PRECISION RELATIVE POSITIONING FOR AUTOMATED AERIAL REFUELING FROM A STEREO IMAGING SYSTEM

THESIS

Kyle P. Werner, 2Lt, USAF

AFIT-ENG-MS-15-M-048

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-15-M-048

PRECISION RELATIVE POSITIONING FOR AUTOMATED AERIAL REFUELING FROM A STEREO IMAGING SYSTEM

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Engineering

Kyle P. Werner, B.S.C.E.

2Lt, USAF

March 2015

AFIT-ENG-MS-15-M-048

PRECISION RELATIVE POSITIONING FOR AUTOMATED AERIAL REFUELING FROM A STEREO IMAGING SYSTEM

THESIS

Kyle P. Werner, B.S.C.E.
2Lt, USAF

Committee Membership:

Maj Brian G. Woolley, PhD
Chair

Dr. John F. Raquet
Member

Maj John M. Pecarina, PhD
Member

Abstract

The United States Air Force relies upon an aerial refueling capability to fulfill its missions of rapid global mobility and global attack. The growing wings of unmanned aerial systems (UAS) and remotely piloted aircraft (RPA) do not currently have access to this capability due to the lack of an on-board pilot to safely maintain an appropriate refueling position and orientation. Future refueling aircraft are likely to employ stereo vision systems to enhance the capability for refueling manned aircraft. This research examines the use of stereo vision for precision relative navigation in order to accomplish the Automated Aerial Refueling (AAR) task. Previous work toward an AAR solution has involved the use of Differential Global Positioning (DGPS), Light Detection and Ranging (LiDAR), and monocular vision. This research aims to leverage organic systems in future aircraft to compliment these solutions. The algorithm used in this thesis generates a point cloud from the disparity between the stereo camera images. The algorithm then fits the point cloud to a digital model using a variant of *iterative closest points* (ICP). The algorithm was tested using simulated imagery of an F-15E rendered in a 3D modeling environment. Experimental results showed a significant increase in accuracy as the receiver aircraft approached the tanker aircraft, reporting accuracies within +/-10 centimeters at distances less than 17 meters. The algorithm's ability to transition to the real world was validated qualitatively using a 1:7 scale F-15E model and 1:7 stereo camera pair.

AFIT-ENG-MS-15-M-048

To my mom, brother(s), advisor, lab partner, and everyone else who helped me through the ups and downs of putting this work together. Thank you for being my (incredibly patient) sounding board.

I'd also like to thank the development and maintenance teams for Notepad++, GIMP, MeshLab, OpenCV, Python, and all of the open source software that this thesis relied upon.

Table of Contents

Page

List of Figures

List of Tables

PRECISION RELATIVE POSITIONING FOR AUTOMATED AERIAL REFUELING FROM A STEREO IMAGING SYSTEM

I. Introduction

The focus of this thesis is the feasibility and accuracy of stereo vision for precision relative navigation, specifically with regard to the automated aerial refueling (AAR) task. Such relative positioning would allow for safe, accurate control of unmanned aerial systems (UAS) and remotely piloted aircraft (RPA) during air-to-air refueling operations, a capability not currently available to the Air Force. Aerial refueling is key to the Air Force's global mobility mission, with AAR expanding that capability to the growing wings of unmanned and remotely piloted aircraft.

Previous efforts at AAR have been based on Differential Global Positioning Systems (DGPS) [26], monocular vision [13, 34], and LiDAR scans from on-board the receiver aircraft [20] The key difference of this effort is that the sensor suite is assumed to be a stereo vision system installed on the refueling aircraft alone. This assumption releases the receiver from carrying any specialized equipment in order to complete the task.

A stereo vision-based AAR solution provides relative navigation capability in situations where previous AAR efforts break down. Such situations include instances where the tanker obscures GPS signals for the receiver and instances where the receiver has not been specifically modified for AAR. While a stereo vision solution can supplement a DGPS solution, this work demonstrates that the latter is not required. Additionally, and perhaps most importantly, a stereo vision solution can operate with the goal of requiring no specialized modification to the receiver aircraft. The system

on board the tanker can calculate the position of the trailing aircraft with respect to where it is expected to be. In this way, a small amount of information can be passed to the receiver, which can then respond with its standard autopilot.

1.1 Problem Statement

This thesis leverages existing technologies in the fields of computer vision and navigation to generate precise relative positioning of the receiving aircraft. Stereo cameras provide the system's sensors, from which depth information is extracted and fit to a computer model of the receiver. The tanker and receiver are assumed to be in a reasonable refueling position and orientation throughout the system's operation. The system reports a simple difference vector that represents the position and orientation of the trailing aircraft with respect to a station keeping point at a fixed location and orientation behind and below the tanker (in range of the refueling boom). This vector simply describes six degrees of freedom (x-axis offset, y-axis offset, z-axis offset, pitch angle, roll angle, yaw angle), which constitute the aircraft pose. In addition to implementation, a key aspect of this thesis then analyzes the experimental results in terms of accuracy, reliability, limitations, and feasibility for full-scale/real world implementation.

The system combines established techniques from computer vision and applies them in a navigation domain to achieve automated pose estimation.

1.2 Overview

This thesis is organized into five chapters: Introduction, Background/Previous work, Methodology, Results/Discussion, and Conclusions. The first chapter is an introduction to the thesis domain, including the problem statement and a thesis overview. Chapter II presents key concepts and algorithms in computer vision and

3D modeling, as well as an overview of previous work in precision relative navigation. Chapter III details the assumptions and limitations of the system and introduces the experimental domain. The overall algorithm workflow is introduced along with specific customizations made to established algorithms. Analysis of the experimental results is presented, followed by a discussion on the impact of those results. This thesis concludes in Chapter V with assessments, suggestions for future work, and final remarks.

II. Background/Previous Work

This chapter describes the established principles and techniques underpinning the relative navigation presented here. Some of the key concepts include fundamentals of stereo vision and 3D modeling. A discussion of coordinate frames and transforms is provided as it pertains to this thesis. This chapter concludes with a discussion of previous work with relative navigation and its applications to the AAR task.

2.1 Stereo Vision

Mankind, along with much of the animal kingdom, has the ability to perceive depth. Humans have a stereo (i.e., binocular) vision system to obtain visual information of their surrounding world. The left and right each provide an image to the brain from a slightly shifted perspective. The differences in these two images provides the brain with information on the distance of objects, which we naturally interpret as depth.

To understand how the slight perspective difference between the left and right eye conveys depth information, try a simple experiment. Holding your index finger at arm's length in front of your nose, focus on a point on an object in the distance. Now, close your right eye. Keeping your focus on the distant point, simultaneously open your right eye and close your left eye. Notice how the distant object appears to move very little (if at all), while your finger appears to jump to the left. The magnitude of this jump is known as the disparity for the object (in this case your finger) and helps describe the distance of the object from your eyes.

The same principles apply to computer vision. However, while the human brain can expertly determine depth information, computer systems are less proficient.

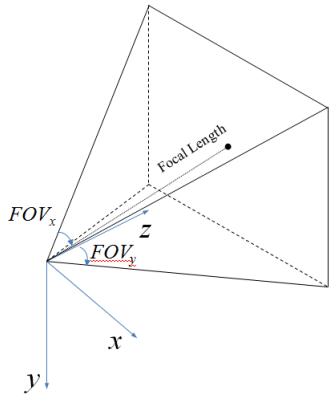


Figure 1. A simple camera model. This camera model illustrates focal length, principle point, and field of view and is assumed for all cameras in this thesis.

2.1.1 Camera Model.

The basic structure of a camera consists of a sensor plane, a lens, and a focal length. This basic arrangement (and accompanying assumptions) is known as the pinhole camera model [35] and serves as the model for all cameras in this thesis.

In the human eye, cones and rods serve as sensors arranged on the retina. Similarly, a digital camera consists of multiple photo sensors also arranged on a plane. In either case, the field of view, or area over which the sensor plane collects data, is determined by a lens. A wider field of view captures more light along the x-axis, and a taller field of view captures more light along the y-axis. The principle point of the camera's field of view is determined by the lens magnification and will be assumed to be the image center for this thesis. The lens also determines the camera's focal length, which measures the distance from the center of the lens to the convergence point of light passing through the lens (Figure 1).

The camera's intrinsic parameters describe the projection from 3D space onto the camera's sensor. The parameters are commonly expressed as a matrix consisting of principle point (along the x- and y-axes) and focal length. The three-dimensional matrix that describes a camera's perspective space is:

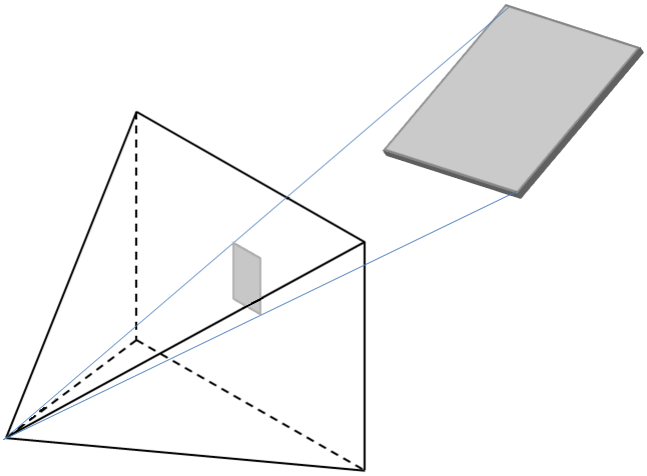


Figure 2. Images do not contain depth information. The position along the z-axis of any point is lost when a 3D scene is captured in a single 2D image.

$$C = \begin{bmatrix} f & s & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where f is the focal length of the camera, s is the skew, and p is the principle point. With the assumption that the principle point is at the image center (valid for an infinitely thin lens assumption), p_x and p_y each become zero.

The resulting image is a projection of spectral information from the three-dimensional (3D) world onto the two-dimensional (2D) sensor plane. It is important to note that depth information (the third dimension) is lost in the projection from 3D to 2D space. A given point in an image may exist at any distance along the projected line from the sensor's center, through the given point, to infinity. Consequently, the images provide no information along the z-axis. Such a projection from 3D-to-2D is illustrated by Figure 2.

2.1.2 Epipolar Constraint.

The line in Figure 2 from the origin through the given point is known as an epipolar line. The vast majority of problems involving positional information from imagery rely upon epipolar geometry. Epipolar geometry describes the relationship between pixels in an image and the camera used to capture the image. Once established, this relationship can be exploited to determine location information. Within an image, each pixel corresponds to an epipolar line, originating at the origin of the camera frame and extending through the given point to infinity. The known origin of the camera frame in addition to the known location of a point within the camera frame allows a precise angle to be defined. The point in 3D space that corresponds to the 2D point exists at some point along the epipolar line.

An image is constructed of points. In a digital image, each pixel represents a single point. Each pixel in the image corresponds to exactly one epipolar line. These lines converge at the sensor's center and diverge approaching infinity. The result is a set of lines that intersect only at the origin, and become increasingly dispersed as distance from the origin increases. Without depth information, the real world distance between two pixels in an image cannot be determined from the positioning of the pixels themselves. This limitation constrains a single image to providing only relative positional (not distance) information along the x- and y-axes. Figure 3 shows a pair of epipolar lines corresponding to points at two differing distances in 3D space. Note that the distance between the pixels in the image (i_1 andi_2) does not equal the true distance between the points (r_1 and r_2).

To regain depth information (and thereby regain distance information along all three axes), multiple images are needed of a single scene. The images can be obtained from a single camera moving in a known manner between images of a static scene

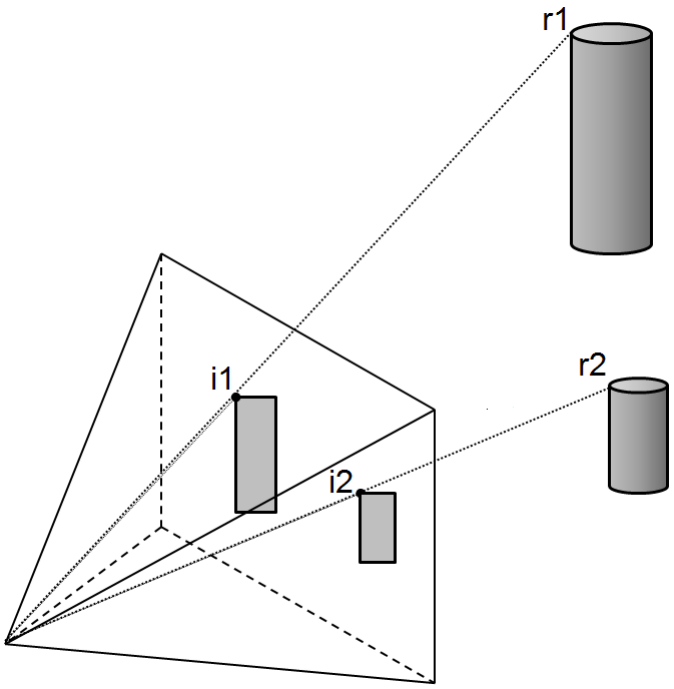


Figure 3. Epipolar lines extending from the camera's origin, through a given point.

(i.e., structure from motion [25]) or from multiple cameras taking images simultaneously. This thesis focuses on the case where pairs of images are taken simultaneously.

2.1.3 Simple Stereo Camera Model.

A simple stereo camera consists of two cameras in a known relative arrangement. The arrangement of the cameras in 3D space defines the pair's extrinsic parameters. Because each camera captures the scene from a unique position, it becomes possible to compute where corresponding epipolar lines intersect in space (i.e., triangulation), thus resolving depth ambiguity present in a single 2D image projection (Figure 4). As a result, each epipolar line from camera A intersects each epipolar line from Camera B at most once (and in many cases, never).

The origins of the camera pair form a triangle with the intersection of a pair of epipolar lines. When combined with knowledge of the pair's extrinsic parameters, this triangular relationship can be resolved to the location of the intersection in 3D space.

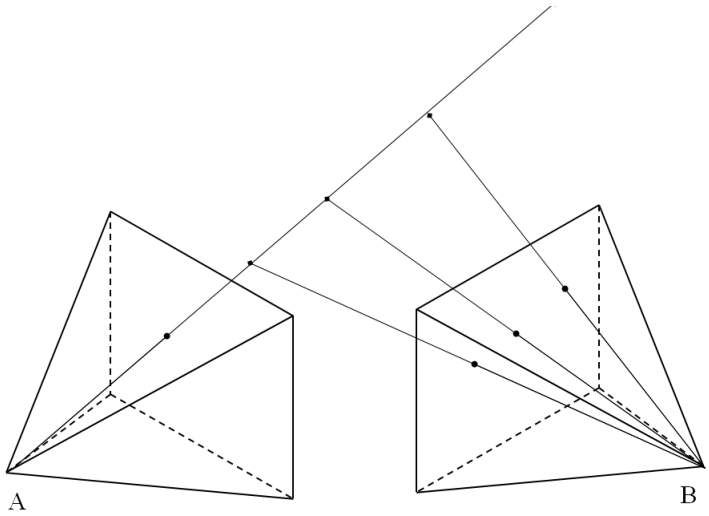


Figure 4. Any two epipolar lines from a stereo camera pair intersect at most once. The highlighted epipolar line from A intersects each highlighted epipolar line from B at in at most one location.

The difficulty lies in determining which epipolar line in camera A is intersecting an epipolar line in camera B (if at all). The one-to-one correspondence between pixels and epipolar lines reduces this difficulty by relating each epipolar line to a single pixel. If a pixel in an image from camera A captures the same point in 3D space as a pixel in an image from camera B, the corresponding epipolar lines must intersect at that point.

2.1.4 Rectification.

Searching an entire image for corresponding pixels in order to find intersecting epipolar lines is computationally expensive. Algorithms exist to reduce the search space by utilizing multiple images, as with optical flow [35]. One of the most efficient methods for simplifying the epipolar intersection calculations is rectification [27]. As described by Loop and Zhang [23], rectification bends (or warps) an image pair to align epipolar lines horizontally with vertical correspondence.

