2005

# Digital Vertical Aerial Camera System for High-resolution Site Inspections in Conservation Easment Monitoring

Kenton Williams

## Recommended Citation

Williams, Kenton, "Digital Vertical Aerial Camera System for High-resolution Site Inspections in Conservation Easment Monitoring" (2005). *Electronic Theses and Dissertations*. 438.
http://digitalcommons.library.umaine.edu/etd/438

# DIGITAL VERTICAL AERIAL CAMERA SYSTEM FOR HIGH-RESOLUTION SITE INSPECTIONS IN CONSERVATION EASEMENT MONITORING

By

Kenton Williams

B.S. University of Maine, 2004

A THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science
(in Forestry)

The Graduate School
The University of Maine
December, 2005

Advisory Committee:

Steven A. Sader, Professor of Forest Resources, Advisor

Raymond Hintz, Associate Professor of Surveying Engineering Technology

Louis Morin, Instructor of Forestry

# DIGITAL VERTICAL AERIAL CAMERA SYSTEM FOR HIGH-RESOLUTION SITE INSPECTIONS IN CONSERVATION EASEMENT MONITORING

By Kenton Williams

Thesis Advisor: Dr. Steven A. Sader

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Science
(in Forestry)
December, 2005

A three-level satellite to ground monitoring scheme for conservation easement monitoring has been implemented in which high-resolution imagery serves as an intermediate step for inspecting high priority sites. A digital vertical aerial camera system was developed to fulfill the need for an economical source of imagery for this intermediate step. A method for attaching the camera system to small aircraft was designed, and the camera system was calibrated and tested.

To ensure that the images obtained were of suitable quality for use in Level 2 inspections, rectified imagery was required to provide positional accuracy of 5 meters or less to be comparable to current commercially available high-resolution satellite imagery.

Focal length calibration was performed to discover the infinity focal length at two lens settings (24mm and 35mm) with a precision of 0.1mm. Known focal length is required for creation of navigation points representing locations to be photographed (waypoints). Photographing an object of known size at distances on a test range allowed

estimates of focal lengths of 25.1mm and 35.4mm for the 24mm and 35mm lens settings, respectively. Constants required for distortion removal procedures were obtained using analytical plumb-line calibration procedures for both lens settings, with mild distortion at the 24mm setting and virtually no distortion found at the 35mm setting.

The system was designed to operate in a series of stages: mission planning, mission execution, and post-mission processing. During mission planning, waypoints were created using custom tools in geographic information system (GIS) software. During mission execution, the camera is connected to a laptop computer with a global positioning system (GPS) receiver attached. Customized mobile GIS software accepts position information from the GPS receiver, provides information for navigation, and automatically triggers the camera upon reaching the desired location.

Post-mission processing (rectification) of imagery for removal of lens distortion effects, correction of imagery for horizontal displacement due to terrain variations (relief displacement), and relating the images to ground coordinates were performed with no more than a second-order polynomial warping function.

Accuracy testing was performed to verify the positional accuracy capabilities of the system in an ideal-case scenario as well as a real-world case. Using many well-distributed and highly accurate control points on flat terrain, the rectified images yielded median positional accuracy of 0.3 meters. Imagery captured over commercial forestland with varying terrain in eastern Maine, rectified to digital orthophoto quadrangles, yielded median positional accuracies of 2.3 meters with accuracies of 3.1 meters or better in 75 percent of measurements made. These accuracies were well within performance requirements.

The images from the digital camera system are of high quality, displaying significant detail at common flying heights. At common flying heights the ground resolution of the camera system ranges between 0.07 meters and 0.67 meters per pixel, satisfying the requirement that imagery be of comparable resolution to current high-resolution satellite imagery.

Due to the high resolution of the imagery, the positional accuracy attainable, and the convenience with which it is operated, the digital aerial camera system developed is a potentially cost-effective solution for use in the intermediate step of a satellite to ground conservation easement monitoring scheme.

# AKNOWLEDGEMENTS

This study was supported by a grant from the New England Forestry Foundation to the University of Maine. Work was performed at the Maine Image Analysis Laboratory (MIAL), Department of Forest Management, University of Maine, in Orono, Maine.

I would like to take the opportunity to thank the many people who were instrumental in helping me complete this thesis. My advisor, Dr. Steve Sader, has been an incredible help not only in advising me on this project but also advising me on other matters relating to my choices in attending graduate school in Forest Management and pursuit of a career. His advice has proven to be immensely valuable and I greatly appreciate the guidance he has provided.

The members of my committee, Dr. Ray Hintz and Louis Morin have also been extremely helpful and I am certain that I could not have finished without their advice and assistance. I wish to thank Ray for assisting me with the technical details of my work, particularly relating to photogrammetry and land surveying. Without his assistance, the quality of the results would not be what they are. Ray also deserves credit for sparking my interest in photogrammetry in undergraduate courses and I am further grateful for his putting me in contact with my advisor, Steve Sader. Louis has been an invaluable resource to lean on due to his knowledge of surveying and photogrammetry as well as his first-hand experience with using small aircraft as an aerial photography platform. I am grateful to the advice he has offered.

I would also like to thank Dr. William Halteman, Professor of Mathematics and Statistics, who provided me invaluable advice on the proper statistical analyses of my data and the way that the results should be presented.

I wish to thank my friend, Patrick Fikes, for his help with virtually all stages of the development of my project. He was responsible for many good ideas that have manifested themselves within the completed work, including assistance with guidance software, electrical concerns, and general engineering advice. I appreciate his help immensely.

Finally, I wish to thank my family, especially my wife Amy and my three girls, Marinne, Kady, and Cambree. I could not have even started on this grand adventure without my wife's encouragement and support. She has made many sacrifices to ensure that I could dedicate the time required to complete this work, and I cannot envision attempting it without her love and patience. I know I could not have done it without her. I also appreciate the sacrifices my daughters have made, giving up playtime with me so I could dedicate my time to completing my studies. I also wish to thank my family, especially my parents, Blaine and Shirley Williams, for supporting our decision to move so far away and undertake such a difficult task.

# TABLE OF CONTENTS

Chapter

# LIST OF TABLES

# LIST OF FIGURES

# Chapter 1

# INTRODUCTION

The digital aerial camera system reported in this thesis was designed in response to the need of conservation groups to develop practical and cost-effective easement monitoring capabilities. The introduction begins with an overview of conservation easements, monitoring needs, and methods for monitoring. The bulk of the thesis focuses upon the development, operation, and testing of the digital camera system as the intermediate component of a satellite to ground monitoring approach.

## 1.1. Background for Working Forest Conservation Easements

In Maine, several non-industrial forestland owners have established working forest conservation easements (WFCEs). A conservation easement is a legally binding agreement that places restrictions on the encumbered land in perpetuity. Generally, the restrictions focus upon preventing development of the land, intending to preserve conservation values of the property (Commonwealth of Massachusetts 2003). The land can be sold, inherited, or transferred, but the restrictions go with the land forever (Sader et al. 2002). A WFCE differs from a conventional conservation easement in that the land will continue to be used as working forestland "actively managed for goods and/or services that have a monetary value in the current marketplace, such as timber, recreation, and water supply protection." (Lind 2001). A landowner encumbering their property with a WFCE normally receives a fee and/or tax credits for granting the easement and contractually agrees to restrictions on development and other uses of the land in perpetuity (Sullivan 2003). In most cases, the grantee of the conservation easement is a conservation organization or land trust that assumes the responsibility of monitoring the easement in accordance with the easement terms. Land trusts may protect large areas of land without having to raise the funds required to purchase the land in fee, as the funds required to purchase a WFCE are significantly less than outright purchase (Block et al. 2004).

## 1.2. Easement Terms and Conditions

The terms of each easement are different and reflect the requirements of both the grantor (landowner) and grantee (usually a land trust or conservation organization). Terms included in WFCEs generally revolve around prohibiting future development with emphasis placed upon allowing forest management to continue. Specific restrictions that may be incorporated include prohibiting subdivision, restrictions on constructing new structures and roads, limitations on the size or total area of active gravel pits, restricting the expansion of camp leases, clearcut size restrictions, and prohibiting cutting within certain sensitive riparian habitat zones (Sader et al. 2002). Easement documents may contain provisions requiring maintaining silvicultural resources, ecosystem health, or quality of timber resources (Block et al. 2004).

### 1.3. Easement Holders' Monitoring/Enforcement Responsibilities

Just as the easement restrictions go with the land in perpetuity, the easement holder assumes the responsibility of monitoring the restrictions and conditions of the easement in perpetuity (Forest Service 2003). Monitoring is required to meet the easement obligations and maintain the tax-exempt status of the conservation organization that holds an easement. In addition to monitoring, the easement holder is responsible for enforcement of the terms of the easement should a violation occur. Monitoring activities are intended to further the integrity of the easement and ensure that the benefits intended by the easement are maximized (Commonwealth of Massachusetts 2003). A conservation easement is only as good as its monitoring and enforcement provisions allow (Interagency Committee for Outdoor Recreation 1999). Without monitoring, easement terms and restrictions cannot be verified and the continuance of the easement is jeopardized (Sader et al. 2002).

### 1.4. Monitoring Methods

With very large easements (tens of thousands of acres or more), monitoring and enforcement of easement conditions can become very expensive and ineffective without proper design and planning. Monitoring conservation easements can be accomplished in a variety of ways, each varying in expense per-site and scale. Three primary groups of methods can be identified: ground visits, aerial photography, and remote sensing via satellite imagery. Although ground visits by land trust staff has been found to be the most effective method of monitoring smaller easements (Block et al. 2004), on-site visits are the most expensive technique for monitoring large scale easements (Levitt 2003) on a per-site basis. Integration of aerial photography into a monitoring scheme allows monitoring of medium to larger easements in a more cost-effective manner compared to ground visits. Aerial photography has been found to be especially appropriate for monitoring of development restrictions, clearcuts, and buffers around water bodies, but can be very expensive for large easements if acquisition and interpretation of the photography is the sole means of monitoring (Block et al. 2004). Finally, remote sensing techniques utilizing medium-resolution (10m to 30m per pixel) satellite imagery represents an effective way to monitor large WFCEs due to the wider geographic extent covered in each satellite image and because satellite change detection techniques allows evaluation of the entire easement quickly at the lowest cost per site (Levitt 2003).

As the size of the easement increases, the importance of appropriate choice of monitoring methods becomes apparent (Sader et al. 2002; Leavitt 2003; Block et al. 2004). The most effective monitoring schemes for large WFCEs include a combination of the three types of monitoring methods (Block et al. 2004). Monitoring schemes incorporating of all three methods at appropriate scales such as the scheme proposed in Sader et al. (2002) are likely to become a model for monitoring landscape scale easements (Levitt 2003; Block et al. 2004) regardless of geographic location.

### 1.5. Three-Level Easement Monitoring Scheme: Satellites to Ground

The largest working forest conservation easement to date in the United States, the Pingree Easement in Northern Maine, encumbering 762,192 acres, was purchased in 2001 by the New England Forestry Foundation of Groton, Massachusetts. A protocol was devised to monitor the Pingree Easement, utilizing a three-level hierarchical scheme designed to minimize costs and maximize efficiency of monitoring activities (figure 1).

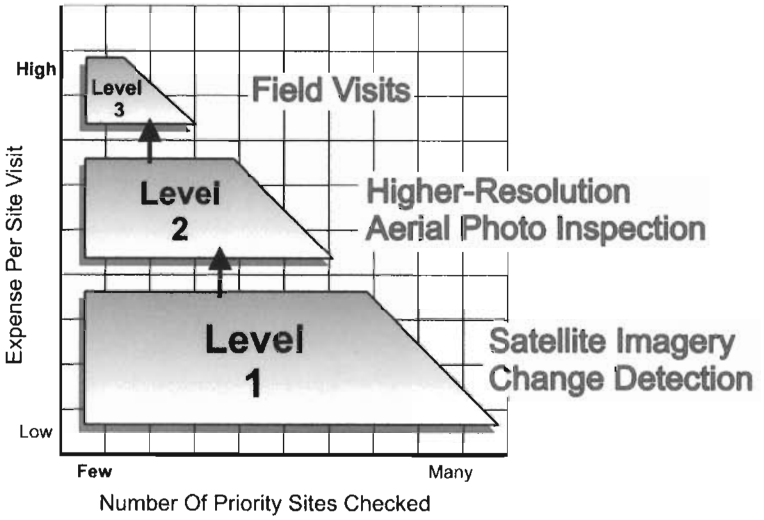


Figure 1. Three-level Conservation Easement Monitoring Scheme

At Level 1, forest change detection techniques are applied using a simple algorithm designed for detecting canopy disturbance on geo-referenced medium-resolution satellite imagery (Sader et al. 2002). Regions of interest ("targets") identified in Level 1 are given higher priority for examination at Level 2, inspected using recent high-resolution imagery such as aerial photography or high-resolution satellite imagery. Camp leases, riparian zone buffers, and clearcuts over a certain size limit are examples of "targets" that may require Level 2 monitoring, especially if canopy change was detected at Level 1. At Level 3, field visits (or measurements) provide the closer scrutiny needed to verify the condition of priority areas identified at Level 1 or 2 (Sader et al. 2002).

The hierarchical multi-resolution monitoring scheme presented is similar to a nested sampling approach and is intended to reduce costs, allowing easement monitoring to occur at frequent intervals. Although the amount of detail with which priority sites can be inspected increases with progression from one level to the next up the hierarchy, the cost per site inspection also increases (figure 1). On a per-site basis, Level 2 visits are more expensive than Level 1 inspections, but Level 3 visits are by far the most expensive, requiring monitoring staff to travel many hours and vehicle miles to remote locations. To reduce the overall cost of monitoring activities, only those sites requiring increased detail to verify the site conditions for easement term compliance are examined at higher levels.

### 1.6. Challenges for Level 2 Image Acquisition

High priority sites are those exhibiting recent change or those ground attributes that cannot be resolved by coarser resolution satellite imagery, therefore requiring recent high-resolution imagery for closer inspection at Level 2. The current sources for imagery required in Level 2 activities include high-resolution satellite images (1-4m ground resolution) or currently available aerial photographs. Existing high-resolution imagery can be utilized, but only if the images are recent and priced within the budget constraints of the project. Unfortunately, due to the remoteness of many locations, recent aerial images are usually not available from aerial survey vendors or landowners. Also, clouds may obscure these areas in recently collected satellite imagery. Furthermore, the high priority sites to be inspected at Level 2 are normally sparsely scattered throughout the easement area rather than appearing only in one region.

Contracting traditional aerial survey photography vendors to acquire imagery may be prohibitively expensive for most easement grantees, especially the smaller land trusts. Because aerial survey vendors' primary mode of operation is to gather imagery of moderate-sized contiguous regions, their operation is not well suited for acquisition of individual sparsely scattered regions. The cost to obtain the required images could amount to many thousands of dollars, depending upon the distance to the target sites, the number of sites to be photographed, photo processing costs, and the scale of the photographs.

While several commercial sources of high spatial resolution satellite imagery suitable for use in Level 2 inspection exist, the cost per-site is extremely high due to the requirement of ordering entire image scenes rather than only the specific areas needed. While the cost-efficiency of these satellite images increases with larger contiguous areas being examined, they are not well suited for economically obtaining imagery of small spot or intermittent areas scattered throughout a large region. As an example, one Quickbird Satellite image covering 16.5 km x 16.5 km (27,225 hectares) at a spatial resolution of 61 cm, corrected for minor radiometric and geometric errors costs US$4,356 ($16 per square km) at the time of writing (source: www.mapmart.com, last accessed November 2005) and US$4,992 for similar minimally-corrected Orbview-3 imagery (source: www.orbimage.com, last accessed November 2005). At the current cost per hour for a small aircraft with pilot and fuel of $150, the expense of just one Quickbird scene equates to over 29 hours of flight time, likely more than sufficient flying time to cover all necessary Level 2 image acquisition costs for an entire year for even the largest conservation easements in the United States (e.g. the Pingree easement in Maine). Performing Level 2 inspections on large easements utilizing Quickbird satellite imagery would require the purchase of many scenes, resulting in high expense.

# Chapter 2

## OBJECTIVES

The objective of this research is to design and test a digital vertical aerial camera system, integrating geospatial technology. The ultimate goal of the camera system is to reduce costs and improve the efficiency of the overall monitoring scheme, while reducing or eliminating reliance upon existing high-resolution imagery sources. The following sections will discuss the process and methods employed to assemble, calibrate, and test the digital camera system developed.

# Chapter 3

# DIGITAL VERTICAL AERIAL CAMERA SYSTEM

To address the requirements of improving Level 2 monitoring capabilities, a digital vertical aerial camera system was assembled, calibrated, and tested. The system is comprised of: a high-quality digital single-lens-reflex (SLR) camera and lens, a vibration-reducing mount, and a laptop computer with a connected Global Positioning System (GPS) receiver for navigation guidance and automated camera triggering. Because the system is dependent upon electricity, the system also includes components to supply the additional power required for long-duration flights. Photogrammetric methods were used in calibration of the camera and are implemented in photography mission planning tools that were developed. The goal in the design was to assemble a compact system that can be transported easily to local airports and mounted on commonly available small aircraft at very short notice, providing images of high-resolution that are readily available to easement monitoring staff in digital format.

Digital camera systems offer several advantages when employed for aerial photography, primarily with respect to convenience, cost savings, and speed. Advantages include: smaller sized cameras which allow use of less expensive and smaller aircraft, immediate access to data, faster delivery of data due to not requiring darkroom or laboratory chemical processing, data compatibility and ease of access in Geographic Information Systems (GIS) and remote sensing software, cost savings due to elimination of film scanning steps for acquisition of digital files, and compact storage and archiving of data (Graham and Koh 2002). Additionally, operation costs are lower with digital aerial camera systems; costing 32 percent less than digitized 9 in. x 9 in. film photographs from a commercial vendor when using high-end expensive digital capture equipment. Digital camera systems have been found to be the most cost-effective means of acquiring imagery for small to medium sized regions, although 9 in. x 9 in. film photography is still the most cost-effective solution for larger regions (Graham and Koh 2002). Using off-the-shelf components that are modestly priced, the digital camera system presented represents significant purchase cost savings and higher operating efficiency, further decreasing operating expenses. Once assembled and calibrated, the expense of obtaining Level 2 imagery is dependent only upon the cost per hour for rental of the aircraft and pilot rather than the cost per image.

# Chapter 4

# PHOTOGRAMMETRY REVIEW

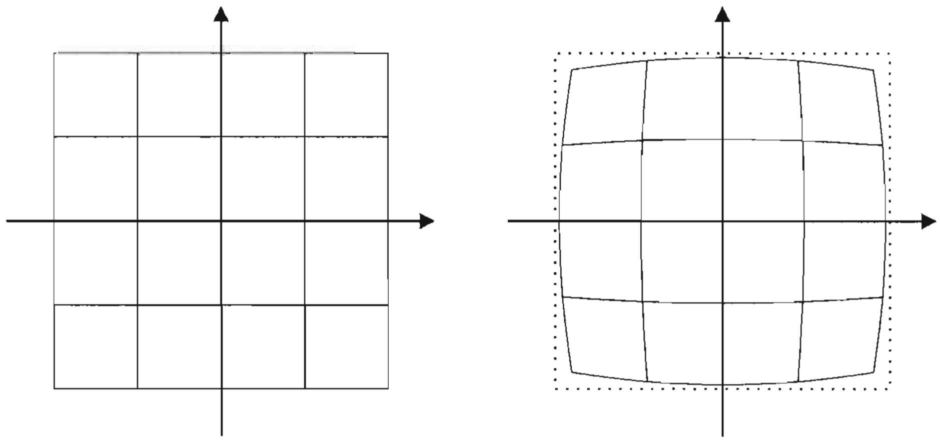
Photogrammetry has been defined as the art, science, and technology of obtaining reliable information about physical objects and the environment through processes of recording, measuring, and interpreting photographic images (ASPRS 1980). Photogrammetry can be further divided into two areas: metric photogrammetry and interpretive photogrammetry. Metric photogrammetry is concerned with making reliable measurements from photographs such as distance, volume, area, size, shape, and elevations of objects appearing in the images being analyzed. Interpretive photogrammetry deals with recognizing and identifying objects and judging their significance (Wolf and Dewitt 2000).

Photogrammetry, both metric and interpretive, is a key technology utilized in the three-level monitoring scheme for large WFCEs at Level 2 (Sader et al. 2002). Interpretive evaluations are made to identify targets for further examination and measurements at Level 2 as well as identification of sites requiring Level 3 inspections. Metric photogrammetry is employed to make measurements of areas such as gravel and borrow pits and linear measurements such as river buffer distances representing riparian habitats.

## 4.1. Methods of Camera Calibration

Interpretive photogrammetry involves mainly visual pattern recognition and analysis, requiring little information regarding the camera system used to capture imagery. In contrast, metric photogrammetry requires knowledge of several parameters unique to the camera and lens, known as interior orientation parameters. These interior orientation parameters are used to correct for imperfections and provide required constants, allowing reliable measurements to be made. The process of obtaining these unique parameters is known as camera calibration.

Camera parameters commonly discovered through calibration procedures include the computed principal distance or focal length ($f$) of the lens, parameters ($x_p$, $y_p$) which denote the coordinates of the center of projection of the image (principal point), and lens distortion coefficients ($k_1$, $k_2$, $k_3$, $p_1$, $p_2$, $p_3$) where the terms $k_i$ represent coefficients of radial lens distortion and $p_i$ terms represent coefficients of decentering distortion caused by a lack of centering of lens elements. Radial lens distortion is normally present in the form of barrel distortion (see Figure 2) while decentering distortion effects are typically very small in magnitude (Fraser 1997).



No Distortion Present

Barrel Distortion Present

Figure 2. Barrel Type Radial Lens Distortion (after BASP's RadCor Software Web Site, www.uni-koeln.de/~a1001/radcor.html, last accessed November 2005)

Radial lens distortion is represented in metric photogrammetry as an odd-ordered polynomial equation where $r$ denotes the radial distance from the principal point (projective center point of the image):

$$\Delta r = k_1 r^3 + k_2 r^5 + k_3 r^7 \tag{1}$$

As an alternative, another representation of radial lens distortion is found by replacing distortion coefficients $k_i$ with slightly different coefficients (*a, b, c, d*), computed using equations 2 and 3.

$$a = k_2 * (w/2)^3 \quad (w \text{ is the width of the image}) \tag{2}$$

$$c = k_1 * (w/2) \quad (w \text{ is the width of the image}) \tag{3}$$

The polynomial distortion function is then composed with these alternative coefficients (equation 4). The quantities found in equation 4 are: $r_{src}$ - the radial distance from the principal point of the source image, $r_{dest}$ - the radial distance in the corrected image, $b$ - normally set to zero, and $d$ - a resampling scaling factor (Dersch 1999).

$$r_{src} = (a * r_{dest}^{\ 3} + b * r_{dest}^{\ 2} + c * r_{dest} + d) * r_{dest} \tag{4}$$

A variety of camera calibration methods have been devised with accompanying complexity and difficulty of implementation. The choice of calibration methods largely depends upon the accuracy requirements of the particular application that the camera system is expected to deliver, more strict accuracy requirements requiring more rigorous calibration methods. The most common calibration methods in use can be grouped into three categories: laboratory methods, field methods, and stellar methods (Wolf and Dewitt 2000).

Laboratory methods require the construction of very precise optical measuring equipment and are usually reserved for large, high accuracy mapping projects due to the expense and difficulty of implementing these techniques. Stellar calibration methods are capable of yielding moderately high accuracy calibration results by photographing many identifiable stars and computing camera parameters using the known ephemeris data of the stars photographed. However, accuracy is affected by atmospheric distortion and inaccuracies in the geographic location of the camera at the time that the exposure is made (ASPRS 1980). Field methods require photographing an array of targets with established characteristics allowing computation of camera calibration parameters by application of projective geometry.

Field calibration methods are normally employed when calibrating camera systems for applications in which extremely high accuracy is not required (Matsuoka et al. 2002). Two common field calibration methods are analytical self-calibration (Kenefick et al. 1972) and analytical plumb-line calibration (Brown 1971).

Analytical self-calibration is characterized by photographing an array of targets with known characteristics from several viewpoints, solving for all of the desired parameters in a simultaneous least-squares adjustment. The advantage to self-calibration is that all interior orientation parameters are discovered with one simultaneous solution, but the disadvantage is that a calibration range must be created and accurately measured (Kenefick et al. 1972).

Analytical plumb line calibration methods are concerned primarily with recovery of lens distortion parameters, using a separate procedure to discover other parameters such as focal length. The premise of a plumb-line calibration is that straight lines in reality should be represented as straight lines in a photograph of the straight lines; any deviations of the lines in the photograph from straightness are caused by distortions present in the lens (Brown 1971). The plumb-line method as proposed by Brown (1971) offers the advantage of a much simpler mathematical model and less complexity in solution, but does not permit recovery of the coordinates of the principal point ($x_p$, $y_p$). Modification of the plumb-line method by adding a curve-fitting process as described in Valkenburg and Evans (2002) allows discovery of the principal point coordinates as well as lens distortion coefficients. Alternative implementations of this procedure based upon Dersch's alternative distortion equation (4) and employing least-squares optimization to discover interior orientation parameters of digital cameras are found in software packages such as PTLens (www.epaperpress.com/ptlens/) and PTGui (www.ptgui.com). Once camera parameters are known, additional software packages can be used to remove the distortion from digital images.

### 4.2. Justifications for Choice of Calibration Methods

Each photogrammetric application carries with it certain expectations for accuracy, whether published or not. The type of application for which photogrammetric measurements will be made dictates the method required for calibration of the camera system to be used. For extremely high-accuracy engineering photogrammetry, accuracy expectations may be in the vicinity of 1 part in 250,000, requiring the most carefully executed and most rigorous methods, generally those performed in high-accuracy laboratories using very expensive equipment. Lower accuracy requirements for the task to be performed allow using methods which are less accurate, but also much more convenient and economical.

While high-accuracy photogrammetric projects are usually well defined, including specification of accuracy requirements, definition of medium-accuracy photogrammetry is somewhat nebulous, falling somewhere in between high-accuracy metric photogrammetry and interpretive photogrammetry. Despite the lack of definition, medium-accuracy photogrammetry still falls into the category of metric photogrammetry, requiring calibration of the camera system. The parameters required in medium-accuracy photogrammetry include many of those previously mentioned, such as focal length, principal point coordinates, and lens distortion coefficients, however, the reduced accuracy requirements permit omission of many lens distortion coefficients. Unless the lens being used is designed specifically for distortion reduction, the distortion function will be nearly totally dominated by the coefficient $k_1$ (Brown 1971). As a result, discovery of the $k_1$ coefficient alone will usually suffice in medium-accuracy metric applications allowing the use of less complex calibration methods (Fraser 1997).

### 4.3. Vertical Aerial Photographs

A vertical aerial photograph is one in which the optical axis of the camera taking the photograph is vertical or as nearly vertical as possible. While truly vertical photographs in which the optical axis is exactly vertical are rarely captured, most vertical aerial photographs deviate only by small amounts from truly vertical, allowing analysis with equations derived from the geometry of the truly vertical condition (Wolf and Dewitt 2000). As the pilot is normally instructed to keep the plane as level as possible, effects of navigation inputs and trim of the aircraft should rarely result in more than 3 degrees of tilt (Wolf and Dewitt 2000, ASPRS 1980). Photographs in which the tilt amount is 5 degrees or less can be utilized for photogrammetric procedures provided that corrective procedures are employed on the imagery (Graham and Koh 2002). Therefore, a maximum tilt error at the moment of exposure of 5 degrees was determined as a reasonable estimate for use in tilt-compensation in mission planning stages.

Using known quantities and making the assumption of perfectly vertical orientation of the camera, the scale of a vertical photograph can be computed using the well-known photogrammetric scale equation (equation 5), where $f$ is the known lens focal length, $ab$ is the measured distance on the photo, $AB$ is the ground distance corresponding to $ab$, $H'$ is the flying height above terrain, and $S$ is the scale of the vertical photo.

$$S = \frac{ab}{AB} = \frac{f}{H'} \tag{5}$$

In addition to computing the scale of vertical photographs, equation 5 can be algebraically manipulated to solve for unknown quantities, such as computing a ground distance ($AB$) if $ab$, $f$, and $H'$ are known or computing focal length if an object of known size is photographed at a known distance ($H'$).

### 4.4. Relief Displacement

Relief displacement is defined as a shift or displacement of an object in a photograph caused by the objects elevation above or below a selected elevation, or datum. Objects that are above datum are displaced outward and objects below datum are displaced inward (Wolf and Dewitt 2000). Relief displacement appears in a radial direction relative to the nadir point of the photograph, with larger magnitudes of displacement occurring on objects that are radially more distant from the nadir point and objects with larger deviations in elevation from datum. The result of these displacements is that objects do not appear in their true horizontal positions in a vertical aerial photograph. In natural landscapes such as those examined in conservation easement monitoring, variations in terrain result in displacements that must be accommodated to maintain horizontal measurement accuracy.

### 4.5. Rectification of Images

Because of the effects of relief displacement and minor deviations from the truly vertical condition, most aerial photographs cannot be used directly as a map, especially important when the intended use of the photograph is to include it in a GIS for analysis. Prior to use as a map or inclusion as a layer in a GIS, the image must be corrected for these displacements as well as any distortions introduced by the camera lens. Correcting an image for horizontal displacements due to terrain relief and tilt error is known as rectification. The process of correcting a nearly vertical aerial photograph for the effects of relief displacement and tilt errors is known as orthorectification or differential rectification (Wolf and Dewitt 2000). Orthorectification is normally performed by collecting coordinates of photo-identifiable ground locations for orientation of the image and mathematically correcting the image based upon a digital elevation model (DEM) of the region appearing in the image (Wold and Dewitt 2000). True orthorectification as described requires significant effort and is highly complex likely requiring expensive ground control and equipment operated by highly trained individuals.

In regions where terrain variation is moderate, rectification alone can be used as a less complex alternative to true orthorectification using a DEM. The process of rectification entails warping the aerial photograph so that known control points visible in the image are presented in correct relation to each other. Known control points are obtained through either surveying on the ground or by using existing orthoimagery such as Digital Orthophoto Quadrangles (DOQs) or existing maps (e.g. 7.5 minute USGS topographic maps). Although rectification can be performed either by optical-mechanical devices or digitally by computers, the most convenient method is digital warping by a computer, making it advantageous to employ digital cameras to capture imagery. Images captured digitally can be immediately processed and rectified without the need for intermediate steps such as digitizing film, thus resulting in time savings and increased convenience when used in a GIS.

### 4.6. Image Warping/Geometric Transformation

Digital images to be used in GIS or image processing software will often require transformation to co-register with existing data, necessitating a rectification process. Regardless of the rectification process chosen, at some point, the image must be warped to correct distortions present in the source image using a geometric transformation. These geometric transformations can be grouped into two categories: simple planar mappings and polynomial transformations (Wolberg 1990).

**4.6.1. Simple Planar Mappings.** Two common simple planar mappings used in image warping are affine warps and perspective transformations. Affine warps are capable of transforming by rotation, translation, scale, and shear. Affine warps preserve parallel lines and preserve relationships among equally spaced points. Perspective transformations can be useful in correcting for tilt error in nearly vertical aerial photographs, and are considered a type of projective mapping. Perspective transformations preserve parallel lines only when they are parallel to the projection plane, otherwise parallel lines converge to a vanishing point. Additionally, perspective transformations preserve lines in all orientation, meaning a straight line in the source is rendered as a straight line in the mapped image (Wolberg 1990).

