by affine transformation estimation of m-points requires precise MSC calibrations beyond the scope of this thesis.

In Step 6, super-resolution based noise removal is used to refine video data. Super-resolution refinement is shown to reduce contrast artifacts, CCD artifacts, and lens artifacts in generated mosaics. We demonstrate mosaic video enhancements at different values of A, a free parameter in cubic convolution. By experimentation, we determine that $A = -11$ provides the best balance between sharpening and high frequency noise levels.

Finally, using objects of known dimension, we demonstrate that generated mosaics facilitate size estimation. We compute inclination angles from mosaics and use angles recorded by a B&S inclinometer to validate mosaic correctness. And, by comparing inclination angles to measurement points within mosaics generated after Step 1 and Step 6, we show that the mosaicking system increases mosaic accuracy by 86%. The mosaicking system provides visual and numerical data that facilitates size estimation by determining precise orientations of video frames.

5.2 Suggestions for Future Work

Current mosaicking results contain imaging artifacts at frame boundaries when mosaic data is taken from scenes containing high contrast change. To expedite further image processing, feathering algorithms and histogram analysis should be considered as methods of resolving imaging artifacts generated by contrast changes. Further visual enhancements must be applied with care to preserve the information within video frames.

The gradient matching system must process large regions of overlap and is, therefore, time consuming. Performance enhancements to the gradient matching system would super-resolution more feasible. Modified calculations of the gradient computation could have a significant impact on execution time. Additionally, the gradient matching system occasionally produces erroneous matches that degrade image quality. Methods of error detection and removal, beyond simple iteration thresholding, should be explored to increase mosaic reliability.

Finally, robust camera calibration should be conducted to determine camera parameters. Given a careful camera calibration, the MSC data collection process would be enhanced considerably. To improve the MSC's angle measurements, the monopod should be replaced with a tripod.

REFERENCES

[1] International Association of Panoramic Photographers, "Panoramic camera information," http://www.panphoto.com/cameras.html

[2] G. Erker, "Circular Fisheye Camera," http://www.angelfire.com/ca/erker/fisheye66.html

[3] T. Svoboda, T. Pajdla, and V. Hlavac, "Omnidirectional vision," http://cmp.felk.cvut.cz/demos/Omnivis/

[4] D. Tegolo and C. Valenti, "A naive approach to compose aerial images in a mosaic fashion," *Image Analysis and Processing*, pp. 512-516, 2001.

[5] A. Krishnan and N. Ahuja, "Panoramic image acquisition", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 379-384, 1996.

[6] L. McMillan and G. Bishop, "Plenoptic modeling: an image-based rendering system," *Proceedings of SIGGRAPH*, pp. 39-46, August 1995.

[7] S. Gumustekin and R. W. Hall, "Mosaic image generation on a flattened gaussian sphere," *IEEE Computer Society Workshop on the Application of Computer Vision*, pp. 50-55, 1996.

[8] B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha, "Universal mosaicing using pipe projection," *IEEE Computer Society International Conference on Computer Vision*, pp. 945-952, 1998.

[9] R. Szeliski and H. Shum, "Creating full view panoramic image mosaics and environment maps," *Proceedings of SIGGRAPH*, pp. 251-258, 1997.

[10] S. Peleg and J. Herman, "Panoramic mosaics with VideoBrush," *DARPA Image Understanding Workshop*, May 1997.

[11] S. Peleg and J. Herman, "Panoramic mosaics by manifold projection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[12] H. Y. Shum and R. Szeliski, "Systems and experiment paper: Construction of Panoramic Image Mosaics with Global and Local Alignment," *International Journal of Computer Vision*, pp.101-130, 2000.

[13] S. Peleg, B. Rousso, A. Rav-Acha, & A. Zomet, "Mosaicing on adaptive manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, pp. 1144-1154, October 2000.

[14] S. E. Chen, "QuickTime VR: an image-based approach to virtual environment navigation," Proceedings of the 22nd annual conference on *Computer Graphics and Interactive Techniques*, pp. 29-38, September 1995.

[15] M. Irani, P. Anandan, J. Bergern, R. Kumar, and S. Hsu, "Efficient representations of video sequences and their applications," *Signal Processing: Image Communication*, Vol. 8, pp. 327-351, 1996.

[16] M. Irani, P. Anandan, "Video indexing based on mosaic representations," *Proceedings of the IEEE,* Volume: 86 Issue 5 , pp. 905-921, May 1998.