Rectification occurs in four main stages: First, epipolar lines for identified features are computed (based on extrinsic camera measurements); next, imagery is trans-

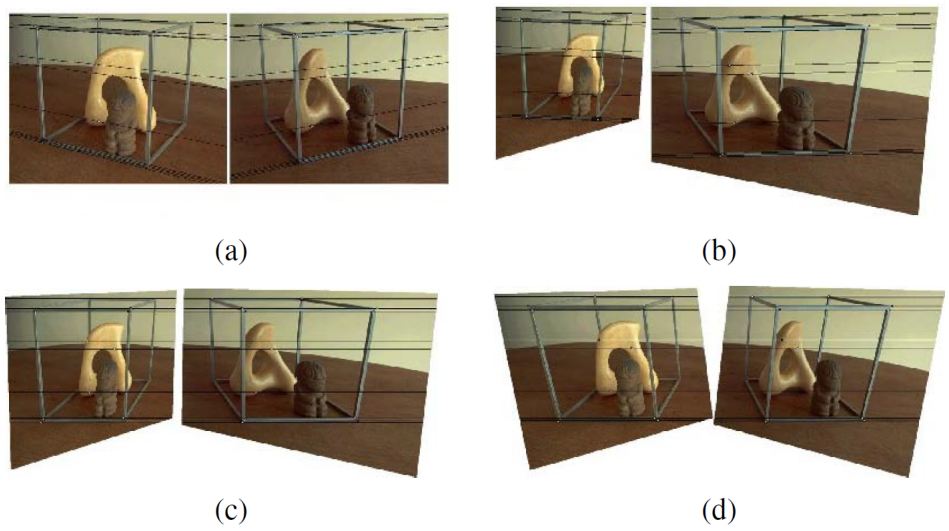


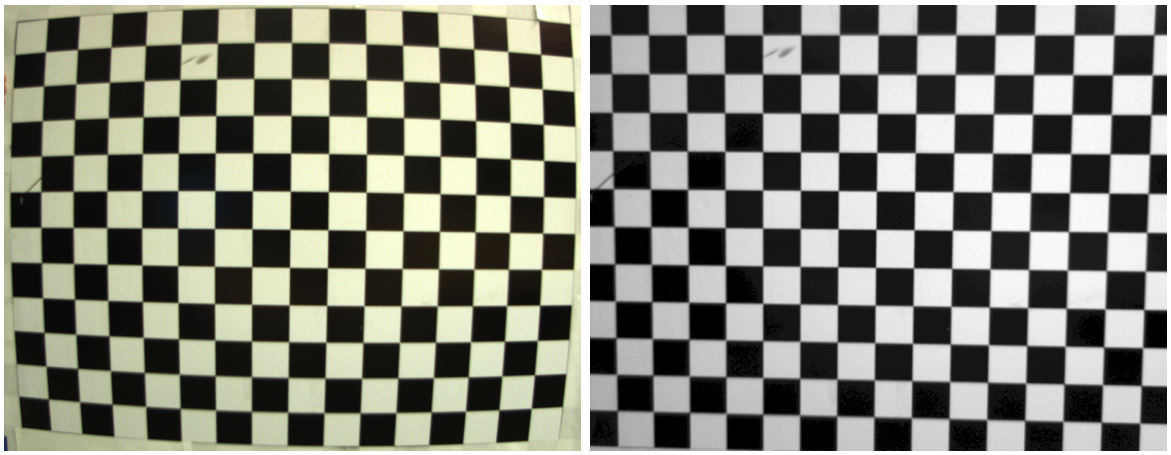
Figure 5. Rectification as performed by Loop and Zhang (1999) [23]. (a) Image pair with epipolar lines shown; (b) Epipolar lines made parallel; (c) Epipolar lines in vertical correspondence; (d) Fully rectified images with horizontal distortion minimized

formed such that the epipolar lines are parallel across the image pair (Figure 5b) and then scaled to vertically align the corresponding epipolar lines (Figure 5c); finally, horizontal distortion between the pair is minimized to give a fully rectified image (Figure 5d).

Stereo Camera Calibration.

In most cases, the cameras' extrinsic parameters are either unknown or cannot be measure explicitly. Alternatively, a stereo camera system's extrinsic (and intrinsic) parameters can be computed via a calibration routine.

There are two sources of distortion commonly addressed by calibration: radial and tangential. Radial distortion results in the "fish-eye" look of an image, where the center of an image appears to have greater magnification than the surrounding edges (Figure 6). Tangential distortion occurs as a result of slight inaccuracy when aligning the camera lens with the image plane. Both forms of distortion are accounted for in the Brown-Conrady model [15].



(a) Raw (distorted) Image (b) Undistorted Image

Figure 6. Distortion causes the center checkerboard squares to appear larger than the outer squares. In reality, the checkerboard is square. Radial distortion gives the "fish-eye" or "barrel" appearance to (a). Undistortion (as part of rectification) corrects for this appearance, as seen in (b).

The Brown-Conrady model assigns five coefficients to describing distortion of a given pixel, p. The coefficients k_1, k_2, and k_3 correct for radial distortion (Equation 2), while p_1 and p_2 correct for tangential distortion (Equation 3). To calculate the coefficients, a simple geometric relationship must be established between the observed pixel coordinate and the ideal pixel coordinate.

$$\begin{aligned} p_{x-corrected} &= p_x * (1 + k_1 r_p ow2 + k_2 r_p ow4 + k_3 r_p ow6) \\ p_{y-corrected} &= p_y * (1 + k_1 r_p ow2 + k_2 r_p ow4 + k_3 r_p ow6) \end{aligned} \tag{2}$$

$$\begin{aligned} p_{x-corrected} &= x + (2p_1 xy + p_2(r_p ow2 + 2x_p ow2)) \\ p_{y-corrected} &= y + (p_1(r_p ow2 + 2y_p ow2) + 2p_2 xy) \end{aligned} \tag{3}$$

A checkerboard provides a set of easily identifiable, precise points in a strict geometric relationship. Any given checkerboard contains c squares arranged into a

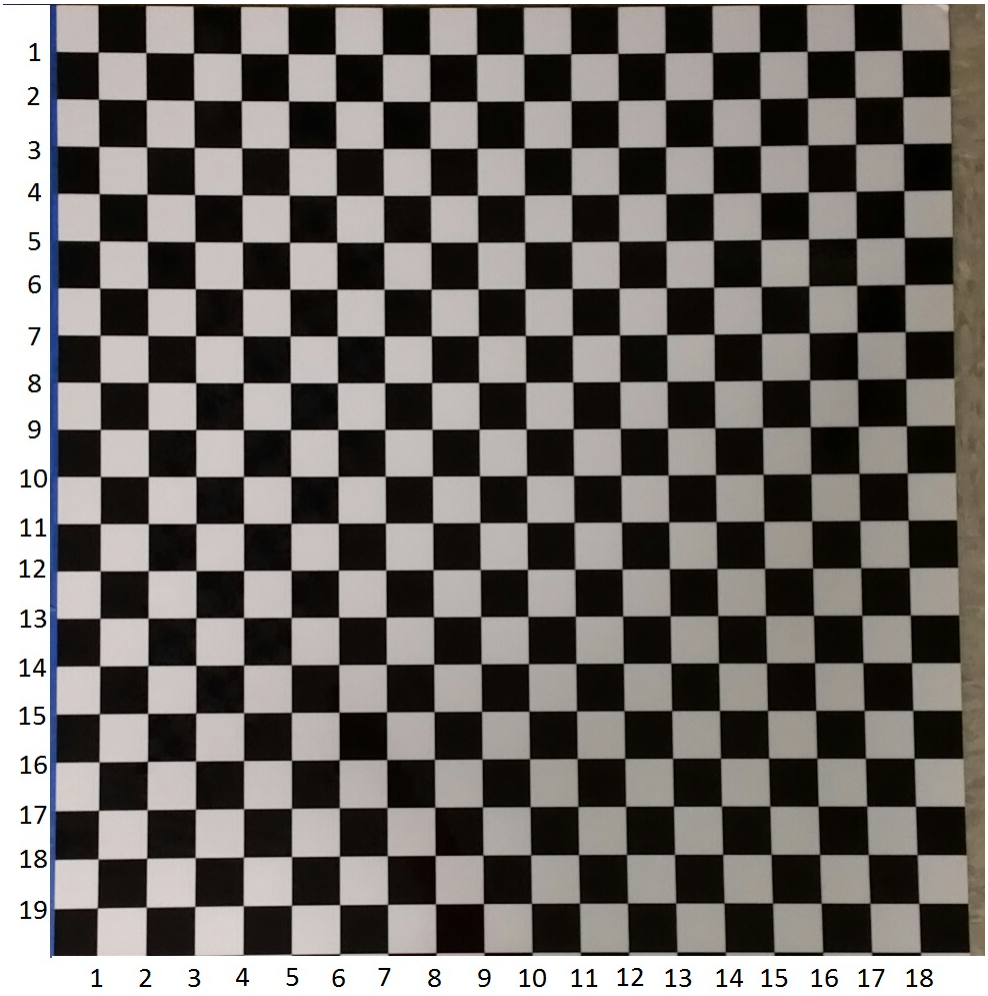


Figure 7. A camera calibration checkerboard (note that the number labels are for reference only and do not contribute to the algorithm). This checkerboard shows 19 rows and 18 columns of inner checkerboard corners. Each square measures precisely 1.5 inches per side. These three known values feed the calibration algorithm.

rows and b columns. This arrangement provides d internal row corners and e internal column corners (Figure 7). The calibration routine identifies these internal corners and is fed the known size of each checkerboard square (measured in side length). This precise, known relationship between points gives the ideal pixel coordinates for an image of the checkerboard. Comparing the ideal coordinates to the observed coordinates in an image defines the geometric relationship to be undistorted by the five coefficients.

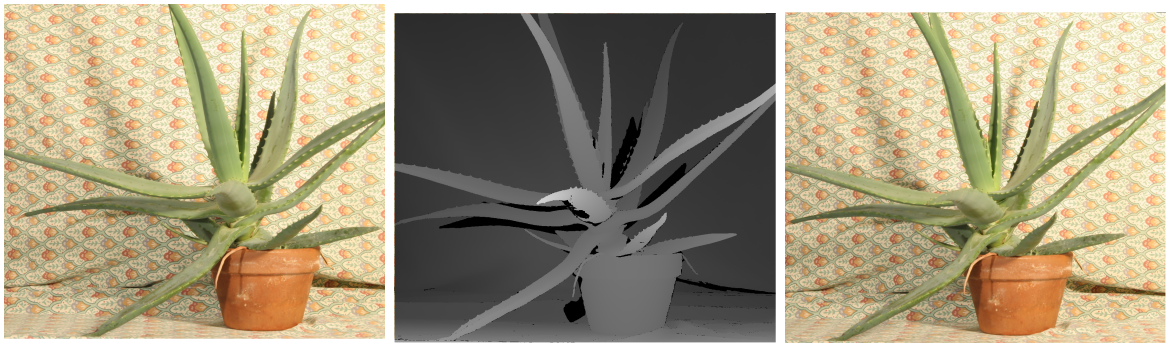
(a) Single Image from Stereo Pair (b) Corresponding Disparity Map

Figure 8. A sample image (a) and its corresponding disparity map (b) [5].

2.1.5 Disparity Maps.

Once rectification is accomplished, epipolar geometry provides a depth value for a matched pixel across two images. Disparity maps, also referred to as depth maps, give a visual representation of depth information within a scene. The color value of each pixel represents the disparity between that pixel and its match in the corresponding image. Through a basic trigonometric relationship (and a known baseline separation between the cameras), disparity gives distance from the camera to the point that the pixel represents in the world space. Grayscale was used for this project, with black representing zero/unknown disparity (infinite/unknown depth) and lighter shades representing increased disparity (decreased depth) (Figure 8).

The precision of a disparity map is determined by a combination of maximum disparity and depth bins. Maximum disparity is the maximum number of pixels between two corresponding points that the matching algorithm will consider a match. The distance is calculated as a simple Manhattan distance from the pixel coordinates of the point in image A and the pixel coordinates in the point fo image B. A smaller maximum disparity results in fewer mismatches, but fewer matches overall. This



(a) Left Image (b) Disparity Map (c) Right Image

Figure 9. A stereo image pair and the resulting disparity map (from the perspective of the right camera) [5]

gives a depth value to points that shift less between the left and right images, while ignoring the rest. A larger maximum disparity allows for more granularity in depth, but will begin to register false matches more frequently as the maximum disparity increases. The algorithm then assigns each match to a disparity bin. Disparity bins range from zero (at the focal length of the camera) to infinity and are measured from the farthest point in the bin. Disparity bins are typically found in multiples of sixteen. Increasing the number of depth bins can result in more accurate depth information, at the expense of both processing time and unused bins.

Disparity map reconstruction algorithms use one image as the base for the depth calculation. Figure 9 is based on the right camera, which serves as the base camera for this application. The points in the base (right) image are compared to the points in the second (left) image when searching for matches. This searching technique leaves an unmatched pixel at each point in the base image where the second image contained no corresponding point. The disparity map measures from the origin of the base camera and can therefore be thought of as an image from the perspective of the base camera.

All disparity map approaches result in areas of no information that may appear to be shadows. These are actually areas of occlusion between the original images.

In the case of Figure 9, the disparity map is generated from the right camera frame. There are points on the right image which cannot be matched to the left image due to change in viewing angle. These points are assigned the maximum disparity value ("infinite", or black in the case of this example).

Nearly all algorithms for identifying corresponding pixels for dense disparity map reconstruction depend upon identifying intensity values within an image [33]. The methods used for identifying and defining intensity values can be grouped into three main categories: Feature-based, area-based, and energy-based [10]. Feature-based approaches identify features based on intensity changes within each image and searches these features for their appearance in multiple images. Edges, lines, corners, and known structures (such as eyes, in the case of facial recognition) are commonly used as features. Area-based approaches compare images through a set of image windows (sized based upon source image and implementation). Energy-based approaches determine correlation iteratively through energy minimization.

Feature-based approaches offer the advantage of reduced comparisons between images, but precision can be limited in cases with few features (such as smooth surfaces), which results in a sparsely distributed set of correlations [37] or occluded features [31]. Area-based approaches function best in highly textured images, but can be limited by areas of high contour (which result in differing windows from differing perspectives) [18, 31]. Energy-based approaches do not require rectification, but can be computationally intensive and are prone to finding local minima (hill-climbing) [31]. This thesis will record points of no known disparity using the maximum possible disparity for that point, with the ability to average nearby disparities in order to smooth the result.

The objective of this thesis is to identify an aircraft-shaped object. The aerodynamic shape of aircraft presents many strong features, as well as areas of high

texture along the aircraft's surface. Initially, a mixed approach was used to exploit both feature- (in the form of SIFT features [24]) and area-based approaches. SIFT features were used to quickly build a sparse disparity map (regional depth information), reinforced by a rolling window to create a dense disparity map. Model-building was eliminated from this thesis in favor of a preexisting aircraft model, thus making the SIFT features less useful. The dense disparity images in this thesis were created using block matching. This allowed for simplification of the disparity map creation algorithm.

2.2 3D Modeling

3D modeling describes the process or method of describing a three-dimensional object in a digital format. 3D modeling approaches utilize various mathematical techniques, which result in models with differing properties.

2.2.1 Solid Models.

Solid models are the most complete physical representations. Solid models describe all surfaces of the object being represented, including internal faces and surface thickness, through verticies connected by edge lines. This results in a model that accurately captures the structure of an object, at the expense of both model storage size and generation complexity.