**4.6.2. Polynomial Warping Transformations.** Polynomial warping transformations commonly encountered in remote sensing and photogrammetric applications map the entire image using low-order polynomial functions shown in equation 6 and 7, where $N$ is the order of the polynomial function, $x'$ and $y'$ are the warped image coordinates, $x$ and $y$ are the source image coordinates, $a_{ij}$ and $b_{ij}$ are constants that define the warping operation. These functions are intended to correct sensor-related distortions, errors due to earth curvature, viewing geometry, camera attitude, and terrain relief variations (Wolberg 1990).

$$x' = \sum_{i=0}^{N} \sum_{j=0}^{N-i} a_{ij} x^i y^j \tag{6}$$

$$y' = \sum_{i=0}^{N} \sum_{j=0}^{N-i} b_{ij} x^i y^j \tag{7}$$

Polynomial warping transformations are able to correct for the smoothly varying distortions mentioned above but cannot correct high-frequency localized deformations such as extreme elevation change over a small region. Because these transformations are able to warp straight lines into curvilinear features, they are well suited to the problem of correcting imagery for slight to moderate terrain relief. Additionally, because of their global nature, other corrections for scale, rotation, shear, and translation can be performed simultaneously by computation of proper coefficients. The order *N* of the polynomial is chosen depending upon the amount of warping desired and the type of errors that need to be corrected. Typically, the order of polynomial rarely exceeds *N*=3 because higher ordered polynomials are capable of severely warping the source image and can yield unexpected and unrealistic results. A first-degree (*N*=1) polynomial is frequently substituted in place of an affine planar mapping, capable of correcting scale, translation, rotation, and skew. For most practical problems, a second order polynomial has been shown to be adequate (Wolberg 1990, Lillestrand 1972).

### 4.7. Stereoscopic Viewing and Measurement

Humans constantly judge distances of objects within their view employing binocular vision, viewing the world with both eyes simultaneously, which is commonly referred to as stereoscopic vision. Binocular vision enables depth perception due to parallax, meaning that each eye sees a slightly different view due to their slightly different vantage point. The phenomenon of stereoscopic depth perception can be achieved with photography using two images that have at least partially overlapping views taken from two different vantage points (Wolf and Dewitt 2000). In the case of aerial photography, these two images are overlapping vertical photographs captured at nearly identical flying heights along the same flight line at a predetermined distance apart. The resulting parallax from photographing objects from two different vantage points is a geometric relationship that can be measured very accurately. Topographic maps created using high-precision cameras and stereo-comparators in conjunction with rigorous computer software have been able to yield maps with precision approaching first-order geodetic surveys (ASPRS 1980). While precision measurements such as these are certainly impressive, not all applications require such rigorous procedures. In the context of conservation easement monitoring, stereoscopic viewing has applications primarily in the realm of interpretive analysis where only qualitative examinations of the landscape are needed instead of precision measurements.

# Chapter 5

# METHODS

A digital vertical aerial camera system was designed and assembled and tested. The system was intended for use in capturing imagery for Level-2 analyses in the three-level system for conservation easement monitoring (Sader et al. 2002). The basic design concept was to assemble and calibrate a digital camera system that can capture high-resolution vertical aerial imagery of desired targets or regions. The camera system must also be compact for convenient transport to the airport, allow rapid capture of desired imagery, provide reliable and reasonably accurate imagery once rectified, deliver high image quality, and be convenient and inexpensive to operate.

The concept for the camera system was that the easement monitoring staff would determine locations requiring Level-2 inspection, then plan photography missions using customized tools in GIS software that implement photogrammetric methods. Finally, monitoring staff would travel to the airport to capture imagery with minimal delay, carrying equipment with them rather than relying on permanently mounted camera equipment and experiencing delays between ordering and delivery of imagery. In addition to the rapid response time between determination of target sites and capture of imagery, using off-the-shelf components and small aircraft allow the system to operate with minimal cost. The system design concept included the criteria that the system be as automated as possible, requiring guidance software to instruct the pilot and trigger the camera at pre-planned locations, thus reducing the possibility of operator errors which could increase costs due to repeated missions being required to photograph target sites.

In addition to image quality concerns and operation convenience, proper methods were required to mount the camera to the aircraft, protecting the camera from vibration or impact damage while holding the camera securely in a near-vertical orientation, all accomplished without structurally weakening the aircraft or impairing flying characteristics. Also, due to the electronic components of the digital aerial camera system and their dependence upon electrical power required that adequate power supply systems be assembled, designed to operate with minimal impact on aircraft performance while providing maximum uninterrupted flying time for photography missions.

## 5.1. Accuracy Requirements

The imagery obtained from the digital camera system are to be used in a GIS for further analysis, requiring that the images be rectified to existing imagery and data so that meaningful measurements and comparisons can be made. Furthermore, the images, once rectified, must have sufficient accuracy and detail to allow Level-2 inspections and reduce the number of required Level-3 ground visits. The criteria used to determine if sufficient accuracy has been achieved is that the coordinates of locations depicted in rectified imagery will be within +/- 5 meters of their true geodetic position, whether on individual images or on image mosaics created from a series of rectified images. While no accuracy limits were specified, the specification of +/- 5 meters was chosen to ensure that the methods used were rigorous enough to provide good quality and yield reliable, quantifiable accuracy without requiring overly complex procedures.

### 5.2. Digital Vertical Aerial Camera System Assembled

The system assembled to perform in Level-2 image acquisition will be described by an operating sequence, followed by detailed descriptions of components and techniques employed. The operating sequence includes the following steps: mission planning, mission execution (image acquisition), and post-mission processing (image rectification). Hardware and software components employed will be described with justification for choice by category: camera and lens, aircraft mounting bracket, GPS receiver, camera control computer and software, and power supply. Following the hardware and software details, camera calibration procedures employed, aircraft mounting bracket design and Federal Aviation Authority approval procedures, and in-flight guidance and camera control software will be described.

**5.2.1. Mission Planning.** Level-1 change detection analyses using satellite imagery remote sensing techniques are performed with GIS software, yielding digital imagery suitable for use in a GIS. Aerial photography mission planning to acquire Level-2 digital imagery is also performed using GIS software. The camera system is designed to capture imagery performing in any of three different modes: 1-single point, user-specified ground resolution, 2-single point, user-specified ground area, and 3-overlapping sequential photographs along a path, allowing stereoscopic coverage of regions around linear features such as rivers. The result of these modes is the creation of a set of individual point locations that will be visited during the photography mission, called waypoints. Waypoints, either individual (modes 1 and 2) or in a series along a path (mode 3), can be used for navigation purposes only or designated as points at which a photograph should be taken. Each waypoint created is added to a database employed by the in-flight camera control software, containing a variety of attributes for each waypoint. For modes 2 and 3 that rely on user-specified ground areas or distances that must appear in the images, compensation procedures designed to include allowances for aircraft position and attitude errors were developed.

*5.2.1.1. Mission Planning Tools in GIS Software.* The tools developed to facilitate mission planning were created using Avenue, the customization scripting/programming language employed by ESRI's ArcView software. Specifically, the version of software employed is ArcView 3.3, although any GIS software that allows customization through scripting of custom tools could be used to recreate these mission-planning tools. The custom tools created include: a camera parameters dialog for viewing and editing known camera parameters stored in a camera parameters text file, a tool to clear (delete) all current waypoints, mission flight path optimization, and tools for creation of waypoints in each of the three operating modes described above. These tools are implemented as custom buttons on the toolbar of an ArcView project as shown in Figure 3, and scripts for the tools appear in appendix A.

*5.2.1.2. Camera Parameters File and Parameter Viewing/Editing Dialog.* For photogrammetric techniques to be employed in creation of waypoints, several parameters unique to the camera and lens must be known, obtained from manufacturer's specifications or discovered through calibration procedures. As system components are changed or upgraded, a simple means of editing these parameters in the GIS software was required, implemented as a dialog box in ArcView 3.3 (figure 4). The fields in this dialog box are filled initially by the script, reading the values from a text file stored on the hard drive of the computer that contains the camera parameters. Table 1 details the structure of the camera parameters text file.

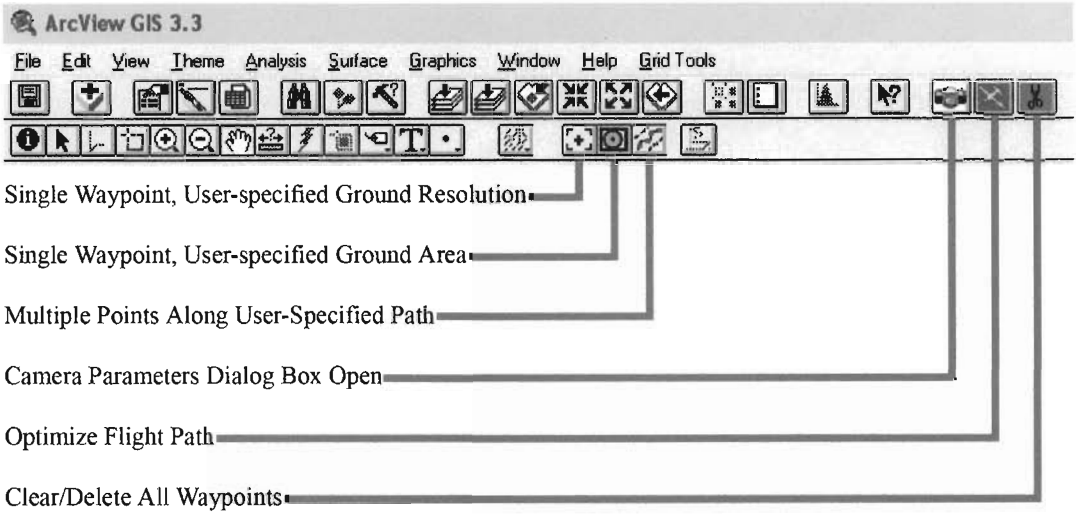
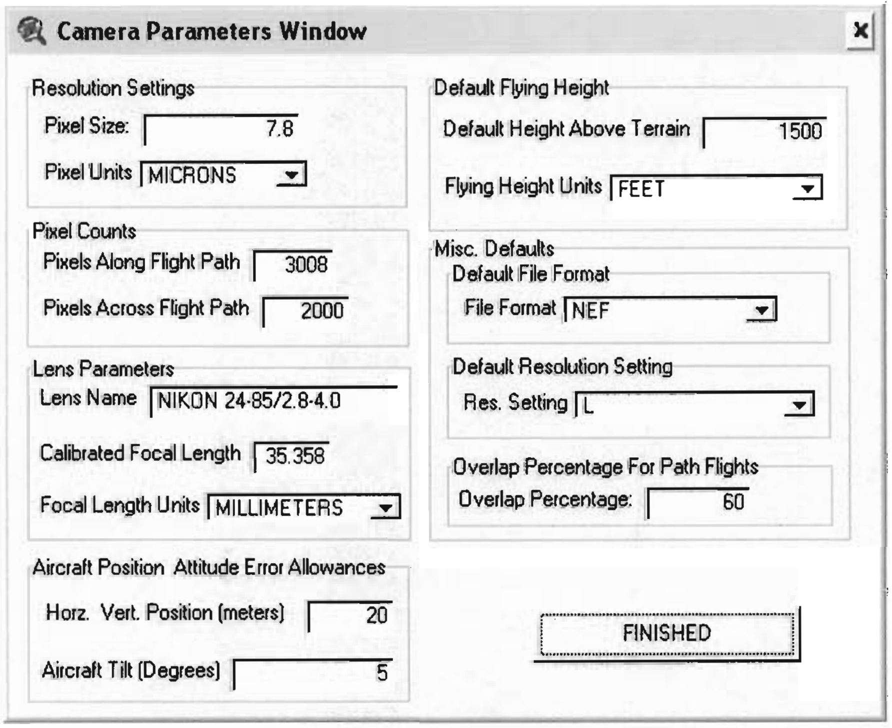


Figure 3. Custom Tools For Photography Mission Planning/Waypoint Creation



Resolution Settings
Pixel Size: 7.8
Pixel Units MICRONS

Pixel Counts
Pixels Along Flight Path 3008
Pixels Across Flight Path 2000

Lens Parameters
Lens Name NIKON 24-85/2.8-4.0
Calibrated Focal Length 35.358
Focal Length Units MILLIMETERS

Aircraft Position Attitude Error Allowances
Horz. Vert. Position (meters) 20
Aircraft Tilt (Degrees) 5

Default Flying Height
Default Height Above Terrain 1500
Flying Height Units FEET

Misc. Defaults
Default File Format
File Format NEF

Default Resolution Setting
Res. Setting L

Overlap Percentage For Path Flights
Overlap Percentage: 60

Figure 4. Camera Parameters View/Edit Dialog Box in ArcView 3.3

Table 1. Camera Parameters Text File

| **Parameter** | **Description** |
|---|---|
| CAMERA NAME | Name of camera for which these parameter apply |
| SENSOR PIXEL SIZE | Size/pitch of each pixel on the camera's sensor |
| SENSOR PIXEL UNITS | Units of pixel size (microns or millimeters) |
| PIXELS ALONG PATH | Number of pixels on the sensor along the flight path |
| PIXELS ACROSS PATH | Number of pixels on the sensor across the flight path |
| LENS NAME | Name of lens being used to fly photography mission |
| CALIBRATED FOCAL LENGTH | Calibrated focal length of the lens being used |
| FOCAL LENGTH UNITS | Units of calibrated focal length (millimeters or inches) |
| DEFAULT FILE TYPE | Default file type setting for the camera (NEF or JPEG) |
| DEFAULT CAMERA RESOLUTION | Default camera resolution/file size setting (S,M, or L) |
| DEFAULT FLYING HEIGHT | Default flying height |
| DEFAULT FLYING HEIGHT UNITS | Units of default flying height (meters or feet) |
| DEFAULT OVERLAP PERCENTAGE | Default overlap percentage for path flights (percent) |
| HORZ. AND VERT. POSITION ERROR ALLOWANCE | Aircraft horizontal & vertical position error allowance (meters) |
| TILT ERROR ALLOWANCE | Aircraft tilt error allowance (degrees) |

*5.2.1.3. Waypoint Database and Attributes.* As each waypoint is created through the use of the custom tools in ArcView, a new shapefile point is created. For each shapefile point, a database entry is created, containing the fields shown in Table 2 with the data type and a brief description of each attribute. Of particular interest are the fields containing the desired flying height, desired airspeed, and the flag indicating whether or not a photo is to be taken at the particular location represented by the waypoint. Additional scripts for optimization of the flight path use other attributes such as the Boolean flag PATH. Regardless of whether waypoints represent an individual location photograph or a photograph within a series of photographs along a path, there is one waypoint present in the database for each photograph to be taken on the mission.

The primary use of the waypoint database is guidance and camera triggering information for use by the in-flight control software. During flights, the pilot is guided to the next waypoint location from the coordinates obtained from the database entry for that waypoint. To ensure that the ground area or distance desired during mission planning stages is captured, the desired elevation and airspeed for the waypoint, computed by mission planning tools in GIS software, is read by the guidance and camera-control software and the pilot is instructed to guide the aircraft to that elevation and airspeed for the particular waypoint. When the aircraft is determined to be at minimum distance from the waypoint's coordinates, the camera is triggered if the PHOTO attribute contains a Boolean value of True, represented as a binary 1. Additional fields read and used by the camera control software allow individual settings for file storage format and camera resolution for each waypoint.

Table 2. Waypoint Database Attributes, Data Types, and Description

| **Field** | **Description** |
|---|---|
| ID | Identification number |
| EASTING | Easting coordinate of waypoint |
| NORTHING | Northing coordinate of waypoint |
| FEAT_NAME | Text name for waypoint |
| PHOTO | Boolean flag: True=take photo here, False=do not take picture |
| CAM_RES | Camera file size setting: S=small, M=medium, L=large |
| FORMAT | File storage format: NEF=camera raw, JPEG=JPEG compressed |
| FLY_HT | Desired flying elevation above mean sea level |
| HT_UNITS | Units for desired flying elevation |
| SPEED | Desired speed over ground for photographing the point in MPH |
| LAT | Latitude for this point in decimal degrees |
| LONG | Longitude for this point in decimal degrees |
| PATH | Boolean flag: True=part of path sequence, False=not part of a path |
| LAT_DMS | Latitude for this point in degrees, minutes, seconds |
| LON_DMS | Longitude for this point in degrees, minutes, seconds |
| CAPTUREX | Easting coordinate at moment of photo capture |
| CAPTUREY | Northing coordinate at moment of photo capture |
| CAPTUREHT | Elevation at moment of photo capture |
| CAPTURECOG | Course Over Ground at moment of capture |
| CAPTURESOG | Speed Over Ground at moment of capture |
| CAPTUREUTC | Universal Time Code at moment of capture |
| VISITED | Flag marking if visited: 0=not visited, 1=visited, 2=skipped/missed |

*5.2.1.4. Correction for Aircraft Position and Attitude Errors.* Ideally, aircraft on photography missions fly at exactly the specified elevation and horizontal position with the aircraft flying perfectly level. In reality, minor errors in horizontal and vertical position are impossible to avoid due to factors such as wind speed, wind direction, the skill of the pilot, reduced precision of GPS guidance equipment, and the precision at which the aircraft can be positioned. Additionally, these same factors cause minor tilt errors to be present at the moment of exposure. Should enough tilt error and position error be present when the exposure is made, the intended ground area may not appear in the photograph in its entirety. For waypoint creation tools that are designed to capture a specific region within each photograph, compensation for these errors must be incorporated to increase the likelihood that the desired areas do, in fact, appear in their respective photographs.

The two modes affected by aircraft position and attitude errors are Mode 2, creation of a single waypoint with user-specified ground area, and Mode 3, creation of multiple waypoints along a user-specified path. In both cases, the user specifies the ground area desired, either by sketch on the map view or by manual entry, expected to appear in their respective photographs. The flying height above terrain is computed based upon these desired ground areas, performed using the photogrammetric scale equation (Eq. 5), solving for height using the ground distance as $AB$ using quantities obtained from the camera parameters text file. Possible aircraft position and attitude errors are corrected separately in the waypoint creation scripts by adding ground distance to the user's specified amount, resulting in a higher computed flying height which in turn increases the area captured in each photograph.

To compensate for horizontal and vertical position error, a predetermined estimate of the horizontal and vertical accuracy within which the pilot is able to guide the plane through the intended location for a photograph. In addition to pilot ability, aircraft capabilities, atmospheric effects such as wind and turbulence, and position error from guidance equipment were considered in determination of the compensation amount required for horizontal and vertical position error. After consulting four experienced pilots at Old Town Aviation, in Old Town, Maine, an estimate of a maximum of 20 meters of horizontal and vertical position error was agreed upon, found in the camera parameters text file and is read by the waypoint creation scripts.

At the moment of exposure, the aircraft may be in any one of the following locations around the desired station: correct horizontal and vertical position; correct horizontal position but too high; correct horizontal position but too low; correct vertical position but incorrect horizontal position; flying too high with incorrect horizontal position; and flying too low with incorrect horizontal position. Flying too high or both flying too high in conjunction with horizontal error are of little concern, as the increased elevation will cause an increase in ground area captured larger than the magnitude of the horizontal position error. Of concern are the conditions in which the ground area captured is reduced, namely horizontal position errors present at correct elevation or flying too low. As illustrated in Figure 5, the combination of flying too low with horizontal position error results in less ground distance lost than the magnitude of horizontal position error alone, therefore, compensation for horizontal and vertical position error is accomplished adequately by adding the horizontal error estimate of 20 meters as previously discussed.

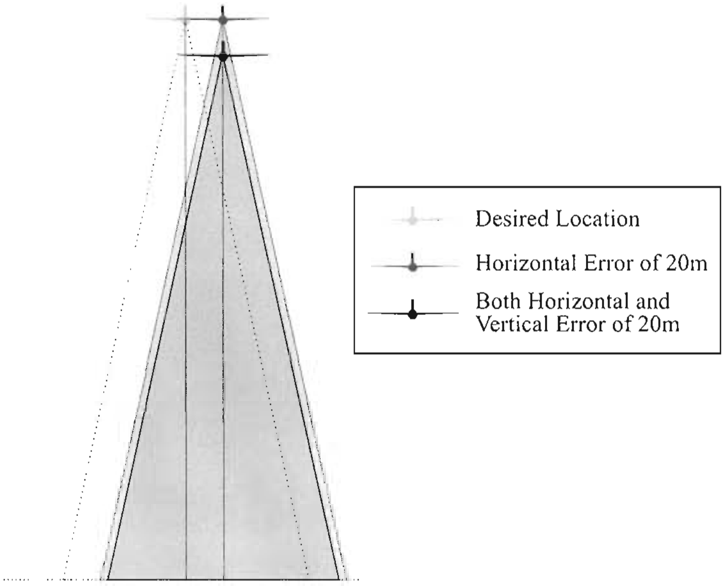


Figure 5. Effects Due to Aircraft Horizontal and Vertical Position Errors

Compensation for aircraft tilt error is accomplished in the same manner as for horizontal and vertical position error: by adding an amount of ground distance to the user's specified ground distance amount, resulting in a higher computed flying height. Causes of tilt error include atmospheric turbulence, wind, flying trim of the aircraft at the moment of exposure, and the pilot's navigation inputs. As the pilot will be instructed to keep the plane as level as possible, effects of navigation inputs and trim of the aircraft should rarely result in more than 3 degrees of tilt (Wolf and Dewitt 2000, ASPRS 1980). Photographs in which the tilt amount is 5 degrees or less can be utilized for photogrammetric procedures provided that corrective procedures are employed on the imagery (Graham and Koh 2002). Therefore, a maximum tilt error at the moment of exposure of 5 degrees was accepted as a reasonable estimate, stored in the camera parameters text file for use by waypoint creation tools. The amount of ground distance added to the user-specified distance is shown as the quantity *I* in Figure 6.

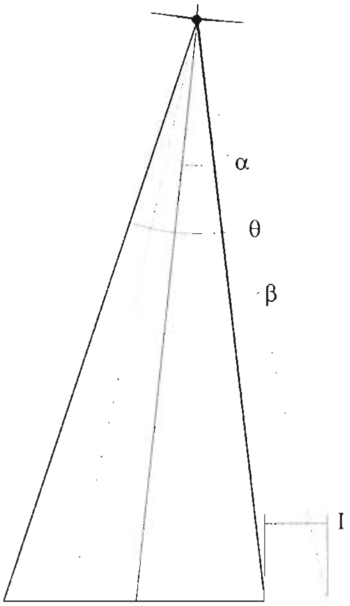


Figure 6. Effects of Aircraft Tilt Error

The quantity α (figure 6) is first computed using equation 8, one-half of the camera across-flight-path angle of view (shown as θ in figure 6), where *ab* is the across-path sensor width and *f* is the calibrate focal length of the lens in use. The tilt error compensation amount *I* to be added to the user-specified ground distance is then computed using the value of α in equation 9, where β is the estimate of maximum tilt error.

$$\alpha = \tan^{-1}\left(\frac{ab}{2*f}\right) \tag{8}$$

$$I = \left(\frac{AB}{2}\right) - \left(\frac{AB*f}{ab}\right) * \tan(\alpha - \beta) \tag{9}$$

*5.2.1.5. Mode 1: Single Waypoint Creation, User-specified Ground Resolution.* Mode 1, intended to be used on small, localized areas for which a desired ground pixel size is the primary concern, rather than the geographic extent visible in the image. The tool is operated in a very simple fashion, requiring the user to activate the tool by clicking upon it on the toolbar, followed by clicking upon the desired location in the map display (illustrated in Figure 7). The user is then prompted for the following: whether the waypoint is a photo location or not, the desired ground resolution, the name for the point, and the desired speed over ground at which the pilot should be flying at the moment of exposure.

Upon obtaining the users desired location and prompted information, the flying height above terrain required to capture the location at the desired resolution is computed using the photogrammetric scale equation (Eq. 5) and camera parameters read from the camera parameters text file. Terrain elevation varies by location, and simply specifying the flying height above terrain to the pilot is not sufficient. Rather than flying height above terrain, the required quantity to be in proper position to photograph the location as desired is the altitude above sea level. For example, if the flying height above terrain is 200 meters and the elevation of the desired location is 300 meters, the altitude at which the aircraft should be flying at the moment of exposure is 200m + 300m = 500 m. Upon computing the flying height above terrain, a digital elevation model (DEM) is queried at the coordinates of the desired location. The terrain elevation at the location is then added to the flying height above terrain yielding the desired elevation above sea level, then written into the waypoint's database table entry with all other attributes as computed and obtained from user prompts.

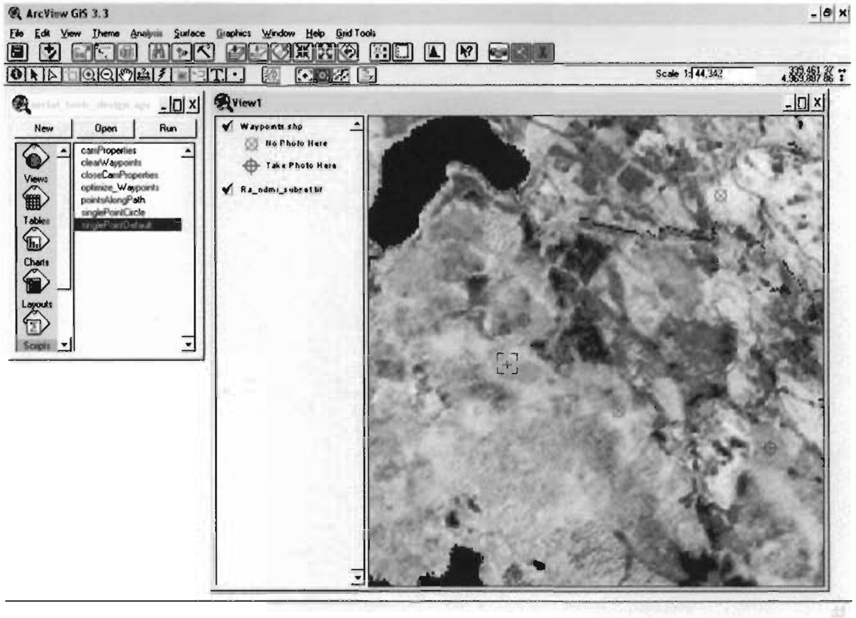


Figure 7. Creating Waypoint with User-Specified Ground Resolution

*5.2.1.6. Mode 2: Single Waypoint Creation, User-specified Ground Area.* Mode 2, creation of a waypoint based upon user-specified ground area, is designed to allow the user to sketch a circle on the GIS view surrounding the area desired to be present in the photograph taken at that location (illustrated in figure 8). The circumference of this circle is used as the user-specified ground distance *AB*, employed to compute the required flying height above terrain (equation 5), solving for *H* using the quantities *ab* and *f* extracted from the camera parameters text file. Prior to computing the required height above terrain, the ground distance *AB* is modified by adding amounts computed as described above to compensate for aircraft position and tilt errors, which may be present at the moment of exposure. As in Mode 1, the user is prompted for additional information such as desired airspeed and whether a photograph should be taken at this point or not. Flying height above terrain is converted into flying elevation in the same manner as Mode 1 by querying a DEM. Remaining fields in the database entry for the waypoint are filled by default values obtained from the camera parameters text file. As a visual cue to the user that Mode 2 is currently active, the pointer cursor symbol used for Mode 2 is different than the symbol used in Mode 1.

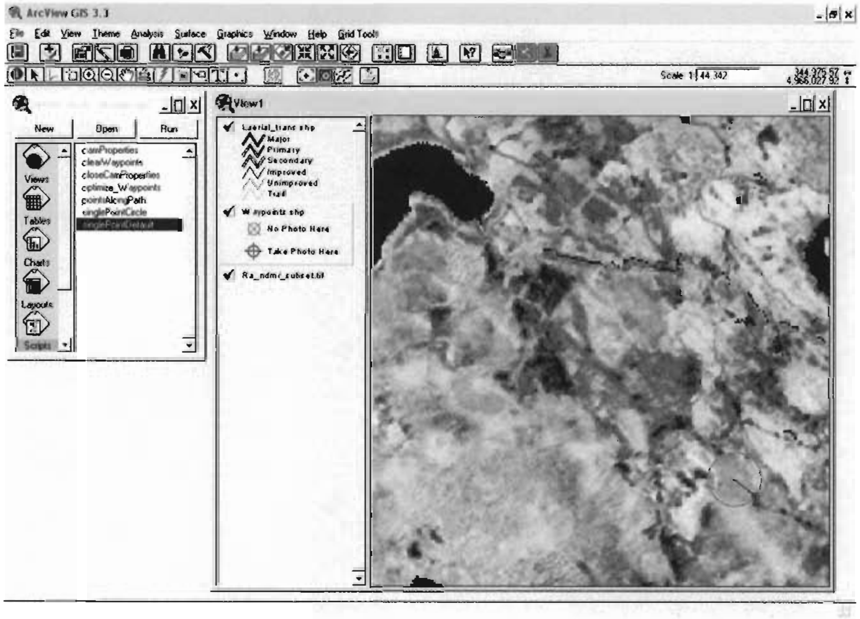


Figure 8. Creating Waypoint with User-specified Ground Area in ArcView 3.3

*5.2.1.7. Mode 3: Series of Waypoints Along a Path.* Mode 3, creation of multiple waypoints along a user-specified path, creates waypoints at computed intervals along linear features such as rivers, allowing photography of buffers surrounding them. For example, a portion of the Saint John River in Maine has a restriction of no harvesting within 1,000 feet of the high water mark. Mode 3 prompts the user to specify the width of buffers on either side of the feature, used to compute flying height above terrain. In addition, the amount of overlap desired for photographs along the line can be entered, allowing stereoscopic coverage of the region (if desired by the user). The operation of this tool is similar to that of Mode 2 with the exception that the user sketches a polyline on the GIS display along the feature that is being monitored. Figure 8 illustrates a series of waypoints created along a user-sketched polyline by the Mode 3 tool implemented in ArcView 3.3. Upon completion of the sketching, the user is prompted for required information such as overlap percentage and the width of the buffer to be photographed. The overlap percentage prompt contains a default overlap amount specified in the camera parameters text file, but can be altered by the user if desired. The flying height above terrain for the path is computed by doubling the user specified buffer width to get ground distance *AB* and adding compensatory amounts to this distance for aircraft position and tilt errors as described above.

Upon computation of the desired flying height above terrain, the ground distance photographed along the flight path of the aircraft is computed (equation 5) and values for focal length *f*, sensor size along flight path are computed from the number of pixels along the flight path in conjunction with the sensor pixel size, obtained from the camera parameters text file. Taking the desired overlap percentage into account, the number of photographs and distance between waypoints required to photograph the entire sketched path is computed. The waypoints are then created along the line sequentially, with two navigation waypoints created prior to the location of the first waypoint to guide the aircraft into position flying at the correct ground track bearing to begin photographing at the first waypoint.

In Modes 1 and 2, database field values for waypoints are filled using values obtained from the camera parameters text file such as default camera resolution settings and default file format. Of particular concern while flying along paths such as those delineated using Mode 3 is the time between each photograph due to the time required for the camera to transfer images to storage after each exposure is made. The time required for file transfer after exposure is dependent upon both the interface employed between the camera and interface and the number of bytes being transferred, which is in turn dependent upon the file storage format. Digital cameras commonly offer several storage options based upon desired output file quality and compression rate. High-quality digital cameras commonly offer an additional raw storage mode that stores un-processed sensor data to be processed later by computer workstation software, yielding a slight detail and resolution advantage over in-camera processing. Other storage options such as in-camera JPEG compression offer smaller file sizes, and in turn shorter file transfer speeds, at a penalty of loss of quality dependent upon the amount of compression chosen. Waypoints created in Modes 1 and 2 are commonly separated by large distances, normally allowing sufficient time between exposures to transfer of files in any storage format, including the camera raw format. In the case of Mode 3 waypoints, flying along paths to photograph narrow buffers with high amounts of overlap may result in waypoints being reached only a few seconds apart, likely at shorter intervals than those required to transfer the higher-quality camera raw files, which will eventually cause missed photographs while waiting for files to transfer. Options for increasing the file transfer speed include: reducing resolution settings, using a higher-speed data interface between the camera and storage device, and employing in-camera data compression such as JPEG compression. Reduction of resolution caused by fliying at a higher altitude is generally undesirable, as the reduction in detail would hinder the image's ability to function in Level-2 inspections. Camera data transfer interfaces cannot normally be upgraded, leaving in-camera data compression as the only viable means of reducing the time required to transfer images after exposure. For this reason, the default file storage format specified in the camera parameters text file is overridden when creating waypoints with the Mode 3 tool, storing the instruction to use in-camera JPEG compression for decreased file transfer times.