[17] M. Jogan and A. Leonardis, "Omnidirectional vision applied to robotic orientation and nondestructive testing (NDT)," *IEEE Computer Society International Conference on Advanced Robotics* pp.31-36, 2001.

[18] S. Thompson, A. Zelinsky, and M.V. Srinivasan, "Automatic landmark selection for navigation with panoramic vision," *Proceedings of Australian Conference on Robotics and Automation*, 1999.

[19] J. Gluckman, S. Nayar, and K. Thoresz. "Real-time omnidirectional and panoramic stereo," *DARPA Image Understanding Workshop*, pp.299-303, November 1998.

[20] T. Takahashi, H. Kawasaki, K. Ikeuchi and M. Sakauchi, "Arbitrary view position and direction rendering for large-scale scenes," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 296-303, June 2000.

[21] Y. Rzhanov, G.R. Cutter, and L. Huff, "Sensor-assisted video mosaicing for seafloor mapping**,**" *IEEE International Conference on Image Processing*, Vol. 2, pp. 411-414, 2001.

[22] N Clark, "Initial results from a video-laser rangefinder device," *Proceedings of the 3rd Southern Forestry GIS Conference*, 2000.

[23] Laser Atlanta, "Advantage Laser Rangefinder" users manual: http://www.laseratlanta.com/Advantage.pdf

[24] J. Shi and C. Tomasi, "Good Features to Track," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600, June 1994.

[25] B. D. Lucas, T. Kanade, "An interactive image registration technique with an application to stereo vision," *The 7th International Joint Conference on Artificial Intelligence*, pp. 674-679, 1981.

[26] J. Shi and C. Tomasi, "Good features to track," Technical Report 93-1399, Cornell U., 1993.

[27] T. Huang and R. Tsai, "Multi-frame image restoration and registration," *Advances in Computer Vision and Image Processing*, Vol. 1, pp. 317-339, 1984.

[28] M. Irani and S. Peleg**, "**Improving Resolution by Image Registration," *CVGIP: Graphical Models and Image Processing*, Vol. 53, pp. 231-239, May 1991.

[29] M. Irani and S. Peleg**, "**Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency," *Journal of Visual Communication and Image Representation*, Vol. 4, No. 4, pp. 324-335, December 1993.

[30] M.C. Chiang and T.E. Boult, "Efficient Image Warping and Super-Resolution," *IEEE Workshop on the Application of Computer Vision*, pp. 56-61, Dec 1996.

[31] B. Bascle, A. Blake, and A. Zisserman, "Motion Deblurring and Super-resolution from an Image Sequence," *European Conference on Computer Vision*, pp. 573-582, 1996.

[32] T. E. Boult and G. Wolberg, "Local image reconstruction and subpixel restoration algorithms," *CVGIP: Graphical Models and Image Processing*, Vol. 55, No. 1, pp. 63-77, 1993.

[33] M.C. Chiang and T.E. Boult. "Local blur estimation and super-resolution," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 821-826, 1997.

[34] M. C. Chiang and T.E. Boult. "Imaging-consistent super-resolution," *DARPA Image Understanding Workshop*, 1997.

[35] S. Baker and T. Kanade, "Limits on Super-Resolution and How to Break Them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 9, pp. 372-379 September 2002.

[36] Z. Lin and H. Y. Shum, "On the fundamental limits of reconstruction-based super-resolution algorithms," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.1171-1176, 2001.

APPENDIX A

This appendix contains mosaics used for testing the mosaicking system. When necessary, mosaics are included as two-part diagrams to show detail. Figure A.1 begins on page 97, to show full detail without text overlapping the image. All mosaics in this appendix are shown in reduced resolution so that they may fit onto the page.

Figure A.1: Mosaic 001evn

Figure A.2: Mosaic 001odd

Figure A.3: Mosaic 091evn

Figure A.4: Mosaic 091odd

Figure A.5: Mosaic 172evn

Figure A.6: Mosaic 172odd.

Figure A.7: Mosaic 252evn (top).

Figure A.8: Mosaic 252evn (bottom).

Figure A.9: Mosaic 252odd (top).

Figure A.10: Mosaic 252odd (bottom).

Figure A.11: Mosaic 328evn (top).

Figure A.12: Mosaic 328evn (bottom).

Figure A.13: Mosaic 328odd (top).