2.2.2 Shell Models.

Shell models describe only the outermost boundary of the represented object. These models can give highly accurate information on the external face(s) of an object, but do not represent the internal structure of the object. Often called "hollow" models, these are the most common type of 3D model because they are simple to

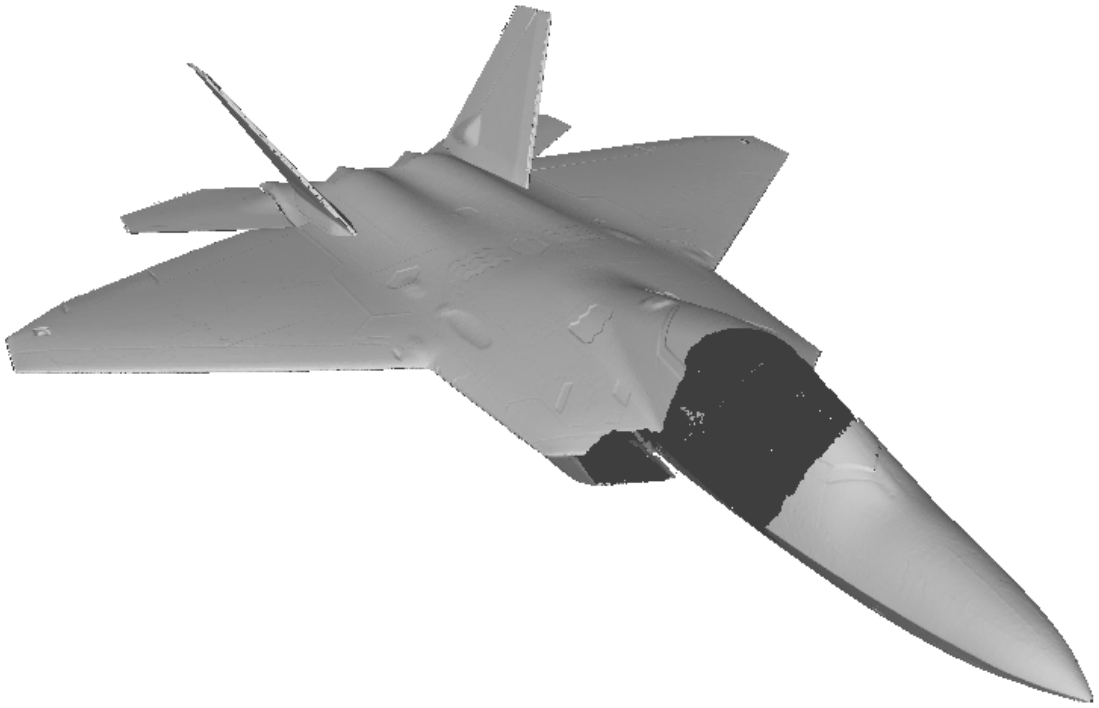


Figure 10. A shell model. The cockpit area has been removed to show that the model represents only the outer surface of the aircraft.

produce. A shell model can be produced by a sensing arm, laser scanner, digital imagery, SONAR, RADAR, or other external sensing device. Most common shell model formats share the vertex-edge structure, including all formats used within this thesis (e.g. .stl, .obj, .mesh).

2.2.3 Point Clouds.

The term "point cloud" can be used to describe any model consisting of a set of points within a given coordinate system (whether or not the points are connected by edges). For this thesis, each point consists of three values, corresponding to the three-dimensional coordinate system (x,y, and z). The density and precision of the points determines the accuracy of the model

Dense Point Clouds.

Dense point clouds are stored as a set of entries, with each entry denoting a three-dimensional position as well as any other information pertaining to the point (such as color, number of facet connections, etc). "Dense" describes the lack of void entries stored in the model.

Sparse Point Clouds.

Sparse point clouds store void entries as part of the model itself. The data structure is larger than the model being represented so that points not needed to describe the model are given no value, while the points describing the model are given a value. Typically, the index of the point within the data structure denotes the 3D location of the point, while the value contains color or other information. This structure is intuitive to work with, but results in much larger models than dense point clouds, due to the addition of many empty entries. Frequently, it becomes desirable to align two point clouds. One common approach is described next.

2.2.4 Model Fitting - ICP.

In this work, a point cloud is created from stereo imagery and then aligned to a preexisting model to determine the receiver pose. ICP works iteratively to converge upon the closest local minimum mean squared distance between two point clouds [11]. To accomplish the fitting of one point cloud, p, to a point cloud, x, the algorithm attempts to minimize the result, f, of

$$f = \mu_p - \mu_x, \tag{4}$$

where μ_p and μ_x are the mean of point clouds p and x, respectively, and are defined as

$$\mu_p = \frac{1}{N_p}\sum_{N_p}^{i=1} p_i \textbf{ and } mu_x = \frac{1}{N_x}\sum_{N_x}^{i=1} x_i. \tag{5}$$

To minimize equation 4, ICP iteratively applies equation 6, where q and t are the quaternion rotation (as a DCM) and translation applied to point cloud p in order to best fit each of the N_p points in cloud x. A significant limit to this approach is that, p and x must contain an equal number of elements.

$$f(q,t) = \frac{1}{N_p}\sum_{N_p}^{i=1}(\|\vec{x_i} - R(q) * \vec{p_i} - t\|)^2 \tag{6}$$

By applying a series of rotations and translations to the point cloud, ICP minimizes the distance from each point in the cloud to the closest point in the model. The model used in this thesis is formed of a mesh of vertices. While this model structure simplifies the model for storage, it does not provide an exact point for each point in the cloud to match to (the size of x does not necessarily equal the size of p). To overcome this limitation, ICP found the closest point within each nearby triangle of the model's mesh [20]. The closest match became the starting point for the next iteration, with the process repeating until the closest triangle did not change or a looping threshold was met (to limit algorithm runtime).

2.3 Coordinate Frames

This thesis aims to describe the location of one object relative to another (i.e., a receiver relative to a tanker). A unique coordinate frame is associated with each observed component of the system [1]. These frames contain unique origin points and axis positions that may or may not align with those of any other frame at a given instance. For each frame, pitch, roll, and yaw are defined relative to the frame's three major axes. Each angular position degree begins at zero and increases to 360

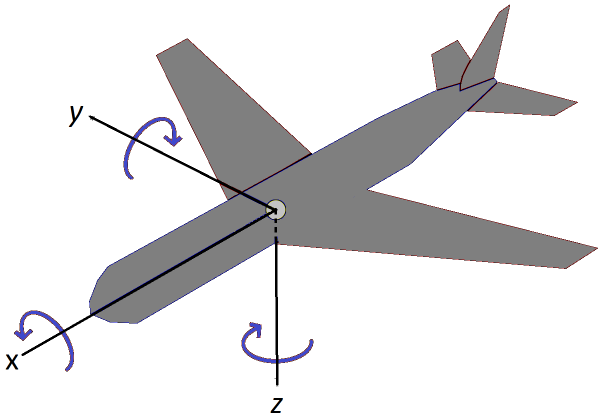


Figure 11. The tanker's body reference frame. A derivative of Yaw_Axis.svg by Auawise, used under Creative Commons Attribution-Share Alike 3.0 Unreported

degrees (2π radians) following the right-hand rule. The relationship between axes and angular position is as follows: (1) Yaw, a rotation about the z-axis. (2) Pitch, a rotation about the y-axis. (3) Roll, a rotation about the x-axis.

All frames used in this thesis follow this standard alignment.

2.3.1 Body Frame.

The body frame describes position from the perspective of a single aircraft. This frame's origin is located at the aircraft's navigational center (reported navigational position), with the positive x, y, and z axes extending in the direction of the aircraft's nose, the aircraft's right wing, and down, respectively. The tanker and receiver comprise the two unique body frames utilized in this thesis. Figure 11 shows the tanker reference frame and tanker aircraft, while Figure 12 shows the same for the receiver aircraft.

2.3.2 Camera Frame.

The physical data collection system is made up of the left and right camera frames. The axes of the two camera frames run parallel. The positive x and y axes

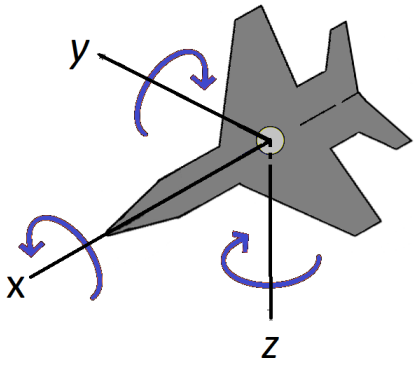


Figure 12. The receiver's body reference frame.

extend to the camera's right and down, respectively. The positive z-axis extends along the center of the camera's field of view. Each camera frame is centered on the sensor surface within the camera. Figure 13 shows the left and right camera frames, as well as their relative positioning within the system.

2.3.3 World Frame.

The above frames use differing origins and axes to describe the same physical space. The world frame ties these frames to a single reference point. In a flying system, this frame would most likely correspond to the WGS [6].

2.3.4 Transforms.

The relative alignment of the above frames is central to describing a precise relative position. With the exception of the camera frames and tanker frame, each frame is free to move independent of the other frames at any time. The camera and tanker frames will remain aligned at all times, with constant relative origins, but may move together as the system moves. In order to achieve the goal of describing the receiver's position in the tanker frame, coordinate transforms are used. For the six-dimensional space used in this thesis, the coordinate transformation matrix is a

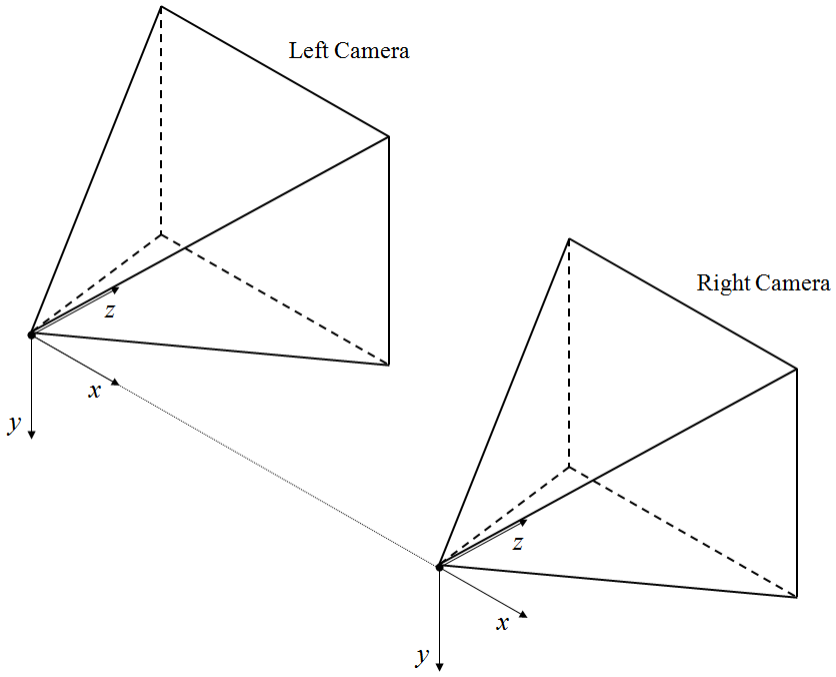


Figure 13. The left and right camera frames, as arranged in a stereo image capture system.

square matrix os six. Coordinate transforms are denoted by C_A^B, where C is the transformation matrix used to convert from A to B (Equation 7).

$$C_R^C \cdot C_C^T = C_R^T \tag{7}$$

Where R, C, and T refer to the receiver frame, camera frame, and tanker frame, respectively.

2.4 Relative Navigation

In the broadest sense, relative navigation is the positioning of an object based upon its placement with respect to an established point. Relative navigation does not rely on a global system or knowledge, which is useful in domains where global systems are not practical, as well as in a reinforcement role to global navigation systems. Relative navigation solutions have been implemented to solve problems

through inertial systems [16], differential global positioning systems [29], monocular vision [34], light detection and ranging systems [20], and other systems.

This thesis expands upon relative navigation solutions applied to close-formation aerial refueling applications. Previously, solutions have focused on three areas: differential GPS, monocular vision, LiDAR, and stereo vision. These areas are addressed in detail below.

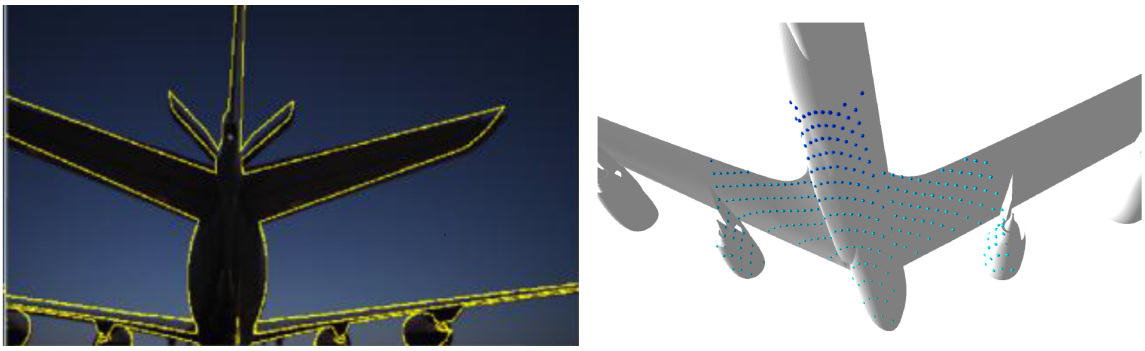
2.4.1 Differential GPS.

Differential GPS (DGPS) blends GPS into a relative navigation solution through the use of "reference receiver" nodes. Wide area DGPS utilizes a set of nodes, while local area DGPS employs a single node [29]. For the purposes of inter-aircraft positioning, local area DGPS solutions are the most researched. In these solutions, a node receives GPS range information from each satellite in view and estimates a correction for the current error. Relative positional information is then passed from the node to a DGPS receiver. Accuracy can be further increased by measuring the phase of the GPS signal at the DGPS receiver in comparison to the phase of the GPS signal at the node (DGPS). Commercial DGPS solutions offer accuracy to 10-15 centimeters [4].

While DGPS provides enhanced accuracy over a single node GPS solution, the system is still limited by the ability of both nodes to receive reliable ranging signals from multiple GPS satellites. DGPS systems also encounter error due to GPS satellite set differences between the receiver and the node, GPS satellite set change during operation, and data loss at system range limits [26].

2.4.2 Monocular Vision.

Monocular vision solutions utilize a single electro-optical (EO) camera to address relative navigation have also been employed to enable relative navigation. Such



(a) Monocular Vision [34] (b) LiDAR Scanning [20]

Figure 14. Previous relative navigation efforts. Monocular vision and LiDAR have been used for relative navigation between a two aircraft.

solutions have been applied to many domains, including terrestrial [21], space [8], and aerial systems. Specifically application of monocular vision to the AAR domain has been demonstrated in both simulation [36] and flight testing [34]. Template matching provides the core of monocular solutions to relative navigation. Although the algorithms vary, monocular solutions employ a preexisting model which is fit to imagery. The translation and rotation required to achieve the best fit provide the relative position of the object being observed (tanker) with respect to the observer (receiver).

Although this thesis shifts the perspective of the problem from the receiver to the tanker, the principles remain the same: localize an object within a scene and employ model fitting to extract positional information. Previous work in applying monocular vision to the AAR domain has resulted in strong relative navigation performance, with accuracy on the order of 35 centimeters at a range of 20 meters [34]. Other efforts have shown accuracy of nearly 3 centimeters under ideal conditions [13]. The main limitations of a monocular vision implementation are in the forward direction, where limited visibility reduces the amount of information available, and weather interference. Sun interference also had a negative effect on monocular vision solutions, due to the system orientation from the receiver looking up at the tanker.

2.4.3 LiDAR.

Light detection and ranging (LiDAR) has also been used to address AAR [20]. Similar to the monocular vision approaches, the LiDAR solution attempted to fit range readings from a receiver-mounted LiDAR to an internal tanker model. This solution offered an accuracy of approximately 35 centimeters from post-processed test flight data. While accurate, the algorithms have not yet offered real-time navigation capability. Furthermore, the approach requires the receiver to be equipped with a LiDAR sensor in the nosecone.

2.4.4 Stereo Vision.

Stereo vision has been proven in aerial refueling applications. In 1999, Boeing patented an aerial refueling system that provides real-time 3D imagery of the receiver aircraft to an operator on board the tanker aircraft [32]. Although not an autonomous solution, the system shows stereo vision successfully employed in a refueling capacity. Perhaps such a system could be leveraged to provide relative navigation cues during refueling operations.

Extensive work has also been done using stereo vision for terrestrial navigation. Navigation of unmanned ground vehicles has been demonstrated using infrared camera systems [28]. Localization of ground vehicles has also been achieved by stereo vision systems in outdoor environments [7, 17]. Finally, GPS-aided [7] and model-based [22] stereo vision solutions provide examples of terrestrial systems achieving portions of what this thesis attempts to achieve in an aerial system.

The background and previous work described here provides verified tools for achieving the goals of this thesis. The following chapter will discuss the methodology used to employ these tools for achieving precision relative positioning through stereo vision toward AAR.