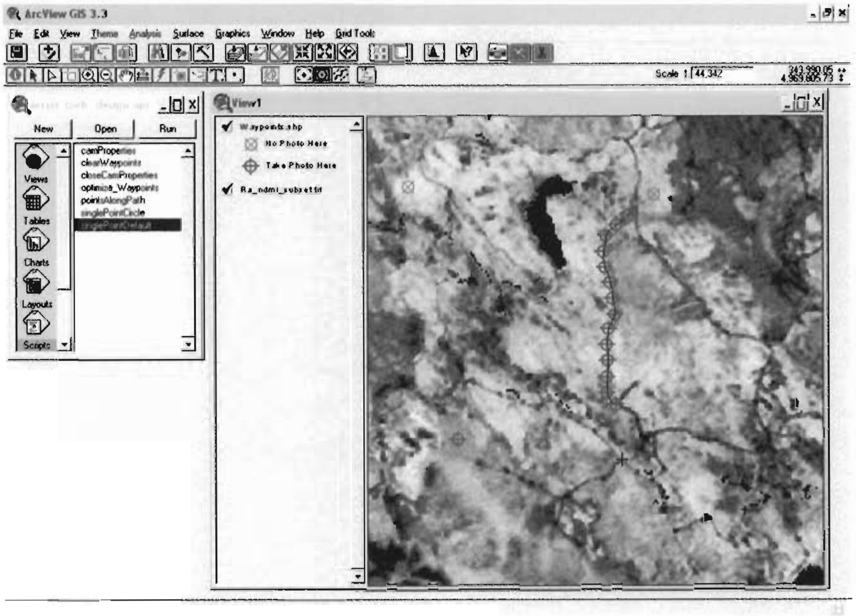


Figure 9. Waypoints Created Along a Path in ArcView 3.3

*5.2.1.8. Flight Path Optimization.* Upon completion of creating waypoints with the tools described above, the user has the option to optimize the sequence in which the aircraft will visit these waypoints during photography missions, activated by clicking upon the customized flight path optimization tool button in ArcView, illustrated in figure 3. The origin point for all photography missions is set to the coordinates of Dewitt Airfield, Old Town, Maine. This location is used as the origin because the aircraft modified to accept the digital camera system is owned and operated by Old Town Aviation at Dewitt Field. The origin point is hard-coded into the optimization script in

ArcView to discourage users from tampering with the origin point, although the coordinates can be altered with little difficulty by editing the script.

The algorithm (flowchart in figure 10) operates on the assumption that the next waypoint to visit from the current location is the waypoint that is closest to the current position, implemented by creating a list of distances from the current location to all remaining waypoints that have not been assigned an ordering number. The list is evaluated and the next waypoint is determined. The order number of the current location is then written to the database entry of that waypoint, the current location is changed to the next closest point, and the new current location is removed from the list of unassigned waypoints. Waypoints created in the waypoint file can contain both individual locations as well as points along a linear path (e.g. for buffer monitoring). It is possible that during flight optimization, the next closest point could be a point within one of these paths. As linear paths are to be flown in sequence from the line start point to end point without interruption, the algorithm was designed to prevent interruption of these sequences. Should the waypoint closest to the current location be part of a path sequence, the algorithm traverses the path back to the first navigation line up waypoint and assigns ordering numbers in sequence for all waypoints of the path, removing them from the list of unassigned waypoints. Upon reaching the last waypoint in the path, the final waypoint in the path is used as the current location and the algorithm returns to normal operation, ceasing when all waypoints have been assigned an ordering number. The waypoint database file can then be sorted by the ordering number and saved for import into in-flight camera control software.

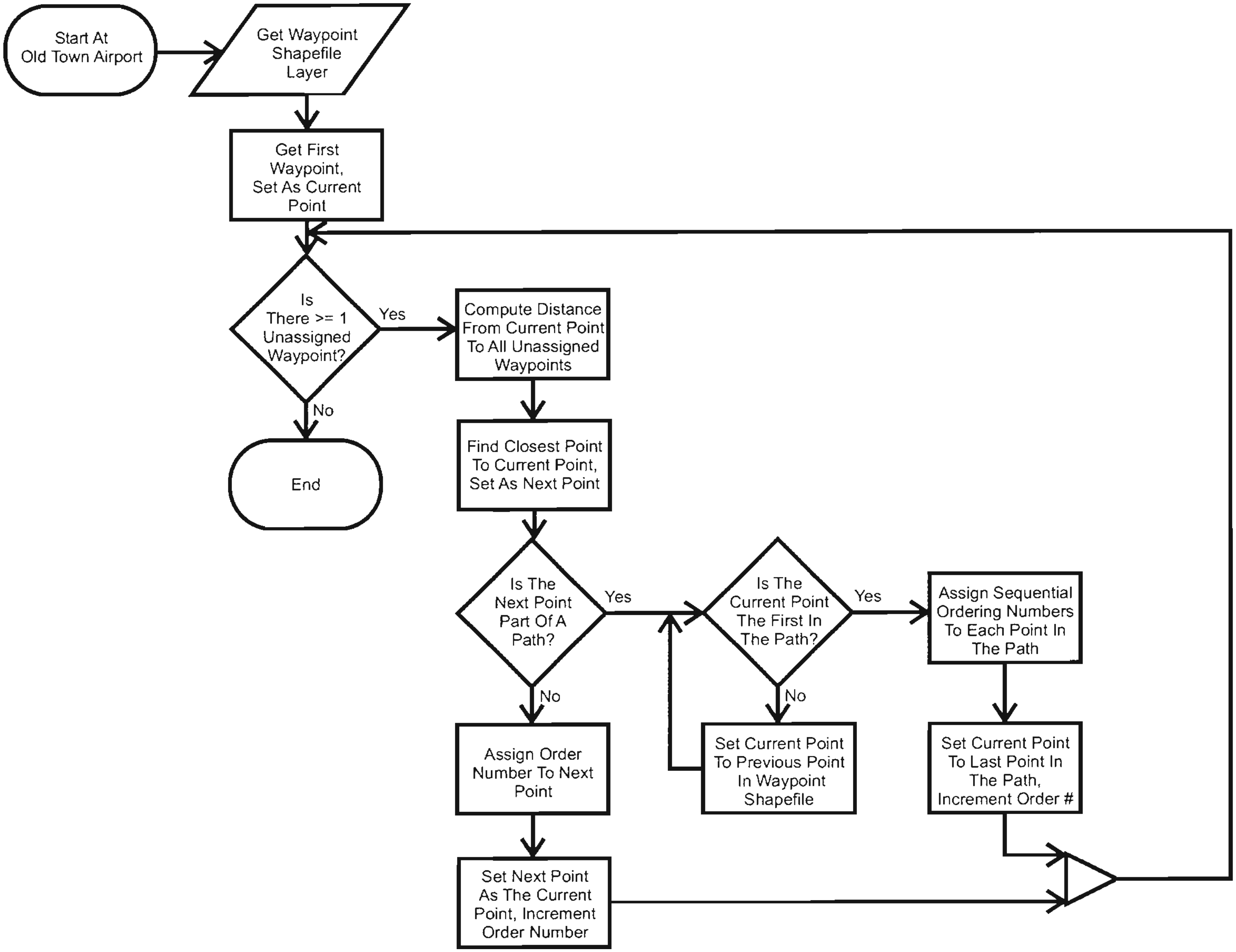


Figure 10. Flowchart For Flight Path Optimization Algorithm

**5.2.2. Mission execution.** Once mission planning is complete, the mission is scheduled and equipment is gathered to make the trip to the airport. The equipment carried by monitoring staff, described in detail in following sections, includes the camera and lens, the mounting bracket required to secure the camera to the aircraft, an external power supply, a small GPS receiver designed to be connected to a computer through a USB port, and a laptop computer for navigation, camera control, and storage of images as they are captured. Installation of the camera into its location on the aircraft takes less than fifteen minutes, including time to clean windows, unpack equipment, and make all required electrical and electronic connections.

During flight, the laptop computer performs the tasks of informing the pilot with course information for navigation, triggering the camera when the aircraft is as close to the waypoints as possible, and storing images transferred from the camera. These operations are carried out by software customized to automate the process of acquiring the desired images, requiring little input from personnel aboard the aircraft. As images are captured, the control system operator reviews the images to verify that the desired target was captured; providing input to the computer if the target was missed and a repeat pass is necessary. To facilitate location of images within existing GIS data in rectification, the control system records the coordinates (obtained from the GPS receiver) at which the image was actually captured. This is useful in creation of hyperlinks in GIS software to more easily locate and add individual images for analysis.

**5.2.3. Post-mission Processing.** Upon returning from the airport, the images captured on the mission are then rectified to assign coordinates and correct for terrain relief effects. The rectification process is carried out first by removing any radial lens distortion present and then rectifying the images using digital orthophoto quadrangles (DOQs) for target coordinate information, supplemented by any other raster or vector GIS data as appropriate. The new images are rectified using at most a second-order ($N=2$) polynomial warping operation in GIS/digital image processing software with the capability of rectifying images for use in GIS software. Third-order polynomial warps will not be employed due to their capacity to warp imagery strongly in unexpected and unrealistic ways. Furthermore, second-order polynomial warping functions have been found to be sufficiently accurate for most purposes as mentioned in Wolberg 1990 and Lillestrand 1972. The rectification process is carried out by carefully selecting locations in both source and target imagery that represent the same location on the ground. Using a least-squares optimization routine, the polynomial coefficients $a_{ij}$ and $b_{ij}$ are calculated and the image is rectified using the resulting polynomial function, resampled with either a bilinear or cubic interpolator. At this time all ancillary files or image header information required to display the image correctly in registration with other GIS data are created as well. Upon completion of rectification, the image is ready to be analyzed in GIS software by monitoring staff for evidence of compliance with or deviation from easement terms as a Level 2 inspection.

### 5.3. Camera System Calibration

The intent of the camera system is to perform as a medium-accuracy digital photogrammetric system, requiring that several parameters unique to the camera be known such as the calibrated focal length and lens distortion characteristics. These parameters, in conjunction with other known quantities such as sensor dimensions and pixel size (provided by the camera manufacturer) are required in mission planning stages to compute the flying height above terrain.

**5.3.1. Focal Length Calibration.** The camera system will be utilized at varying large distances from the object being photographed, requiring discovery of the lens infinity focal length. While the focal length of the lens is commonly found engraved upon the barrel of the lens, the true focal length of the lens can vary from these values. The true focal length value is not normally provided to the purchaser of the lens. Deviations from the engraved focal length of a lens result in variances in manufacturing of each element making up the lens and variations in the positioning of these elements within the lens barrel, with each manufacturer applying their own tolerance ranges. For example, production lenses manufactured by Carl Zeiss may have a statistical variation in focal length between engraved and actual values of between 0.5 and 2.0 percent, with the engraved focal length deviating by as much as 5 percent from the designed value of the lens. In the case of a 24mm lens, the deviation could be as much as 1.2mm using the 5% deviation from design specifications (Dr. Hubert Nasse, Carl Zeiss Camera Lens Division Laboratory, personal communication September 2005). While deviations such as this have little or no impact on normal photographic situations, applications such as photogrammetry require knowledge of these parameters with much higher precision, dictating that calibration procedures be carried out to discover the actual focal length of the lens in use.

Discovery of an approximation of the infinity focal length of the lens in use was performed for the shortest focal length setting and for a slightly longer focal length setting used for all test photos and future photography missions. The focal length at each setting was to be computed to a repeatability of 0.1mm. Discovery of the focal length within 0.1mm allows computations of flying height above terrain to be performed within +/- 0.28% at the 35mm lens setting and +/- 0.40% at the 24mm setting. This requirement results in a maximum error in ground area photographed of +/- 3.87 meters with the 35mm setting and 7.58 meters at the 24mm setting at 10,000 feet above terrain, the maximum height allowed by waypoint creation tools. Algebraically solving equation 5 for $f$ yields equation 10, allowing the computation of focal length at each setting. Photographing an object of known size ($AB$) at measured distances ($H$), measuring the object's size on-sensor ($ab$), and substituting these quantities into equation 10 yields the lens focal length at that distance.

$$f = \frac{H * ab}{AB} \tag{10}$$

*5.3.1.1. Focal Length as a Function of Distance.* Representing a lens as a simple, or thin lens with a thickness of zero, allows representation of the relationship between lens focal length, object distance, and image distance, yielding the well-known lensmaker's formula or thin lens equation (equation 11) found in many introductory textbooks on optics. The particularly important factor of equation 11 is that, as the object distance $O$ approaches infinity, the image distance $i$ approaches the focal length $f$ of the lens, converging upon the true focal length. When the object distance is sufficiently large and the image distance does not change significantly with a further increase in object distance, the approximate infinity focus has been discovered and the computed focal length at that distance can act as a substitute for the true focal length of the lens.

$$\frac{1}{f} = \frac{1}{i} + \frac{1}{O} \tag{11}$$

To compute the focal length at the two lens settings, an object of known size was photographed at known measured distances. The sensor size is known from the camera manufacturer's specifications, allowing measurement of image distances (*ab*) captured by the sensor by using digital image processing software. Photographing an object of known length (*AB*) on a test range at known, measured distances (*H*) and substituting these known quantities into equation 10 allows computation of the focal length of the lens.

*5.3.1.2. Focal Length Testing.* The test range was Morse Field at the Alfond Stadium at University of Maine. This football field is an artificial turf surface that has been repeatedly surveyed by intermediate land surveying students for several years and has been verified that the yard markings are accurate within the measuring ability of modern surveying instruments. Additional real-time kinematic (RTK) global positioning system (GPS) surveying equipment has been employed by advanced land surveying students to verify that the field is also flat and level within measuring limits.

The camera was set up on a Bogen/Manfrotto 3011 tripod directly above the field's center marking along the end-zone line (zero-yard line) and leveled for both forward/back tilt and left/right tilt using a hot-shoe mounted bubble level. The camera rotation was adjusted by observing the angles indicated on the tripod head while the camera central focusing point intersected both end-zone corners in the viewfinder. By bisecting the total angle, the camera was aligned with the center markings and the opposite end zone's uprights, assumed to be the correct rotation angle to ensure that the sensor surface is parallel to the marked lines on the field.

The object photographed was a 4.3-meter (14-foot) maximum adjustable length black painted aluminum Bogen/Manfrotto Auto-pole, with white tape attached at either end for contrast in measuring its length. The length of the pole was measured, prior to the tests, to a precision of 0.003 meters (0.01 feet) with a steel measuring tape, repeated again after the test to ensure that the length did not change during the test due to temperature variations or slippage in the adjustment mechanism. At each distance, the pole was photographed a minimum of four times, varying the position slightly in the camera's viewing area as well as photographing the pole in horizontal orientation to check that the focal length did not change along the axes of the sensor. The pole was leveled carefully for each photograph using a small bubble level. An assistant triggered the camera when the signal was given that the pole was level. For each focal length setting, the pole was photographed at distances of 18.3, 27.4, 36.6, 45.7, 54.9, and 100.6 meters from the camera location to discover at what object distance the computed focal length no longer changes within the 0.1mm precision specification, indicating the approximated infinity focus distance.

Upon completion of taking photographs, the images were retrieved from the camera and the object size was measured using Adobe Photoshop's distance measurement tool. For each image, the auto-pole's image length was measured between the white tape markings, measured in number of pixels. These image length measurements (pixel counts) were entered manually, one-by-one into a Microsoft Excel spreadsheet, and converted to sensor distance units by multiplying the pixel count by the physical size of the pixels as provided by the camera manufacturer. Converting all known quantities (*ab, AB, H*) to meters and substituting these values into equation 10 allowed computation of the focal length at each distance, computed to the nearest tenth of a millimeter as previously specified. To compute an approximation of the infinity focal length of the lens setting, all computed focal lengths for object distances equal to and greater than the approximate infinity focus distance were analyzed using an $F$-test to determine if a statistical relationship between computed focal length and object distance existed. If none existed, these computed focal lengths were then used in the computation of the mean infinity focal length with a 95% confidence interval.

**5.3.2. Distortion Characteristics Calibration.** Upon returning from a photography mission, the images captured require rectification in order to be used in GIS software with other geographic data in proper co-registration. Prior to performing the actual rectification, any distortions in the image caused by lens aberrations should be removed. To perform distortion removal procedures, the distortion characteristics of the lens must be known and expressed using parameters that distortion removal software can accept as input. Radial lens distortion characteristics were discovered at both the 24mm and 35mm marked focal length settings of the lens. For convenience, the alternative radial lens distortion expression in equation 4 was used. The goal of the camera system is to allow medium accuracy photogrammetric operations, requiring only the discovery of one radial lens distortion parameter, $c$. Due to its relationship with $k1$, the $c$ parameter accounts for the majority of radial lens distortion (Brown 1971), however, because software allows convenient computation of multiple parameters, the distortion parameters were computed twice, once for only the $c$ parameter, and once for both $a$ and $c$.

Distortion function coefficients were discovered using a plumb-line calibration procedure to discover one distortion function parameter or a combination of two distortion function parameters by photographing a large building in downtown Bangor, Maine (see figure 11) from a large distance as such a height that the camera was able to be pointing directly toward the building without convergence of vertical or horizontal lines. The building selected was chosen for its ample supply of straight horizontal and vertical lines along windows and facades, photographed from the third level of a parking garage across the street. The camera was attached to a Bogen/Manfrotto 3011 tripod at 60 inches height and carefully leveled in both left/right and forward/backward tilt directions. The camera horizontal rotation was adjusted by aligning several lines of windows with superimposed grid lines in the camera viewfinder to minimize convergence of lines toward the left or right, bringing the sensor surface plane as close as possible to the plane of the building face. A cable release was used to trigger the camera and the camera raw storage format was used to maximize sharpness and detail in the images. The photographs were converted to TIFF images with no compression to preserve detail, followed by a high-pass sharpening filter with a radius of 1.3 pixels.

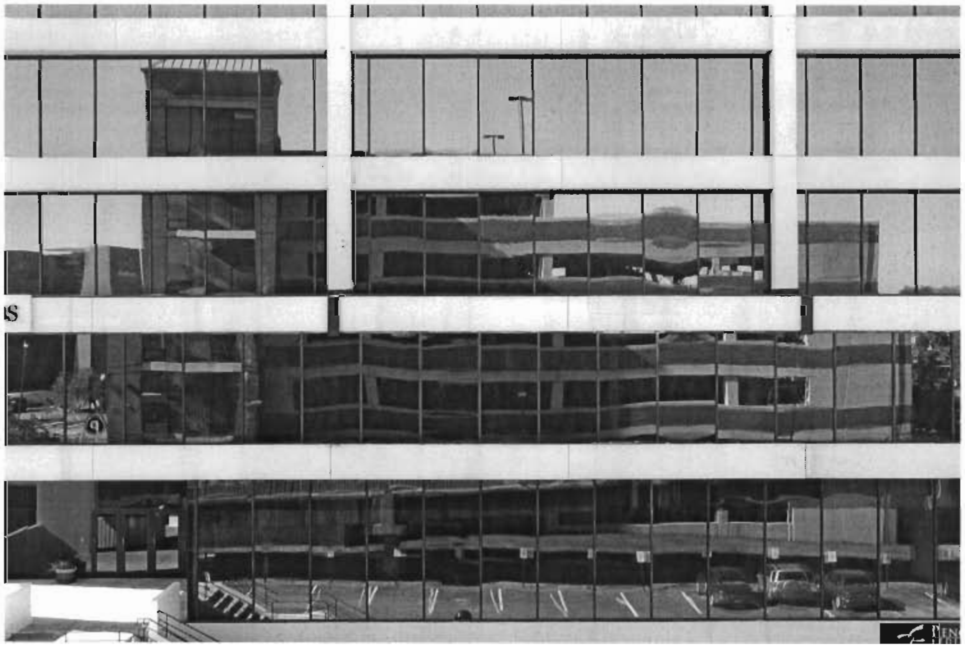


Figure 11. Building Face Used for Plumb-Line Calibration

Panoramic stitching software PTGUI was used to discover lens distortion function parameters using its built-in least-squares optimizer. Control points were added, defining lines distributed throughout the image that should be rendered as straight lines as described by Thomas Neimann (http://epaperpress.com/ptlens/calibrate.html, last accessed November 2005). Particularly, lines toward the edges of the image were included where radial lens distortion effects tend to have the greatest magnitude. For the 24mm lens setting, a total of 138 control points were created, defining 12 separate horizontal and vertical lines. For the 35mm lens setting, a total of 140 control points were identified, defining 8 separate horizontal lines. To complete plumb-line procedures, the distortion function parameters were computed with the least-squares optimizer built into PTGUI. Upon discovery of distortion function parameters, removal of distortion was performed using the Panoramic Tools PTCorrect plugin for Adobe Photoshop, which accepts the parameters *a, b, c,* and *d* as input.

### 5.4. Mounting Camera System to Aircraft

To be useful in capturing imagery to be incorporated in GIS databases for making measurements, the photographs taken by the aerial camera system must be nearly vertical as defined previously. Additionally, the camera system will be exposed to potentially damaging shock forces and constant vibrations due to the aircraft engine. Not only can the vibrations from the engine damage sensitive electronic and mechanical components inside the camera, but the vibrations can introduce image degradation in the form of motion blur, resulting in reduced detail and sharpness. The camera system must be held securely in vertical orientation and protected from shock forces and vibrations to prevent damage and image degradation. Furthermore, the camera system must be mounted in a location on the aircraft that affords an unobstructed view of the target area and does not alter the aircraft's structural integrity or flying characteristics significantly. For the purpose of securely holding the camera in vertical orientation and isolating the camera from vibration and shock forces, a mounting bracket was designed and built, constructed to allow multiple possible mounting locations on the aircraft. Once the mounting bracket was constructed, the mounting location on the aircraft was determined and the proper safety certification process from the Federal Aviation Administration (FAA) was completed.

**5.4.1. Mounting Bracket.** The camera mounting bracket designed to secure the camera in vertical orientation was constructed with the following design ideals: securing the system in vertical orientation, allowing multiple mounting locations and orientations, and protecting the camera from vibration and shock force. The mounting bracket constructed appears in figure 12, shown with the camera system installed.

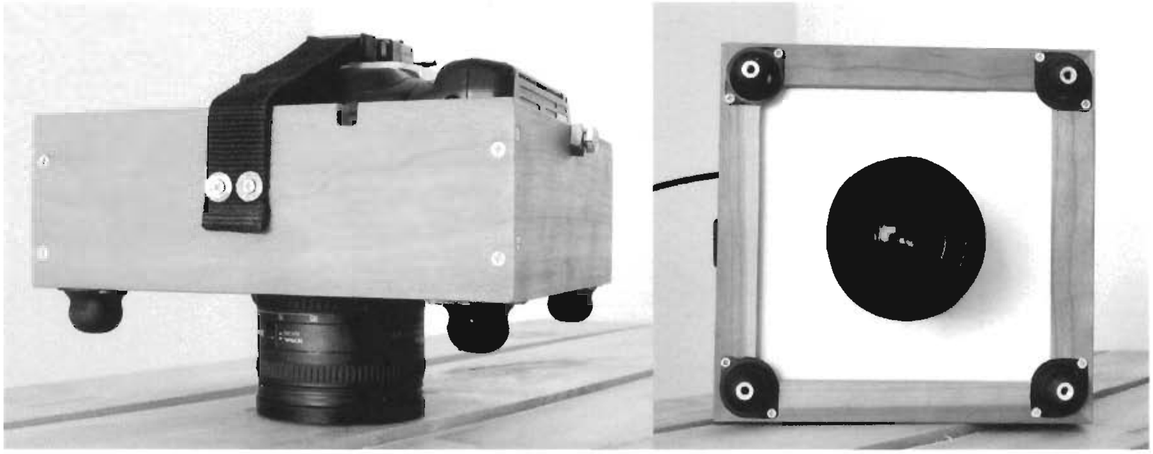


Figure 12. Camera Mounting Bracket with Camera Installed

To accommodate positioning the camera with the sensor's maximum dimension either across or along the flight path, the mounting bracket was designed to be square in the width and length dimensions, with the default orientation along the flight path to facilitate capture of overlapping images. The bracket is constructed of solid precision-milled hardwood shadowbox picture frame moulding. The walls of the mount are 1/2 inches thick and 2-3/4 inches tall with a 1/4-inch thick lip at the bottom that protrudes 7/8 inches from the outer edge of the wall, forming a strong "L" shape. The four wall sections were cut to size using a precision knife chopper at a local custom frame shop and assembled using high-strength wood construction glue and four brass screws in each corner for strength. A platform was fabricated to drop into the bracket to rest the camera on, built using 1/4-inch thick solid wood. A 3-1/4 inch diameter hole was cut into the center of the platform to allow the lens to protrude through, and the platform was secured in place using four metal "L" brackets, screwed to the inside of the camera bracket to the walls.

To ensure that the camera is held securely and cannot easily become unattached to the bracket, combinations of methods were used to secure the camera. The main attachment is accomplished by a ¼-20 bolt threaded into the tripod mount on the camera base through a hole drilled in the bracket wall. This attachment method is very strong and will likely never fail, however, vibration could potentially loosen the bolt on long flights, so a secondary system was added to secure the camera, consisting of a padded adjustable 1-inch wide nylon webbing belt screwed to the walls of the mounting bracket and closed by Fastex plastic quick-release buckles.

Protection from shock force and vibrations was accomplished in two ways: neoprene rubber vibration isolation mounts on the external corners of the bracket and closed-cell foam padding within the bracket to accommodate the shape of the camera and provide extra shock protection. The closed-cell foam within the bracket was cut to fit the shape of the camera body to fill gaps between portions of the camera and the birchwood platform, further protecting the camera from flexing and potentially impacting the platform in a shock force event such as heavy turbulence. Without this extra protection, the camera tripod mount would be required to absorb all shock forces encountered, potentially flexing the tripod mount enough to bend or break. To protect the entire camera mounting bracket from shock and vibration, neoprene vibration isolation cup mounts (figure 13) were attached the base of the bracket at each of the four corners, each with a threaded bolt hole in the center to allow the bracket assembly to be firmly attached to mounting points on the aircraft. The mounts are designed specifically to shield sensitive electronic equipment in ambulances and small aircraft from physical shock and vibrations, rated to absorb over 80% of vibrations and support a maximum vertical load of 4 pounds each. With the camera, lens, batteries, and bracket weighing a total of 4.5 pounds, the vibration mounts are only experiencing roughly 1/4 of their rated maximum load, leaving ample absorption capacity in the event of severe turbulence.

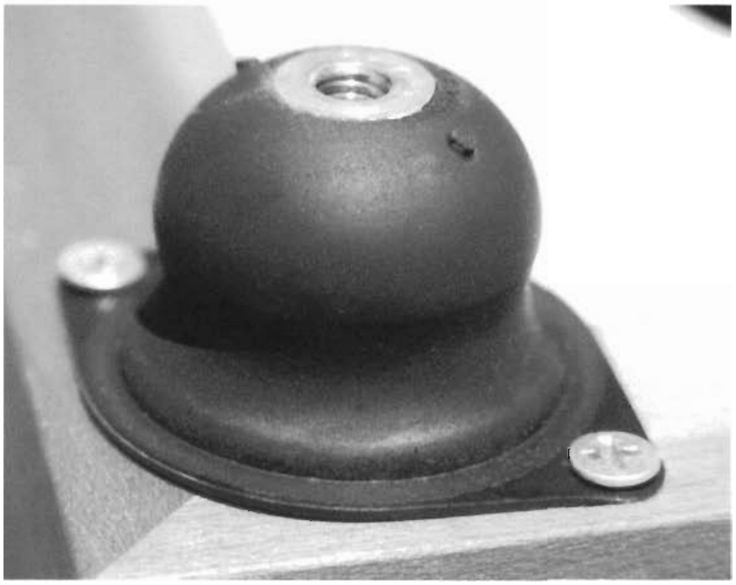


Figure 13. Neoprene Rubber Shock/Vibration Absorption Mounts

**5.4.2. Possible Mounting Locations.** Choice of proper mounting locations on the aircraft to be used for flying was performed with consultation of Old Town Aviation in Old Town Maine. Several potential mounting locations were identified, each with their own advantages and disadvantages. The criteria used in choosing a mounting location included ease of mounting, minimal modification to the aircraft, lack of obstructions in the camera field of view, structural integrity for mounting location to prevent loss of or damage to equipment or the aircraft, minimizing impact upon aircraft flight characteristics, avoiding interference with aircraft control surfaces, and minimizing expense for construction of mounting assemblies and modifications to aircraft. Upon repeated consultation with Old Town Aviation owners and maintenance staff, two candidates for the camera system mounting location emerged: mounting the camera outside the baggage compartment door and modification of the aircraft to allow mounting the camera system inside the aircraft's baggage compartment.

*5.4.2.1. Mounting Out Baggage Compartment Door.* Mounting the camera system outside the baggage door entails removing the baggage door and mounting the camera to an assembly fastened to hard points on the floor of the baggage compartment. This location affords an unobstructed view beneath the aircraft and would permit photographs to be made without image degradations caused by photographing through glass or plexiglass panels. Structurally, the aircraft would not require modification other than removal of the baggage compartment door, minimizing costs. The camera system would only be required to protrude out of the aircraft a small amount and would not interfere with control surfaces on the aircraft; however, the external mounting location places the camera system in a more precarious place. In the event of mechanical or structural failure of the mounting assemblies, the external location would almost certainly result in complete loss of the camera system unless adequate redundant securing methods were employed.

*5.4.2.2. Mounting Inside Aircraft in Baggage Compartment.* Mounting the camera system within the aircraft offers several advantages including drastically improved security for the camera system, eliminating the need to fly with the baggage door removed and thereby leaving the aircraft's flight characteristics intact, and simpler mounting that only requires attachment of the existing camera mounting bracket to the floor of the baggage compartment. The camera system could be mounted very quickly and easily, eliminating the need for bulky or inconvenient mounting assemblies and, due to the camera system being completely enclosed within the aircraft, eliminates the chance of losing the camera system or damaging control surfaces of the aircraft in the event of structural failure. The disadvantages to this mounting location are that cutting holes in the aircraft would be necessary and FAA approval would be required for the modifications made to the aircraft, causing delays and increasing expense.

The FAA approval process for modifications to aircraft can be complex and expensive. The owners of Old Town Aviation stated that for the baggage compartment internal mounting location, the modifications would require the completion of one of two possible FAA approval processes: a Form 337 Field Approval or a Supplimental Type Certificate (STC). These processes are intended to ensure the safety and integrity of the aircraft when major modifications are made, such as cutting holes in the aircraft, whether internal or external. While both processes can be expensive, the Field Approval process is much quicker and far less expensive, requiring only minimal engineering and no flight-testing. The plan proposed by the owner of Old Town Aviation was to cut two holes in the baggage compartment, one in the floor of the compartment and one on the exterior skin of the aircraft, which are the proper size to be covered using standard inspection plates. The theory behind this use of standard inspection plate holes is that only a Field Approval would be required as many aircraft operators routinely install new inspection plates on their aircraft. The hole cut into the skin of the aircraft would be covered with a specially modified inspection plate cover that has a round port cut into it and covering the port with optically transparent material to allow photographs. While the owner felt that the probability of being able to employ a Field Approval for this modification is very high, there was still the danger that the FAA inspectors may decide that the much more complex, expensive, and time-consuming STC process would be required. An initial estimate of $1500 was given for the baggage compartment mounting location's modification and FAA Field Approval, with a possibility of much greater expense if more rigorous FAA approval procedures are necessary.

**5.4.3. Aircraft Modification and FAA Approval Process.** Regardless of the mounting location chosen, the mounting of the camera system to the aircraft requires approval from the Federal Aviation Administration, as do all modifications to aircraft in the United States. The objectives behind the approval process are to ensure aviation safety while allowing innovation in civil aviation. The safety standards defined by the FAA are contained within Title 14 of the Code of Federal Regulations (CFR), Part 21. FAA approval is required prior to making any modifications to an aircraft and the aircraft owner is required to demonstrate that the modification is in compliance with the regulations outlines in 14 CFR Part 21 to gain FAA approval.

*5.4.3.1. Types of Certification Approvals.* Modifications to aircraft are subdivided into minor and major changes (14 CFR 21§21.95). Minor changes are defined as those that do not appreciably affect weight, balance, structural strength, reliability, operational characteristics, airworthiness characteristics, power and noise characteristics, or emissions. Major changes are defined as modifications that are not minor, meaning any change that alters any of the above characteristics. While minor changes may be approved by a variety of methods, major modifications require either a Form 337 Field Approval for Major Modification or a Supplemental Type Certificate (STC) that allows changes to be made to the particular aircraft/engine/propeller's Type Certificate (14 CFR 21 §21.19).