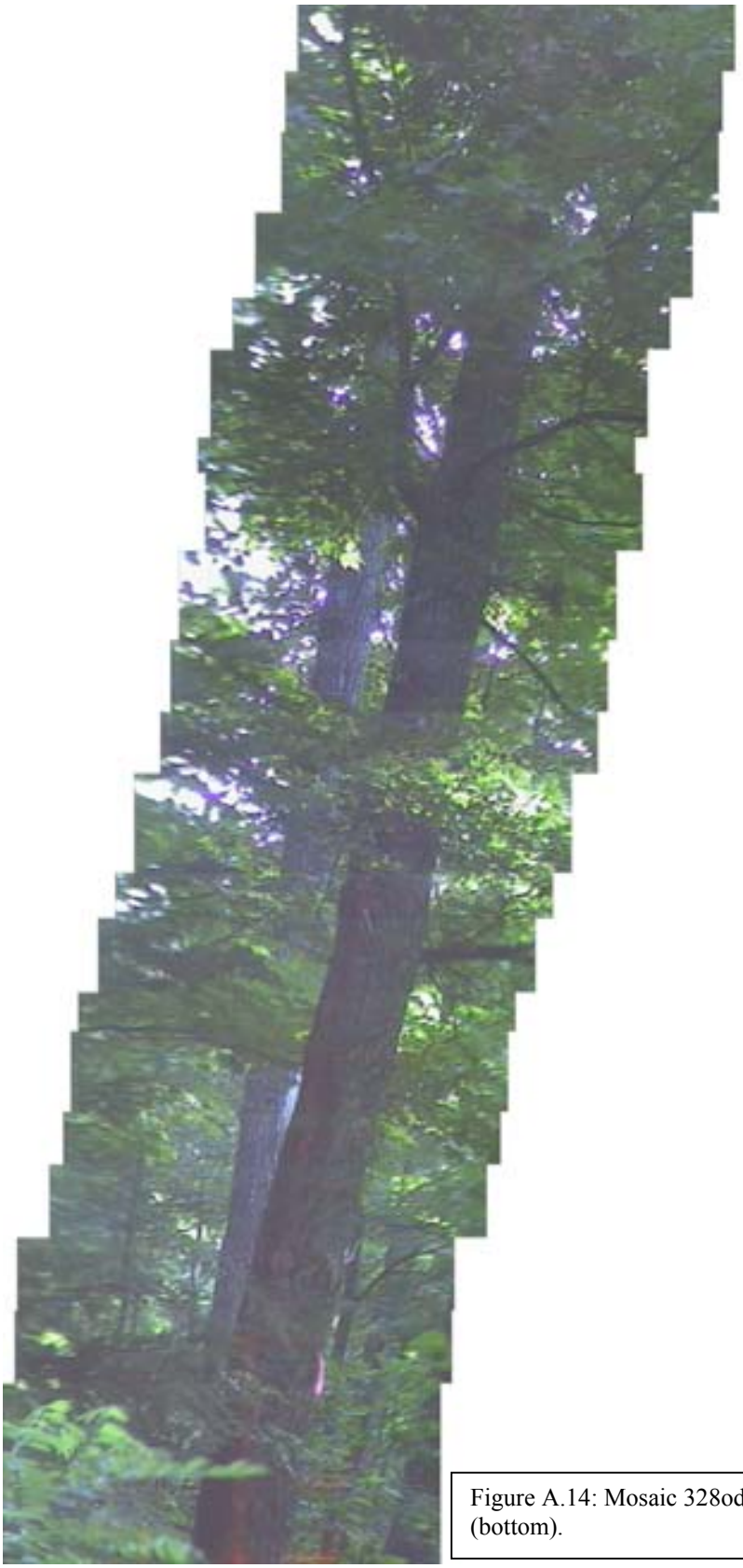

Figure A.14: Mosaic 328odd (bottom).

Figure A.15: Mosaic 423evn (top).