III. Methodology (Algorithms and Analysis)

This chapter details the concepts and approaches used to determine relative position from stereo vision. The chapter is organized as follows. First, the assumptions and limitations of the algorithm and experiments are outlined. The experimental domain portion covers the algorithms implemented for this thesis as well as the experiments performed. Finally, intended measurements describes the quantitative assessments to be performed.

3.1 Algorithm Assumptions and Limitations

The implemented algorithm consists of six main stages: (1) stereo imagery capture, (2) rectification, (3) disparity map generation, (4) point cloud projection, (5) model fitting, and (6) result return (Figure 15). The performance of the algorithm presented in this thesis was quantitatively tested in a simulated environment. The simulation environment aims to accurately represent the real world, but requires some assumptions.

Vision System Configuration.

The stereo camera pair is located at the rear of the tanker with a fixed position and orientation relative to the tanker. All relative orientation information between the camera pair and the tanker is known. The left and right cameras are separated along the x-axis by a known baseline. The camera pair is aligned along the y- and z-axes. The cameras share an overlapping field of view.

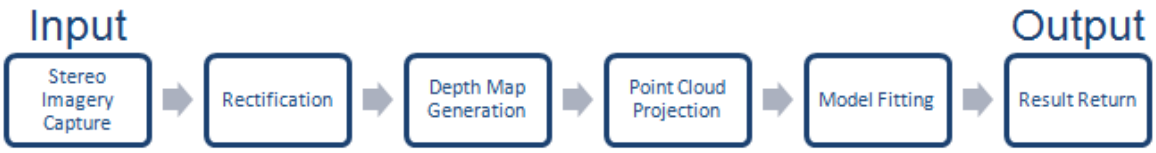


Figure 15. The basic algorithm flow.

Camera Model.

The cameras used in this thesis are assumed to follow the simple camera model, with a fixed focal length and field of view. The principle point of all images is assumed to be the image center.

Rigid Body.

The receiver is represented by a rigid model, which does not flex during flight. Similarly, the receiver is assumed to be rigid, with no changes to its form (wing defection, flaps, ailerons, landing gear, etc.).

System and Error are Scalable.

The 1:7 scale camera rig and receiver model represent a full-scale system. The error reported by the 1:7 scale system can be linearly scaled.

3.2 Experimental Domain

Simulation Environment.

For testing purposes, a precision 3D modeling environment served as the world frame. Within this environment, a rendering point served as each camera and was able to accurately model the camera properties of the real system (Figure 16). The simulation environment was used to obtain truth data for evaluation of the algorithm in this thesis.

Table 1. Stereo camera parameters for the simulated imagery (in Lightwave).

Parameter	Value
camera angles	90°,-124°,180°(x, y, z)
camera baseline	498mm
horizontal FOV	56°
vertical FOV	43.38°
focal length	18.8mm
resultion	1024x768 pix
F-15E dimensions	19m long, 13m wide
Altitude	10,000 feet

The system was run on input images from two sources: computer-generated imagery and real world imagery. The computer-generated imagery provided a simulation environment which allowed for precise knowledge and control of aircraft position, camera properties, lighting, and other properties. The simulation environment provided a full-scale representation of the system, including a full-scale receiver aircraft and camera system. For the simulated imagery, a highly detailed F-15E Strike Eagle model filled the role of receiver aircraft, with imagery from Google Earth [2] providing background and land. The tanker system was represented by two simulated cameras from which the 3D scene was rendered simultaneously. The simulated cameras were positioned within a cube representing the tanker, to ensure a precise stereo baseline and relative viewing angle. Figure 16 shows the simulation environment with the receiver in the contact position. The cameras remained fixed within the simulation environment, while the receiver moved to simulate varying relative receiver-tanker positions.

All simulation data collects and measurements utilized the Lightwave3D animation suite [3]. This allowed for rapid, identically reproducible data collects with precise truth information. The environment parameters for all simulated data collects match ideal real world parameters as shown in table 1.

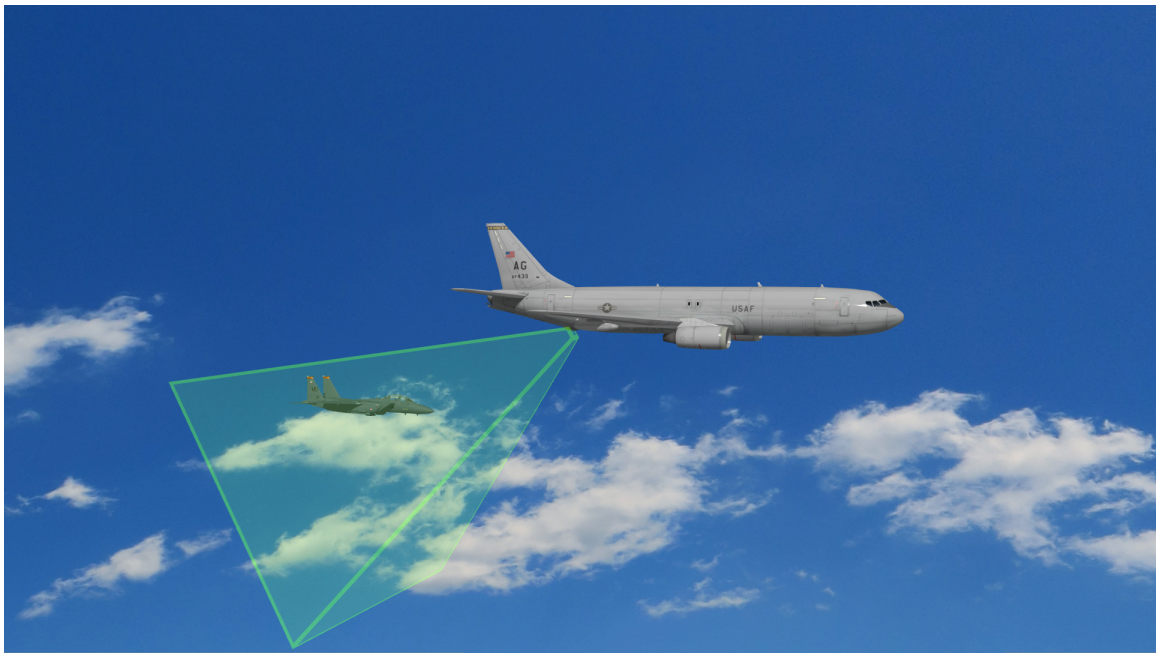
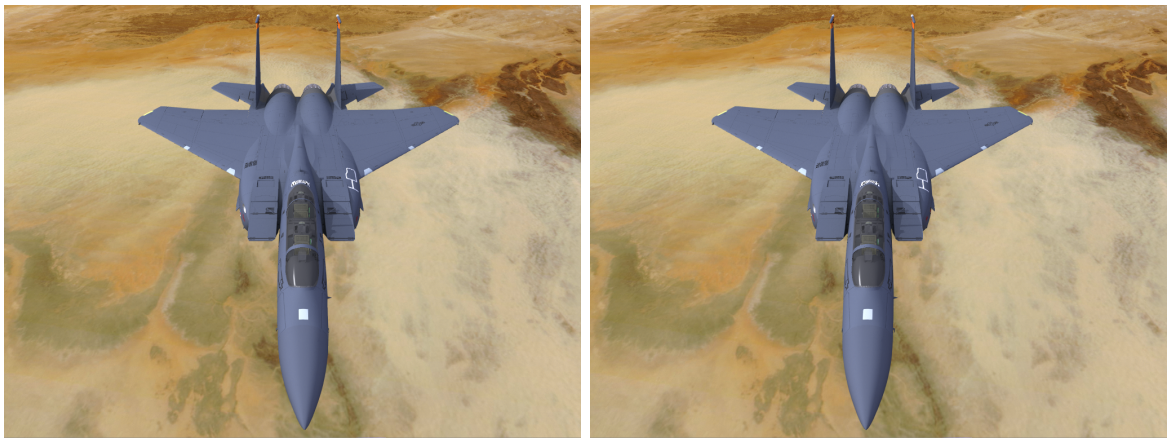


Figure 16. A view of the simulation environment with the receiver approaching the tanker. The box (left) represents the camera system housing and contains the points from which the imagery was rendered.

Each simulation data collect contained an image pair as well as the relative pose of the receiver. Because the F-15 was not modeled with a refueling port, pose was measured relative to a "refueling port" location at the base of the small antenna at the rear of the canopy. The center of a small white ball marked the precise "refueling port" location and was flagged not to render in the simulated images. A corresponding node was added to the model in the algorithm to ensure a consistent origin for reporting relative pose. The F-15 model used for the simulated imagery was the same model used by ICP for point cloud fitting of all imagery. Figure 17 shows a simulated image pair as well as the reference point used for describing receiver pose.

In addition to receiver data collects, twenty checkerboard images provided the calibration data set for the simulation environment. Each image pair captured a checkerboard in a unique position and orientation. These images were used to calibrate the simulated camera via the same algorithm used for real world imagery.



(a) Left Camera View (b) Right Camera View

Figure 17. A pair of images taken from the simulated tanker system with a highlight of the "refueling port" location highlighted (near the rear of the canopy)

Real World Camera System.

A 1:7 scale setup provided real world imagery. The scale setup consisted of a 1:7 scale tanker camera system (Figure 19a)and a 1:7 scale receiver aircraft model (Figure 19b).

The camera system consisted of a pair of Allied Vision Technologies' Prosilica CG1020C cameras. These cameras captured images at 1024x768 resolution using a full color CCD sensor.

The cameras were secured within a metal frame to ensure a constant baseline and relative positioning. A tripod mount allowed the system to be adjusted and aimed for data collection at a known position and angle. The EO cameras captured images simultaneously by triggering the slave camera from the master camera. The master camera was triggered by a host computer running Robot Operating System (ROS) Indigo Igloo [30]. Images were stored on the host hard drive for post-processing.

A 1:7 scale F-15E Strike Eagle provided the receiver model for the scaled data collects. The F-15E model, was built to scale by Fly Eagle Jet Model Factory. The combination of a 1:7 scale model and a 1:7 scale camera pair resulted in imagery that

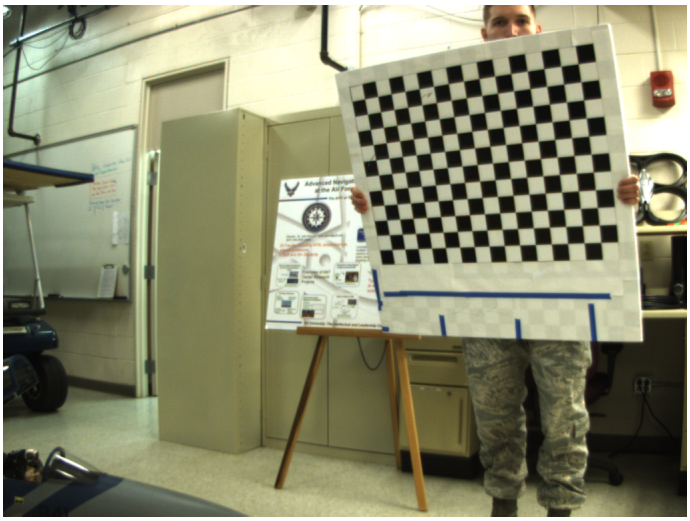


Figure 18. An example of a checkerboard image. Such images were used to verify the simulated camera system's construction, as well as allow the simulated imagery to utilize the same algorithm as the real world imagery. 20 simulated image pairs and 10 real world image pairs were used.

appeared optically full-scale. This appearance allowed the disparity maps generated from the scaled system to be projected into point clouds according to a full-scale projection matrix. Using a full-scale projection matrix resulted in a full-scale point cloud, which could be compared to the full-scale model used for ICP.

3.3 Camera Calibration and Disparity Map Calculation

This section outlines the tools and processes used to create a disparity map from the stereo image pairs. Generating a disparity map from a stereo image pair first required calibrated cameras. OpenCV v2.49 [14] provided tools for camera calibration and disparity map generation. OpenCV contains a set of calibration tools which calculate both intrinsic and extrinsic camera parameters based on a series of checkerboard images.

From a series of 20 simulated checkerboard image pairs (Figures 18 and 20), OpenCV produced an accurate pair of intrinsic and extrinsic camera matrices for

(a) Scale Camera System (b) Scale F-15 model

Figure 19. 1:7 Scale Test Setup. (a) The 1:7 scale camera system with a Prosillica EO camera pair mounted above an IR camera pair. (b) A 1:7 scale F-15 model, used in conjunction with a 1:7 scale stereo camera setup to produce image pairs representative of a full-scale system.

the simulated camera. A second calibration with 15 real world checkerboard images provided the camera matrices for the real world imagery.

Once generated, the calibration matrices were fed into a modified semi-global block matching algorithm (SGBM) [19] with the following modifications to improve runtime: (1) matching was performed between square pixel blocks (not individual

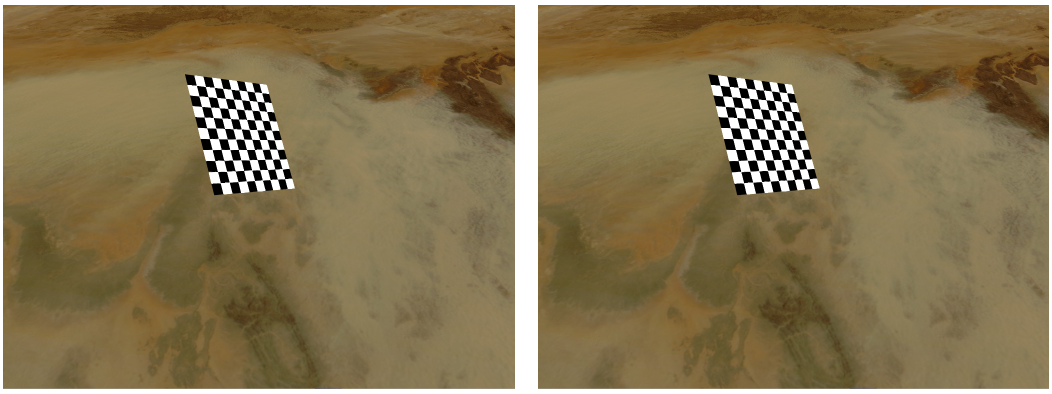


Figure 20. A pair of left and right calibration checkerboard images rendered in the simulation environment.

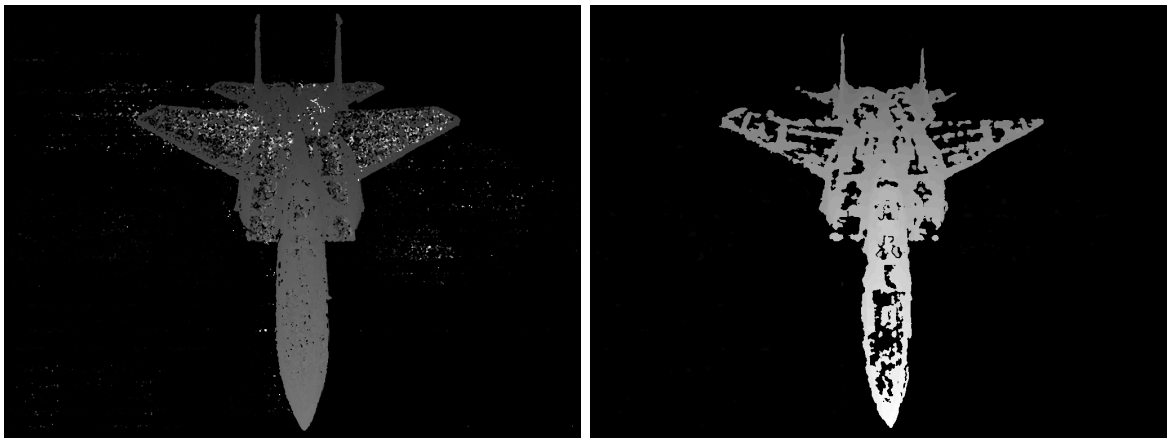
Table 2. A table listing the SGBM parameters as used in this thesis.

Parameter	Value
Window size	9
minimum disparity	0
number of disparities	96
Uniqueness ratio	40%
Speckle Window Size	50
Speckle Range	100
Single-pixel smoothness penalty	1
Multi-pixel smoothness penalty	18,816

pixels); (2) comparisons were made in five of the eight cardinal directions; and (3) a simplified Birchfield-Tomasi [12] cost function was implemented.