*5.4.3.2. Form 337 Field Approval for Major Modification.* The FAA website defines a field approval as a maintenance performance approval for a major repair or major alteration that is performed by a Flight Standards Service, Aviation Safety Inspector. Some major alterations are not eligible for field approval and must be approved under another method such as STC. These alterations are identified in FAA Order 8300.10, Vol. 2, and include modifications such as engine changes or employing a non-standard propeller. The steps required in obtaining a Form 337 Field Approval are found in table 3.

*5.4.3.3. STC Application Process.* Application for an STC is accomplished in four basic steps: 1-Submission of FAA Form 8110-12, Application for Type Certificate, Production Certificate, or Supplemental Type Certificate, with accompanying data, 2-Inspection and testing of detail parts, components, and subassemblies, 3-Inspection and tests of the complete assembly, modification and installation, and 4-Issuance of the STC.

Information required to submit Form 8110-12 includes a description of the project, the type of aircraft involved, a schedule for completion, where the work will be conducted and installation performed, detailed drawings of modifications to be made and schematics of all affected aircraft components, testing plans, and plans for inspection. Additionally, a local FAA engineer must evaluate the modifications and testing data.

**5.4.4. Mounting Location Selection.** After repeated consultations with the owner and mechanics of Old Town Aviation the decision was made to attempt to employ the baggage door mounting location because permanent modifications to the aircraft would not be required. Old Town Aviation technicians constructed an aluminum mounting assembly to securely suspend the camera bracket out the baggage door, protruding out the doorway to prevent the doorframe from appearing in photographs, constructed in such a manner to allow rapid installation and removal from the aircraft.

Table 3. Steps of The Field Approval Process

(Source: http://www.faa.gov/ aircraft/air_cert/design_approvals/field_approvals, Last Accessed October 5, 2005):

- The applicant proposes to repair or alter one serial numbered aircraft.
    - The applicant must determine that the change is a major alteration or repair per 14 CFR 1.1 and 14 CFR part 43, Appendix A;
- The change is annotated on a FAA Form 337, Major Repair and Alteration;
- The applicant submits FAA Form 337 annotating the change with the data package to the Flight Standards District Office;
- The Flight Standards District Office may meet to assess the scope, complexity of change in light of 14 CFR 1.1 definitions and 14 CFR part 43, Appendix A. The Flight Standards District Office determines that either:
    - The data are adequate and no field approval is required.
    - The Aviation Safety Inspector can sign Block 3 of FAA Form 337 to approve the repair or alteration, or
    - Additional data from the applicant are needed if the original data package are found to be inadequate, or
    - The data needs Aircraft Certification Office review in light of its complexity or adequacy, or
    - The alteration is of a type listed in FAA Orders 8300.10 which exceed the basic scope of a Field Approval and must be processed as an STC.
- If the Aircraft Certification Office reviews the data, they may:
    - Determine that the data package is acceptable as is and can be approved as a Field Approval;
    - Support the field approval with engineering review, advocate additional data or testing, assist with the flight test and Airplane Flight Manual supplements;
    - Recommend that the project should be an Aircraft Certification Office managed Supplemental Type Certification project, and should proceed with the Supplemental Type Certification process.
- The Inspector approves the repair or alteration by signing block 3 of Form 337.
- Owners, operators, and persons who repair or alter aircraft, FAA Flight Standards Inspectors, FAA Aircraft Certification Office Engineers, and DERs need to know when a field approval is made.

Upon completion of the aluminum mounting assembly, the camera bracket was attached in position on the aircraft to determine the amount of the bracket that would be exposed to the slipstream during flight. Exposing the camera bracket to the slipstream introduces problems: the air temperature outside the aircraft can be very low, reducing the camera batteries' run-time and potentially rendering the camera inoperable, and exposure of the camera bracket assembly to strong shearing forces due to the drag of the bracket in the slipstream. Additionally, the vibration/shock isolation mounts employed on the camera bracket are not capable of withstanding the high shear forces encountered due to wind resistance and drag in the slipstream, either eliminating their usefulness or placing the camera system in danger of being torn off the aircraft. To deal with the problems introduced by choosing the external mounting location, a protective shield or "fairing" was required.

A fairing was designed and built for the camera system to overcome the problems of wind resistance and cold temperatures. A composite fiberglass over insulating foam airfoil fairing was constructed to solve these problems. The combination of fiberglass and resin skin over lightweight rigid insulation foam has been employed for decades to construct complex and lightweight but extremely rigid shapes in aeronautics and hydrodynamics. Additionally, the insulation foam would serve to limit the camera battery capacity reducing effects of low temperatures encountered in the slipstream as well as reducing the amount of wind allowed to enter the cabin through the open baggage door.

The fairing required using an aerodynamic shape designed to minimize the drag introduced as a result of placing the camera bracket into the slipstream of the aircraft.

Research into aerodynamic shapes was undertaken and after consultation with senior-level aeronautical engineering students at Utah State University, in Logan Utah, the decision was made to employ an NACA 0035 airfoil shape for the fairing, a well-known and effective airfoil. The NACA designation 00xx indicates that this is a zero-lift airfoil when the flow of air is presented directly along the chord axis (zero degrees angle of attack) with the largest width 35 percent of the chord length of the airfoil, yielding an airfoil profile that is 24.2 inches long and 8.5 inches tall when fitting a chamber within sufficiently large enough to allow proper movement space for the vibration isolation mounts to operate correctly, illustrated in figure 14. Point coordinates defining the shape of the NACA 0035 airfoil were generated using Javafoil online airfoil software (http://www.mh-aerotools.de/airfoils/javafoil.htm, last accessed November 2005). A template for the airfoil was printed and transferred onto nine pieces of one-inch thick rigid insulation foam, cut with a band saw and sandwiched together using high-strength epoxy to form the final foam airfoil shape, 24.2 inches long, 8.5 inches tall, protruding out of the aircraft 9 inches.

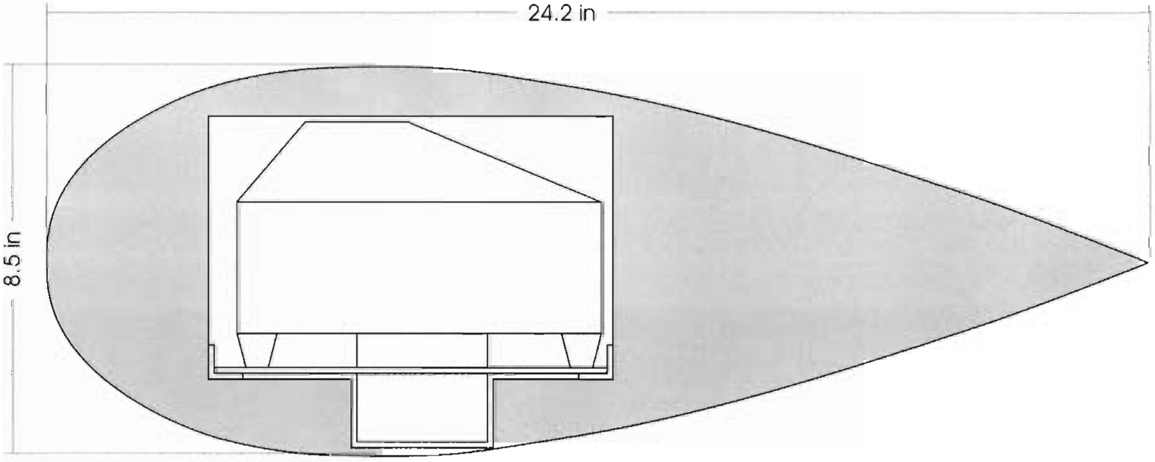
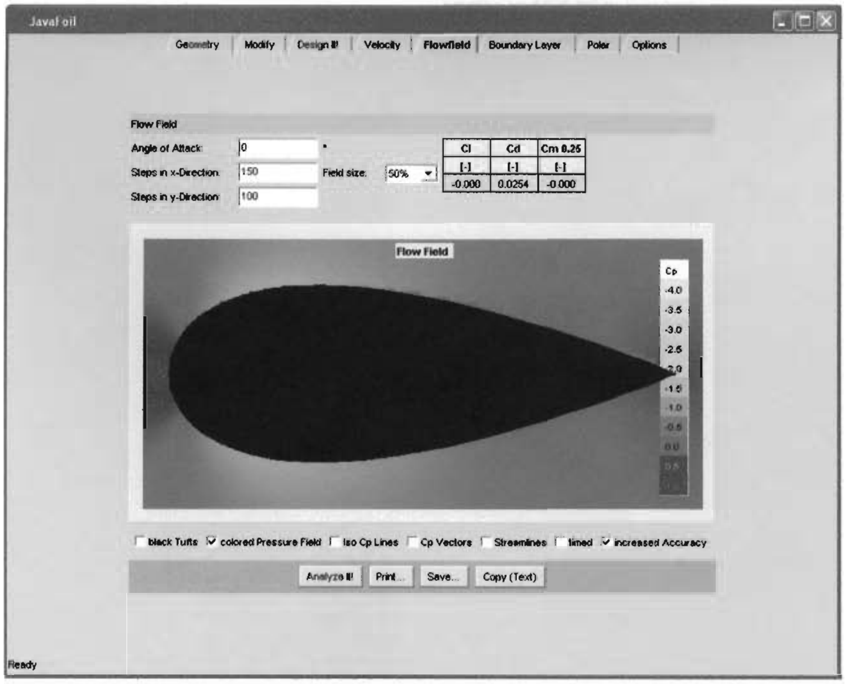


Figure 14. NACA 0035 Airfoil Profile with Camera Chamber Outline

Upon completion of initial construction stages, the foam structure, aluminum mounting assembly, and camera bracket was delivered to Old Town aviation for evaluation by FAA inspectors along with estimates of drag and lift during takeoff and cruising. As the aircraft begins to lift off during takeoff, the nose of the aircraft is raised, temporarily changing the angle at which the slipstream encounters the shape of the airfoil of the fairing, resulting in generation of lift force due to the reduction of air pressure on the upper surface of the airfoil. To compute the lift and drag forces encountered at takeoff, aeronautical engineering students at Utah State University were consulted for an estimate of the angle of attack at which the slipstream would encounter the fairing. An estimate of ten degrees angle of attack was suggested, as larger angles would require excessively steep rates of climb that pilots normally avoid. Coefficients for lift and drag computations at zero degree and ten degree airflow angles of attack were obtained to allow computation of drag and lift forces while cruising and while taking off. The accompanying plots were generated to illustrate the pressure field formed around the airfoil at zero degrees angle of attack and 10 degrees angle of attack, appearing in figure 15 and 16 respectively.



Flow Field

Angle of Attack: 0 °

Steps in x-Direction: 150

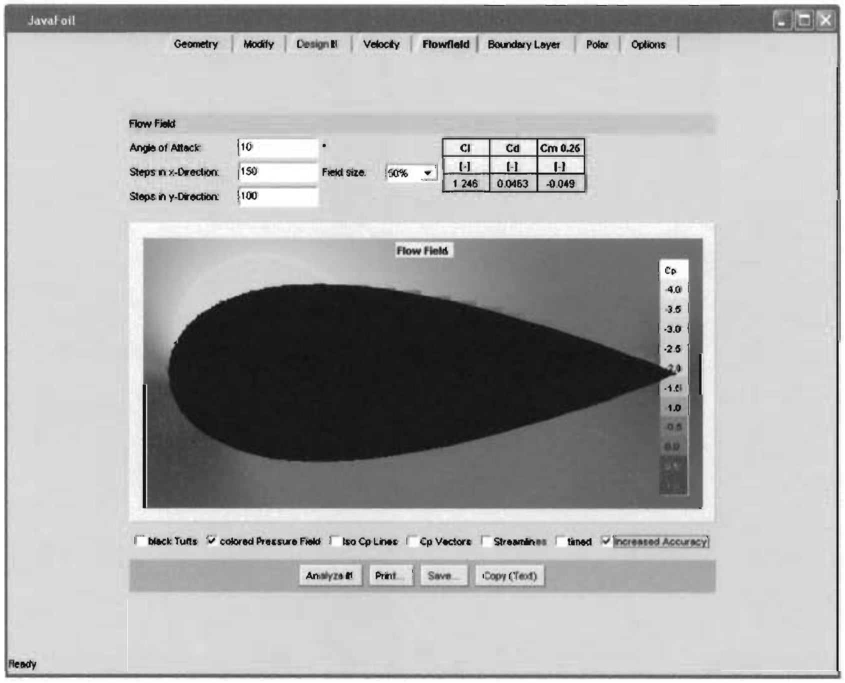
Steps in y-Direction: 100

Field size: 50%

| Cl | Cd | Cm 0.25 |
|---|---|---|
| [-] | [-] | [-] |
| -0.000 | 0.0254 | -0.000 |

☐ black Tufts ☑ colored Pressure Field ☐ Iso Cp Lines ☐ Cp Vectors ☐ Streamlines ☐ timed ☑ increased Accuracy

Figure 15. Javafoil Flow Field Pressure Plot For NACA 0035 Airfoil, 0-Degree Angle of Attack (Cruising Orientation)



Flow Field

Angle of Attack: 10 °

Steps in x-Direction: 150

Steps in y-Direction: 100

Field size: 50%

| Cl | Cd | Cm 0.25 |
|---|---|---|
| [-] | [-] | [-] |
| 1.246 | 0.0463 | -0.049 |

☐ black Tufts ☑ colored Pressure Field ☐ Iso Cp Lines ☐ Cp Vectors ☐ Streamlines ☐ timed ☑ increased Accuracy

Figure 16. Javafoil Flow Field Pressure Plot For NACA 0035 Airfoil, 10-Degree Angle of Attack (Takeoff Orientation)

The computed drag and lift estimates for takeoff and cruising appear in table 4 where $\alpha$=angle of attack, $V$=airflow velocity, $\rho$=air density, $C_l$=coefficient of lift, $C_d$=coefficient of drag, $A_p$=planform wing area, $A_f$=wing frontal area, $L$=computed lift, and $D$=computed drag. The values used for these quantities were obtained in the following fashion: $V$ for takeoff and cruising was based upon specifications obtained from Old Town Aviation pilots, $\rho$ was interpolated from an air density at altitude table (www.coolingzone.com) using estimates for elevation at takeoff and common cruising altitude, $C_l$ and $C_d$ were obtained from Javafoil output, and $A_p$ and $A_f$ were computed using measurements obtained from the foam airfoil fairing design drawing in CorelDraw, rotated 10 degrees for takeoff area estimates. Equations 12 and 13, obtained from NASA's Glenn Research Center basic aerodynamics web pages, allowed computation of the lift and drag estimates found in table 4, presented to FAA inspectors.

$$L = C_l * \rho * \frac{V^2}{2} * A_p \tag{12}$$

$$D = C_d * \rho * \frac{V^2}{2} * A_f \tag{13}$$

FAA inspectors expressed concerns regarding the size of the airfoil and its position relative to tail control surfaces, resulting in the opinion that approval from the FAA was highly unlikely for the baggage door mounting location. Despite the low computed drag introduced by the airfoil, the estimates of the amount of lift generated at takeoff suggests that this airfoil shape at this size generates a noteworthy amount of lift. The concern expressed by FAA inspectors was that despite the short duration in which lift force would be generated, repeated applications of the lift force would be more force than the aluminum mounting assembly could withstand over time, causing it to be damaged or break off entirely and impact the tail of the aircraft, potentially rendering the plane uncontrollable. Furthermore, no engineering or wind tunnel studies had been performed to estimate the likelihood of catastrophic failure of the airfoil structure from normal use and the estimated amount of operation time before structural failure was likely to occur. Expensive and time-consuming engineering and testing would likely be required in order to show that FAA regulations could be satisfied.

Table 4. Lift and Drag Estimates for NACA 0035 Airfoil Camera Bracket Fairing

| | **Takeoff** | **Cruising** |
|---|---:|---:|
| $\alpha$ | 10 deg. | 0 deg. |
| $V$ | 30.8 m/s | 61.7 m/s |
| ***Elev.*** | 120 ft = 36.6 m | 3000 ft = 914.4 m |
| $\rho$ | 1.221 kg/m$^3$ | 1.122 kg/m$^3$ |
| $C_l$ | 1.2460 | 0.0000 |
| $C_d$ | 0.0463 | 0.0254 |
| $A_p$ | 0.1382 m$^2$ | 0.1405 m$^2$ |
| $A_f$ | 0.0521 m$^2$ | 0.0465 m$^2$ |
| $L$ | 99.7 N = 22.41 lbf | 0.0 N = 0.0 lbf |
| $D$ | 1.4 N = 0.31 lbf | 2.5 N = 0.57 lbf |

Due to the opinion of the FAA inspectors regarding the external baggage door mounting location, the alternate internal baggage compartment mounting location became the only remaining option. Drawings and sketches of the desired modification were created, submitted to the regional FAA engineer by Old Town Aviation for analyses in structural integrity and proper installation procedures for submission to satisfy the supporting engineering data requirement for the Form 337 Field Approval from the FAA. The results from the engineer, including computations, schematic drawings, and part listings were then submitted to the FAA for field approval. The FAA granted Old Town Aviation the field approval required to modify the Cessna 172 named 8HE by cutting the required holes in the skin and baggage floor and installation of a round plexiglass window to operate as a camera port. Upon completion of the modifications, FAA inspectors examined the modifications for compliance with the engineer's installation guidelines and the aircraft was ready for fitting of the camera-mounting bracket.

The mounting bracket built is designed to securely hold the camera so that the imaging sensor plane is as nearly parallel as possible to the plane formed by the four vibration absorption mount contact points. The baggage compartment floor in a Cessna 172 is not level during flight, inclined by several degrees. Without compensation for the baggage floor inclination, the camera lens could never point in a near vertical direction. Additionally, the threaded mounts of the vibration absorption mounts would now be inaccessible to thread the bolts due to the baggage compartment floor, so an adapter assembly, shown in figure 17 was constructed to allow rapid and convenient mounting of the camera and bracket to threaded inserts in the baggage compartment floor. In order to compensate for the inclination of the baggage compartment floor, the aircraft was lifted on large adjustable jacks and leveled according to Cessna factory-installed level points that indicate the designed in-flight orientation of the aircraft. Once level, the camera bracket was attached using only two rearward bolts and the entire assembly tilted at the front until level, requiring 0.75 inch spacers on forward mounting bolts to position the camera in near-vertical orientation while in flight. Images of the camera system installed in the aircraft are found in figure 18.

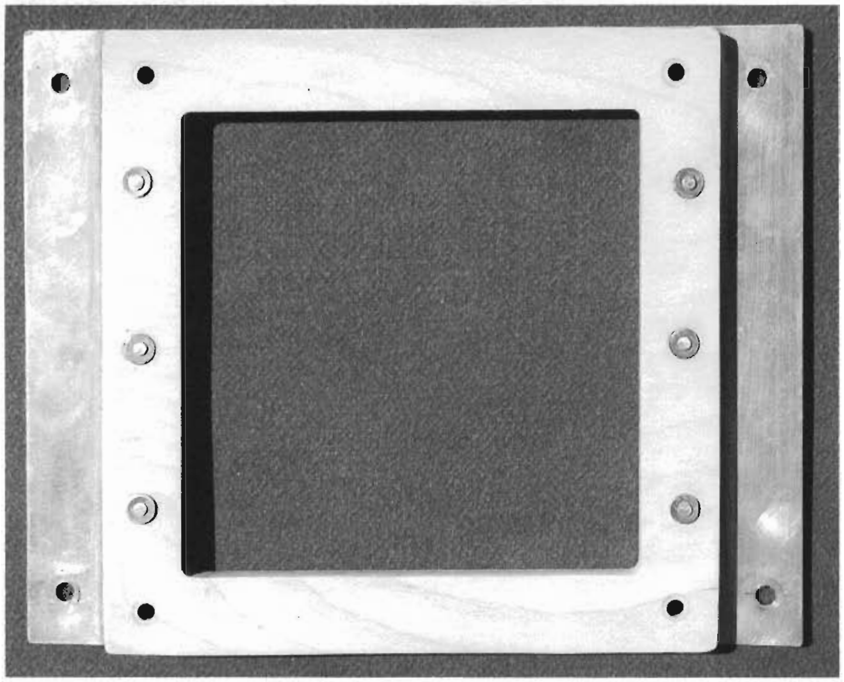


Figure 17. Camera Bracket Mounting Adapter Assembly

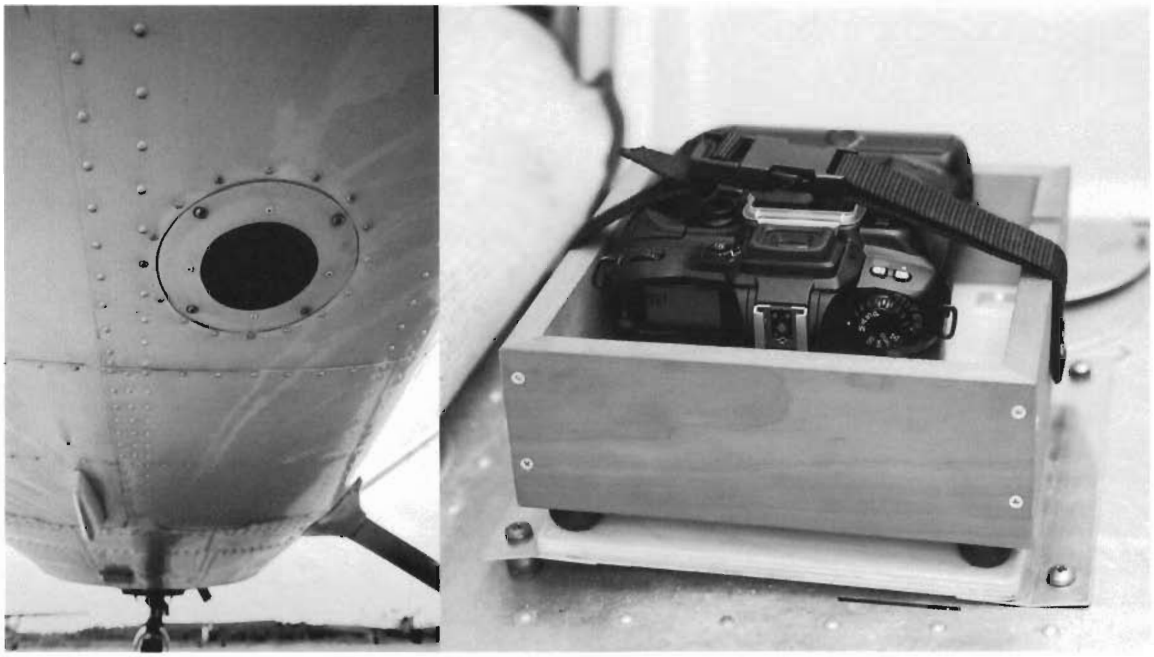


Figure 18. Plexiglas Port on Exterior of Aircraft, Camera System Installed in Aircraft

### 5.5. In-flight Guidance and Camera Control System

The tasks of determining whether the aircraft is on the correct bearing to fly directly over the target, visually examining current coordinates and comparing them to desired coordinates, computing the moment at which the aircraft is at its closest point to the desried target, and triggering the camera at that moment are too many items for most humans to deal with simultaneously. To remove most of these tasks from the camera operator, a laptop computer with a connected GPS receiver is employed to control the camera, store captured imagery, track the location of the aircraft, and provide guidance information to the pilot, accomplished through the use of customized mobile GIS software.

**5.5.1. GPS Receiver.** Without input from external devices, the laptop computer used to control the camera system is unable to do much other than wait for input from the operator, triggering the camera only when instructed. Prior to the development of radio positioning technologies such as GPS, aerial photography missions depended upon optical sights and intervalometer timing units to trigger the camera. By connecting a GPS receiver to the computer used to control the camera, the coordinates at which the aircraft is currently located as well as the speed and heading at which the aircraft is traveling can be monitored and the image capture process can be automated, triggering the camera at the moment the aircraft is in proper position, eliminating the need for optical sights or interval timers. A Delorme Earthmate GPS receiver (figure 19) was acquired, which is small, capable of delivering real-time coordinate accuracy of 2-5 meters (www.delorme.com, last accessed November 2005), and is powered by the USB connection used to communicate with the laptop computer.

Figure 19. Delorme USB Earthmate GPS Receiver Used for Navigation

**5.5.2. Determination of the Location at which to Trigger the Camera.** During flight, the current bearing or course along which the aircraft is traveling changes constantly due to atmospheric effects such as wind and turbulence and navigation equipment accuracy error, requiring the pilot to make frequent course corrections. The result of these errors and course corrections is that the aircraft will likely never pass directly through the coordinates that the pilot is attempting to navigate to, always missing the intended target location by some amount. The software employed was designed to trigger the camera automatically when the aircraft reaches the closest point that is possible along the current ground track, triggering the camera system at the moment the aircraft reaches this location.

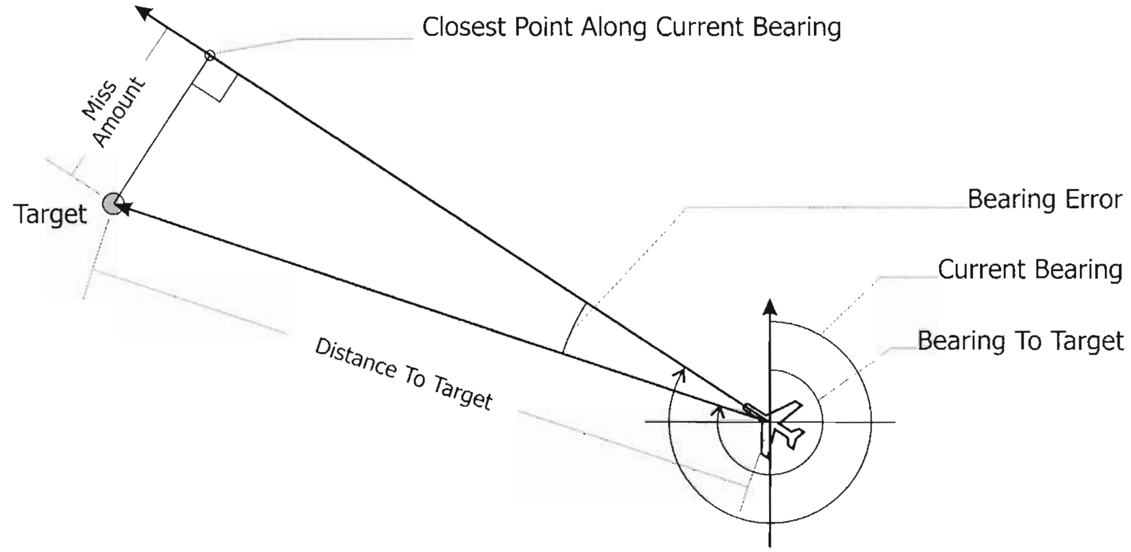


Figure 20. Horizontal Miss Due to Bearing Error

The point at which the aircraft will be closest to the intended target is located at the intersection of the current ground track (current bearing) and a perpendicular bearing that intersects the intended target (figure 20). The current ground track bearing along which the aircraft is traveling (CB), current speed over ground (SPD) in units per second, current position coordinates, distance to the intended target (DT), and bearing from current location to the intended target (BT) are obtained from the GPS receiver. The distance between the closest location at which the aircraft will pass along the current bearing and the target itself due to bearing error (BE), called the miss amount (MA), is computed using equation 15. Computation of the distance to the closest point (DCP) and the time required to reach this closest point (TCP) is then accomplished (equations 16 and 17). If the miss amount is less than the maximum amount allowed, the message to fire the camera can be sent.

$$BE = |CB - BT| \tag{14}$$

$$MA = DT * Sin(BE) \tag{15}$$

$$DCP = DT * Cos(BE) \tag{16}$$

$$TCP = \frac{DCP}{SPD} \tag{17}$$

**5.5.3. In-flight Software.** As an alternative to the significant task of creating completely new navigation and camera triggering software, mobile GIS software was customized to perform the required tasks, freeing the operator to provide guidance directions to the pilot and review photographs as they are taken. ESRI ArcPad 6.0.3 mobile GIS software, originally designed to operate on mobile computing devices such as Pocket PCs and laptop computers, was chosen to act as the host for navigation and camera control customization due to its simple interface and customization capabilities through Microsoft Visual Basic/VBScript using ESRI ArcPad Studio. Factors influencing the choice of ArcPad over other GIS software packages include: built-in GPS handling, built-in Navigation objects, and support for inclusion of GIS data layers to be used as a background and moving map display without requiring additional software coding. Additionally, because the University of Maine possesses a site license for ESRI products including ArcPad, no additional expenditures for software were required. To handle camera control functions, a copy of Nikon Capture software was acquired, allowing full control of camera settings and quick review of photographs captured in-flight.

Customization of ArcPad was accomplished using ESRI ArcPad Application Builder's ArcPad Studio software, designed specifically for customizing ArcPad. A series of VBScript subroutines and functions found in appendix B were created to handle the computation of the closest point to the current target along the current bearing, computation of the miss amount, computation of the time required to reach the closest point to the intended target, display of navigation information, sending messages to trigger the camera system, and changing the navigation target to the next waypoint. Upon reaching the computed closest location to the intended target waypoint, the ArcPad script sends a message to Nikon Capture to take a photograph, providing visual confirmation to the operator that these messages are being sent.

A custom toolbar was created in ArcPad Studio (figure 21) containing commonly used tools for zooming, panning, identification, selection, and GPS control, as well as several custom tool buttons specific to navigation and camera control during aerial photography missions. Customized icons were developed for the newly created control buttons, color-coded for quick identification.

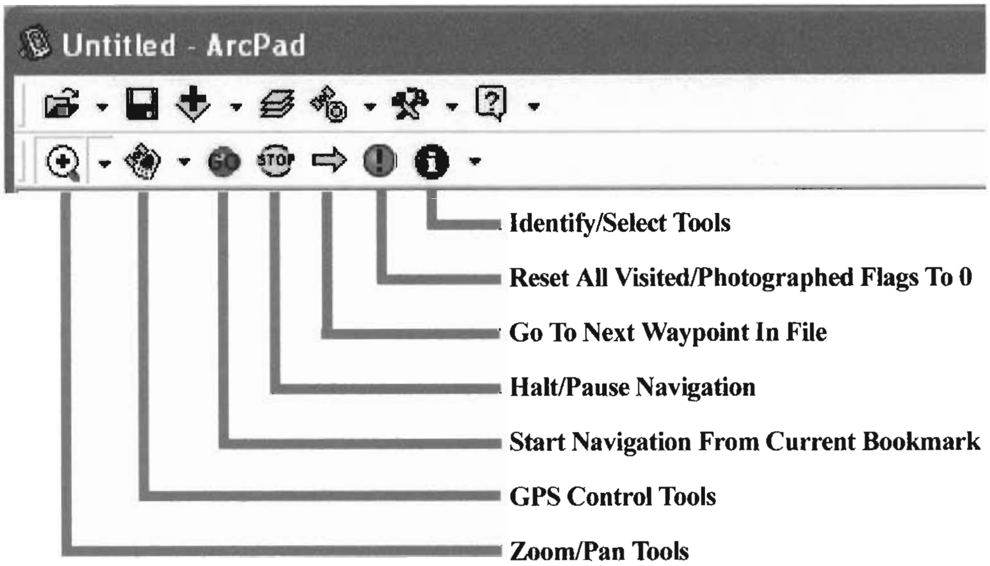
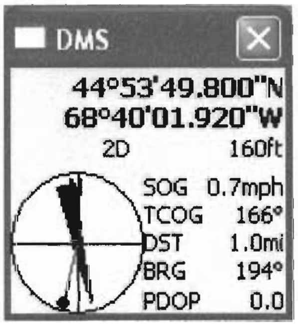


Figure 21. ArcPad Custom Toolbar for Aerial Photography Missions

*5.5.3.1. ArcPad Navigation and GPS Handling Functionality.* GPS receiver connection and NMEA string parsing are built into ArcPad software, as are functions to Navigate to specified features or locations, including graphical displays for navigation and position information. These GPS and Navigation objects are readily accessible throught script code, simplifying the process of creating the in-flight control software considerably. While the GPS receiver is active and transmitting position information to ArcPad, application events fire in ArcPad, capable of executing script code. If a target is sent to the ArcPad navigation object while the GPS object is active, computation of bearing to target and distance to target is completed upon receipt of every new GPS position computation. Graphic display of these quantities as a navigation compass is built in, providing a simple and effective means of communicating course information to the pilot (figure 22). The larger black arrow indicates the current true-North course over ground (TCOG), displayed numerically immediately to the right of the compass arrow. The bearing to the current navigation target is indicated by the red line with a red-filled dot at its end, indicated numerically on the navigation compass window as the quantity BRG. Navigation to the target requires simply providing directions to the pilot to change course either right or left so that these indicators coincide.