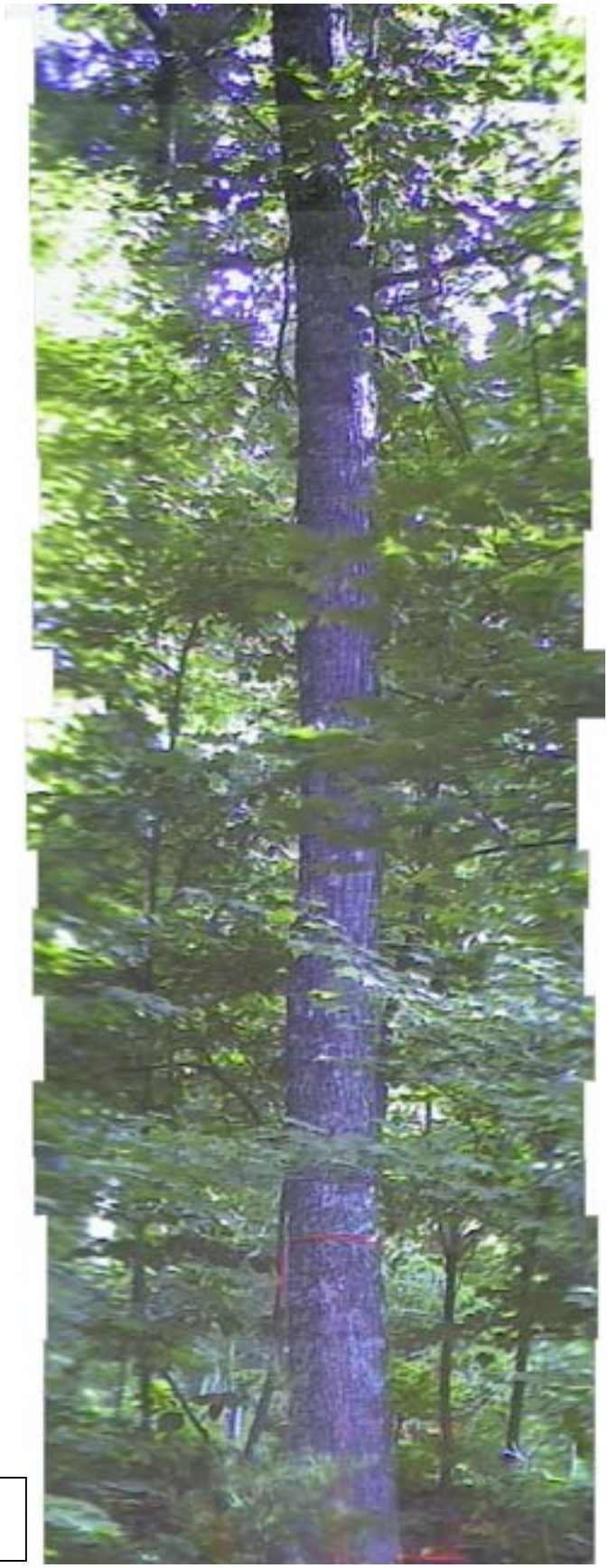

Figure A.16: Mosaic 423evn (bottom).

Figure A.17: Mosaic 423odd (top).

Figure A.18: Mosaic 423odd (bottom).

Figure A.19: Mosaic 500evn (top).

Figure A.20: Mosaic 500evn (bottom).

Figure A.21: Mosaic 500odd (top).

Figure A.22: Mosaic 500odd (bottom).

Figure A.23: Mosaic 590evn.

Figure A.24: Mosaic 590odd.

Figure A.25: Mosaic 8000.

Figure A.26: Mosaic 8047 (top).

Figure A.27: Mosaic 8047 (bottom).

Figure A.28: Mosaic 8090.

Figure A.29: Mosaic 8131.

Figure A.30: Mosaic 8173 (top).

Figure A.31: Mosaic 8173 (bottom).

Figure A.32: Mosaic 8207 (top).

Figure A.33: Mosaic 8207 (bottom).

Figure A.34: Mosaic 4005evn (top).

Figure A.35: Mosaic 4005evn (bottom).

Figure A.36: Mosaic 4039odd (top).

Figure A.37: Mosaic 4039odd (bottom).

Figure A.38: Mosaic 4062evn (top).

Figure A.39: Mosaic 4062evn (bottom).

VITA

Eric Kee was born in January 1978 in Franklin Virginia. As he finished his highschool education, he developed a keen interest in algorithms and logic. In 1996, Eric moved to pursue his technical interests and began his Bachelors in computer engineering at Virginia Tech. During his undergraduate work, Eric augmented his studies with a position as an embedded software developer at IBM. In his Junior year, Eric was accepted into the 5 year Bachelors/Masters program at Virginia Tech and chose to continue his studies in computer engineering by studying computer vision and image analysis. Funded by the Brooks Forest Products center at the USFS Southern Research Station in Blacksburg Virginia, Eric developed his technical skills in the form of this thesis and defended his work in January of 2003.

Eric now works in aerospace developing satellite systems and continues to develop ideas for future Ph.D. work in the fields of artificial intelligence, computer vision, and image analysis.