To handle disparity maps with a significant amount of high-disparity noise (Figure 21a), a speckle filter is applied. The speckle filter removes noise by eliminating matches without a sufficient number of similar-disparity matches within a defined window. Such filtering will inevitably remove some valid matches, but the quantity of noise removed was shown to outweighed the loss of clarity from filtering some valid matches.

The tuned parameters for the SGBM used on all simulated imagery were determined empirically (Table 2), to provide the most accurate disparity values when the receiver is in the refueling position.



(a) Unfiltered Disparity Map (b) Speckle Filtered Disparity Map

Figure 21. Disparity map filtering. (a) shows an unfiltered disparity map. The speckling surrounding the aircraft as well as the bright points on the aircraft show some of the noise present. Additional noise is present in the black, but the color variation is difficult to detect without altering the image contrast. (b) shows a disparity map after applying a speckle filter. Histogram spreading applied to emphasize detail. Note the reduction in noise, as well as some loss of useful disparity information. The black areas are now a uniform disparity.

In this way, SGBM outputs a disparity value for each block, which was assigned to the disparity value of the block's center pixel. Rendering the disparity values gives a disparity map (Figure 21b).

3.4 Point Cloud Generation

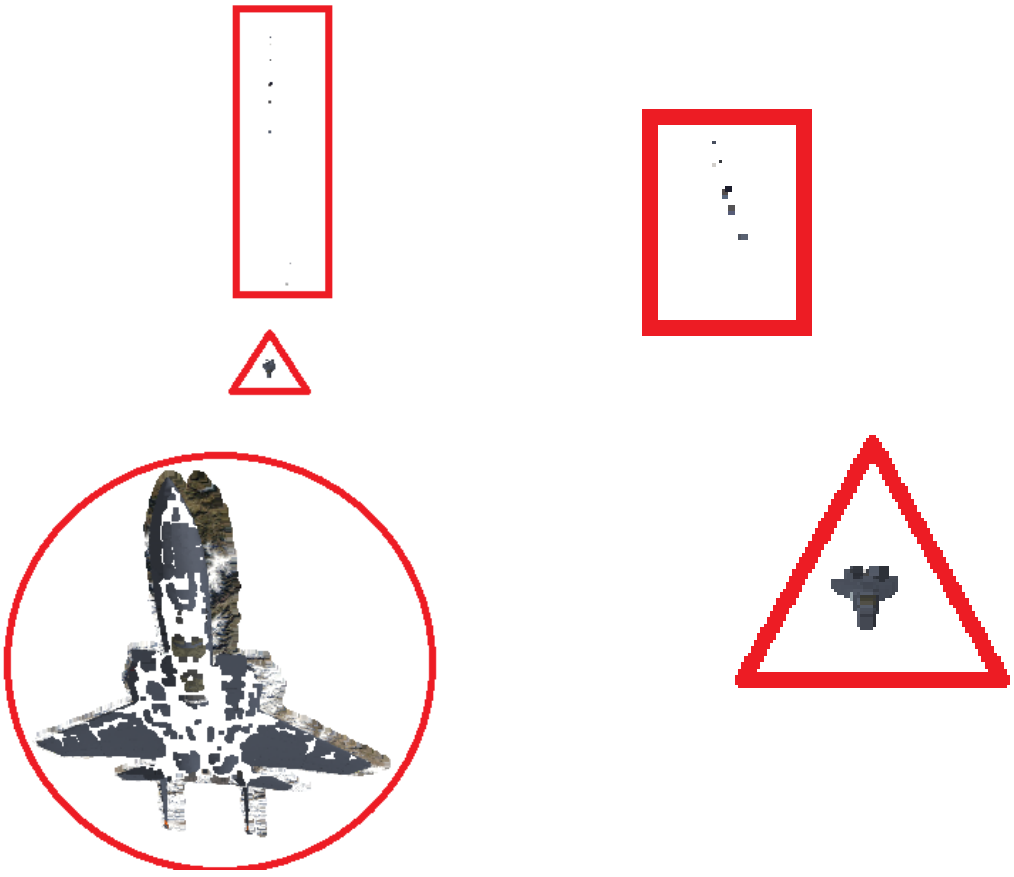
A point cloud was generated for each disparity map by projecting pixels along their epipolar line, according to its disparity value. The OpenCV function ReprojectTo3D [14] was used to generate a disparity-to-depth-mapping matrix from intrinsic and extrinsic camera matrices. However, the returned disparity-to-depth-map matrix contained an unclear discrepancy that caused the point cloud projections to result in clouds of noise. In response, the disparity-to-depth-map (Q) matrix was manually overridden using calculated values for principal point and focal length. The principal point was assumed to be the image center (an assumption commonly made in computer vision systems), with a focal length defined by

$$F = \frac{(width)_{pixels}}{2 \tan FOV}, \tag{8}$$

where F is the focal length in pixels, $width$ is the image width in pixels, and FOV is the horizontal field of view, measured in radians.

Applying the disparity-to-depth matrix to the disparity map resulted in three-dimensional projected position for each pixel, including remaining noise not removed by the speckle filter. A series of filters were required to isolate the receiver within the point cloud. First, points with infinite disparity values were removed (Figure 22a), followed by points with depth values less than a minimum threshold (Figure 22b). Finally, a moving average filter was applied to filter out remaining points located more than 75% of the receiver's length behind the point cloud's center of mass. The application of all three filters provided a clear point cloud of solely the receiver aircraft (Figure 23).

This filtered point cloud could be passed into the ICP stage in order to obtain a translation between the receiver and tanker aircraft. ICP works iteratively and each iteration requires a per-point distance calculation. The less distance between the point cloud and model, the fewer iterations ICP requires to reach a solution. Additionally, the longer the distance between the point cloud and model, the longer runtime required for each iteration. To reduce the number of iterations (i.e., runtime) required by ICP, a constant offset was created to apply an initial translation to each point cloud. This offset translates the point cloud's center of mass to or near to the camera frame's origin with a single translation. Additional translations applied by ICP were then added to the initial translation to obtain the total translation between the receiver and tanker. The constant offset selected was the translation required to move a point cloud at the refueling position to the camera frame's origin. Another



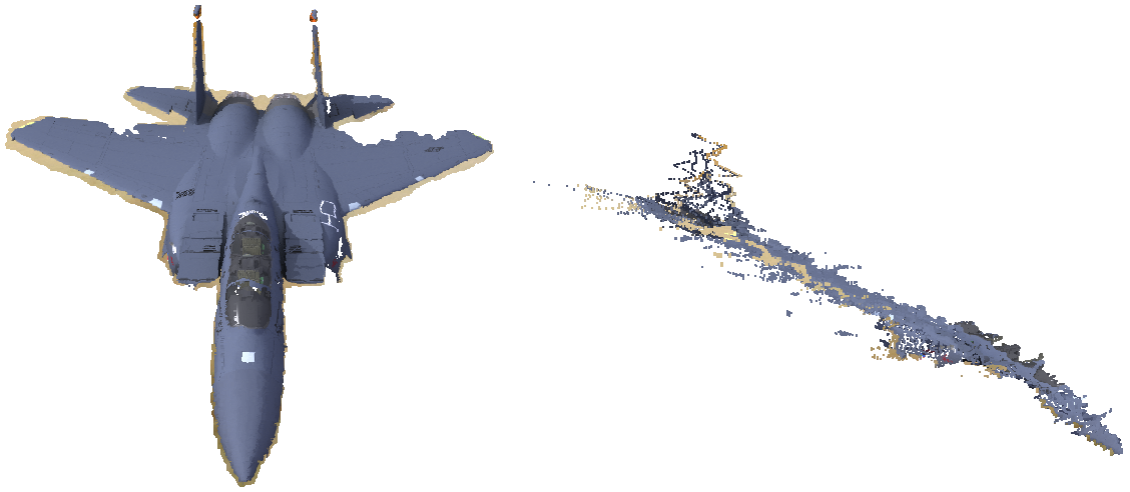
(a) Infinity Filtered (b) Infinity and Near Field Filtered

Figure 22. A point cloud at two filtering stages. The circled area shows points projected opposite the z-axis. The triangle marks the location of the receiver. The square shows trailing noise toward infinity. Filtering to remove points at infinity ((a)), within the camera focal length ((b)), and trailing toward infinity isolated the receiver within the point cloud.

key factor in the performance of ICP is the number of points that must be matched. Thus an approach for selecting keypoints is presented next.

3.5 Stochastic Universal Sampling

The point clouds generated for a simulated full-scale F-15 in a typical refueling position (depending on position relative to the tanker and block matching parameters) contained between 50,00 and 100,000 points. Because ICP operates on each point, larger point clouds result in long runtimes. Thus downsampling is employed to speed alignment. Initial test cases were run to observe the appearance of point clouds at



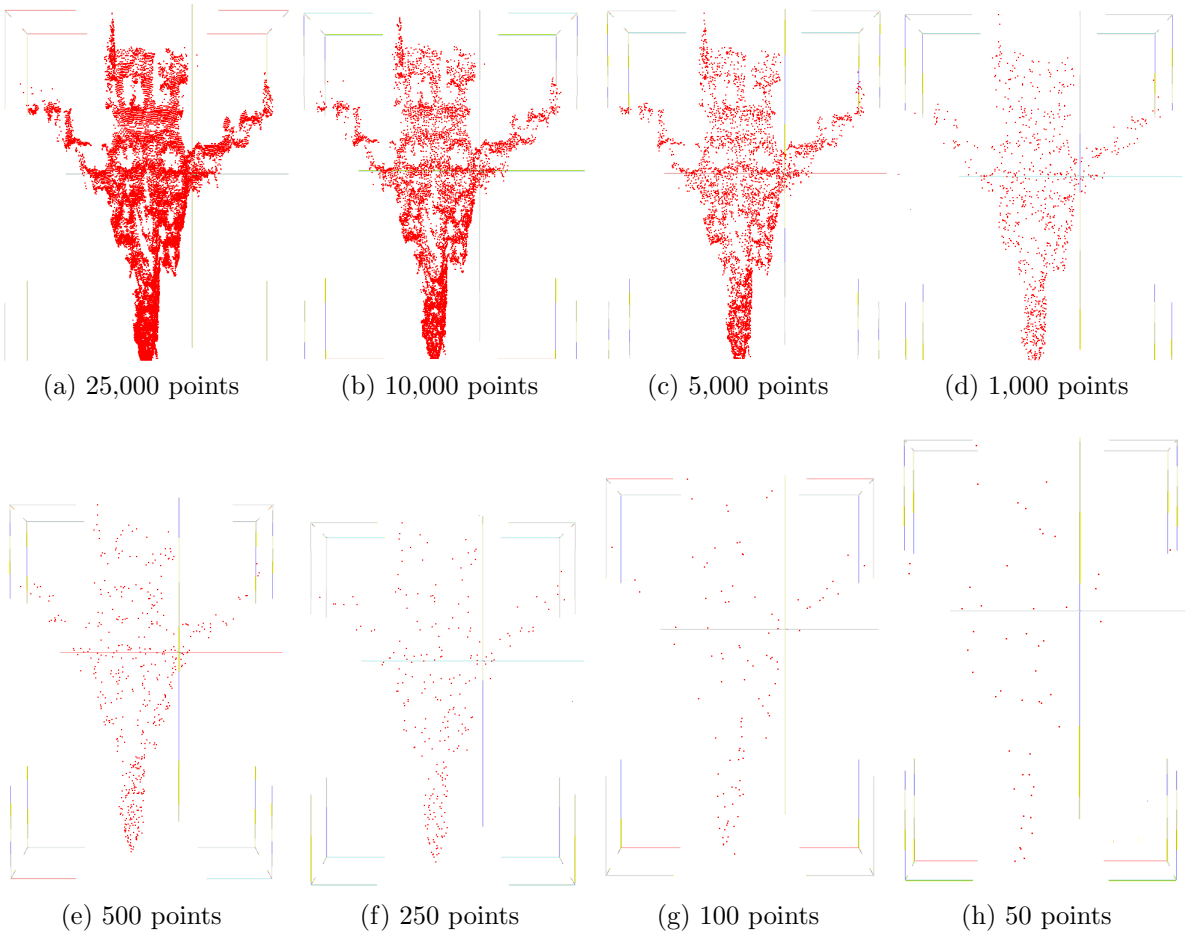
(a) Frontal View (b) Side View

Figure 23. A filtered point cloud generated from a stereo image pair, as seen from the left camera (a). The side view (b) shows that the point cloud represents only surfaces of the receiver that are visible to the camera pair.

downsampled sizes of 50,000+ (full size), 25,000, 10,000, 5,000, 1,000, 500, 250, 200, and 50 pixels (Figure 24). To gain a significant ICP speedup without risking over-downsampling, 10,000 points was selected empirically as the size of all point clouds.

Initially, random sampling was used to quickly select 10,000 points from the filtered point cloud. This resulted in downsampled point clouds with regions of high point density as well as regions of low point density. In some cases, entire sections of major features such as a rudder or wingtip failed to be represented after random downsampling, resulting in a more difficult match for ICP.

In order to ensure that all areas of the original point cloud are represented, this thesis implements stochastic universal sampling (SUS) [9]. SUS allows for random sampling to occur uniformly across the point cloud by ensuring that points are randomly selected from segmented areas covering the entire model. Each segment forms a bin containing all of the points in that segment. SUS randomly selects one point from each segment, looping back to the first segment as many times as necessary to obtain the desired number of points. Points are selected without replacement, and



(a) 25,000 points (b) 10,000 points (c) 5,000 points (d) 1,000 points

(e) 500 points (f) 250 points (g) 100 points (h) 50 points

Figure 24. SUS downsampling on a point cloud. SUS ensures that the downsampled point cloud represents points from all areas of the original cloud, regardless of the number of points removed. (b) was selected to obtain a speedup in ICP runtime, without causing the point cloud to appear drastically different from the original.

the size of each segment can be adjusted according to the level of detail in the initial point cloud.

3.6 Model Fitting and Iterative Closest Point

Most model fitting algorithms work by exploiting a predetermined correspondence between the points each model. Two of the most popular model fitting algorithms, Active Shape Modeling (ASM) and ICP are no exception. The problem domain of this thesis does not guarantee any knowledge of point correspondence

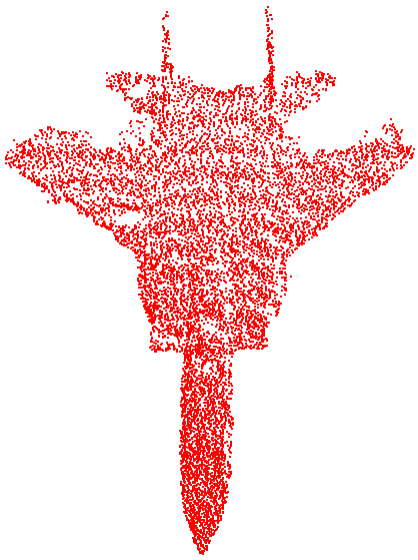


Figure 25. A point cloud downsampled to 10,000 points. (color values are discarded for ICP).

between the point cloud generated from stereo imagery and the model prior to execution. For this reason, a model-based variant of ICP was selected. Model Based ICP (MBI) [20] overcomes the lack of established point correspondence between point clouds by sampling points from the model in order to create a correspondence with the stereo-generated point cloud. The algorithm locates the nearest neighbor in the model to each point in the stereo-generated point cloud. The nearest neighbor may correspond to a vertex in the model or any point along an edge joining two vertices. The nearest neighbors form a temporary point cloud to which MBI minimizes the error with the stereo-generated point cloud (Figure 26).