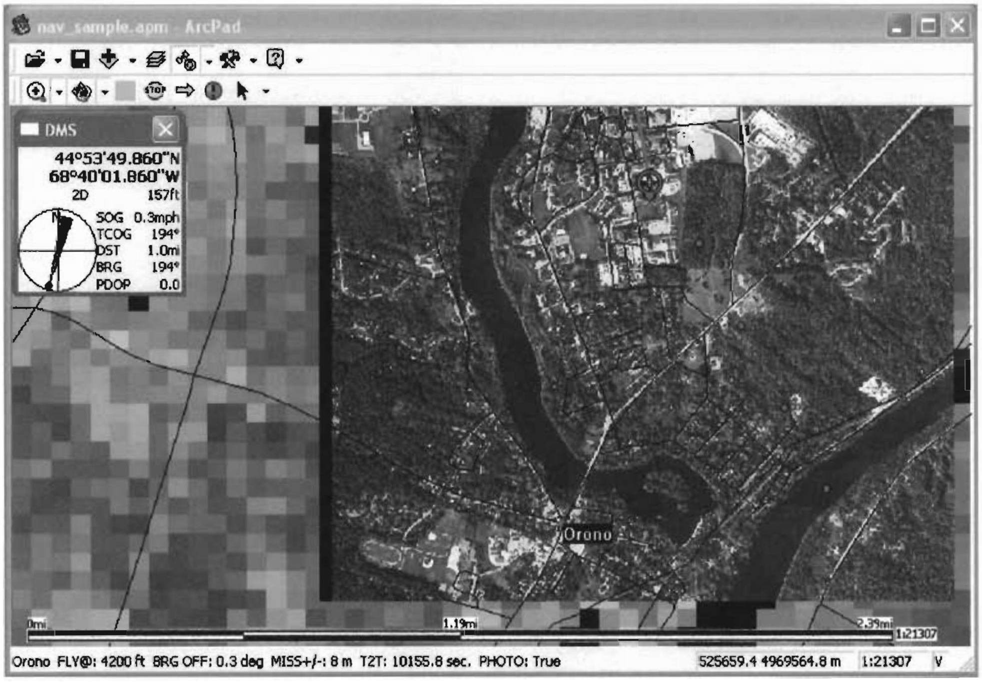
44°53'49.800"N
68°40'01.920"W
2D 160ft
SOG 0.7mph
TCOG 166°
DST 1.0mi
BRG 194°
PDOP 0.0

Figure 22. ArcPad Navigation Compass

*5.5.3.2. Computation of Miss Amount, Distance to Target Waypoint, Display of Computed Quantities.* Upon starting an aerial photography navigation mission, a global system variable is changed to indicate that a flight is in progress. The GPS receiver, when active, fires an application event every time a new valid position is computed, normally at an interval of once per second. ArcPad was customized to execute VBScript code immediately upon the firing of these GPS position events, but only if a flight is in progress. If a flight is in progress, the computation of the closest point to the current navigation target will be computed with accompanying time required to reach the closest point, completed at the receipt of every new valid GPS position using the methods described in section 5.4.2. Computing these quantities at the receipt of every GPS position allow continuous compensation for errors in bearing as the current course and speed continuously vary in time as the aircraft is navigated to the intended target.

The GPS receiver sends new position calculations to the computer at a rate of once per second. At a rate of travel of 75 miles per hour, a common flight speed while capturing photographs, one second of travel equates to 33.5 meters of ground distance covered. If the position for triggering of the camera is less than but nearly a full second away from the current location, then the camera will not be triggered until after the next GPS position update, meaning that the aircraft has passed over the closest location in which the camera should be triggered by as much as over 30 meters. To compensate for this problem, a system timer event is used to trigger the camera at the proper moment rather than relying on the one-second interval GPS positions. If the miss amount is within 3 times the horizontal ground position error estimate in the camera parameters file and the aircraft position, speed, and ground track bearing result in computation of time to nearest point of less than 3 seconds, the system timer is activated, set to fire a timer event after the amount of time remaining has elapsed. This solution allows the camera system triggering to be based upon the time to the closest point in milliseconds rather than full second intervals.

In addition to computing the distance and time required to reach the closest point to the target along the current ground track bearing, the miss amount is computed using the methods described in section 5.4.2. To provide the operator feedback on how far away the target is and how much the aircraft will miss the intended navigation target waypoint, the miss amount and time required to reach the closest point are displayed on the ArcPad window status bar (figure 23). Other information such as the name of the waypoint, the desired flying elevation, how much ground track bearing error is present, and whether or not to take a photograph are displayed as well.



```
44°53'49.860"N
68°40'01.860"W
2D           157ft
SOG      0.3mph
TCOG       194°
DST       1.0mi
BRG        194°
PDOP        0.0
```

```
Orono FLY@: 4200 ft BRG OFF: 0.3 deg MISS+/-: 8 m T2T: 10155.8 sec. PHOTO: True     525659.4 4969564.8 m   1:21307   V
```

Figure 23. ArcPad In-Flight Control Application in Operation

*5.5.3.3. Triggering The Camera System, Changing to Next Waypoint.* As the aircraft approaches the computed closest point and the system event timer is enabled, the OnTimer event triggers the execution of the script that fires the camera. If the Photo flag in the current waypoint's database record contains a value of 1, Nikon Capture software is sent a message to shoot a photograph and a message is sent to the ArcPad status bar indicating that it is time to take a photograph. Upon completion of sending the message to take the photograph, the current coordinates, ground track bearing, speed, elevation, and time are stored in the waypoint's database record in the appropriate fields, useful for post-processing steps and creation of hyperlinks in GIS software for management of photographs.

For individual waypoints, the user will be prompted asking if the desired location was captured in the photograph. If the location was missed for some reason, the record can be marked as missed if desired or kept as the current navigation target so that a repeat pass can be made to attempt another capture. Some photographs taken will be part of a series of photographs along a path, which would result in a series of annoying prompts along the path asking if the location was captured, so provisions are incorporated to only display verification prompts if the waypoint is not within a path, indicated by the Path field in the waypoint's database record.

Upon completion of the capture and storage of the coordinate information at which the timer event was fired, if the point was captured adequately or is in part of a path, the next point in the waypoint file (if present) is obtained and sent to the navigation object. The cycle of computation of the closest point, distance to the closest point, and so on continues until the last waypoint target has been visited. Upon completion of the flight mission, the global system variable indicating if a flight is in progress is set to indicate that the flight is complete and the pilot can be instructed to return back to the airfield.

### 5.6. Digital Aerial Camera System Hardware Components

Digital photogrammetric aerial camera systems in use by aerial survey companies are large, high precision instruments that are extremely expensive to acquire. In addition to the high purchase cost, the large size of aerial survey cameras requires use of larger aircraft and accompanying high costs for installation and operation. Furthermore, the complexity of digital aerial camera systems requires that staff be trained in the correct operation of the equipment with continued updating of training as upgrades are made.

The digital aerial camera system designed and presented herein was assembled with the intent that the system be inexpensive, capable of performance allowing imagery to be used in Level 2 monitoring activities, compact, durable, and easy enough to operate that even untrained staff would be capable of carrying out photography missions if required. In keeping with these ideals, the components selected are of high quality but are available off-the-shelf. Employing off-the-shelf components allows significant cost savings by avoiding the use of highly specialized limited production equipment. Because the components are designed for common consumers, the system is small, portable, and relatively easy to use, allowing installation in small aircraft with minimal modification. Automation provided by custom software running on an attached computer removes most tasks from the operator during photography missions.

The camera chosen for the digital vertical aerial system was the Nikon D100 digital single lens reflex (SLR) camera body. This camera was chosen for high-quality image capabilities afforded by the 6.1 million pixel imaging sensor, the durable construction of the body, the expandability provided by accessories, and its ability to be remotely controlled by connection to a computer through USB interface. The sensor employed in this camera body captures images at maximum resolution of 3008 by 2000 pixels, with each pixel occupying 7.8 microns by 7.8 microns of space on the sensor, yielding images of very high quality and low noise even at high ISO sensitivity ratings. As the camera will be expected to operate for long durations of up to five hours on flights, extra battery power was required. To ensure that sufficient charge is available, the MB-D100 vertical grip was purchased, which allows simultaneous installation of two lithium ion EN-EL3 battery packs for a total of 2800 milliamp hours of power available, providing for hundreds of photographs per charge.

Innovations in consumer level 35mm SLR camera lenses have resulted in the creation of lenses able to compensate for movement of the camera during exposures resulting in sharper photographs under these conditions, commonly referred to as Vibration Reducing (VR) or image stabilizing lenses. Initial research into VR lenses available for the camera body chosen revealed that, while the ability to reduce image degradation from camera movement during exposure would likely be an advantage, no information on whether the VR mechanisms would operate in vertical orientation, resulting in the decision to not employ a VR lens, instead purchasing a lens with a larger maximum aperture. With the increased amount of light transmitted through a large-aperture lens, a higher shutter speed on the camera can be used, “freezing” any movement and allowing higher-quality imagery. Due to VR lens operability questions, a non-VR lens with a large maximum aperture was chosen, the Nikon 24-85mm zoom lens with a large variable maximum aperture of *f*/2.8 to *f*/4.0. In addition to the large maximum aperture, this lens offers a long focal length setting of 85mm, useful for handheld oblique aerial photographs on qualitative inspection aerial missions.

Additional camera and computer accessories purchased include a high-capacity flash memory storage card for storing image files, a Dell laptop computer, and power supply equipment to augment the laptop computer's internal battery. The laptop computer purchased is a Dell Inspiron 600m with 512 Megabytes of RAM, a 40 Gigabyte hard drive, and a 1.4 Meghertz Pentium M processor, providing ample storage space for photography missions and sufficient processing power for post-mission rectification.

Although laptop computers are available with substantially longer battery runtimes, the laptop computer acquired was found to be capable of operating for 1-1/2 hours or less, depending upon exact settings used. The maximum in-flight duration that passengers in the aircraft can comfortably tolerate is approximately 5 hours. Additional power supply equipment was required to provide the additional operating power for the laptop computer. To supply this power, a 600VA uninterruptible power supply (UPS) was acquired, designed to power computers during power outages. During flight, the UPS provides power for the laptop and any connected devices, including the GPS receiver. Once the battery is depleted in the UPS, the laptop computer automatically switches to its internal battery power for the remainder of the flight. Repeated tests operating the laptop computer at full performance with the display lamp on high, running external USB bus-powered devices, and wireless radio communication active revealed that the UPS battery is capable of providing just over three hours of operation time until depleted. Tests of the system with the wireless radio communications disabled and display lamp at half power revealed that the UPS battery is capable of providing nearly five hours of power before the laptop switches to internal battery power. As the maximum flight time rarely will exceed 5 hours due to passenger fatigue, the combination of the UPS battery and laptop internal battery provide more operating time than will likely ever be required.

The total expenses for equipment purchased, flight-testing, and the FAA form 337 approval process appears in table 5. The University of Maine currently possesses a license for the ESRI ArcPad and ArcPad Application Builder software used to develop the in-flight camera control software; therefore, expenditures for license updates were not required. However, pricing for these software titles and the ArcView desktop GIS software are included in table 5 for reference.

Table 5. Expenses for Acquisition and Testing of Digital Aerial Camera System Components

**EXPENSES**

| Item | Cost |
|---|---|
| Nikon D100 Camera Body | $1300 |
| Nikon MB-D100 vertical/battery grip | 220 |
| Nikon EN-EL3 spare battery | 45 |
| 512MB Compact Flash storage card | 50 |
| Nikon 24-85/2.8-4.0 lens | 490 |
| Dell Inspiron 600m laptop computer | 1190 |
| Delorme Earthmate | 100 |
| Nikon Capture 4.0 Software | 100 |
| Materials for construction | 15 |
| 5 Anti-vibration mounts (4 lb max load, neoprene rubber cup mounts) | 35 |
| Adapter mount materials and machining | 24 |
| Uninterruptible power supply/line conditioner, 600VA 345 Watts | 109 |
| FAA Engineer review | 600 |
| Modification to aircraft | 500 |
| 20 hours at $125 per hour, Old Town Aviation | 2500 |
| ESRI ArcPad 6.0.3 software | 475 |
| ESRI ArcPad Application Builder software | 1350 |
| ESRI ArcView software, single-user license | 1290 |
| **Total** | **$10,393** |

### 5.7. Positional Accuracy Testing

Photographs taken by the digital aerial camera system will frequently be used to make measurements and assist in navigation to locations depicted, requiring an estimate of the accuracy of points depicted in the rectified photograph. For the purposes of this discussion, accuracy is defined as the amount of difference between the coordinates of any given point in the rectified image and the corresponding field surveyed or published coordinates of the point. To evaluate the accuracy attainable from the camera system in conjunction with rectification using only a second-order polynomial warping operation, registered to existing DOQ imagery, two separate testing procedures were carried out in order to discover the accuracy that can be expected in both an ideal situation and situations more commonly encountered. A nearly ideal situation, with relatively flat terrain, many photo-identifiable ground control targets, and very accurate control point coordinates, was tested to provide an estimate of the maximum accuracy attainable. To simulate the conditions that will commonly be encountered in monitoring activities, the estimated expected accuracy was tested in separate locations having varying terrain relief, fewer ground control targets, and less accurate ground coordinates gathered using mapping grade GPS equipment.

**5.7.1. Flat Terrain, Many Targets, Accurate GCPs.** The ideal situation for testing the accuracy attainable would be to photograph a perfectly flat region in which elevation is constant throughout the view of the camera, therefore the effects of relief displacement are not present, and many clearly identifiable ground control points that can be measured with high accuracy exist on the ground. Finding such a location that perfectly complies with these characteristics and is large enough to fill the camera's viewing angle from common flying heights is practically impossible, however suitable substitutes that are nearly perfect are readily available. The University of Maine Morse Football Field used for focal length calibration lies in a region that is surrounded by parking lots and other athletic fields, with very slight elevation changes totaling approximately 10 meters (32.8 feet) throughout an area large enough to fill the camera's view at common flying heights. Additionally, aside from stadium structures, few objects are present that would exhibit relief displacement and many easily identifiable features are present that can be conveniently accessed to survey their ground coordinates. For these reasons, the field and its surroundings were again used for camera system testing.

Acquiring high-resolution digital imagery of unknown accuracy for the region of the football field allowed use of the single-point, user-specified waypoint creation tool to generate rough coordinates of the center point of the football field and computation of the estimated flying height of 1219 meters (4000 feet) above terrain required to capture the desired region surrounding the field. A test flight was carried out in which repeated passes over the football field were made at the computed flying height above terrain, allowing selection from several images for the image possessing the highest quality and containing the most identifiable points on the ground.

Upon returning from the test flight, the images were evaluated and the most desirable image was chosen. Locations in which ground coordinates would be collected were then chosen, with predetermined names for each point. 40 ground control points were measured using an Ashtech Real-Time Kinematic surveying GPS receiver and data collector, each predetermined ground point was visited and coordinates measured with an instrument-indicated accuracy of 0.012 meters (0.04 feet). These coordinates were then converted into a GIS shapefile with database fields containing x coordinates, y coordinates, and point names for use in GIS software processing. Following the creation of the ground control point shapefile, the chosen aerial image of the region captured in the test flight was rectified to 9 of the surveyed control points using a second order warping operation and bilinear image resampling in Hypercube software with an RMS error of 0.065 meters (0.213 feet) and an output pixel size of 0.22 meters (0.72 feet) per pixel, the maximum resolution that the source image was capable of providing.

After completion of image rectification, the image was opened in ArcView GIS software and a new point shapefile was created, with points created at the depicted locations of 33 surveyed ground control points, with northing and easting coordinate fields and identical names for points as contained within the ground control point shapefile created from RTK GPS surveying data as described above. The surveyed ground control points shapefile was loaded and the database tables for both the rectified image points and the surveyed points were joined on the point name attribute field. Figure 24 depicts the result of overlaying both the surveyed control points and the corresponding rectified image points upon the rectified image. Figure 25 shows a smaller portion of the total image area to illustrate the difference in planimetric position between the surveyed and rectified image control points. New database fields for the distance (error amount) between surveyed points and image points in northing, easting, and total overall distance were created, with values populated using Arcview's field calculator. Total distance was computed using the Pythagorean distance formula and the Northing and Easting distances were computed as the magnitudes of the difference in the coordinates. Northing and easting difference fields were created in addition to total distance as a check that error magnitudes were not dominating in either northing or easting directions.

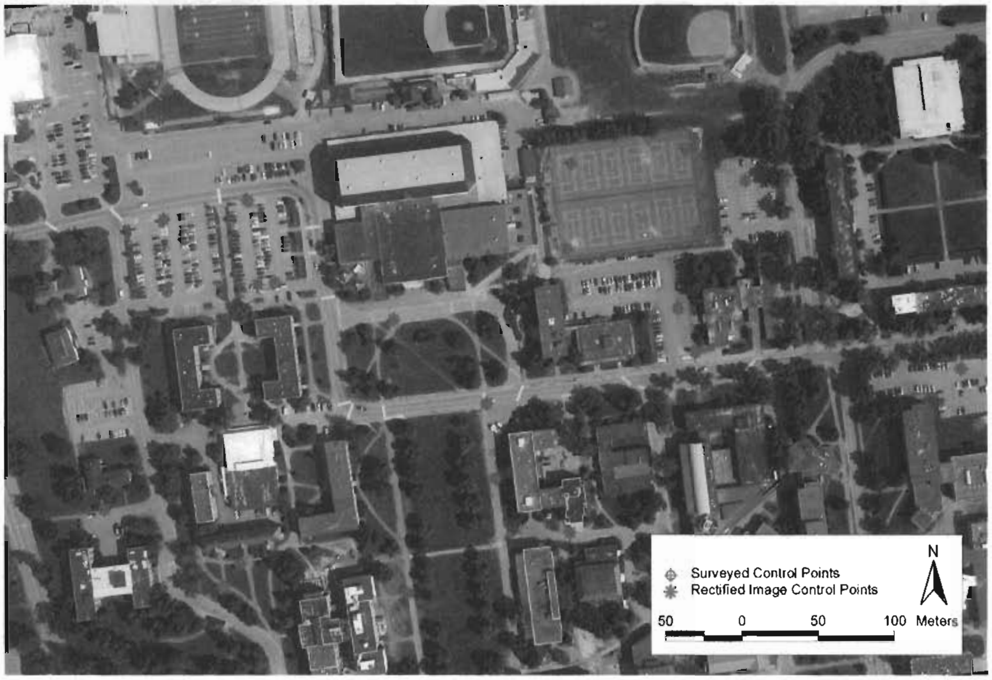


Figure 24. Flat Terrain Accuracy Test - Surveyed Ground Control Points and Rectified Image Control Points Superimposed Upon Rectified Image

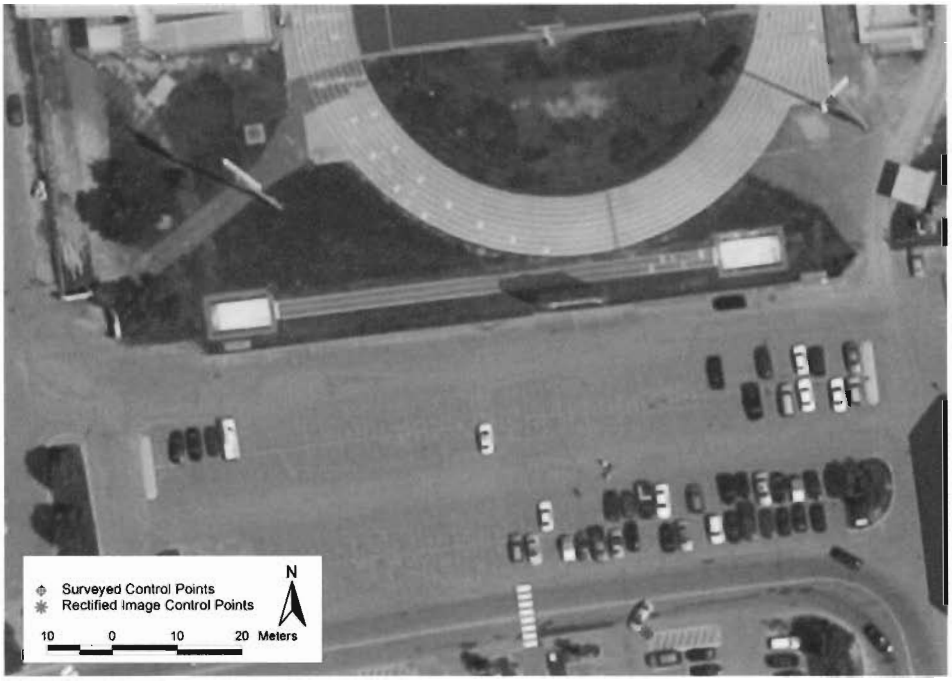


Figure 25. Flat Terrain Accuracy Test - Surveyed Ground Control Points and Rectified Image Control Points Superimposed Upon Rectified Image (Closer View)

**5.7.2. Varying Terrain, Few Targets, Rectified to Digital Orthophoto Quadrangles (DOQs).** While the accuracy test using high-precision equipment over a relatively flat region with many identifiable targets provides an idea of the maximum accuracy attainable with the system assembled, it does not provide an estimate of the accuracy that can be expected under normal Level-2 inspection operating conditions in which it will be used. Specifically, the camera system will be employed to photograph regions in which the terrain can vary in elevation throughout the image by moderate amounts, the number of clearly identifiable targets are far fewer, and any ground checking will likely be performed using GPS equipment capable of delivering far less positional accuracy. It is assumed that users will most likely be employing mapping grade or consumer grade GPS receivers, delivering accuracies of 2 to 5 meters (6.6 to 16.4 feet) for mapping grade receviers and 2 to 7 meters (6.6 to 23.0 feet) for consumer grade receivers (Wing et al. 2005). Furthermore, high-accuracy, photo-identifiable target data to use in rectification of imagery will likely not exist, requiring the use of alternative coordinate source imagery such as DOQs.

To test the accuracies that can be expected under these conditions, a series of photographs were taken at photo-identifiable locations along a road passing through terrain similar to the regions in which monitoring activities will be carried out. The road chosen to fly along was the Studmill Road in Eastern Maine, commonly used by timber companies to access lands in which forest product harvesting is underway. The photographs were taken along a 45 mile long segment of the Studmill Road from the abandoned airstrip at Pickerel Pond in Township 32MD to locations along Route 9 next to Crawford Lake in Crawford Township.

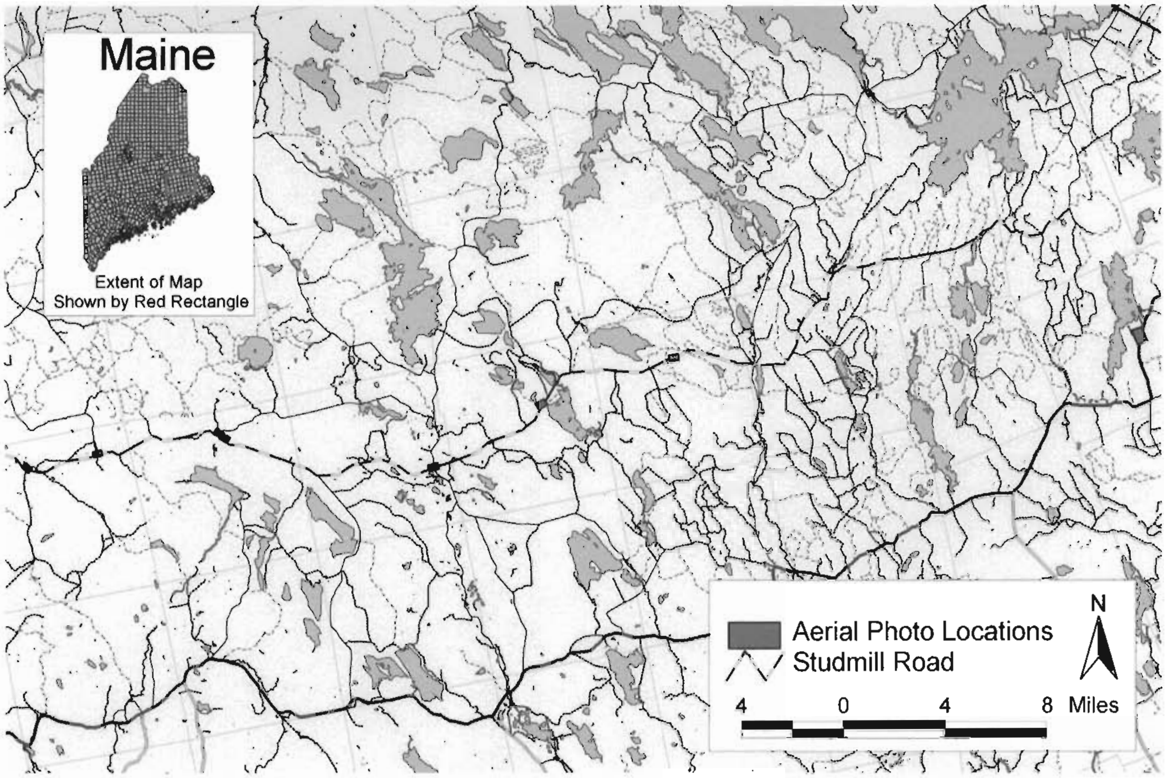


Figure 26. Varying Terrain Accuracy Testing – Locations of Aerial Photos

A total of 8 photographs were used from those captured, geo-referenced to DOQs downloaded from the Maine Office of GIS, and rectified using second-order polynomial resampling functions to compensate for image pixel displacements due to terrain variation. Upon completion of the rectification process, locations in the photographs that appeared accurately identifiable on the ground were selected and coordinates were obtained, created as a GIS point shapefile with each point named for easy identification. A total of 50 control points were identified, distributed throughout the 8 photographs. The selected points were visited on the ground and coordinates obtained using a Trimble GeoXT GPS receiver, post-processed to provide sub-meter coordinate accuracy. The ground coordinate file was converted to a GIS point shapefile with each point given the same name as the image reference points to allow comparison of ground coordinates to rectified image coordinates. New fields were created in the image shapefile database for the difference in northing, difference in easting, and difference between points as total distance, with values populated using Arcview's field calculator. Northing and easting difference fields were created in addition to total distance as a method to discover if error magnitudes were dominating in either northing or easting directions.

# Chapter 6

# RESULTS AND DISCUSSION

## 6.1. Calibration Results

Lens calibration procedures were initially carried out for the 24mm focal length setting of the lens acquired because wide angle lenses allow flying at lower altitudes, reducing the amount of image degradation due to atmospheric haze. Unfortunately, upon installation of the camera system in the aircraft upon completion of modifications, the external hole cut into the inspection plate was found to be too far away from the front element of the lens, resulting in a large amount of vignetting and image cutoff. It was discovered that the 35mm focal length setting of the lens allows full-frame image capture while still offering a reliable physical stopping mechanism that could be used to ensure that the focal length was set at a repeatable length that could be relied upon for calibration procedures as well as flights, therefore the 35mm lens setting was calibrated as well. The results of calibration procedures are presented in tables 6 and 7 for both the 24mm and 35mm focal length settings despite the limitations on the usable focal length due to the size of the hole cut into the aircraft.

**6.1.1. Lens Focal Length.** Lens focal length calibration techniques (described in section 5.2.1) were employed to discover an approximation of the infinity focal length of the lens at both the 24mm and the 35mm focal length settings of the lens. As described in section 5.2.1, focal length is a function of distance and, as the object distance approaches infinity, the resulting focal length approaches the lens' true focal length. If a statistical linear regression analysis is performed on a response variable and the slope of the regression line is found to be zero at a particular level of significance, then it can be said that no linear relationship between the response and the independent variables was found. In the case of attempting to discover an approximation of the lens' infinity focal length, the computed focal length at an object distance is the response variable while the object distance is the independent variable. If the slope of the regression line computed for various distances is found to be zero, then the focal length is independent of the object distance, meaning that an approximation for the infinity focal length has been found – increasing the object distance further does not change the resulting focal length significantly.

A linear regression was performed on computed focal lengths (*f*) at various distances, for both the 24mm and the 35mm focal length settings on the lens. For the 24mm lens setting, the analysis yielded the regression line found in equation 18, where *d* is the object distance in yards from the camera. The squared multiple R correlation coefficient from regression output was very small at 0.181, and an F test revealed that the F* was greater than the F distribution at a significance level of 0.05 with a P-value of 0.038, indicating that a linear relationship between distance and focal length exists. The regression output for the analysis indicated that two observations were outliers, one 25.0mm computation and one 25.2mm computation. All other computations at all distances were found to be identical values of 25.1mm. Removal of the two outlying measurements meant that for all object distances used, the computed focal length was independent of the object distance, so it can be safely assumed that the computed focal length of 25.1mm can be used as an approximation for the infinity focal length for all distant object distances such as those encountered during aerial photography missions.

For comparison's sake, basic statistics detailing the mimimum, mean, median, maximum, and 95% confidence interval for the mean focal length were computed using statistics software (table 6). Even if the outliers identified by regression analysis were retained, the confidence interval when rounded to the nearest tenth of a millimeter becomes the single value of 25.1mm, which was accepted as the approximation for infinity focal length at this lens setting.

$$f = 25.117463 - 0.000403 * d \qquad (18)$$

Linear regression analysis was completed for the 35mm lens focal length setting to discover if an approximation for the infinity focal length at this setting had been discovered as well. The computed focal lengths at distances in the case of the 35mm lens setting were smaller at nearer distances, increasing to a maximum at around 45.7 meters, then decreasing slightly for large distances. Initial analysis using the computed focal lengths yielded a regression line function that was positively increasing for all distances, which did not agree with the trends visually identified within the computed focal lengths. A transformation of the computed focal lengths of $\frac{1}{f}$ was employed in an attempt to model the converging characteristics of the focal length as a function of distance as illustrated in the thin lens equation (equation 11). Regression analysis was performed on the transformed data, yielding a regression line with an extremely small negative slope, less than $1x10^{-9}$, meaning that as distance increases to infinity, the focal length converges upon the constant value as predicted by the thin lens equation. In the case of the transformed data, the squared R correlation coefficient was found to be very small at 0.01193, the F* statistic much smaller than the F distribution at a significance level of 0.05, with a large P-value of 0.58, indicating that no significant linear relationship was found. As there was some variability in the focal lengths at distances in this case, basic statistics for the focal length were computed with a confidence interval for the mean computed focal length (table 6). The 95% confidence interval for the mean computed focal length at the 35mm lens setting includes the values of 35.3 and 35.4 when rounded to the desired precision of one tenth of one millimeter, with the mean at 35.35mm, slightly less than the median. Because rounding the mean focal length down to 35.3mm would mean that less ground area would be used in flying height computations, the mean was rounded up to 35.4mm and accepted as the approximate infinity focal length for waypoint creation purposes.

Table 6. Focal Length Computations from Calibration

| | 24mm Setting | 35mm Setting |
|---|---:|---:|
| N of Measurements | 24 | 28 |
| Minimum | 25.00 | 35.20 |
| Maximum | 25.20 | 35.50 |
| Median | 25.10 | 35.40 |
| Mean | 25.10 | 35.35 |
| 95% CI Upper | 25.11 | 35.38 |
| 95% CI Lower | 25.09 | 35.33 |
| Standard Dev. | 0.029488 | 0.063413 |

**6.1.2. Distortion Characteristics.** Distortion characteristics for the lens at both 24mm and 35mm settings were computed using the procedures in section 5.2.3 above. Lens distortion function coefficients were computed for both one-parameter (*c*) and two-parameter (*a*, *c*) instances. Upon completion of plumb-line calibration procedures in

PTGUI, the distortion characteristics (table 7) were computed using the PTGUI least-squares optimizer.