Many other ICP variations also exist, but all share the same basic, iterative process: match, minimize, transform. Consider two point clouds, A and B. In the matching step, for each point in A, the nearest neighbor point in B is determined. Then, the algorithm minimizes an error metric between A and B (commonly euclidean distance between nearest neighbors, but may be implementation-specific). In the final

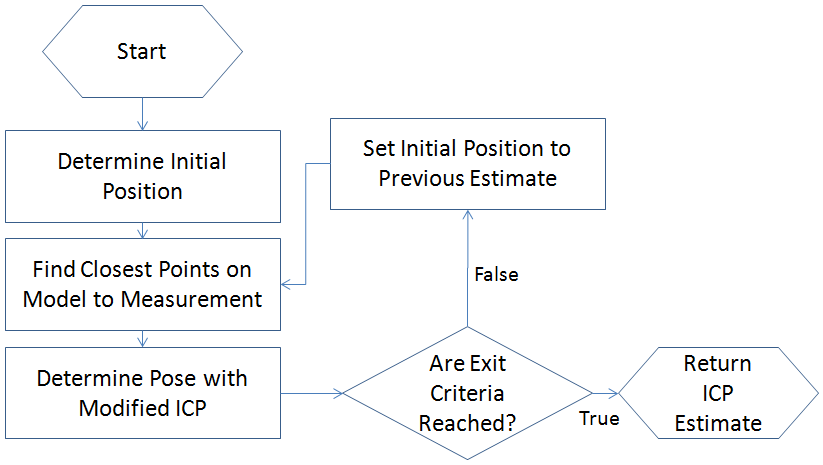


Figure 26. MBI Algorithm Flow. The algorithm iteratively matches a point cloud (A) to a model by sampling a point cloud from the model (B).

step, ICP applies a transform to A that results in minimized error, A_1. ICP concludes if the exit criteria is met, otherwise, the algorithm restarts with A_1 and B.

While the lack of per-established point correspondence grants MBI greater flexibility than standard ICP, long (sometimes infinite) runtimes hinder the algorithm's usefulness. To ensure a maximum allowable runtime, MBI contains two exit criteria: a minimum error threshold and an iteration threshold.

Noise is present in the formation of a point cloud generated from a stereo image pair, A. Although visualizations show A and B appear similar, the relative positioning of any two points in A will not match the corresponding positioning in B. This creates a minimum error that can be achieved when matching A and B. Therefore, ICP must cease on a minimum error threshold. With the goal of achieving results accurate to the centimeter level, this error threshold was set to 1 millimeter. Due to the nature of noise in the point cloud, there is no guarantee that any given point cloud will be matched to the model with an error less than 1 millimeter. To avoid, ICP becoming caught in an infinite loop as it makes minor adjustments between two alignments that minimize error nearly equally, but are above the threshold, a maximum iteration

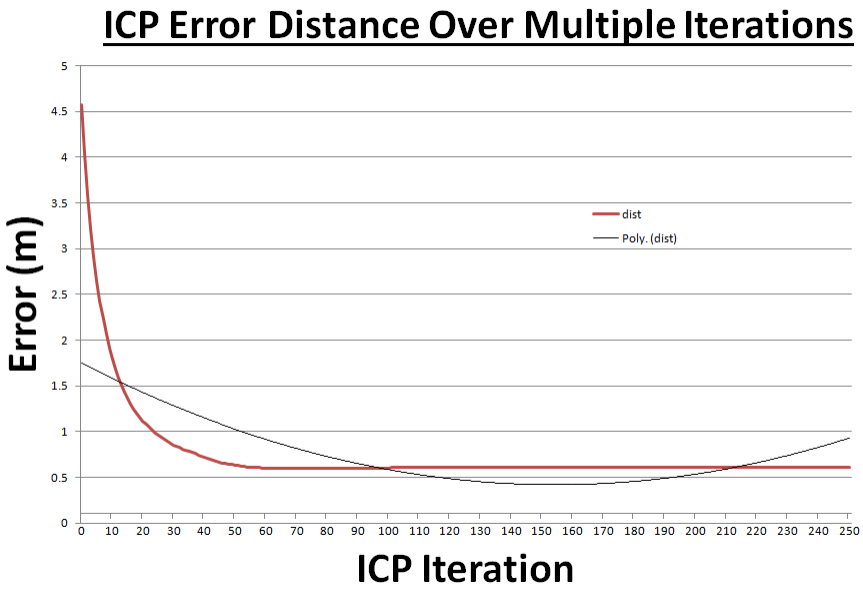


Figure 27. A graph showing the translation applied for each ICP iteration (Note the magnitude decay and trending increase for overfitting)

count is set as an exit criteria. Initial testing revealed ICP returned results (regardless of receiver position or orientation) which first quickly approached, then slowly met, and ultimately moved past the true translation, which signifies overfitting (Figure 27). After 140 iterations, each additional iteration offers a minor pose adjustment (on the order of <4mm), thus a maximum of 200 iterations were allowed due to the tradeoff between runtime and accuracy.

An ideal environment for ICP consists of two point clouds, between which a one-to-one point correspondence exists. In such a case, the total error will be minimized minimum error only when each point in A is paired with the corresponding point in B. Additional points of noise in either cloud distorts this property. After despeckling and filtering, additional points of noise remained around the edges of the point cloud. Removing such noise entirely (manually or algorithmically) was not practical. The remaining noise after minimization can bee seen in 28. With noise in the point cloud, the total error takes into account the error in fitting each point of noise to the model. This can result in significant misalignment that meets the exit criteria. These cases

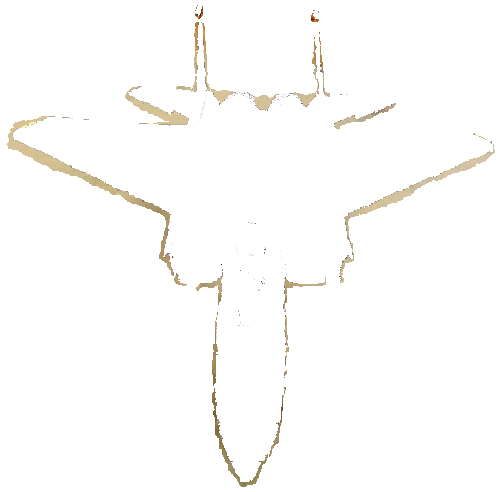


Figure 28. Point cloud error after filtering. A filtered point cloud generated from a stereo image pair, with useful points removed to show remaining erroneous points.

are local minima in the error space of ICP solutions and ICP can become caught in a local minimum. For instance, matches inverted about the y- or z-axis resulted in small error for much of the model's fuselage and wings. Such misalignments occurred rarely when the point cloud and model were in rough alignment at the beginning of ICP. Significantly misaligned starting positions or large distances between the point cloud and model frequently resulted in ICP returning a local minimum alignment.

3.7 Results Format

The final result provided by the algorithm consists of the sum of two translation matrices: a constant offset translation, and an ICP alignment (Equation 9). The algorithm begins with the point cloud at some point in the system frame, and the model loaded at the system origin. The constant offset translation translates the point cloud's center of mass toward the system origin. This translation accelerates computation by performing a rough alignment of the point cloud and model at the origin. From the translated position, ICP then calculates an additional translation

and rotation to align the model to the point cloud. The final relative position estimate is calculated as

$$Result = TR_{constant} + TR_{ICP} \tag{9}$$

where $TR_{constant}$ is a the translation and rotation estimated from the point cloud center of mass and TR_{ICP} is the translation and rotation computed by ICP.

3.8 Intended Measurements

The experiments performed here intend to measure the instantaneous accuracy of the algorithm in determining aircraft pose along six degrees of freedom. We will examine the accuracy measurements for each degree of freedom independently as a simulated receiver moves along a known flight path. In this way, the performance of the algorithm can be evaluated and compared against the point cloud center of mass (direct point cloud) estimation approach to determine the relative accuracy of each solution. The algorithm runtime will also be analyzed for runtime and accuracy under various point cloud sizes. Real world imagery will then be used for a qualitative analysis of the algorithm's feasibility to transition from the simulation domain into the real world.

The direct point cloud estimation attempts to give the receiver's position by computing the center of mass of the point cloud. This approach is less complex than ICP and may provide useful data to seed ICP and reduce overall algorithm runtime.

The isolated DOF test aims to provide accuracy measurements for each of the six degrees of freedom used to describe receiver pose. In this test, motion along a single degree of freedom will be applied for each degree of freedom.

The simulated flight path provides a 25-second video consisting of 750 unique stereo frames at 30 frames per second (Figure 29). During the course of the video,

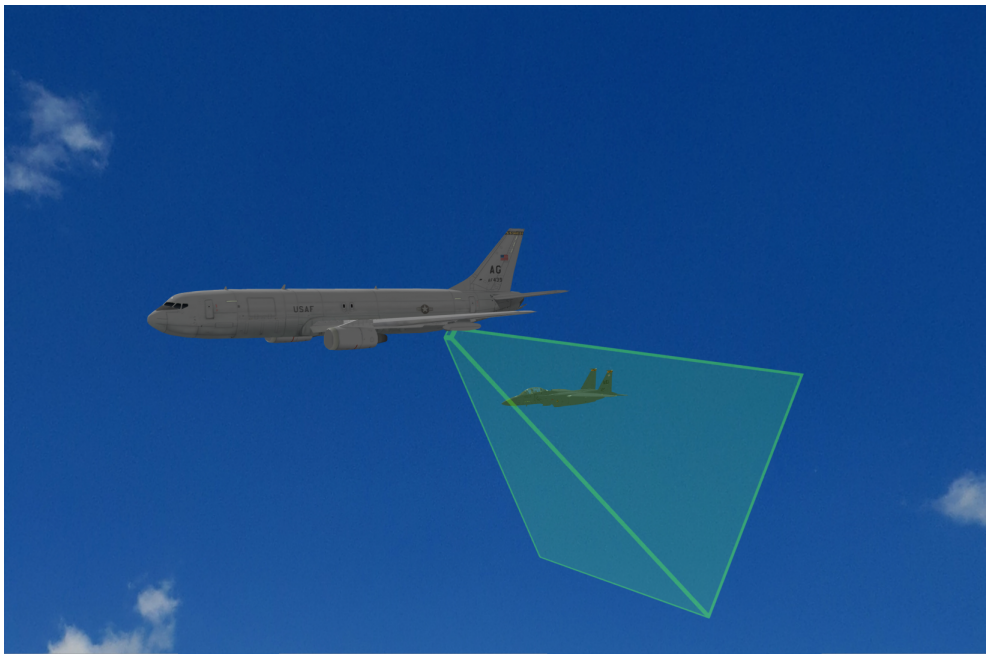


Figure 29. A screen capture of the simulation environment. The green pyramid represents the stereo camera pair's field of view.

the receiver approaches the tanker from behind and positions itself for refueling. The tanker will be traveling at 350 knots at an altitude of 10,000 feet. The receiver will move along in all six degreed of freedom. The tanker will contain the stereo camera pair and will also be subject to random movement in all degrees of freedom. The combined movements of the tanker and receiver aim to simulate a typical refueling operation from approach to the contact position.

Runtime.

The goal of an AAR application requires a minimum of video-rate results in order to operate usefully in real-time. With this requirement in mind, the algorithm's performance should be analyzed. Each phase of the algorithm will be examined for real world runtime while processing the simulated data.

IV. Results/Discussion

This chapter analyzes the results of two tests performed within the simulation environment and presents the initial results on 1/7th scale real world imagery. The experimental results demonstrate the degree to which the algorithm can describe movement in six degrees of freedom, as well as in a simulated refueling scenario.

Because point clouds generated from stereo imagery are not perfect, we can expect errors due to various anomalies. One of the key factors contributing to such error is the fact that digital imagery is a discretization of the world space. In particular, a pixel representing a feature close to the camera describes a small area in the world space, while farther features may exist in a much larger area in the world-space. Binning the world-space into pixels thus results in some x- and y-axis error for each point. Such error naturally creates error in the resulting depth map, which is then propagated into the point cloud representation when projected from the disparity map. For example, the simulated camera in this experiment uses pixels that cover a 2.7 centimeter square in the world space to describe a feature at a range of 50 meters. At 15 meters, the area represented by each pixel has reduced to 1.4 cm. The use of depth bins to describe pixels is a necessary step in translating pixel disparity to a depth value, but the binning adds an additional error in the z-axis, as each depth bin represents a range of depths in the world-space. Combined, these potential binning errors result in an increasing potential for error in each point cloud point as distance along the z-axis increases. Figure 30 shows how the point cloud is "smeared" toward the tail of the aircraft (furthest along the z-axis).

Although smearing reduces accuracy toward the tail of the aircraft, the majority of the point cloud remains an accurate representation of the receiver aircraft. ICP is able to operate on the point clouds despite the smearing, because ICP can make many close matches throughout the accurate portions of the point cloud, while accepting

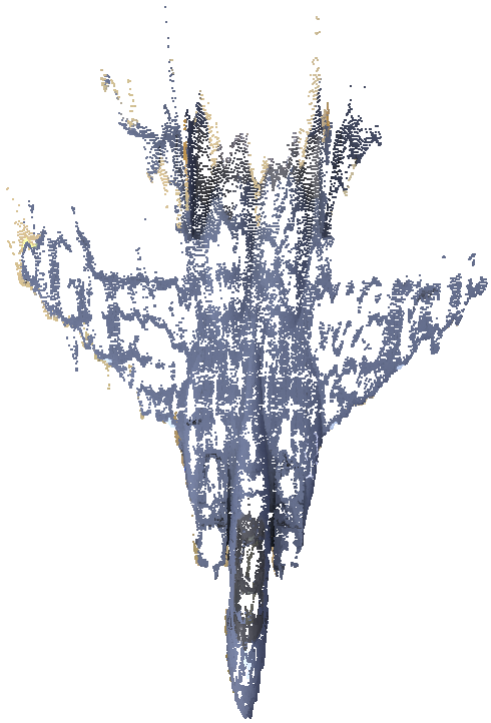


Figure 30. A point cloud showing smearing toward the tail section, as a result of imprecision in the pixel representation of the source images.

less accurate matches toward the rear of thew point cloud. Weighting the cloud's points in ICP in hopes of giving priority to more-accurate portions of the point cloud did not have a noticeable impact on ICP alignment. Trials included weighting pixels between 1 and 0 as well as 1 and 0.5.

4.1 Results

4.1.1 Direct Point Cloud Estimation.

The direct point cloud estimation returns a estimated receiver position using only the point cloud measurements. Each filtered point cloud returns a calculated center of mass (COM), which is taken as the estimated receiver position. The center of mass can be calculated from an existing point cloud in O(n) time, offering a much more rapid runtime than an ICP-based estimation.

Table 3. Direct point cloud estimation successfully reports an estimated relative position of the receiver, with increasing accuracy as the receiver approaches the tanker. Note that error increases sharply when portions of the aircraft are occluded (Frames 50 and 749).

Frame	True Distance	Error Length (m)
50	48.64	13.68
100	39.20	11.36
200	25.44	9.09
300	21.45	4.42
400	19.61	1.20
500	18.52	1.64
600	17.69	1.17
700	14.65	0.60
749	13.24	1.21

The accuracy of direct point cloud estimation varied greatly based upon two main factors: distance from the camera pair and receiver occlusion. At the beginning of the flight path, with the receiver > 25 meters from the camera pair, estimates placed the receiver 9 or more meters from its true position. As the distance between the receiver and camera pair decreased, accuracy increased rapidly. Direct point cloud estimation returned relative receiver positions accurate within +/-2 meters when the receiver moved within 20 meters of the camera pair. Accuracy remained within +/-2 meters for the reminder of the flight path (see Table 3).

Occlusion of the receiver occurred in the beginning and ending portions of the flight path, as the receiver entered from beyond pair's field of view and began to move underneath the cameras' field of view. During these periods, the estimated position error was greater than when the receiver was fully visible at a similar distance. Other occlusions such as the refueling boom or an aircraft with a wingspan greater than the horizontal field of view allows (not simulated here) could be expected to produce similar spikes in error.

The above direct point cloud estimation accuracies are somewhat misleading for an AAR implementation. The simulated F-15 refueling port is not located at the

Table 4. Although the center of mass is not the refueling port, adjusting the COM results by the COM-to-refueling-port lever arm increased error an average of > 50 centimeters.

Frame	Error Length Difference (Adjusted-Raw)
50	0.49
100	0.25
200	0.24
300	0.45
400	0.94
500	0.74
600	0.92
700	0.61
749	0.92

COM, which introduces some known error into using the center of mass estimate. A calculated translation between the cut model's center of mass and the simulated refueling port attempted to account for this error. However, the high degree of fluctuation in center of mass from frame-to-frame rendered the adjusted results less accurate along the z-axis in all test cases. The z-axis is the primary axis of motion for refueling, so the adjusted results were less helpful and therefore not used (Table 4).

This approach is too underdeveloped and imprecise for a direct AAR implementation, but the data is still useful. Direct point cloud estimation may be used to seed ICP in order to minimize the number of time-intensive ICP iterations required per estimation.