Table 7. Lens Distortion Function Coefficients from Plumb-Line Calibration

| Lens Setting | *One Parameter (c)* | *Two Parameter (a)* | *Two Parameter (c)* |
|---|---|---|---|
| 24mm | -0.028026 | -0.000116 | -0.027310 |
| 35mm | -0.002652 | 0.000821 | -0.002494 |

While the coefficients computed for the 24mm lens setting are in the range of expected values that would indicate that image distortion correction procedures should be applied to captured images, the 35mm lens setting coefficients are considerably smaller, indicating that correction for distortion is likely unnecessary at the 35mm setting. Visual examination of the building face photographs at both focal length settings supports the outcome of the distortion coefficient computation results. Lines that should be rendered perfectly straight in the image captured at the 24mm setting display some noticeable curvature with no visibly noticeable distortion present in the image captured at the 35mm setting. The difference between the un-corrected images and those corrected with one radial lens distortion coefficient using the PTCorrect utility are illustrated in figure 27 for the 24mm lens setting and figure 28 for the 35mm lens setting.

Figure 27. 24mm Lens Setting - Distortion Calibration Images Before and After Correction

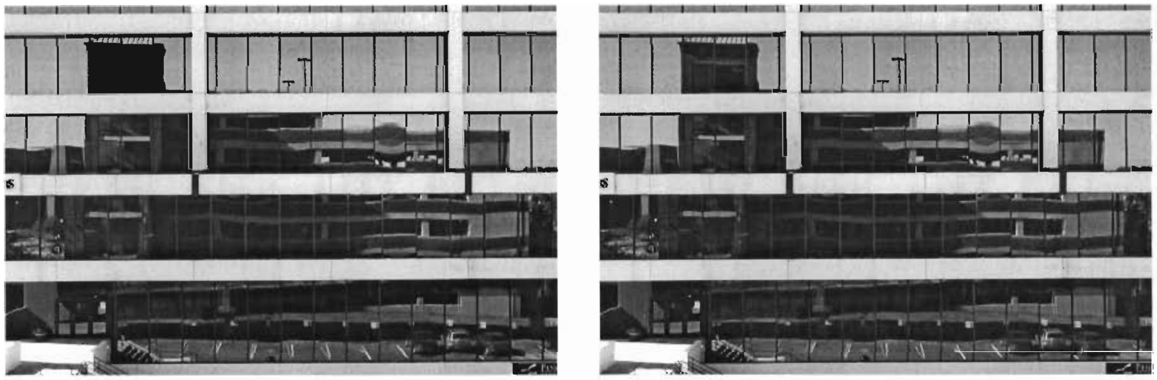


Figure 28. 35mm Lens Setting - Distortion Calibration Images Before and After Correction

As can be seen in figure 27, the correction applied to the 24mm lens setting image appears very effective at removing the distortion present while virtually no difference is seen between the uncorrected and corrected images at the 35mm setting (figure 28). To quantify the difference between the corrected and uncorrected images, a difference image for each focal length setting was created (figure 29). Because of the high contrast between the straight-line features in the image and their respective surroundings, the displacement due to distortion correction becomes visible in the difference image, with dark values indicating larger differences between distorted and corrected imagery due to pixel displacement caused by distortion correction procedures and white pixels indicating that little or no difference is found that region. A difference image that is almost or completely blank indicates that little or no difference exists between the corrected and original source images. Grey or black pixels in the image indicate that differences exist, interpreted as displacement of pixels by distortion correction procedures. The maximum displacement measured in the 24mm lens setting image after distortion removal using one distortion function coefficient was found to be 22 pixels while the 35mm lens setting image resulted in a maximum displacement of 3 pixels.

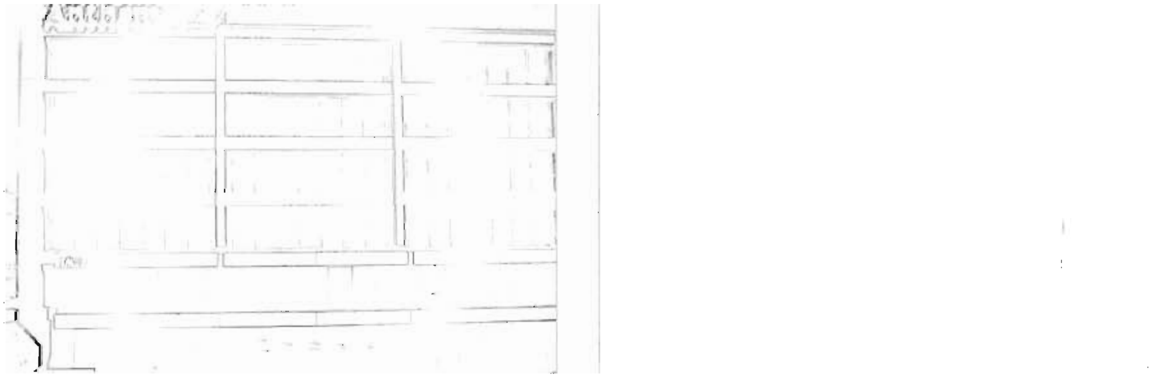


Figure 29. Difference Images Illustrating Displacement Due to Distortion Correction Procedures for 24mm Lens Setting (Left) and 35mm Lens Setting (Right)

Examining the magnitude of the distortion coefficient *a* in comparison to coefficient *c* reveals that, as expected, the *c* parameter dominates the distortion function in both cases, indicating that employing one parameter will correct for the majority of the distortion present. To quantify the amount of difference yielded between employing one parameter (*c*) and two parameters (*a, c*) in the distortion removal process, the lens distortion process was completed for both the 24mm and 35mm focal length settings using one and two-parameter distortion correction functions in PTCorrect. As can be seen in the difference images created for both focal length settings (figure 30), the maximum amount of difference between the one-parameter and two-parameter distortion removal functions was very small, resulting in almost completely blank difference images. The 24mm lens setting yielding a maximum displacement of 3 pixels, while the 35mm lens setting image displaying a maximum measured difference of 2 pixels. These results of one and two-parameter comparisons validate the statement in Brown (1971) and Fraser (1997) that the use of only one coefficient in the distortion removal function is likely sufficient for most applications. Furthermore, because the magnitude of the distortion present at the 35mm lens setting images is so small, the images may be rectified and geo-referenced immediately upon return from photography missions without distortion correction.

Figure 30. Difference Between One and Two-Parameter Distortion Removal Functions, 24mm Lens Setting (Left) and 35mm Lens Setting (Right).

**6.1.3. JPEG Compression Versus Camera Raw Files.** The difference in transfer time from the camera to the laptop computer between camera raw sensor data files and in-camera JPEG compression is substantial, with JPEG files requiring roughly 3 seconds to transfer and camera raw files requiring 12 to 13 seconds. While flying along a path with overlapping coverage at lower altitudes, the time between photo stations can be as little as 5 seconds with the preset minimum flying height above terrain, meaning that for flight paths, photographs would be captured at a rate faster than the image files can be transferred and stored. Additionally, camera raw files require four times as much storage space. While many digital camera reviewers have found that camera raw files yield a slight resolution advantage over in-camera JPEG compression, the limited amount of time between photo stations while photographing along a path dictates that in-camera JPEG compression should be employed.

The digital aerial camera system as presented captures photographs through plexiglass port with forward motion during exposure, resulting in some loss of image detail, possibly resulting in more image degradation than that introduced by using in-camera JPEG. In order to evaluate whether the losses of detail due to in-camera compression were of concern, photographs at the same elevation of the same region were taken using both camera raw file storage and in-camera JPEG compression. Care was taken that the aircraft was flying along a similar ground track bearing at as close to the same altitude between photographs to ensure that any detail differences would be a result of in-camera processing rather than elevation, rotation, or forward motion differences.

Upon returning from the test flight, the images were evaluated in image-processing software, rotated so that the features in the image were presented in the same orientation. The images in figure 30 illustrate the differences between camera raw storage and in-camera JPEG compression, shown at high magnification so that image pixels are visible. As can be seen in figure 31, the differences between the camera raw and in-camera JPEG compression images are minimal, with the camera raw file displaying very slightly higher detail and slightly reduced noise in shadowed regions. Whenever possible, individual location photographs should be captured using camera raw storage formats to take advantage of the slight increase in detail, but the default use of in-camera JPEG compression was judged to be sufficient for path series photographs.

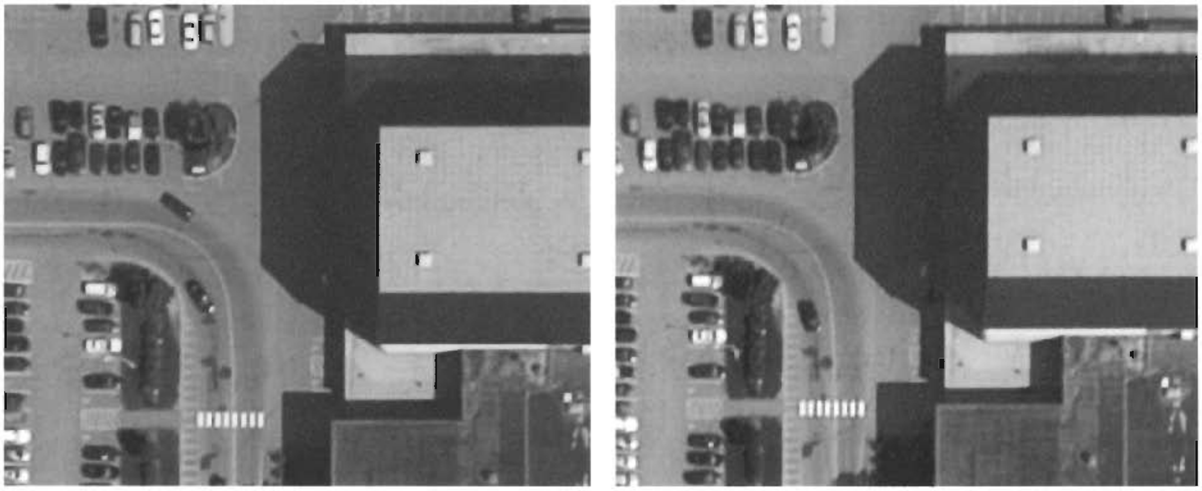


Figure 31. Image Quality Comparison Between Camera Raw (Left) and In-Camera JPEG Compression (Right)

### 6.2. Accuracy Test Results

Accuracy test results provided some unexpected findings with regards to expected accuracies attained and variations due to elevation, especially in the images taken along the Studmill Road to simulate real-world results. In general, the accuracy attained was higher than expected, considering the off-the-shelf components used, the plexiglass port photographs are captured through, and the lack of any forward motion compensation devices.

**6.2.1. Ideal Case: Campus Image.** In the ideal-case test using high accuracy control points and many photo-identifiable targets distributed throughout an image in which little elevation change is present, the mean magnitude of control point deviation from their locations in the rectified image compared to their surveyed ground coordinate location was found to be 0.29 meters. This low result was unexpected, as previous experiments with similar resolution imagery from a commercial vendor only resulted in accuracies of 1.5 meters using the same ground control points. Examination of the statistics for the magnitudes of error (shown in table 8) reveals that the median error is slightly lower than the mean, indicating that the error magnitudes tend to be lower than the mean, shown visually in the error magnitude distribution plot in figure 32.

As can be seen in the distribution plots, the error magnitudes tend to be lower than the mean in overall magnitude as well as magnitude in Northing or Easting directions. Examination of the magnitudes of Northing and Easting error reveals that the error magnitudes tend to be slightly higher in the Easting direction, not unexpected as the images were rectified into a Universal Transverse Mercator (UTM) coordinate reference system.

Table 8. Basic Statistics for Magnitude of Error (in Meters), Ideal Case Image of Campus

| | Overall Error | Northing Error | Easting Error |
|---|---:|---:|---:|
| Minimum | 0.088 | 0.005 | 0.030 |
| Maximum | 0.762 | 0.509 | 0.728 |
| Median | 0.238400 | 0.132500 | 0.182400 |
| Mean | 0.292 | 0.177 | 0.209 |
| Standard Dev | 0.170 | 0.127 | 0.152 |

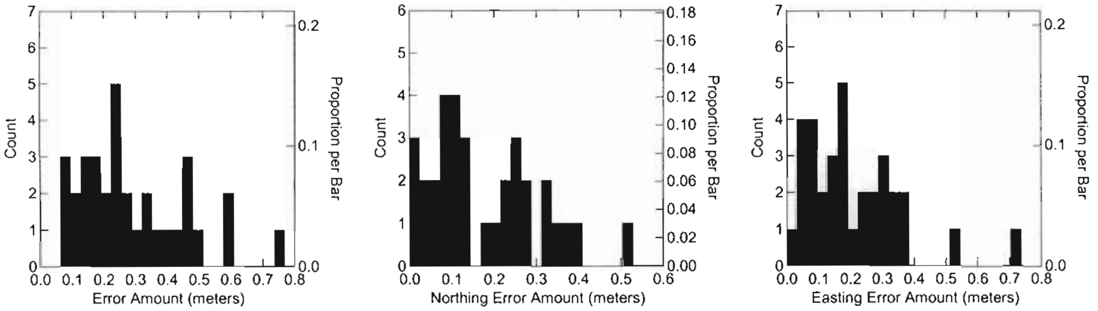


Figure 32. Error Magnitude Distributions For Ideal Case Image of Campus, Overall Error (Left), Northing Error (Center), Easting Error (Right)

Because the error magnitudes show a slight bias toward being slightly less than the median, the error magnitudes were computed separately for overall error, Northing error and Easting error at the 25th, 50th, and 75th percentiles (table 9). The magnitudes of error in the case of many accurate control points were surprisingly small, with error magnitudes of less than 0.24 meters in 50 percent of error measurements. The conclusion from these analyses was made that the accuracy limits of the system as currently assembled had been reached, and that increasing the maximum accuracy attained by even a small amount would require significantly more expensive equipment and more complex processing methods.

Table 9. Magnitudes of Error (in Meters) for Ideal-Case Image by Percentile

| Percentile | Overall Error | Northing Error | Easting Error |
|---|---|---|---|
| 25 | 0.164 | 0.089 | 0.093 |
| 50 | 0.238 | 0.133 | 0.182 |
| 75 | 0.407 | 0.257 | 0.301 |

### 6.2.2. Real-world Case: Studmill Road and Crawford Lake Photographs.

Because of the high number of variables in real-world test images, it is impractical or even nearly impossible to repeatedly capture images of any particular area for repeated tests. Additionally, there are a limited number of photo-identifiable points that can be used for accuracy testing in any given region, therefore, rather than analyzing the real-world case images individually, they were analyzed as a group. The difference between the coordinates of control point positions in the rectified image and the corresponding measured coordinates yields the magnitude of error, computed for overall error, Northing error, and Easting error as in the ideal-case analysis (table 9). The basic statistical analysis of the error magnitudes for the 50 control points that were used appears in table 10. While the magnitudes of error are much larger than those obtained from the campus image, similar trends are apparent. The median magnitude of error tends to be less than the mean, but only slightly. Also, the Easting error tends to be slightly larger than the Northing error by a similar amount as the campus image, again likely due to the use of UTM coordinates as a target reference system in the rectification process.

Table 10. Basic Statistics for Magnitude of Error (in Meters) Between Surveyed Control Points and Rectified Image Control Points, Real-World Case Images

| | Overall Error | Northing Error | Easting Error |
|---|---|---|---|
| Minimum | 0.149700 | 0.006300 | 0.075000 |
| Maximum | 5.292500 | 3.954700 | 5.061900 |
| Median | 2.272900 | 1.311250 | 1.164200 |
| Mean | 2.360582 | 1.408996 | 1.569344 |
| Standard Dev | 1.342162 | 1.048468 | 1.359889 |

Distribution histograms for the error magnitudes were generated (figure 33), verifying visually that the error magnitudes tend to smaller more often than they tend to be larger, identified by the median value being slightly less than the mean. Because of this tendency, the error magnitude percentiles were computed (table 11). The magnitude of errors between surveyed control points and their corresponding rectified image points considered as a group was found to be 2.3 meters or less in 50 percent of the measured locations, surprisingly low when considering that the test images were rectified using DOQ imagery of published accuracy of 7.0 meters or less to obtain target coordinates.

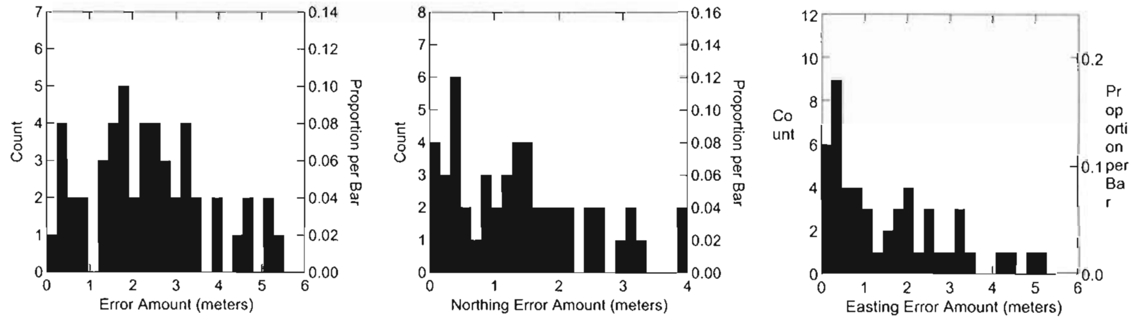


Figure 33. Error Magnitude Distributions for Images Along Studmill Road, Overall Error (Left), Northing Error (Center), Easting Error (Right)

Table 11. Magnitude of Error (in Meters) by Percentile Between Surveyed Control Points and Rectified Image Control Points, Real-World Case Images

| Percentile | Overall Error | Northing Error | Easting Error |
|---|---|---|---|
| 25 | 1.4541 | 0.4366 | 0.4587 |
| 50 | 2.2729 | 1.3113 | 1.6420 |
| 75 | 3.1928 | 2.0536 | 2.3768 |

Several anomalies emerged in the analysis of mean error magnitudes for the individual test images when compared to the elevation ranges for those images. Using ArcView software and DEM data, the range of elevation present within each image was obtained. In regions displaying larger ranges of elevation, it might be expected that the images displaying the largest amount of error would be related to terrain effects. It was discovered that the opposite was found in some cases: images displaying the lowest mean error magnitude were those containing the largest range of elevation, and those images with the most error contained the least range of elevation (figure 34, campus image included for comparison).

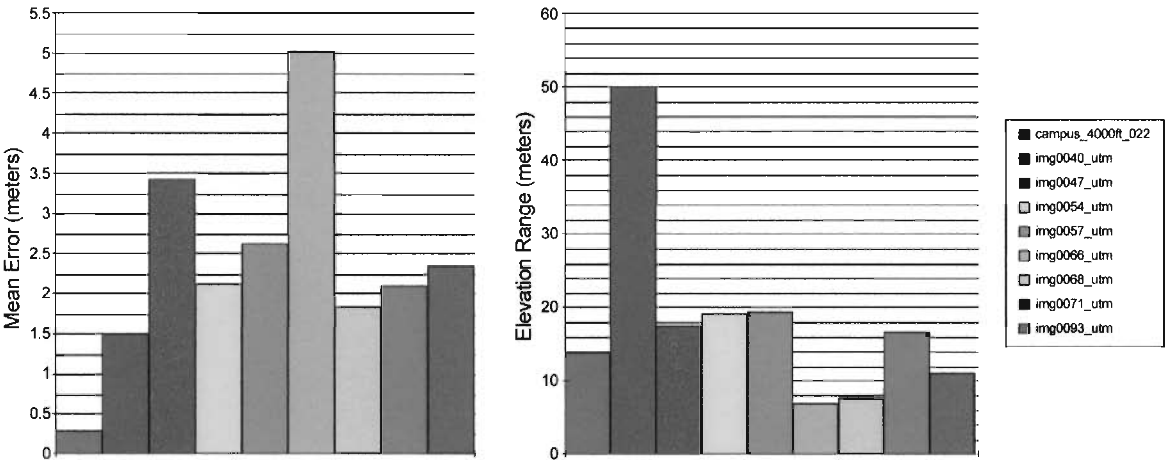


Figure 34. Mean Error and Elevation Range for Accuracy Test Images

The results seem to contradict the concept of relief displacement, however a variety of factors must be included in the analysis. While img0066 displays the least elevation range and the largest mean error magnitude, only 3 control points could be used for accuracy testing due to a large percentage of the image being covered in trees, limiting the number of photo-identifiable locations. Furthermore, the three control points are located very closely together, therefore any systematic error in ground coordinate computation from the GPS receiver and errors in control point placement during rectification would be magnified. While this explanation seems plausible for the discrepancy found in img0066, the discrepancy found in img0040 cannot be explained in this fashion. For img0040, the elevation range was large and many well-distributed control points were used in accuracy testing. The resulting low magnitude of error in this image can be explained, however, in that the region within which the control points fall displays only slight elevation variation. In extreme corners of img0040, the elevation drops substantially, well away from any measured control point. Because the control points in this image fall within a region of relatively constant elevation, errors introduced by terrain variation are not present. Additionally, the ground control points employed are newly paved and newly painted within days of photo acquisition and ground checking, making them easy to locate on the ground with accuracy as well as in photographs, resulting in a low error magnitude.

Taking the effects of the anomalous img0040 and img0066 out of the analysis yielded little difference in the mean and median error magnitudes, resulting in the conclusion that an expected accuracy estimate for normal operations had been achieved. Images captured and rectified using DOQ imagery for target coordinate reference should yield images upon which measurements can be made in which positional accuracy can be expected to be within 2.3 meters (7.6 feet) of ground position 50 percent of the time and within 3.2 meters (10.5 feet) 75 percent of the time. Barring extreme terrain features such as cliffs or heavy ground cover, field inspectors required to make ground visits to a location should have no difficulty locating the place of interest as its location can be provided within a circular region 4.6 to 6.4 meters (15.2 to 21.0 feet) in diameter or less.

### 6.3. Image Suitability For Level 2 Checks

Imagery suitable for use in Level 2 conservation easement inspections includes high-resolution satellite imagery (ground spatial resolution of 0.6 to 4.0 meters per pixel) or existing (scanned) aerial photographs (Sader et al. 2002). In the case of scanned aerial photographs the ground pixel size depends upon the scale of the photograph. For the purpose of comparison, a ground resolution of 1 meter per pixel (the ground spatial resolution of a DOQ) would certainly be suitable for Level 2 inspections and can be attained from scanned aerial imagery. For the imagery from the digital vertical camera system presented herein to be suitable for use in Level 2 inspections, the ground resolution attained should meet or exceed the resolution capabilities of these existing sources.

The ground pixel size (spatial resolution) for various common flying heights above terrain were computed using the approximate infinity focal length for the 35mm lens setting (obtained from calibration procedures) in conjunction with the photogrammetric scale equation (equation 5), yielding the pixel sizes and ground areas covered by the entire digital camera sensor (table 12). Even at the flying height of 3,048 meters (10,000 feet) above terrain, the digital vertical aerial camera system yields images in which the ground pixel size is 0.67 meters (2.2 feet), only slightly higher than the minimum pixel size in current high-resolution satellite imagery. Additionally, the ground pixel size at 3,048 meters (10,000 feet) above terrain is smaller than the 1-meter (3.3 feet) pixels found in DOQ imagery. Based on pixel size alone, the indication is that the digital camera system is certainly suitable for Level 2 site inspections, and much higher spatial resolution can be attained if desired at lower flying heights above terrain (table 12).

Although the computed ground pixel sizes are a good indication of the capability of the camera to deliver imagery with enough detail to allow Level 2 inspections, the image is degraded slightly during capture by such factors as forward motion of the aircraft and the plexiglass window photographs are captured through. To verify that images from the digital vertical camera system are suitable for Level 2 inspections, the images captured during accuracy testing were examined at high magnification. An example image of Pickeral Pond along the Studmill Road in Township 32 MD, Maine, captured at roughly 3,500 feet above terrain (spatial ground resolution of 0.22 meters (0.72 feet) per pixel) is shown in figure 35 with a portion magnified to display the detail present in the image. As can be seen in figure 35, a high amount of detail is captured. This image was captured while flying at a ground speed over terrain of about 75 miles per hour with the camera sensitivity set at ISO200. Despite forward motion of the aircraft, the combination of large lens aperture and the light sensitivity of the camera sensor allowed use of a shutter speed high enough to prevent significant motion blur in the image. Furthermore, while the plexiglass window does reduce the captured images' contrast significantly, the detail lost due to the imperfect optical characteristics of the plexiglass is minimal. Post-mission processing allows correction for most of the decreased contrast and blurring due to forward motion and the plexiglass window. The imagery obtained from the digital vertical aerial camera system in common operating conditions exhibits very high detail, well suited to support Level 2 site inspections.

Table 12. Ground Pixel Sizes and Entire Image Ground Areas at Common Flying Heights, 35mm Lens Setting

| Flying Height Above Terrain | Ground Pixel Size (m) | Entire Frame Ground Area (ha) | Entire Frame Ground Area (ac) |
|---|---|---|---|
| 1000 ft = 304.9 m | 0.07 | 2.95 | 7.29 |
| 2000 ft = 609.8 m | 0.13 | 10.17 | 25.12 |
| 3000 ft = 914.7 m | 0.20 | 24.06 | 59.46 |
| 4000 ft = 1219.6 m | 0.27 | 43.86 | 108.38 |
| 5000 ft = 1524.4 m | 0.34 | 68.32 | 168.81 |
| 6000 ft = 1829.3 m | 0.40 | 96.26 | 237.85 |
| 7000 ft = 2134.1 m | 0.47 | 132.90 | 328.39 |
| 8000 ft = 2439.0 m | 0.54 | 175.43 | 433.48 |
| 9000 ft = 2743.9 m | 0.60 | 216.58 | 535.16 |
| 10000 ft = 3048.8 m | 0.67 | 270.06 | 667.33 |

Figure 35. Example Image from Digital Vertical Aerial Camera System with Magnified Portion (Inset) for Detailed Viewing

### 6.4. Guidance System Test Results

Many variables are encountered during flight missions that prevent the aircraft from passing directly over a desired location, including wind, temperature, turbulence, pilot skill, and instrument error. In designing the waypoint creation tools, most of these position error sources have been compensated for, and the ArcPad camera control and navigation assistance scripts provide immediate feedback on how far off the aircraft will be, allowing course correction to be made. Because the coordinates of the aircraft's location are captured within 1 second (the sampling rate of the GPS receiver) of the moment the camera is sent a message to capture a photograph, they can be compared to the waypoint coordinates created in GIS software to discover how closely the pilot is able to maneuver the plane into position using the inputs from the ArcPad guidance display.

Examination of the coordinates from test flights in which photographs were taken as compared to the desired waypoint coordinates reveal that the pilot was able to maneuver the aircraft to within 44.5 meters (146 feet), more than double the amount included to compensate for pilot error. Fortunately, the pilot was able to control the amount of tilt within a few degrees, resulting in photographs still containing the desired target area. As the pilot and laptop operator became more familiar with using the displays provided and communicating course correction information, aircraft positional accuracy increased significantly.

Ground based walking tests of the system were carried out as well to discover if the camera system was being triggered at the correct moment. Delays between the moment the message is sent to take a photograph and the moment in which the camera actually fires were estimated at roughly one half of one second. When computing the time interval to send to ArcPad scripts for triggering the camera, 500 milliseconds (0.5 seconds) are subtracted from the computed time to compensate for this delay. In walking tests on the ground, the navigation system was able to guide the operator to within 2 meters (6.6 feet) of the intended target 80 percent of the time, with the camera firing the moment that the operator's foot reached the intended location despite the fact that successive GPS position computations were each made within the error limits of the previous observed coordinates. In the case of photographs taken in flight, at 33.5 meters per second (75 mph) forward speed over ground, if the camera can be triggered within 0.5 seconds of the desired instant, the camera will be positioned a maximum of 16.8 meters (55.1 feet) from the location in which the control software expects the camera to be located for capturing photographs along the flight path. Because the long dimension of the sensor in the camera is, by default, aligned with the aircraft's flight path and a portion of the sensor size along that length is ignored during waypoint creation (the sensor is treated as a square rather than a rectangle), it is safe to assume that the positional error introduced by timing inaccuracy will only result in shifting of the scene along the long dimension of the sensor by an amount that can be ignored.

# Chapter 7

# CONCLUSIONS

As the number of working forest conservation easements increases in the future, efficient and economical methods of monitoring these easements will become increasingly important. Employing higher-resolution imagery to inspect high-priority sites identified in Level-1 procedures of the three-level satellite to ground easement monitoring scheme (Sader et al. 2002) allows inspection of sites at a lower cost, facilitating monitoring of large easements while minimizing financial burdens. The digital aerial camera system presented in this document was designed for this purpose, enabling monitoring staff to acquire images of high-priority sites scattered throughout the easement conveniently and efficiently without requiring purchase of unneeded and expensive commercial imagery.

Using the camera calibration procedures described above, the imagery obtained from the digital camera system can be processed to remove distortion effects, facilitating the rectification process and improving the accuracies attainable from the camera system. Additionally, discovery of lens approximate infinty focal lengths enables GIS software tool creation that allows simple and convenient creation of waypoints for regions requiring Level-2 inspections, eliminating the need for the user to perform computations or make guesses about the proper elevation at which any given waypoint should be overflown. Test flights verify that the altitudes for flying over waypoints, computed using parameters resulting from camera calibration procedures, are sufficient to ensure capturing desired regions in photographs with a high success rate.

To simplify in-flight operation procedures, the software customizations created assist users by providing automation of in-flight navigation and camera control, removing a substantial number of tasks from camera system operators. Removing these tasks frees the operator to provide better guidance to the pilot prior to reaching the desired location. In conjunction with the increased precision of the automated camera triggering, the in-flight software increases the success rate of aerial photography missions, ensuring that economic advantages of the system are maintained.

High positional accuracies (sub-meter) are attainable in imagery rectified with ground control points collected through the use of high-precision instruments and large numbers of easily photo-identifiable control points throughout the images. In normal easement monitoring activities, the absence of identifiable control points due to heavy tree canopy cover and the lack of high-precision control points for use in rectification steps will limit the accuracy attainable. Despite the reduction in accuracy resulting from the lack of high-accuracy ground control, images obtained from the digital camera system rectified using DOQ imagery and second-order polynomial warp operations are capable of delivering coordinate accuracies of 3.2 meters or less 75 percent of the time, well within the original specification of +/- 5 meters. Due to the accuracies attainable, field workers should have little difficulty locating targets of interest on the ground using coordinates obtained from rectified digital aerial images when called upon to perform ground checks. Furthermore, the detail captured at common flying heights ensures that the images are capable of supplying adequate information to complete Level 2 analyses regardless of whether in-camera compression algorithms are employed or not.

The detail captured by the digital vertical aerial camera system at common flying heights above terrain is significant, providing imagery of high quality suitable for use in Level 2 inspections. The spatial resolution of the imagery captured is of a magnitude that, in order to reach the spatial resolution provided in high-resolution satellite imagery, the digital camera system would have to fly at roughly 2,743 meters (9,000 feet) above the terrain being photographed. At the more common lower altitudes of the Cessna aircraft employed in test flights, the ground resolution of the images captured by the digital camera system far exceeds the resolving abilities of commercial high-resolution satellites.

The camera system developed and presented in this thesis fulfils the requirements of the intermediate Level 2 component in a satellite to ground monitoring approach, due to increases in convenience, high image quality, high positional accuracy attainable, cost-effectiveness, flexibility, and ease of use.

# REFERENCES

American Society of Photogrammetry and Remote Sensing (ASPRS). 1980. "Manual of Photogrammetry", 4th ed.

Block, A., Hartigan, K., Heiser, R., Horner, G., Lewandowski, L., Mulvihill-Kuntz, J., Thorn, S. 2004. "Trends in Easement Language and the Status of Current Monitoring on Working Forest Conservation Easements." Master of Science Thesis, University of Michigan School of Natural Resources & Environment.

Bonn Archaeology Software Package (BASP), online at http://www.uni-koeln.de/~al001/radcor.html , last accessed November 2005.

Brown, Duane C. 1971. "Close-Range Camera Calibration". *Photogrammetric Engineering,* 37(8), 855-865.