4.1.2 Isolated DOF Accuracy.

A set of simulated images were produced to replicate movement isolated to each of the six degrees of freedom. A receiver positioned at a typical refueling location and orientation provided the "base" position for all isolated DOF tests. For each image pair in the set, a perturbation of between -3 and 3 meters or 10 to 15 degrees was

applied to a single DOF. These images sought to quantify the algorithm's accuracy in each DOF when the receiver is being refueled.

The result obtained from adding the initial translation (constant offset) to the ICP alignment gave a high amount of error in each DOF. Further analysis revealed that while the error values along corresponding DOF's were consistent among trials, the x-axis, y-axis and z-axis reported estimations with an observable bias across all trials. To account for the consistent error, the average error of each DOF was calculated from a set of six trials (one per isolated DOF). The inverse of these average errors is the bias and is applied uniformly to each result as

$$Result = TR_{CO} + TR_{ICP} + TR_{bias}, \tag{10}$$

where TR_{CO}, TR_{ICP}, and TR_{bias} are the translation and rotation matrices representing the constant offset, ICP output, and bias, respectively.

Applying the bias to the sum of the initial translation and ICP alignment resulted in a reduction in per-DOF error from 40 to 100 centimeters to 1 to 14 centimeters. With the bias applied, the isolated DOF results remained accurate to within 14 centimeters and 6 degrees, regardless of the DOF perturbed. Table 5 shows the results of a single trial configuration (perturbed along the z-axis). The roll, pitch, and yaw estimations varied greatly trial-to-trial. Slight variations in the point cloud caused by the points selected during downsampling caused the rotational estimates to swing by up to 3 degrees above or below the truth value.

4.1.3 Flight Path Accuracy.

The flight path test provided 750 stereo frames, each depicting the receiver and tanker in unique relative pose. Analysis of the video provided a pose estimation at each frame. Similar to direct point cloud estimation, the system's accuracy improved

Table 5. The results of an isolated 2m movement along the z-axis.

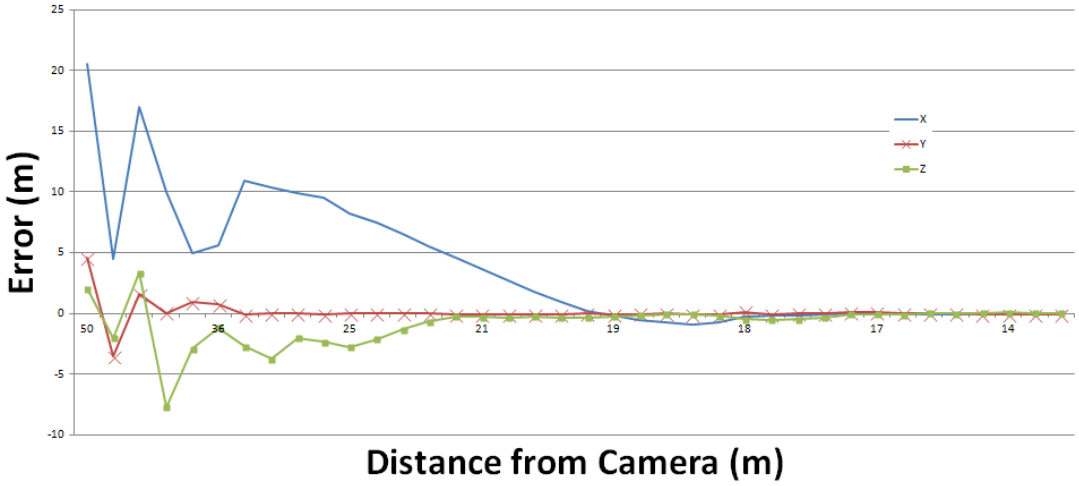
DOF	Truth Value (meters)	Error (cm or degrees)
x-axis	0	1.71
y-axis	9	2.38
z-axis	20	13.6
Roll	0	0.29
Pitch	0	0.37
Yaw	0	0.28

Table 6. Position estimation accuracy improves as the receiver approaches the tanker.

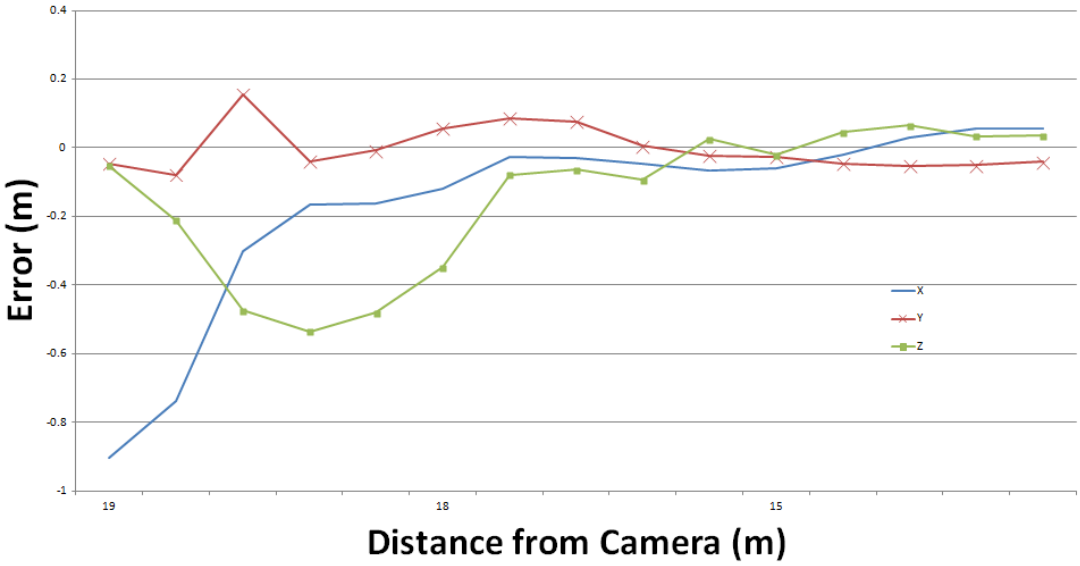
Frame	Distance to Receiver	Error Length (m)
50	48.64	7.96
100	39.20	5.83
200	25.45	9.78
300	21.45	4.56
400	19.61	0.37
500	18.52	0.78
600	17.69	0.12
700	14.65	0.06
749	13.24	0.08

as the receiver approached the refueling position (Figure 32). However, the results of the full algorithm were more accurate throughout the flight path, including areas of partial receiver occlusion. The error remains within 10 centimeters of the true position when the receiver is within 17 meters of the camera pair (Figure 31).

A selection of position estimation errors can be found in Table 6. The results show that centimeter-level accuracy can be achieved when the receiver is near the refueling position. Additionally, the algorithm's level of accuracy remains stable at a given receiver distance, regardless of partial receiver occlusions. Frame 749 (Figure 33) captures the receiver with the nosecone beyond the field of view. Despite this occlusion, the algorithm is able to estimate the receiver position to within the decimeter level.



(a) Full Approach Error: 50 to 13 meters



(b) Contact Position Error: 19 to 13 meters

Figure 31. Raw pose estimate graph. Estimates along the x-axis, y-axis, and z-axis show a steady reduction in error as the receiver approaches. (b) shows a focused view of the results when the receiver is within 19 meters. The error along all three axes remains within 10 centimeters of the true position when the receiver is within 17 meters.

Figure 34 shows the pose estimation error reducing rapidly as the receiver approaches the tanker. When the receiver is within 20 meters, the pose estimates are accurate within 1 meter. The pose estimations continue to improve to within one decimeter as the receiver reaches 16 meters from the tanker. Error was not noticeably impacted by receiver or tanker rotations.

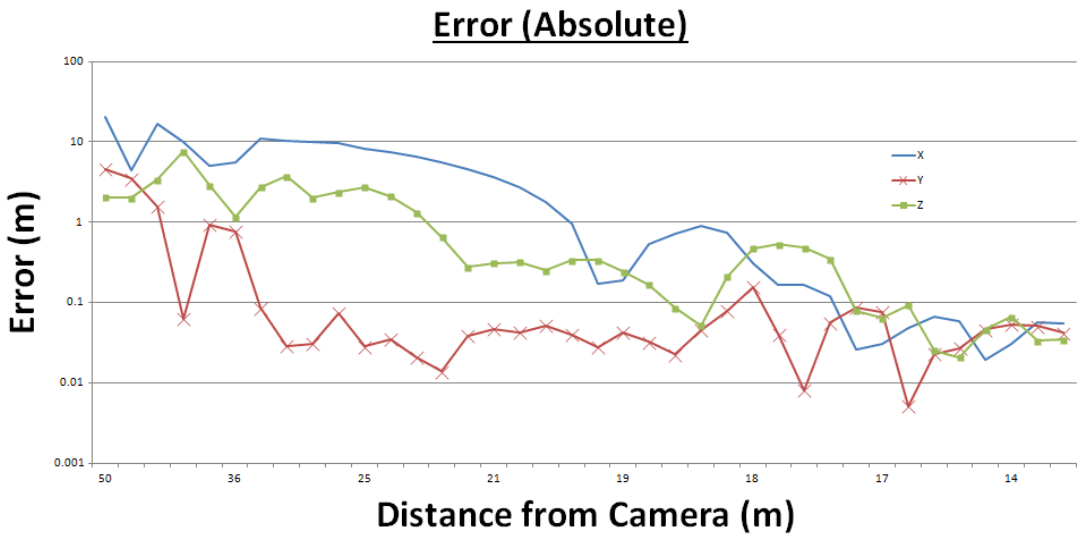
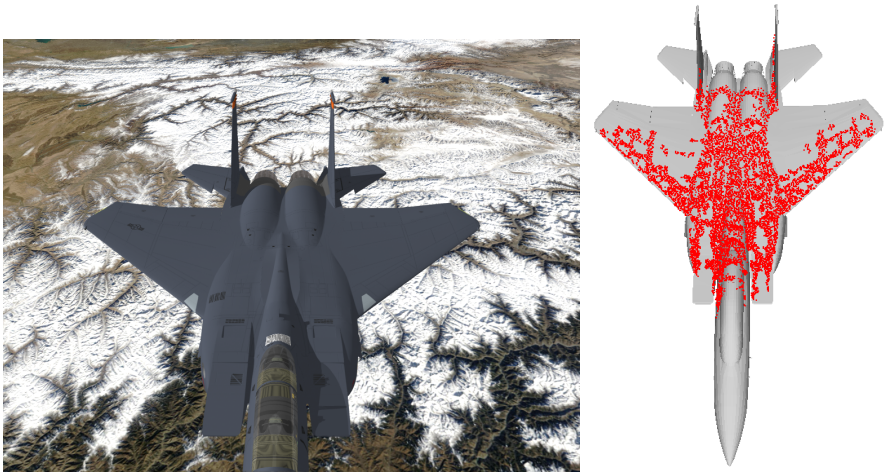


Figure 32. The corrected pose estimates (offset+ICP+bias). The graph shows error decreasing as the reciever approaches the camera pair. In this graph, the absolute value of the error is presented to emphasize the error decay as distance decreases.



(a) Occluded Nosecone

(b) Point Cloud Fit to Model

Figure 33. Flight path frame 749. Even with partial aircraft occlusions (such as the nosecone, seen here) ICP is able to accurately match the point cloud to the model. The reported accuracy with such occlusions is similar to the accuracy reported for a completely visible aircraft at the same range. (a) shows how ICP aligns the point cloud to the model with the nosecone area missing.

Error in roll, pitch, and yaw also improved as the receiver approached the tanker (Fig 35). Beyond 40 meters, 180 degree rotation errors occurred normally. In these cases, ICP rotated the model to fit the wings to the point cloud. These massive

Figure 34. ICP vs Direct Point Cloud Estimation The absolute value of the Euclidian distance error for both direct point cloud estimation and the algorithm with ICP. While both methods improve as distance to the reciever decreases, ICP results in a much more accurate estimation when the reciever is within 20 meters of the tanker.

rotations fell off sharply, with the yaw error remaining within one degree once the receiver reached a distance of 25 meters. Pitch estimation gave the least stable and least accurate results. Noise along the x-axis and z-axis was small in relation to the range of values along those respective axes. However, the shell created by the point cloud represented only the top surface of the receiver (and some of the vertical rudders). This shell contained very little range y-axis, which increased the significance of the noise. For example: In most cases, erroneous points along the y-axis appeared below the receiver's tail area. The nose section would remain properly aligned, but the rigidity of the model forced ICP to apply a pitch in order to minimize the distance between all points at the tail of the aircraft wile keeping the nose aligned.

4.2 Timing Profile

In order to be applied to the AAR task, the algorithm must be capable of operating in real time (30 frames per second or greater). Although the algorithm used in

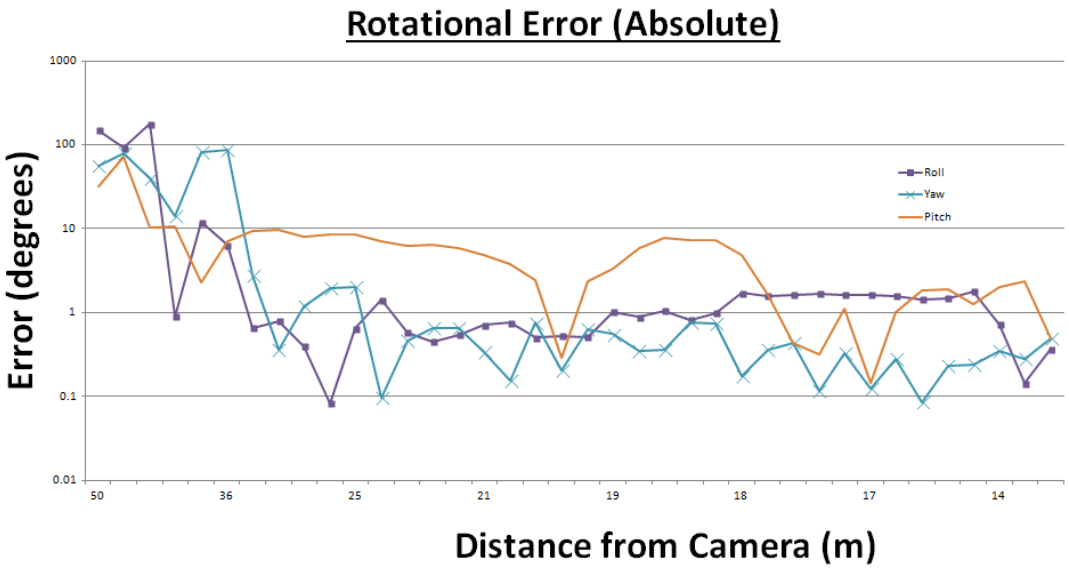


Figure 35. A graph of rotational error over the flight path.

this thesis post-processed data, the relative bottlenecks of algorithm operation reveal where the greatest potential for speedup lies. The algorithm was not parallelized for this thesis.

The first three algorithm stages (image capture, rectification, and disparity map generation) are O(n) operations where n is the number of pixels in the input image. The runtime of the fourth stage (point cloud generation) is also O(n), but may take up to O($4n$) depending upon filtering parameters and the number of points in the downsampled result. The first four stages required 1.685 seconds per image to produce a 10,000-point cloud on a 3.20GHz dual-core system (Table 7). This point cloud is of the same quality cloud seen in Figure 25.

The most time-intensive stage was point cloud generation, which accounted for the majority of the runtime fo the first four stages. Point cloud generation consists of point projection, filtering, and downsampling (in order). Point projection required and average of 0.586 seconds. Downsampling to 10,000 points required 0.47 seconds. Downsampling to 1,000 decreased the average downsampling time to 0.04 seconds.

Table 7. Runtime of the first four algorithm stages. All times measured in wall-clock time (from image load to result return). Point cloud generation makes up nearly 77% of the algorithm runtime without ICP and represents the best candidate for runtime reduction in future work.

Algorithm Step	Runtime (%)	Runtime (seconds)
Rectification	0.675	0.0093
Disparity Map Generation	22.59	0.311
Point Cloud Generation	76.73	1.056
Overall	1.0	1.367

Parallelization of the point cloud generation and downsampling to run on a GPU would offer a significant speedup toward real time operation.

Model fitting made up the remainder of the algorithm runtime. Model fitting runtime depended upon two main factors: constant offset accuracy and point cloud size. Although not explicitly tested, constant offset accuracy offered the single greatest impact on runtime. The constant offset applied a transform to each point cloud in order to bring the center of mass to the designated refueling position. ICP failed to complete a single iteration on a 10,000-point cloud in five minutes when presented with a point cloud that had not been adjusted by a constant offset. Once the offset was applied, ICP returned 63 iterations in 5.22 minutes before exiting due to the minimum movement threshold.

Point cloud size offered the second largest runtime variation. Point clouds of varying sizes were generated for multiple images. ICP was performed on these point clouds with a minimum movement threshold of 1 millimeter. Reducing the number of points in the final point cloud offered a linear speedup to ICP with larger point clouds (Table 8). Frame 300 of the flight path video represents the results seen throughout the test cases. For frame 300, a 1,000-point cloud required 33.98 seconds to complete ICP, while a 10,000-point cloud required 313.32 seconds.

Table 8. ICP runtime and the number of points in the point cloud are linearly related for larger point clouds.

Number of Points in Cloud	ICP Runtime
50	1.12 sec
100	6.53 sec
250	8.86 sec
500	14.22 sec
1000	33.98 sec
5000	2.52 min
10000	5.22 min
25000	12.30 min
50000	22.80 min

4.3 Point Density Impact on Accuracy

Interestingly, the reduction in number of points did not have an adverse effect on matching accuracy. Starting with a point cloud of nearly 75,000 points, various downsampled clouds were created using SUS on the original cloud (Figure 24). The downsampled clouds were each matched using ICP and the accuracies were recorded (Table 9). Although severe downsampling caused the point cloud to appear less like an aircraft to the naked eye, ICP was able to make an accurate alignment, and at an increased speed.

Again, frame 300 of the flight path video provides a strong example. The resulting euclidean error for 50,000, 25,000, 10,000, and 5,000-point clouds each fell within 1 centimeter of each other. The smallest point cloud, at 50 points, still returned an estimate within 6 centimeters of the average. Error variance remained consistent when the receiver was within 26 meters. Beyond 26 meters, downsampling below 1,000 points was not always possible as the full point cloud often contained 500-2,000 points.

The relatively small decrease in accuracy compared to the linear decrease in runtime makes downsampling a strong candidate for consideration in creating a real-

Table 9. A comparison between point cloud size and Euclidean error length for a single frame.

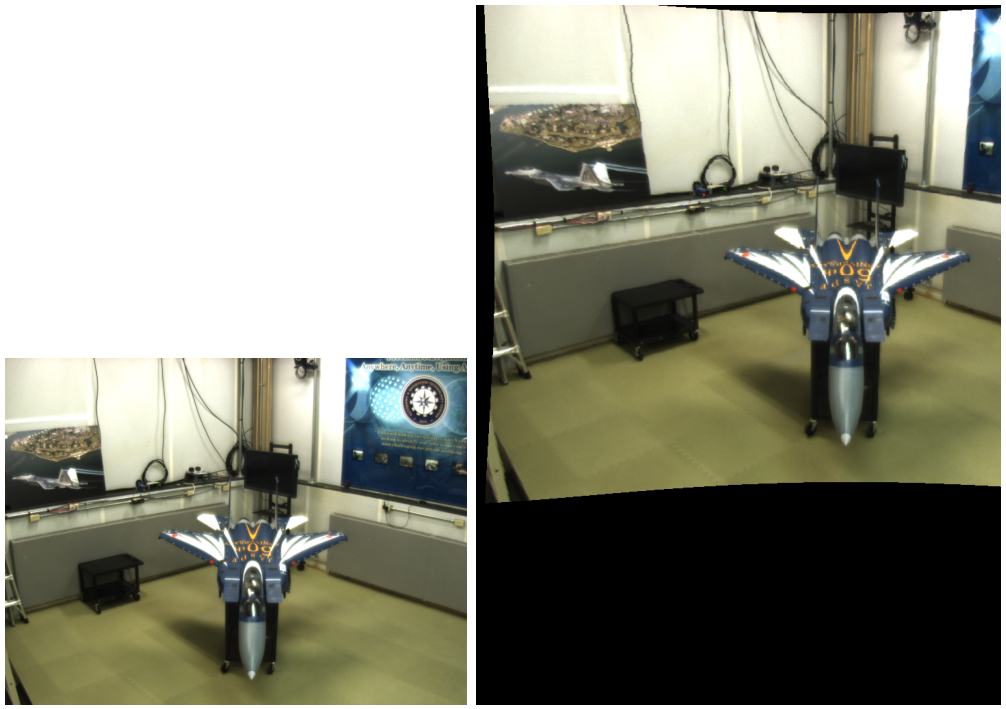
Number of Points	Error Length
50	0.1478
100	0.1368
250	0.0128
500	0.1195
1000	0.0489
5000	0.0867
10000	0.0798
25000	0.0864
50000	0.0828

time stereo vision solution. Greater accuracy is preferable, but establishing a desired accuracy range would allow downsampling to be maximally exploited to reduce run-time.

4.4 Real-world Imagery Examples

Qualitative analysis shows that this approach is capable of translating from the simulation domain to the real world. The Prosillica cameras captured images of a 1/7th scale F-15 on a 1 meter stand (Figure 36a). The algorithm was run on the real world imagery without modification from the simulation domain experiments, with the exception of the camera calibration matrices and point cloud filter parameters. All calibration matrices were recomputed using a checkerboard pattern (through the same algorithm that generated the calibration matrices for the simulated imagery).

After calibration, the real world images resulted in a disparity map in which the F-15 is distinguishable (Fig 36c). The disparity map suffers from vertical bars of high disparity along the bottom and left sides of the image. These areas are artifacts from the rectification process (Figure 36b). Because the locations of the artifact areas is known, the bars of high disparity are easily cropped from the point cloud.



(a) Image (b) Rectified

(c) Disparity Map (d) Point Cloud

Figure 36. Real World Imagery. (a) shows the 1:7 scale receiver as captured by the right camera of the 1:7 scale stereo camera pair. (b) shows the image after rectification to remove both radial and tangential distortion. (c) shows the disparity map created from the left and right rectified images. (d) shows the point cloud projected from the disparity map. The gradient background was added to increase contrast between the background and the largely white wings and elevators.

The point cloud generation required adjustment to the filtering parameters to eliminate background objects from the point cloud. This adjustment is specific to the test case, where the receiver is surrounded by walls, a floor, a stand, and other objects (rather than flying). Once the filter parameters were empirically set, the resulting point cloud clearly isolated the receiver (Figure 36d).

The resulting point cloud shows the feasibility of this algorithm to be applied to full-scale, real world imagery. Although truth information was not available a the time of image capture, the initial ICP results are of the correct magnitude.

V. Conclusion

This thesis presented an algorithm for determining relative aircraft position using stereo vision. In conclusion, the position estimates returned by both direct point cloud estimation and ICP estimation demonstrate that relative positioning from stereo vision is possible. Experimental results show that pose estimates on the order of +/-10cm are achievable while the receiver is in the contact position. Although measured in simulation, the very same algorithm is also able to tender pose estimates from a 1:7 scale vision system that are qualitatively correct.

5.1 Future Work

This section outlines areas for future work and research. Many areas of improvement exist if we are to continue moving towards the AAR task.

Real-world Test and Analysis. The algorithm analyzed in this thesis has been tested qualitatively on real world imagery. Moving forward, quantifiable results on images captured int the real world would greatly improve the confidence in this algorithm for real world application. A precision measurement system such as a VICON chamber would be ideal for gathering quantitative test data, as sensor balls could be placed at easily identifiable points on the model, allowing for a direct translation between truth data and algorithm output.

Real-Time Operation. This thesis post-processed all images using a single process. The timing profile clearly shows that the current algorithm runs too slowly for flight operation. In order to decrease the algorithm's runtime, many of the stages could be parallelized and optimized for running on a GPU. A well-parallelized algorithm would be able to significantly reduce the runtimes required for this thesis.

Multiple-Sensor Systems. Other AAR solutions (such as DGPS) may benefit from a confidence-check performed by a stereo vision system. In the event that the primary system fails, or provides erroneous data, a comparison against the stereo vision system would result in a low confidence, signaling the receiver and tanker to take appropriate action.

Generalized Implementation. The current algorithm relies heavily on the open-source rendering environment Ogre. While Ogre is helpful for simplifying three-dimensional model manipulations and key to real-time visualization of the algorithm's results, Ogre is not required from a theory perspective. Moving to a more generalized, array-based solution would offer much more flexibility in upgrading the algorithm and translating it to varying platforms, as well as offer a significant speed improvement.

Reduce the Number of Approximations. In addition to the approximations made by the simulation environment, calculating a true disparity-to-depth matrix would offer more precise results, especially on real world imagery where the approximation used in this thesis will be less representative of the true matrix.

Seeding ICP. As mentioned earlier, direct point cloud estimation may provide a fast method of seeding ICP in order to reduce the number of ICP iterations required per frame. The time required for each iteration, as well as the total number of iterations required to reach a given threshold, is reduced when the point cloud is fed to ICP in rough alignment with the model. Direct point cloud estimation may be able to provide ICP with a point cloud that is aligned to the model much more closely in all cases than the constant offset applied in this thesis.

5.2 Final Remarks

Aerial refueling capability is key to the Air Force's missions of rapid global mobility and global attack. This thesis shows that a stereo vision systems can be used to provide precision relative navigation to a receiver aircraft. Position estimates accurate to +/-10 centimeters have been shown in simulation, and the same algorithm has been used to process scaled real world imagery with qualitatively correct results. With future work, stereo vision may be leveraged to aid in bringing an AAR capability to the growing wings of UAS and RPA, which currently lack an aerial refueling capability.

Bibliography

1. Coordinate system.

2. Google earth.

3. Newtek lightwave 3d animation suite.

4. Gps.gov: Augmentation systems. World Wide Web Page. Available at *http://www.gps.gov/systems/augmentations/*.

5. The middlebury computer vision pages. World Wide Web Page. Available at *http://vision.middlebury.edu/stereo/data/*.

6. Department of defense world geodedic system 1984. Technical report, National Imagery and Mapping Agency, January 2000. Third edition.

7. M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1063–1068. IEEE, 2006.

8. R. Alonso, J. L. Crassidis, and J. L. Junkins. Vision-based relative navigation for formation flying of spacecraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Denver, CO*, 2000.

9. J. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceeding of the Second International Conference on Genetic Algorithms and their Application*, pages 14–21, Hillsdale, New Jersey, 1987.

10. S. T. Banard and M. A. Fischler. Computational stereo. In *Computational stereo*, pages 553–572, 1982.

11. P. Besl and N. D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, Feb 1992. ISSN 0162-8828. doi: 10.1109/34.121791.

12. S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on pattern Analysis and Machine Intelligence*, 1998.

13. R. E. Bowers. Estimation algorithm for autonomous aerial refuelling using a vision based relative navigation system. Master's thesis, Texas A&M University, 2005.

14. G. Bradski. *Dr. Dobb's Journal of Software Tools*.

15. D. C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 32 (3):444–462, May 1966.

16. J. L. W. David Titterton. *Strapdown Inertial Navigation Technology, 2nd Edition*. The Institution of Engineering and Technology, London, United Kingdom, 2004.

17. U. Franke and A. Joos. Real-time stereo vision for urban traffic scene understanding. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 273–278, 2000.

18. E. Guelch. Results of test on image matching of isprs. Master's thesis, University of Stuttgart Institute for Photogrammetry, 1988.

19. H. Hirschmuller. Stereo processing by semiglobval matching and mutual information. *PAMI*, 30(2):328–341, Feb 2008.

20. J. A. C. II. Automated aerial refueling position estimation using a scanning lidar. Master's thesis, Air Force Institute of Technology, 2012.

21. J. J. Koenderink, A. J. Van Doorn, et al. Affine structure from motion. *JOSA A*, 8(2):377–385, 1991.

22. A. Kosaka and A. C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: Image understanding*, 56(3):271–329, 1992.

23. C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, 1(1):125–131, 1999.

24. D. G. Lowe. Distinctive image features from scale-invariant key points. *International Journal of Computer Vision*, 60(2):91–110, 2004.

25. L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, 1989. ISSN 0920-5691. doi: 10.1007/BF00133032. URL `http://dx.doi.org/10.1007/BF00133032`.

26. *Comparison of Relative Navigation Solutions Applied Between Two Aircraft*. National Aeronautics and Space Administration, Dryden Flight Research Center, Edwards, Claifornia 93523-0273, June 2002. Available at http://www.nasa.gov/centers/dryden/pdf/88740main_H-2498.pdf.

27. D. Oram. Rectification for any epipolar geometry. In *BMVC*, volume 1, pages 653–662, 2001.

28. K. Owens and L. Matthies. Passive night vision sensor comparison for unmanned ground vehicle stereo vision navigation. In *Computer Vision Beyond the Visible Spectrum: Methods and Applications, 1999.(CVBVS'99) Proceedings. IEEE Workshop on*, pages 59–68. IEEE, 1999.

29. B. W. Parkinson and P. K. Enge. Differential gps. *Progress in Astronautics and Aeronautics: Global Positioning System Theory and Applications*, 2(164):3–32, 1996.

30. C. K. G. B. P. F. J. F. T. L. J. W. R. Quigley, Morgan. and A. Y. Ng. Ros: an open-source robot operating system. *IRCA Workshop on Open Source Software*, 2009.

31. L. Robert and R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In B. Buxton and R. Cipolla, editors, *Computer Vision ECCV '96*, volume 1064 of *Lecture Notes in Computer Science*, pages 439–451. Springer Berlin Heidelberg, 1996. ISBN 978-3-540-61122-6. doi: 10.1007/BFb0015556. URL `http://dx.doi.org/10.1007/BFb0015556`.

32. D. Ruzicka. Automated director light system for aerial refueling operations, May 18 1999. URL `http://www.google.com/patents/US5904729`. US Patent 5,904,729.

33. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo corresondence algorithms. *International Journal of Computer Vision*, 47 (1-3):7–42, 2001.

34. J. C. Sean M. Calhoun, John Raquet. Flight test evaluation of image rendering navigation for close-formation flight. In *Proceedings of the 25th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2012)*, pages 826–832, Nashville, TN, September 2012.

35. R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer Publishing, Anytown, State, 2010.

36. A. Weaver. Using predictive rendering as a vision-aided technique for autonomous aerial refuelling. Master's thesis, Air Force Institute of Technology (AFIT), 2009.

37. Q. Yang, C. Engels, and A. Akbarzadeh. Near real-time stereo for weakly-textured scenes. In *BMVC*, pages 1–10, 2008.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
26–03–2015	Master's Thesis	Sept 2013 — Mar 2015

4. TITLE AND SUBTITLE

Precision Relative Positioning for Automated Aerial Refueling from a Setero Imaging System

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Werner, Kyle, P. 2d Lt, USAF

5d. PROJECT NUMBER

15-245

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-15-M-048

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Caroline King
Aerospace Systems Sirectorate
Air Force Research Laboratory (AFRL/RQQC)
2210 8TH ST
WPAFB OH 45433
937-938-4644 caroline.king@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQQC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S. Government and is not sunject to copyright protection in the United States

14. ABSTRACT The United States Air Force relies upon aerial refueling to fulfill its missions. Unmanned aerial systems (UAS) and remotely piloted aircraft (RPA) do not currently have access to this capability due to the lack of an on-board pilot to safely maintain a refueling position. This research examines stereo vision for precision relative navigation in order to accomplish the Automated Aerial Refueling (AAR) task. Previous work toward an AAR solution has involved the use of Differential Global Positioning (DGPS), Light Detection and Ranging (LiDAR), and monocular vision. This research aims to leverage organic systems in future aircraft to compliment these solutions. The algorithm presented here generates a point cloud from the disparity between stereo camera images. The algorithm then fits the point cloud to a digital model using a variant of *iterative closest points* (ICP). The algorithm was tested using simulated imagery of an F-15E rendered in a 3D modeling environment. Experimental results showed a significant increase in accuracy as the receiver aircraft approached the tanker aircraft, reporting accuracies within +/-10cm at distances less than 17m. The algorithm's ability to transition to the real world was validated qualitatively using a 1:7 camera and model setup.

15. SUBJECT TERMS

aerial refuelling, AAR, stereo vision, position estimation, relative navigation, model fitting

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Maj Brian G. Woolley, PhD, AFIT/ENG
U	U	U	U	77	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4555; brian.woolley@afit.edu