Commonwealth of Massachusetts. 2003. "Conservation Restriction Stewardship Project – A Pilot Initiative". Published by the Commonwealth of Massachusetts Executive Office of Environmental Affairs.

Dersch, Helmut. 1999. "Correcting Barrel Distortion" Technical publication for proper use of PTCorrect. Found online at www.all-in-one.ee/~dersch/barrel/barrel.html, last accessed November 2005.

Dersch, Helmut, Mirror of old PTools website including distortion correction documents, online at http://www.path.unimelb.edu.au/%7Edersch/, last accessed November 2005.

EPaper Press lens distortion calibration procedures using Helmut Dersch's Panoramic Tools, online at http://epaperpress.com/ptlens/calibrate.html, last accessed November 2005.

Fraser, Clive S. 1997. "Digital Camera Self-Calibration". *ISPRS Journal of Photogrammetry and Remote Sensing,* 52 (1997), 149-159.

Graham, R., Koh, A. 2002. "Digital Aerial Survey Theory and Practice". CRC Press LLC, Boca Raton, FL.

Interagency Committee for Outdoor Recreation. 1999. "Guidelines for Use of Conservation Easements". Washington State Interagency Committee for Outdoor Recreation, Manual 6.

Javafoil online airfoil software, online at http://www.mh-aerotools.de/airfoils/javafoil.htm, last accessed November 2005.

Kenefick, J., Gyer, M., Harp, B. 1972. "Analytical Self-Calibration", *Photogrammetric Engineering*, Vol. 38, 1117-1126.

Levitt, James N. 2003. "The Next Level: The Pingree Forest Partnership as Private Lands". Conservation Innovation. N.p.: The Program on Conservation Innovation at the Harvard Forest, Harvard University.

Lillestrand, R. 1972. "Techniques for Change Detection", *IEEE Trans. On Computers*, 21(7), 654-659.

Lind, Brenda. 2001. "Working Forest Conservation Easements: A Process Guide for Land Trusts, Landowners, and Public Agencies". Land Trust Alliance, Washington, D.C.

Matsuoka, R., Fukue, K., Cho, K., Shimoda, H., Matsumae, Y., Hongo, K., and Fujiwara, S. 2002. "A Study on Calibration of Digital Camera". *ISPRS Commission III Symposium Proceeding, IAPRS,* Vol. XXXIV, part 3A/B.

NASA Glenn Research Center, aerodynamics equations, online at http://www.grc.nasa.gov, last accessed September 2005.

Sader, S.A., Ross, K., Reed, F.C. 2002. "Pingree Forest Partnership: Monitoring Easements at the Landscape Level". *Journal of Forestry,* 100(3), 20-25.

Sader, S.A., Bertrand, M., Wilson, E.H. 2003. "Satellite Change Detection of Forest Harvest Patterns on an Industrial Forest Landscape". *Forest Science,* 49(3), 341-353.

Sullivan, P. 2003. *Conservation Easements Resource Series.* Available online at ATTRA website at www.attra.ncat.org. Last accessed May 2005.

Valkenburg, R. and Evans, P. 2002. "Lens Distortion Calibration by Straightening Lines". Proceedings of Image and Vison Computing, New Zealand 2002.

Wing, M.G., Eklund, A., Kellog, L.D. 2005. "Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability", *Journal of Forestry,* 103(4), 169-173.

Wolberg, G. 1990. "Digital Image Warping", IEEE Computer Society Press.

Wolf, P.R., Dewitt, B.A. 2000. "Elements of Photogrammetry With Applications in GIS", 3rd Ed., McGraw Hill, Inc.

# APPENDICES

# APPENDIX A: ArcView Avenue Scripts

## Script for Reading Camera Parameters File and Placing in Camera Properties Dialog for Verification or Editing

```
theProj=av.GetProject
theDialog=av.FindDialog("CameraParameters")
camFile=LineFile.Make("C:\camera_file.txt".AsFileName,#FILE_PERM_READ)
'set the dialog as not modified
contrPanel=theDialog.FindByName("aControlPanel7")
contrPanel.SetModified(False)

' Read Camera parameters from camera file
camName=camFile.ReadElt
pixRes=camFile.ReadElt
pixUnits=camFile.ReadElt
alongPath=camFile.ReadElt
acrossPath=camFile.ReadElt
lensName=camFile.ReadElt
cFl=camFile.ReadElt
cFlUnits=camFile.ReadElt
fileType=camFile.ReadElt
resSet=camFile.ReadElt
flyHt=camFile.ReadElt
flyHtUnits=camFile.ReadElt
overlap=camFile.ReadElt
horzVert=camFile.ReadElt
tiltErr=camFile.ReadElt
camFile.close

theDialog.open
pixelSz=theDialog.FindByName("pixSize")
pixelSz.SetText(pixRes)
pixelUnit=theDialog.FindByName("pixUnits")
pixelUnit.DefineFromList ({"MICRONS","MILLIMETERS"})
pixelUnit.FindByValue(pixUnits)
pixelUnit.SelectCurrent
pixAlPath=theDialog.FindByName("pixAlongPath")
pixAlPath.SetText(alongPath)
pixAcrPath=theDialog.FindByName("pixAcrossPath")
pixAcrPath.SetText(acrossPath)
lensNm=theDialog.FindByName("lensName")
lensNm.SetText(lensName)
focLen=theDialog.FindByName("lensFocalLength")
focLen.SetText(cFl)
focLenUnit=theDialog.FindByName("focLenUnits")
focLenUnit.DefineFromList({"MILLIMETERS","INCHES"})
focLenUnit.FindByValue(cFlUnits)
focLenUnit.SelectCurrent
defHgt=theDialog.FindByName("defaultHeight")
defHgt.SetText(flyHt)
hgtUnits=theDialog.FindByName("heightUnits")
hgtUnits.DefineFromList({"FEET","METERS"})
```

```
hgtUnits.FindByValue(flyHtUnits)
hgtUnits.SelectCurrent
fileForm=theDialog.FindByName("fileFormat")
fileForm.DefineFromList({"NEF","JPEG"})
fileForm.FindByValue(fileType)
fileForm.SelectCurrent
rSet=theDialog.FindByName("resSetting")
rSet.DefineFromList({"L","M","S"})
rSet.FindByValue(resSet)
rSet.SelectCurrent
ovLap=theDialog.FindByName("OverlapPerct")
ovLap.SetText(overlap)
HV=theDialog.FindByName("posError")
HV.SetText(horzVert)
TEB=theDialog.FindbyName("tiltErrBox")
TEB.SetText(tiltErr)
```

## Script for Clearing All Existing Waypoints in the Waypoints Shapefile

```
theView=Av.GetActiveDoc
theThemes=theView.GetThemes
theView.GetGraphics.Empty

goAhead=msgBox.YesNo("This tool will clear all existing waypoints."+NL+NL+"Do you really
want to clear all waypoints?","Clear Waypoints?",False)

if (goAhead=False) then
  exit
end
' check to see if the waypoint layer exists in the TOC
' if it doesn't, add it to the project
theWaypointLayer=theView.FindTheme("Waypoints.shp")
if (theWaypointLayer=nil) then
  theFilePath=FileDialog.Show("waypoints.shp","Waypoints","Locate The Waypoint File")
  if (theFilePath=nil) Then
    exit
  end
  theSrcname=SrcName.Make(theFilePath.AsString)
  theWaypointLayer=Theme.Make(theSrcname)
  theView.AddTheme(theWaypointLayer)
end

' begin edit session and delete the records
theFTab=theWaypointLayer.GetFTab
theFTab.StartEditingWithRecovery
theFTab.BeginTransaction
for each i in 0..(theFTab.GetNumRecords-1)
  theFTab.RemoveRecord(i)
end
theFTab.UpdateSelection
theFTab.EndTransaction
theFTab.StopEditingWithRecovery(True)
```

### Script to Close the Camera Properties View/Edit Dialog and Save Edits if Made

```
theProj=av.GetProject
theDialog=av.FindDialog("CameraParameters")
camFile=LineFile.Make("C:\camera_file.txt".AsFileName,#FILE_PERM_READ)

' Read Camera parameters from camera file
camName=camFile.ReadElt
pixRes=camFile.ReadElt
pixUnits=camFile.ReadElt
alongPath=camFile.ReadElt
acrossPath=camFile.ReadElt
lensName=camFile.ReadElt
cFl=camFile.ReadElt
cFlUnits=camFile.ReadElt
fileType=camFile.ReadElt
resSet=camFile.ReadElt
flyHt=camFile.ReadElt
flyHtUnits=camFile.ReadElt
overlap=camFile.ReadElt
camFile.close

'get handles on camera parameters dialog objects
pixelSz=theDialog.FindByName("pixSize")
pixelUnit=theDialog.FindByName("pixUnits")
pixAlPath=theDialog.FindByName("pixAlongPath")
pixAcrPath=theDialog.FindByName("pixAcrossPath")
lensNm=theDialog.FindByName("lensName")
focLen=theDialog.FindByName("lensFocalLength")
focLenUnit=theDialog.FindByName("focLenUnits")
defHgt=theDialog.FindByName("defaultHeight")
hgtUnits=theDialog.FindByName("heightUnits")
fileForm=theDialog.FindByName("fileFormat")
rSet=theDialog.FindByName("resSetting")
ovLap=theDialog.FindByName("OverlapPerct")

'check to see if camera parameters have changed
uChanged=False

if (pixelSz.GetText<>pixRes) Then
  uChanged=True
end
if (pixelUnit.GetSelection<>pixUnits) Then
  uChanged=True
end
if (pixAlPath.GetText<>alongPath) Then
  uChanged=True
end
if (pixAcrPath.GetText<>acrossPath) Then
  uChanged=True
end
if (lensNm.GetText<>lensName) Then
  uChanged=True
```

```
end
if (focLen.GetText<>cFl) Then
  uChanged=True
end
if (focLenUnit.GetSelection<>cFlUnits) Then
  uChanged=True
end
if (defHgt.GetText<>flyHt) Then
  uChanged=True
end
if (hgtUnits.GetSelection<>flyHtUnits) Then
  uChanged=True
end
if (fileForm.GetSelection<>fileType) Then
  uChanged=True
end
if (rSet.GetSelection<>resSet) Then
  uChanged=True
end
if (ovLap.GetText<>overlap) Then
  uChanged=True
end
saveChange=True
if (uChanged) Then
  saveChange=msgBox.YesNo ("Camera Parameters Were Altered.  Do You Wish To Save The
Changes?","Save Changes?",True)
end

'save the changes if desired, overwriting old file
if (saveChange) then
  camFile=LineFile.Make("C:\camera_file.txt".AsFileName,#FILE_PERM_WRITE)
  camFile.WriteElt(camName)
  camFile.WriteElt(pixelSz.GetText)
  camFile.WriteElt(pixelUnit.GetSelection)
  camFile.WriteElt(pixAlPath.GetText)
  camFile.WriteElt(pixAcrPath.GetText)
  camFile.WriteElt(lensNm.GetText)
  camFile.WriteElt(focLen.GetText)
  camFile.WriteElt(focLenUnit.GetSelection)
  camFile.WriteElt(fileForm.GetSelection)
  camFile.WriteElt(rSet.GetSelection)
  camFile.WriteElt(defHgt.GetText)
  camFile.WriteElt(hgtUnits.GetSelection)
  camFile.WriteElt(ovLap.GetText)
  camFile.Close
end

av.FindDialog("CameraParameters").Close
```

## Script for Mode 1 Waypoint Creation – Create Waypoint With User-specified Ground Resolution

```
theView=Av.GetActiveDoc
theThemes=theView.GetThemes
theView.GetGraphics.Empty
' get the user's point
newPoint=theView.GetDisplay.ReturnUserPoint
' open camera file or locate it if necessary
camFile=LineFile.Make("C:\camera_file.txt".AsFileName,#FILE_PERM_READ)
while (camFile=nil)
  aFile=FileDialog.Show("camera_file.txt","Text Files","Locate Camera File")
  if (aFile=nil) then
    exit
  end
  camFile=LineFile.Make(aFile,#FILE_PERM_READ)
end
' read camera parameters file
camName=camFile.ReadElt
pixRes=camFile.ReadElt
pixUnits=camFile.ReadElt
alongPath=camFile.ReadElt
acrossPath=camFile.ReadElt
lensName=camFile.ReadElt
cFl=camFile.ReadElt
cFlUnits=camFile.ReadElt
fileType=camFile.ReadElt
resSet=camFile.ReadElt
flyHt=camFile.ReadElt
flyHtUnits=camFile.ReadElt
overlap=camFile.ReadElt
camFile.close
' check to see if the waypoint layer exists in the TOC
' if it doesn't, add it to the project
theWaypointLayer=theView.FindTheme("Waypoints.shp")
if (theWaypointLayer=nil) then
  theFilePath=FileDialog.Show("waypoints.shp","Waypoints","Locate The Waypoint File")
  if (theFilePath=nil) Then
    exit
  end
  theSrcname=SrcName.Make(theFilePath.AsString)
  theWaypointLayer=Theme.Make(theSrcname)
  theView.AddTheme(theWaypointLayer)
end
' define the projection
tR=Rect.MakeXY(-66,80,-72,-80)
thePrj=TransverseMercator.Make(tR)
thePrj.SetCentralMeridian(-69)
thePrj.SetReferenceLatitude(0)
thePrj.SetFalseEasting(500000)
thePrj.SetFalseNorthing(0)
thePrj.SetScale(0.9996)
thePrj.SetSpheroid(#SPHEROID_GRS80)
```

```
' check to see that the DEM is present in a view named "DEM"
' if not, add the view, get the path to the DEM, and add to DEM view
theDEMview=Av.FindDoc("DEM")
if (theDEMview=nil) then
  theDEMview=View.Make
  theDEMview.SetName("DEM")
  demFile=SourceDialog.ShowClass("Locate The DEM",Grid)
  if (demFile=nil) then
    theDEMview.SetName("error upon loading DEM")
    exit
  end
  while (demFile.Count>1)
    demFile=SourceDialog.ShowClass("Select Only One DEM",Grid)
  end
  theDEMlayer=Theme.Make(demFile.Get(0))
  theDEMview.AddTheme(theDEMlayer)
  theDEMlayer.SetVisible(True)
end
theDEM=theDEMview.GetThemes.Get(0).GetGrid
' get desired ground pixel size
cellSize=msgBox.Input("Enter pixel ground size (meters): ","Pixel Size","0.2").AsNumber

' convert pixel size to meters
if (pixUnits="MICRONS") then  ' pixels are in microns
  ab=pixRes.AsNumber*0.000001
else                          ' pixels are in millimeters
  ab=pixRes.AsNumber*0.001
end

if (cFlUnits="MILLIMETERS") then
  focLen=cFl.AsNumber*0.001    ' converts focal length to meters
else                           ' focal length is in inches, convert to meters
  focLen=cFl.AsNumber*0.0254
end

' compute flying height above terrain
theElev=(cellSize*focLen)/ab

' start editing session and add the point
theFTab=theWaypointLayer.GetFTab
theFTab.StartEditingWithRecovery
theFTab.BeginTransaction
theField=theFTab.FindField("Shape")
rec=theFTab.AddRecord
theFTab.SetValue(theField,rec,newPoint)
theField=theFTab.FindField("Easting")
theFTab.SetValue(theField,rec,newPoint.GetX.SetFormat("ddddddd").AsString)
theField=theFTab.FindField("Northing")
theFTab.SetValue(theField,rec,newPoint.GetY.SetFormat("dddddddd").AsString)
newPoint=newPoint.ReturnUnprojected(thePrj)
theField=theFTab.FindField("Long")
theFTab.SetValue(theField,rec,newPoint.GetX.SetFormat("ddd.dddddddd"))
```

```
theField=theFTab.FindField("Lat")
theFTab.SetValue(theField,rec,newPoint.GetY.SetFormat("dd.dddddddd"))

' now convert DD to DMS
latDeg=newPoint.GetY.Truncate
latMin=((newPoint.GetY-latDeg)*60).Truncate
latSec=(((newPoint.GetY-latDeg)*60)-latMin)*60
lonDeg=newPoint.GetX.Truncate
lonMin=((newPoint.GetX-lonDeg)*60).Truncate
lonSec=(((newPoint.GetX-lonDeg)*60)-lonMin)*60
lonDMS=lonDeg.AsString+" d "+lonMin.AsString+" m "+lonSec.SetFormat("dd.dddd").AsString+"
s"
latDMS=latDeg.AsString+" d "+latMin.AsString+" m "+latSec.SetFormat("dd.dddd").AsString+"
s"
theField=theFTab.FindField("Lat_dms")
theFTab.SetValue(theField,rec,latDMS)
theField=theFTab.FindField("Lon_dms")
theFTab.SetValue(theField,rec,lonDMS)
ftName=MsgBox.Input("Enter A Name For This Point","Name Point","")
theField=theFTab.FindField("Feat_name")
theFtab.SetValue(theField,rec,ftName)
photoBool=MsgBox.YesNo("Is This A Photo Point?","Take A Photo Here?",True)
if (photoBool) then
  theField=theFTab.FindField("Photo")
  theFTab.SetValue(theField,rec,"1")
  airspeed=MsgBox.Input("Enter Desired Airspeed (mph):","Airspeed","75")
  theField=theFTab.FindField("Airspeed")
  theFtab.SetValue(theField,rec,airspeed)
else
  theField=theFTab.FindField("Photo")
  theFTab.SetValue(theField,rec,"0")
  theField=theFTab.FindField("Airspeed")
  theFtab.SetValue(theField,rec,"75")
end
theElev=theElev+theDEM.CellValue(newPoint,Prj.MakeNull)
if (flyHtUnits="FEET") then
  theElev=Units.Convert(theElev,#UNITS_LINEAR_METERS,#UNITS_LINEAR_FEET)
  multiplier=100
else
  multiplier=10
end
' round the flying height down to nearest 100 feet or 10 meters
theElev=((theElev/multiplier).Floor)*multiplier
theField=theFTab.FindField("Fly_ht")
theFTab.SetValue(theField,rec,theElev)
theField=theFTab.FindField("Ht_units")
theFTab.SetValue(theField,rec,flyHtUnits)
theField=theFTab.FindField("Cam_res")
theFTab.SetValue(theField,rec,resSet)
theField=theFTab.FindField("Format")
theFTab.SetValue(theField,rec,fileType)
theFTab.UpdateSelection
theFTab.EndTransaction
```

```
theFTab.StopEditingWithRecovery(True)
```

## Script for Mode 2 Waypoint Creation – Create Waypoint With User-specified Ground Area

```
theView=Av.GetActiveDoc
theThemes=theView.GetThemes
theView.GetGraphics.Empty

' get the user's point
newCircle=theView.GetDisplay.ReturnUserCircle
newPoint=newCircle.ReturnCenter

' open camera file or locate it if necessary
camFile=LineFile.Make("C:\camera_file.txt".AsFileName,#FILE_PERM_READ)
while (camFile=nil)
  aFile=FileDialog.Show("camera_file.txt","Text Files","Locate Camera File")
  if (aFile=nil) then
    exit
  end
  camFile=LineFile.Make(aFile,#FILE_PERM_READ)
end

' read camera parameters file
camName=camFile.ReadElt
pixRes=camFile.ReadElt
pixUnits=camFile.ReadElt
alongPath=camFile.ReadElt
acrossPath=camFile.ReadElt
lensName=camFile.ReadElt
cFl=camFile.ReadElt
cFlUnits=camFile.ReadElt
fileType=camFile.ReadElt
resSet=camFile.ReadElt
flyHt=camFile.ReadElt
flyHtUnits=camFile.ReadElt
overlap=camFile.ReadElt
posError=camFile.ReadElt
tiltErrorDeg=camFile.ReadElt
camFile.close

' check to see if the waypoint layer exists in the TOC
' if it doesn't, add it to the project
theWaypointLayer=theView.FindTheme("Waypoints.shp")
if (theWaypointLayer=nil) then
  theFilePath=FileDialog.Show("waypoints.shp","Waypoints","Locate The Waypoint File")
  if (theFilePath=nil) Then
    exit
  end
  theSrcname=SrcName.Make(theFilePath.AsString)
  theWaypointLayer=Theme.Make(theSrcname)
  theView.AddTheme(theWaypointLayer)
end
```

```
' define the projection
tR=Rect.MakeXY(-66,80,-72,-80)
thePrj=TransverseMercator.Make(tR)
thePrj.SetCentralMeridian(-69)
thePrj.SetReferenceLatitude(0)
thePrj.SetFalseEasting(500000)
thePrj.SetFalseNorthing(0)
thePrj.SetScale(0.9996)
thePrj.SetSpheroid(#SPHEROID_GRS80)

' check to see that the DEM is present in a view named "DEM"
' if not, add the view, get the path to the DEM, and add to DEM view
theDEMview=Av.FindDoc("DEM")
if (theDEMview=nil) then
  theDEMview=View.Make
  theDEMview.SetName("DEM")
  demFile=SourceDialog.ShowClass("Locate The DEM",Grid)
  if (demFile=nil) then
    theDEMview.SetName("error upon loading DEM")
    exit
  end
  while (demFile.Count>1)
    demFile=SourceDialog.ShowClass("Select Only One DEM",Grid)
  end
  theDEMlayer=Theme.Make(demFile.Get(0))
  theDEMview.AddTheme(theDEMlayer)
  theDEMlayer.SetVisible(True)
end
theDEM=theDEMview.GetThemes.Get(0).GetGrid

' convert radius of newCircle to meters, double to get width of extent
maxExtent=Units.Convert((2*newCircle.GetRadius),theView.GetUnits,#UNITS_LINEAR_METERS)

' determine least sized dimension of sensor
pixelSize=pixRes.AsNumber
sizeAcross=acrossPath.AsNumber*pixelSize
sizeAlong=alongPath.AsNumber*pixelSize
if (sizeAcross<sizeAlong) then
  ab=sizeAcross
else
  ab=sizeAlong
end

' convert minimum pixel width to meters
if (pixUnits="MICRONS") then ' pixels are in microns
  ab=ab*0.000001
else ' pixels are in millimeters
  ab=ab*0.001
end

' convert focal length to meters
focalLength=cFl.AsNumber
```

```
if (cFlUnits="MILLIMETERS") then ' focal length is in millimeters
  focalLength=focalLength*0.001
else ' focal length is in inches
  focalLength=focalLength*0.0254
end

' compute padding amount to add to ground distance in order to
' compensate for horizontal & vertical position error, also
' for tile error, using error estimates read from camera file
hvCorrect=((2*focalLength-ab)*posError.AsNumber)/(2*focalLength)
tiltRadians=tiltErrorDeg.AsNumber.AsRadians
tiltCorrect=(maxExtent/2)*((-
1*((ab^2)+4*(focalLength^2))*tiltRadians.Sin)/(2*ab*(ab*tiltRadians.Sin-
2*focalLength*tiltRadians.Cos)))
maxExtent=maxExtent+hvCorrect+tiltCorrect

' compute flying height (meters) using photogrammetric scale equation
h=(focalLength*maxExtent)/ab

' Check to see if computed height is less than the minimum flying height allowed
' Change to minimum height if needed
' Minimum height above terrain is 1000 feet=305 meters
if (h<305) then
  h=305
end

' convert flying height to feet if needed according to camera file
if (flyHtUnits="FEET") then
  flyHt=Units.Convert(h,#UNITS_LINEAR_METERS,#UNITS_LINEAR_FEET)
else
  flyHt=h
end

' start editing session and add the point
theFTab=theWaypointLayer.GetFTab
theFTab.StartEditingWithRecovery
theFTab.BeginTransaction
theField=theFTab.FindField("Shape")
rec=theFTab.AddRecord
theFTab.SetValue(theField,rec,newPoint)
theField=theFTab.FindField("Easting")
theFTab.SetValue(theField,rec,newPoint.GetX.SetFormat("ddddddd").AsString)
theField=theFTab.FindField("Northing")
theFTab.SetValue(theField,rec,newPoint.GetY.SetFormat("dddddddd").AsString)
newPoint=newPoint.ReturnUnprojected(thePrj)
theField=theFTab.FindField("Long")
theFTab.SetValue(theField,rec,newPoint.GetX.SetFormat("ddd.dddddddd"))
theField=theFTab.FindField("Lat")
theFTab.SetValue(theField,rec,newPoint.GetY.SetFormat("dd.dddddddd"))
' convert decimal degree coordinates to degrees, minutes, seconds string
latDeg=newPoint.GetY.Truncate
latMin=((newPoint.GetY-latDeg)*60).Truncate
latSec=(((newPoint.GetY-latDeg)*60)-latMin)*60
```

```
lonDeg=newPoint.GetX.Truncate
lonMin=((newPoint.GetX-lonDeg)*60).Truncate
lonSec=(((newPoint.GetX-lonDeg)*60)-lonMin)*60
lonDMS=lonDeg.AsString+" d "+lonMin.AsString+" m "+lonSec.SetFormat("dd.dddd").AsString+"
s"
latDMS=latDeg.AsString+" d "+latMin.AsString+" m "+latSec.SetFormat("dd.dddd").AsString+"
s"
theField=theFTab.FindField("Lat_dms")
theFTab.SetValue(theField,rec,latDMS)
theField=theFTab.FindField("Lon_dms")
theFTab.SetValue(theField,rec,lonDMS)

ftName=MsgBox.Input("Enter A Name For This Point","Name Point","")
theField=theFTab.FindField("Feat_name")
theFtab.SetValue(theField,rec,ftName)
photoBool=MsgBox.YesNo("Is This A Photo Point?","Take A Photo Here?",True)
if (photoBool) then
  theField=theFTab.FindField("Photo")
  theFTab.SetValue(theField,rec,"1")
  airspeed=MsgBox.Input("Enter Desired Airspeed (mph):","Airspeed","75")
  theField=theFTab.FindField("Airspeed")
  theFtab.SetValue(theField,rec,airspeed)
else
  theField=theFTab.FindField("Photo")
  theFTab.SetValue(theField,rec,"0")
  theField=theFTab.FindField("Airspeed")
  theFtab.SetValue(theField,rec,75)
end

' get the elevation of the point from DEM, add to height above terrain
theElev=theDEM.CellValue(newPoint,Prj.MakeNull)
if (flyHtUnits="FEET") then
  theElev=Units.Convert(theElev,#UNITS_LINEAR_METERS,#UNITS_LINEAR_FEET)
  multiplier=100
else
  multiplier=10
end
flyHt=flyHt+theElev

' round the flying height up to nearest 100 feet or 10 meters
flyHt=((flyHt/multiplier).Ceiling)*multiplier

theField=theFTab.FindField("Fly_ht")
theFTab.SetValue(theField,rec,flyHt)
theField=theFTab.FindField("Ht_units")
theFTab.SetValue(theField,rec,flyHtUnits)
theField=theFTab.FindField("Cam_res")
theFTab.SetValue(theField,rec,resSet)
theField=theFTab.FindField("Format")
theFTab.SetValue(theField,rec,fileType)
theFTab.UpdateSelection
theFTab.EndTransaction
theFTab.StopEditingWithRecovery(True)
```

## Script for Mode 3 Waypoint Creation – Create Multiple Waypoints Along User-specified Path

```
theView=Av.GetActiveDoc
theThemes=theView.GetThemes
theView.GetGraphics.Empty

' get the user's polyline
newPoly=theView.GetDisplay.ReturnUserPolyline
grShape=GraphicShape.Make(newPoly)
grList=theView.GetGraphics
grList.Add(grShape)

' open camera file or locate it if necessary
camFile=LineFile.Make("C:\camera_file.txt".AsFileName,#FILE_PERM_READ)
while (camFile=nil)
  aFile=FileDialog.Show("camera_file.txt","Text Files","Locate Camera File")
  if (aFile=nil) then
    exit
  end
  camFile=LineFile.Make(aFile,#FILE_PERM_READ)
end

' read camera parameters file
camName=camFile.ReadElt
pixRes=camFile.ReadElt
pixUnits=camFile.ReadElt
alongPath=camFile.ReadElt
acrossPath=camFile.ReadElt
lensName=camFile.ReadElt
cFl=camFile.ReadElt
cFlUnits=camFile.ReadElt
fileType=camFile.ReadElt
resSet=camFile.ReadElt
flyHt=camFile.ReadElt
flyHtUnits=camFile.ReadElt
overlap=camFile.ReadElt
posError=camFile.ReadElt
tiltErrorDeg=camFile.ReadElt
camFile.close

' check to see if the waypoint layer exists in the TOC
' if it doesn't, add it to the project
theWaypointLayer=theView.FindTheme("Waypoints.shp")
if (theWaypointLayer=nil) then
  theFilePath=FileDialog.Show("waypoints.shp","Waypoints","Locate The Waypoint File")
  if (theFilePath=nil) Then
    exit
  end
  theSrcname=SrcName.Make(theFilePath.AsString)
  theWaypointLayer=Theme.Make(theSrcname)
  theView.AddTheme(theWaypointLayer)
end
```

```
' define the projection
tR=Rect.MakeXY(-66,80,-72,-80)
thePrj=TransverseMercator.Make(tR)
thePrj.SetCentralMeridian(-69)
thePrj.SetReferenceLatitude(0)
thePrj.SetFalseEasting(500000)
thePrj.SetFalseNorthing(0)
thePrj.SetScale(0.9996)
thePrj.SetSpheroid(#SPHEROID_GRS80)

' check to see that the DEM is present in a view named "DEM"
' if not, add the view, get the path to the DEM, and add to DEM view
theDEMview=Av.FindDoc("DEM")
if (theDEMview=nil) then
  theDEMview=View.Make
  theDEMview.SetName("DEM")
  demFile=SourceDialog.ShowClass("Locate The DEM",Grid)
  if (demFile=nil) then
    theDEMview.SetName("error upon loading DEM")
    exit
  end
  while (demFile.Count>1)
    demFile=SourceDialog.ShowClass("Select Only One DEM",Grid)
  end
  theDEMlayer=Theme.Make(demFile.Get(0))
  theDEMview.AddTheme(theDEMlayer)
  theDEMlayer.SetVisible(True)
end
theDEM=theDEMview.GetThemes.Get(0).GetGrid

' Get overlap percentage, default from camera file
ovLapPerct=MsgBox.Input("Enter your desired overlap percentage","Overlap
Percentage",overlap)
while ((ovLapPerct.AsNumber<10) OR (ovLapPerct.AsNumber>80))
  ovLapPerct=MsgBox.Input("Overlap must be between 10 and 80 percent. Re-enter: ","Re-enter
Overlap Percent","10")
end
overLap=ovLapPerct

' get buffer width and setup constants needed for scale equation computations
bufferWidth=MsgBox.Input("Please enter your desired buffer width for one side of the line
(feet):","Enter Buffer Width","1000")
while (bufferWidth.AsNumber<300)
  bufferWidth=MsgBox.Input("Buffer width too small.  Must be 300 feet or more"+NL+"Re-enter
your buffer (feet):","Re-enter Buffer Width","1000")
end

' convert buffer width to meters
maxExtent=2*(Units.Convert(bufferWidth.AsNumber,#UNITS_LINEAR_FEET,#UNITS_LINEAR_M
ETERS))

' compute physical size of camera sensor
```

```
pixelSize=pixRes.AsNumber
sizeAcross=acrossPath.AsNumber*pixelSize
sizeAlong=alongPath.AsNumber*pixelSize
if (sizeAcross<sizeAlong) then
  ab=sizeAcross
else
  ab=sizeAlong
end

' convert pixel widths to meters
if (pixUnits="MICRONS") then ' pixels are in microns
  ab=ab*0.000001
  sizeAlong=sizeAlong*0.000001
  sizeAcross=sizeAcross*0.000001
else ' pixels are in millimeters
  ab=ab*0.001
  sizeAlong=sizeAlong*0.001
  sizeAcross=sizeAcross*0.001
end

' convert focal length to meters
focalLength=cFl.AsNumber
if (cFlUnits="MILLIMETERS") then ' focal length is in millimeters
  focalLength=focalLength*0.001
else ' focal length is in inches
  focalLength=focalLength*0.0254
end

' add 15% to total buffer to include extra area around buffer
maxExtent=maxExtent*1.15

' compute padding amount to add to ground distance in order to
' compensate for horizontal & vertical position error, also
' for tile error, using error estimates read from camera file
hvCorrect=((2*focalLength-ab)*posError.AsNumber)/(2*focalLength)
tiltRadians=tiltErrorDeg.AsNumber.AsRadians
tiltCorrect=(maxExtent/2)*((-
1*((ab^2)+4*(focalLength^2))*tiltRadians.Sin)/(2*ab*(ab*tiltRadians.Sin-
2*focalLength*tiltRadians.Cos)))
maxExtent=maxExtent+hvCorrect+tiltCorrect

' compute flying height from maxExtent (buffer width), ab, and focalLength
flyHt=(focalLength*maxExtent)/ab

' check to see if computed height is less than the minimum flying height allowed
' change to minimum height if needed (minimum height is 305 meters=1000 feet)
if (flyHt<305) then
  flyHt=305
end

' compute ground distances (map units) along & across path using flyHt
groundAlong=(flyHt*sizeAlong)/focalLength
groundAcross=(flyHt*sizeAcross)/focalLength
```

```
' compute amount of distance not overlapped (5% error margin added)
lapPerct=(100-overlap.AsNumber)*0.01
lapDist=groundAlong*lapPerct
lapDist=lapDist*0.95

' compute estimated time between photo points
timebetpts=lapDist/33.528
msgBox.Info("At 75 MPH, time between photo points is " + timebetpts.AsString +
"Seconds","Time Interval")

' compute number of points needed along line
lineLength=Units.Convert(newPoly.ReturnLength,theView.GetUnits,#UNITS_LINEAR_METERS)
numPoints=(lineLength/lapDist).Ceiling

aPoint=newPoly.Along(0)
aPoint=aPoint.ReturnUnprojected(thePrj)
' compute the flying height based upon elevation obtained from DEM
theElev=theDEM.CellValue(aPoint,Prj.MakeNull)
flyHt=flyHt+theElev

' convert flying height to feet if necessary
if (flyHtUnits="FEET") then
  flyHt=Units.Convert(flyHt,#UNITS_LINEAR_METERS,#UNITS_LINEAR_FEET)
end

if (flyHtUnits="FEET") then
  theElev=Units.Convert(theElev,#UNITS_LINEAR_METERS,#UNITS_LINEAR_FEET)
  multiplier=100
else
  multiplier=10
end
flyHt=flyHt+theElev

' round the flying height up to nearest 100 feet or 10 meters
flyHt=((flyHt/multiplier).Ceiling)*multiplier

' start editing session and add the points
lineName=msgBox.Input("Enter a name for this line: ","Line Name","")
theFTab=theWaypointLayer.GetFTab
theFTab.StartEditingWithRecovery
theFTab.BeginTransaction

'create two lineup points at the head of the line
newPoint=newPoly.Along(100/numPoints)
originPoint=newPoly.Along(0)
deltaX=2*(newPoint.GetX-originPoint.GetX)
deltaY=2*(newPoint.GetY-originPoint.GetY)

for each i in 1 .. 2
theField=theFTab.FindField("Shape")
  newPoint=Point.Make(originPoint.GetX-((3-i)*deltaX), originPoint.GetY-((3-i)*deltaY))
  rec=theFTab.AddRecord
```

```
  theFTab.SetValue(theField,rec,newPoint)
  theField=theFTab.FindField("Easting")
  theFTab.SetValue(theField,rec,newPoint.GetX.SetFormat("ddddddd").AsString)
  theField=theFTab.FindField("Northing")
  theFTab.SetValue(theField,rec,newPoint.GetY.SetFormat("dddddddd").AsString)
  newPoint=newPoint.ReturnUnprojected(thePrj)
  theField=theFTab.FindField("Long")
  theFTab.SetValue(theField,rec,newPoint.GetX.SetFormat("ddd.dddddddd"))
  theField=theFTab.FindField("Lat")
  theFTab.SetValue(theField,rec,newPoint.GetY.SetFormat("dd.dddddddd"))
  ftName=lineName + " Lineup Point " + " " + i.SetFormat("ddd").AsString
  latDeg=newPoint.GetY.Truncate
  latMin=((newPoint.GetY-latDeg)*60).Truncate
  latSec=(((newPoint.GetY-latDeg)*60)-latMin)*60
  lonDeg=newPoint.GetX.Truncate
  lonMin=((newPoint.GetX-lonDeg)*60).Truncate
  lonSec=(((newPoint.GetX-lonDeg)*60)-lonMin)*60
  lonDMS=lonDeg.AsString+" d "+lonMin.AsString+"
m"+lonSec.SetFormat("dd.dddd").AsString+" s"
  latDMS=latDeg.AsString+" d "+latMin.AsString+" m "+latSec.SetFormat("dd.dddd").AsString+"
s"
  theField=theFTab.FindField("Lat_dms")
  theFTab.SetValue(theField,rec,latDMS)
  theField=theFTab.FindField("Lon_dms")
  theFTab.SetValue(theField,rec,lonDMS)
  theField=theFTab.FindField("Feat_name")
  theFtab.SetValue(theField,rec,ftName)
  theField=theFTab.FindField("Photo")
  theFTab.SetValue(theField,rec,"0")
  theField=theFTab.FindField("Airspeed")
  theFtab.SetValue(theField,rec,75)
  theField=theFTab.FindField("Fly_ht")
  theFTab.SetValue(theField,rec,flyHt)
  theField=theFTab.FindField("Ht_units")
  theFTab.SetValue(theField,rec,flyHtUnits)
  theField=theFTab.FindField("Cam_res")
  theFTab.SetValue(theField,rec,resSet)
  theField=theFTab.FindField("Format")
  theFTab.SetValue(theField,rec,"JPEG")
  theField=theFTab.FindField("Path")
  theFTab.SetValue(theField,rec,1)
end

' create the points along the line now
for each i in 0..numPoints
  newPoint=newPoly.Along((100/numPoints)*i)
  theField=theFTab.FindField("Shape")
  rec=theFTab.AddRecord
  theFTab.SetValue(theField,rec,newPoint)
  theField=theFTab.FindField("Easting")
  theFTab.SetValue(theField,rec,newPoint.GetX.SetFormat("ddddddd").AsString)
  theField=theFTab.FindField("Northing")
  theFTab.SetValue(theField,rec,newPoint.GetY.SetFormat("dddddddd").AsString)
```

```
  newPoint=newPoint.ReturnUnprojected(thePrj)
  theField=theFTab.FindField("Long")
  theFTab.SetValue(theField,rec,newPoint.GetX.SetFormat("ddd.dddddddd"))
  theField=theFTab.FindField("Lat")
  theFTab.SetValue(theField,rec,newPoint.GetY.SetFormat("dd.dddddddd"))
  ftName=lineName + " " + i.SetFormat("ddd").AsString
  latDeg=newPoint.GetY.Truncate
  latMin=((newPoint.GetY-latDeg)*60).Truncate
  latSec=(((newPoint.GetY-latDeg)*60)-latMin)*60
  lonDeg=newPoint.GetX.Truncate
  lonMin=((newPoint.GetX-lonDeg)*60).Truncate
  lonSec=(((newPoint.GetX-lonDeg)*60)-lonMin)*60
  lonDMS=lonDeg.AsString+" d "+lonMin.AsString+" m
"+lonSec.SetFormat("dd.dddd").AsString+" s"
  latDMS=latDeg.AsString+" d "+latMin.AsString+" m "+latSec.SetFormat("dd.dddd").AsString+"
s"
  theField=theFTab.FindField("Lat_dms")
  theFTab.SetValue(theField,rec,latDMS)
  theField=theFTab.FindField("Lon_dms")
  theFTab.SetValue(theField,rec,lonDMS)
  theField=theFTab.FindField("Feat_name")
  theFtab.SetValue(theField,rec,ftName)
  theField=theFTab.FindField("Photo")
  theFTab.SetValue(theField,rec,"1")
  theField=theFTab.FindField("Airspeed")
  theFtab.SetValue(theField,rec,75)
  theField=theFTab.FindField("Fly_ht")
  theFTab.SetValue(theField,rec,flyHt)
  theField=theFTab.FindField("Ht_units")
  theFTab.SetValue(theField,rec,flyHtUnits)
  theField=theFTab.FindField("Cam_res")
  theFTab.SetValue(theField,rec,resSet)
  theField=theFTab.FindField("Format")
  theFTab.SetValue(theField,rec,"JPEG")
  theField=theFTab.FindField("Path")
  theFTab.SetValue(theField,rec,1)
end

theFTab.UpdateSelection
theFTab.EndTransaction
theFTab.StopEditingWithRecovery(True)
theView.Invalidate
```

## Script for Photography Mission Flight Path Optimization

```
theView=Av.GetActiveDoc
theThemes=theView.GetThemes

' check to see if the waypoint layer exists in the TOC
' if it doesn't, add it to the project
theWaypointLayer=theView.FindTheme("Waypoints.shp")
if (theWaypointLayer=nil) then
  theFilePath=FileDialog.Show("waypoints.shp","Waypoints","Locate The Waypoint File")
  if (theFilePath=nil) Then
    exit
  end
  theSrcname=SrcName.Make(theFilePath.AsString)
  theWaypointLayer=Theme.Make(theSrcname)
  theView.AddTheme(theWaypointLayer)
end

numRecords=theWaypointLayer.GetFTab.GetNumRecords

' begin editing and clear out any previous route optimizations
theFTab=theWaypointLayer.GetFTab
theFTab.StartEditingWithRecovery
theFTab.BeginTransaction
theField=theFTab.FindField("Id")

For Each i in 0..(numRecords-1)
  theFTab.SetValue(theField,i,0)
end

' make a list of all waypoints that are currently set
theList=List.Make
For Each i in 0..(numRecords-1)
  theList.Add(i)
End

' set the origin at the Old Town Airport and make a list for distances
theCurrOrigin=Point.Make(525816,4977778)
theDistanceList=List.Make
currNumRecords=numRecords

i=0
While (i<numRecords)
  ' on each pass of the loop, initialize the lowest distance to the
  ' distance between the current origin and the first record in theList
  theField=theFTab.FindField("Easting")
  theEasting=theFTab.ReturnValue(theField,theList.Get(0)).AsNumber
  theField=theFTab.FindField("Northing")
  theNorthing=theFTab.ReturnValue(theField,theList.Get(0)).AsNumber
  theNewPoint=Point.Make(theEasting,theNorthing)
  lowestVal=theCurrOrigin.Distance(theNewPoint)
  lowestValIndex=0
```

```
' compute all the distances for the current collection of waypoints
' from the current origin
For each k in 0..(currNumRecords-1)
  theField=theFTab.FindField("Easting")
  theEasting=theFTab.ReturnValue(theField,theList.Get(k)).AsNumber
  theField=theFTab.FindField("Northing")
  theNorthing=theFTab.ReturnValue(theField,theList.Get(k)).AsNumber
  theNewPoint=Point.Make(theEasting,theNorthing)
  theDistanceList.Add(theCurrOrigin.Distance(theNewPoint))
  if (theCurrOrigin.Distance(theNewPoint)<lowestVal) then
    lowestVal=theCurrOrigin.Distance(theNewPoint)
    lowestValIndex=k
  end
end

' check to see if the next closest point is part of a path
' if so, back up to start of that line and add order numbers to all
' points in the line, removing them from theList
theField=theFTab.FindField("Path")
theValue=theFTab.ReturnValue(theField,theList.Get(lowestValIndex))
theField=theFTab.FindField("Feat_name")
theFeatName=theFTab.ReturnValue(theField,theList.Get(lowestValIndex))

if (theValue=1) then  ' closest point is in a path
  ' check to see if the closest point is the beginning of a path
  theString=theFeatName.Right(16)
  if (theString="Lineup Point  01") then
    theBoolean=False
  else
    theBoolean=True
  end
  While (theBoolean)
    lowestValIndex=lowestValIndex-1
    theFeatName=theFTab.ReturnValue(theField,theList.Get(lowestValIndex))
    theString=theFeatName.Right(16)
    if (theString="Lineup Point  01") then
      theBoolean=False
    else
      theBoolean=True
    end
  end

  ' now at beginning of path, add the path point-by-point until end of path
  ' start by adding the first point of the path and remove it
  theField=theFTab.FindField("Id")
  theFTab.SetValue(theField,theList.Get(lowestValIndex),i)
  theList.Remove(lowestValIndex)
  theDistanceList.Remove(lowestValIndex)
  i=i+1
  currNumRecords=currNumRecords-1

  ' now add the remaining points in the path, stopping when either the line
  ' ends (Path is no longer=1) or the beginning of a new line is detected
```

```
  theField=theFTab.FindField("Path")
  theValue=theFTab.ReturnValue(theField,theList.Get(lowestValIndex))
  theField=theFTab.FindField("Feat_name")
  theFeatName=theFTab.ReturnValue(theField,theList.Get(lowestValIndex))
  theString=theFeatName.Right(16)
  if ((theString="Lineup Point  01") Or (theValue=0)) then
    theBoolean=False
  else
    theBoolean=True
  end
  While (theBoolean)
    theField=theFTab.FindField("Id")
    theFTab.SetValue(theField,theList.Get(lowestValIndex),i)
    theField=theFTab.FindField("Path")
    theValue=theFTab.ReturnValue(theField,theList.Get(lowestValIndex))
    theField=theFTab.FindField("Easting")
    theEasting=theFTab.ReturnValue(theField,theList.Get(lowestValIndex)).AsNumber
    theField=theFTab.FindField("Northing")
    theNorthing=theFTab.ReturnValue(theField,theList.Get(lowestValIndex)).AsNumber
    theNewPoint=Point.Make(theEasting,theNorthing)
    theCurrOrigin=theNewPoint
    theList.Remove(lowestValIndex)
    theDistanceList.Remove(lowestValIndex)
    i=i+1
    currNumRecords=currNumRecords-1
    ' check to see if the end of the list has been reached
    if (lowestValIndex<theList.Count) then
      theField=theFTab.FindField("Feat_name")
      theFeatName=theFTab.ReturnValue(theField,theList.Get(lowestValIndex))
      theString=theFeatName.Right(16)
      if ((theString="Lineup Point  01") Or (theValue=0)) then
        theBoolean=False
      else
        theBoolean=True
      end
    else ' the end of the list has been reached
      theBoolean=False
    end
  end
else
  ' write the current i (order) to the ID field of the record that is
  ' closest to the current origin
  theField=theFTab.FindField("Id")
  theFTab.SetValue(theField,theList.Get(lowestValIndex),i)
  currNumRecords=currNumRecords-1

  ' set the current origin to the next closest point and remove the next
  ' closest point's indices from theList and theDistanceList
  theField=theFTab.FindField("Easting")
  theEasting=theFTab.ReturnValue(theField,theList.Get(lowestValIndex)).AsNumber
  theField=theFTab.FindField("Northing")
  theNorthing=theFTab.ReturnValue(theField,theList.Get(lowestValIndex)).AsNumber
  theNewPoint=Point.Make(theEasting,theNorthing)
```

```
    theCurrOrigin=theNewPoint
    theList.Remove(lowestValIndex)
    theDistanceList.Remove(lowestValIndex)
    i=i+1
  end
end

' stop editing and save changes
theFTab.UpdateSelection
theFTab.EndTransaction
theFTab.StopEditingWithRecovery(True)
```

# APPENDIX B: VBScript Code for ArcPad Customizations

## Script for Navigation and Camera Triggering Operations

```
Sub startNav
    Dim theLayer, theRecords, estg, nrthg, ptName, layerCount, i, theHandle
'If Nikon Capture is not running, run it
    theHandle=System.FindWindow ("#32770", "Nikon Capture Camera Control")
    If theHandle=0 Then
        Application.Run("C:\Program Files\Nikon\NCapture4\Control\NControl.exe")
    End If

'If there are more than 0 layers and the first layer is a point layer, navigate
    If (Map.Layers.Count > 0) Then
        Set theLayer=Map.Layers(1)
        Set theRecords=theLayer.Records

    'make sure that a waypoint layer exists at layer 1
        If theRecords.Fields("Visited") Is Nothing Then
            MsgBox "Waypoint Layer Must Be Loaded At Top of TOC!"_
                  , vbExclamation, "ERROR"
            Application.UserProperties("FlightInProgress")="NotFlying"
            Exit Sub
        End If

    'Disable the toolbar button for starting navigation, enable the button to stop
        Application.Toolbars("Flight").Item("NavigateCapture").Enabled=False
        Application.Toolbars("Flight").Item("StopNavigation").Enabled=True
        Application.Toolbars("Flight").Item("NextPoint").Enabled=True
        Application.Toolbars("Flight").Item("ResetFlags").Enabled=True

    'set all layers identifiable property to false except the waypoint layer
        layerCount=Application.Map.Layers.Count
        i=2
        Do While i<=layerCount
            If Application.Map.Layers(i).Layertype=0 And _
            (Not Application.Map.Layers(i).Comments="ECW Local Raster File") Then
                Application.Map.Layers(i).Identifiable=False
            End If
            i=i+1
        Loop

    'check to make sure that the layer is editable
        If Not theLayer.Editable Then
            theLayer.Editable=True
        End If

    'start the GPS if it is not already active
        If Not Application.GPS.IsOpen Then
            Application.GPS.Open
            Application.ExecuteCommand("gpspositionwindow")
        End If
```

```
    ' start the GPS tracklog if not active
        Application.ExecuteCommand("gpstracklog")
        ' if the navigation is not active, activate it and send the bookmarked waypoint
        ' to navigation.  The bookmarked waypoint may not be the first in file - it is
        ' the current bookmarked waypoint.  Upon loading the waypoint layer, the first
        ' waypoint will automatically be bookmarked.  If there is a selected record in
        ' the waypoint layer, use that as the starting navigation point
        Dim theSelLayer, yn, currBookmark
        If Not Navigation.Active Then
            Set theSelLayer=Map.SelectionLayer
            ' if a record is selected, navigate to it if ok with user
            If (Not theSelLayer Is Nothing) Then
                currBookmark = theRecords.Bookmark
                theRecords.Bookmark=Map.SelectionBookmark
                yn = MsgBox ("Waypoint " & theRecords.Fields("Feat_name") _
                    & " Is Currently Selected.  Start Navigation There?"_
                    ,vbYesNo,"Navigate To Selected?")
                If yn=7 Then
                    ' NO was clicked, return bookmark to previous record
                    theRecords.Bookmark = currBookmark
                End If
            End If
            estg=theRecords.Fields("Easting").Value
            nrthg=theRecords.Fields("Northing").Value
            ptName=theRecords.Fields("Feat_name").Value
            Navigation.GotoXy estg, nrthg, ptName
        End If
        Application.UserProperties("FlightInProgress")="FlightInProgress"
    Else
        Application.UserProperties("FlightInProgress")="NotFlying"
    End If
End Sub
```

## Script, Execute on GPS Position Message Receipt, Update Status Information, Compute Closest Point Along Current Bearing, Trigger Camera if In Position

```
Sub gpsPositionUpdate
    Dim theLayer, theRecords, estg, nrthg, ptName
    Dim photoFlag, flyHt, htUnits, flySpeed, ptId, posError
    Dim currX, currY, currAlt, currCOG, currSOG, brg2pt, dist2pt, time2target
    ' if not in flight mode, then exit the sub, operate as normal
    If Application.UserProperties("FlightInProgress") = "NotFlying" Then
        Exit Sub
    End If

    On Error Resume Next
    ' get current X and Y coordinates, COG, height, and speed from GPS
        currX = GPS.X
        currY = GPS.Y
        currAlt = GPS.Altitude
        currCOG = GPS.Properties("COG") ' COG is Course Over Ground
        currSOG = GPS.Properties("SOG") ' SOG is Speed Over Ground (in km/h)
```

```
' read attributes for current waypoint from attribute table
    Set theLayer=Map.Layers(1)
    Set theRecords=theLayer.Records
    estg=theRecords.Fields("Easting").Value
    nrthg=theRecords.Fields("Northing").Value
    ptName=theRecords.Fields("Feat_name").Value
    If theRecords.Fields("Photo").Value=0 Then
        photoFlag=False
    Else
        photoFlag=True
    End If
    flyHt=theRecords.Fields("Fly_Ht").Value
    htUnits=theRecords.Fields("Ht_Units").Value
    flySpeed=theRecords.Fields("Airspeed").Value
    ptId=theRecords.Fields("ID").Value

    If Not Navigation.Active Then
        Navigation.GotoXY estg, nrthg, ptName
    End If
' get the bearing And distance To current waypoint
    brg2pt=Navigation.Bearing
    dist2pt=Navigation.Distance

' compute location of nearest point to target along current bearing (COG)
    Dim closestBrg, brgDiff, missAmt

' compute difference in bearing of current COG from bearing to target
' and convert the difference To radians
    brgDiff = bearingDiff(brg2pt, currCOG) * 0.0174532925

' if current bearing is less than 90 degrees off target, compute closest
' point data, otherwise set time to large number, indicating bearing error
    If brgDiff<90 Then
' compute miss amount due to difference between current COG and bearing to point
        missAmt = dist2pt * Sin(brgDiff)

' create a point at the intersection of the current COG line and the bearing line
' perpendicular to the current bearing line that passes through the waypoint
        Dim ptNewPoint1, ptNewPoint2, alpha, bd, hu
        Set ptNewPoint1 = Application.CreateAppObject("Point") ' current location
        Set ptNewPoint2 = Application.CreateAppObject("Point") ' closest point
        ' make a point at the waypoint location
        ptNewPoint1.X=estg
        ptNewPoint1.Y=nrthg

' compute bearing of line from waypoint to intersection with current bearing line
        closestBrg = currCOG + 270

' if adding 270 degrees puts closestBrg over 360 degrees, convert to less than 360
        If closestBrg >= 360 Then
            closestBrg = closestBrg - 360
        End If
```

```
' make the closest point
        If closestBrg = 0 Then
            ptNewPoint2.X = ptNewPoint1.X
            ptNewPoint2.Y = ptNewPoint1.Y + missAmt
        Elseif (closestBrg>0 And closestBrg<90) Then
            alpha=(90 - closestBrg) * 0.0174532925 ' convert to radians
            ptNewPoint2.X = ptNewPoint1.X + missAmt * Cos(alpha)
            ptNewPoint2.Y = ptNewPoint1.Y + missAmt * Sin(alpha)
        Elseif closestBrg = 90 Then
            ptNewPoint2.X = ptNewPoint1.X + missAmt
            ptNewPoint2.Y = ptNewPoint1.Y
        Elseif (closestBrg>90 And closestBrg<180) Then
            alpha=(180 - closestBrg) * 0.0174532925 ' convert to radians
            ptNewPoint2.X = ptNewPoint1.X + missAmt * Sin(alpha)
            ptNewPoint2.Y = ptNewPoint1.Y - missAmt * Cos(alpha)
        Elseif closestBrg = 180 Then
            ptNewPoint2.X = ptNewPoint1.X
            ptNewPoint2.Y = ptNewPoint1.Y - missAmt
        Elseif (closestBrg>180 And closestBrg<270) Then
            alpha=(270 - closestBrg) * 0.0174532925 ' convert to radians
            ptNewPoint2.X = ptNewPoint1.X - missAmt * Cos(alpha)
            ptNewPoint2.Y = ptNewPoint1.Y - missAmt * Sin(alpha)
        Elseif closestBrg = 270 Then
            ptNewPoint2.X = ptNewPoint1.X - missAmt
            ptNewPoint2.Y = ptNewPoint1.Y
        Else
            alpha=(360 - closestBrg)* 0.0174532925 ' convert to radians
            ptNewPoint2.X = ptNewPoint1.X - missAmt * Sin(alpha)
            ptNewPoint2.Y = ptNewPoint1.Y + missAmt * Cos(alpha)
        End If
    ' move ptNewPoint1 to current location
        ptNewPoint1.X=currX
        ptNewPoint1.Y=currY

    ' compute distance to nearest point along current bearing
        dist2pt = ptNewPoint1.DistanceTo(ptNewPoint2)

    ' compute time to target (1 km/h = 0.277778 m/s)
        If currSOG<>0 Then
            time2target=round(dist2pt/(currSOG*0.277778),1)
        Else
            time2target=888888
        End If
    Else
        time2target=999999
    End If

    ' convert current speed over ground (SOG) to MP/H from KM/H
    currSOG = Round( (currSOG * 0.62) ,0)

    If htUnits="FEET" Then
        currAlt = Round( (currAlt * 3.28) , 0)
```

```
            hu=" ft"
        Else
            currAlt = Round(currAlt, 0)
            hu=" m"
        End If

        bd=Round(bearingDiff(brg2pt, currCOG), 1)
        If bd > 90 Then
            missAmt=777777
        Else
            missAmt= Round(missAmt, 0)
        End If

    ' send data to the status bar
        Application.Statusbar.Text = ptName & "  FLY@: " & flyHt & hu _
            & "  BRG OFF: " & bd & " deg" & "  MISS+/-: " & missAmt _
            & " m" & "  T2T: " & time2target & " sec." & "  PHOTO: " & photoFlag

    ' if the time is less than three seconds and miss amount is less than 3*position error, take
photo
    ' by starting system timer, which will trigger the takePhoto script in the time2target

        If (time2target < 3 And (Not Application.Timer.Enabled) And (missAmt<60)) Then
            ' convert from seconds to milliseconds
            time2target = Round((time2target * 1000)-500,0)
            Application.Timer.Interval = Cint(time2target)
            Application.Timer.Enabled=True
        End If

End Sub
```

## Script to Trigger Camera, Store Coordinates at Moment of Exposure, Move on to Next Waypoint

```
Sub takePhoto
' this script turns off the system timer, sends the instruction to take the photo,
' stores the coordinates in the current bookmarked waypoint's record, moves the
' bookmark to the next waypoint (if one exists), and sends the new point to the
' navigation object

    Dim currX, currY, currAlt, currCOG, currSOG, theLayer, theRecords, theHandle
    Application.Timer.Enabled=False

' obtain gps coordinates, altitude, COG, and SOG
    currX = GPS.X
    currY = GPS.Y
    currAlt = GPS.Altitude
    currCOG = GPS.Properties("COG") ' COG is Course Over Ground
    currSOG = GPS.Properties("SOG") ' SOG is Speed Over Ground (in km/h)

    Set theLayer=Map.Layers(1)
    Set theRecords=theLayer.Records
```

```
    ' if this is a photo point and no valid photo taken here, take the photo
    If (theRecords.Fields("Photo")<>0 And theRecords.Fields("Visited").Value<>1) Then
        theHandle=System.FindWindow("#32770", "Nikon Capture Camera Control")
        System.PostMessage theHandle, 273, 1160
        ' show message to take a photo, store the coordinates, ht, COG, SOG
        Application.Statusbar.Text = "        !!! Take Photo !!!"
        theRecords.fields("CaptureX").Value=currX
        theRecords.fields("CaptureY").Value=currY
        If theRecords.Fields("Ht_Units").Value="FEET" Then
            currAlt = currAlt * 3.28
        End If
        theRecords.Fields("CaptureHt").Value=currAlt
        theRecords.Fields("CaptureCOG").Value=currCOG
        ' convert current speed over ground (SOG) to MP/H from KM/H
        currSOG = currSOG * 0.62
        theRecords.Fields("CaptureSOG").Value=currSOG
        theRecords.Fields("CaptureUTC").Value=GPS.Properties("UTC")
    End If

' mark the point as visited
    Dim yn
    theRecords.fields("Visited").Value=1
    If (theRecords.Fields("Path").Value=0 And _
            theRecords.Fields("Photo").Value=1) Then
        ' the photo is not in a path and is not a navigation point, verify OK
        yn = MsgBox("Photo OK?", vbYesNo, "Verify")
        ' a vbYesNo message box returns 6 for yes, 7 for no
        If yn=7 Then ' mark as missed/skipped
            theRecords.Fields("Visited").Value=2
        End If
    End If
' save the updates to the waypoint record attributes
    theRecords.Update
' now move bookmark to next waypoint if present, otherwise clear Navigation
    Dim theX, theY, pointName, boolFlag
    If theRecords.Fields("Visited").Value=2 Then
        yn=MsgBox("Move On To Next Point?", vbYesNo, "Skip Missed Point?")
    Else
        yn=6
    End If
    If yn=6 Then
        boolFlag=moveToNext(theRecords)
    End If
    ' if yn<>6, the current point will stay as the current navigation target,
    ' allowing another try to capture it
End Sub
```

## Script to Compute Absolute Bearing Difference Between Current Bearing (COG) and Bearing to Target

```
Function bearingDiff(brg2target, theCOG)
' this function returns the difference in bearing between the current course
' over ground (COG) and the bearing to the target.
   Dim diff
   diff = Abs(brg2target-theCOG)
   If diff > 180 Then
      bearingDiff = 360 - diff
   Else
      bearingDiff = diff
   End If
End Function
```

## Script to Move Current Point Bookmark to Next Waypoint Record Upon Skipping/Missing a Waypoint

```
Sub moveNextPt
' this function moves the bookmark to the next point in the waypoint layer
' if it exists after writing a code 2 (missed/skipped) to the
' current bookmarked record's visited field
   Dim theLayer, theRecords, theX, theY, pointName, boolFlag
   Set theLayer=Map.Layers(1)
   Set theRecords=theLayer.Records

   ' check to make sure that the layer is editable
   If Not theLayer.Editable Then
      theLayer.Editable=True
   End If
   ' set the current bookmarked record's Visited field to 2
   theRecords.Fields("Visited").Value=2
   theRecords.Update
   ' move to the next record in the list if it exists, otherwise
   ' terminate navigation and reset button controls, uses moveToNext function
   boolFlag=moveToNext(theRecords)
End Sub
```

## Script to Move Current Waypoint Bookmark to Next Waypoint Record

```
Function moveToNext(theRecords)
' this subroutine moves the waypoint layer record bookmark to the next non-visited
' record, skipping those that have been visited (Visited=1) or already skipped
' (Visited=2).  If no other records exist, the toolbuttons are reset and the
' Navigation is disabled, but GPS activity is maintained.
   Dim theX, theY, pointName, boolFlag
' move to next record that has not been visited or skipped
   Do While theRecords.Fields("visited").Value>0
      If Not theRecords.EOF Then ' there is at least one more record
         theRecords.MoveNext
```

```
            theX=theRecords.Fields("Easting").Value
            theY=theRecords.Fields("Northing").Value
            pointName=theRecords.Fields("Feat_name").Value
            Navigation.GotoXY theX, theY, pointName
            boolFlag=True
        Else ' the end of the waypoint list has been reached
            Navigation.Clear
            Application.Toolbars("Flight").Item("NavigateCapture").Enabled=True
            Application.Toolbars("Flight").Item("StopNavigation").Enabled=False
            Application.Toolbars("Flight").Item("NextPoint").Enabled=False
            Application.Toolbars("Flight").Item("ResetFlags").Enabled=True
            boolFlag=False
            Application.UserProperties("FlightInProgress")="NotFlying"
            Exit Do
        End If
    Loop
    moveToNext=boolFlag
End Function
```

## Script for Resetting All Visited Flags in the Waypoint Layer

```
Sub resetFlags
    Dim theLayer, theRecords
    Set theLayer=Map.Layers(1)
    Set theRecords=theLayer.Records
' make sure the layer is editable
    If Not theLayer.Editable Then
        theLayer.Editable=True
    End If
' cycle through all points and reset the Visited field to zero
    theRecords.MoveFirst
    Do While Not theRecords.EOF
        theRecords.Fields("Visited").Value=0
        If Not theRecords.EOF Then
            theRecords.MoveNext
        End If
    Loop
' commit the changes
    theRecords.Update
End Sub
```

# BIOGRAPHY OF THE AUTHOR

Kenton Williams was born in Ogden, Utah on October 29, 1970. He grew up living in the town of Layton, Utah, where he graduated from Layton High School in 1988. In 1996, he received his Associate of Science degree from Weber State University in general studies with an emphasis on photography.

He worked as a professional photographer from 1993 until the year 2000, when he relocated to the State of Maine with his wife and daughter to pursue additional education.

In May, 2004, he received his Bachelor of Science Degree from The University of Maine College of Engineering, Department of Spatial Information Science and Engineering, summa cum laude.

In September 2003, he began work in the Maine Image Analysis Lab in the Department of Forest Management while completing his undergraduate studies, where he continued onward to complete graduate school studies.

He is a candidate for the Master of Science Degree in Forestry from The University of Maine in December, 2005.