TOWARD AUTOMATED AERIAL REFUEING: RELATIVE NAVIGATION WITH STRUCTURE FROM MOTION

THESIS

Kevin W Colson Jr, Capt, USAF

AFIT-ENG-MS-16-M-009

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-16-M-009

TOWARD AUTOMATED AERIAL REFUEING: RELATIVE NAVIGATION WITH STRUCTURE FROM MOTION

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 Electrical Engineering

Kevin W Colson Jr, B.S.E.E.

Capt, USAF

March 24, 2016

AFIT-ENG-MS-16-M-009

TOWARD AUTOMATED AERIAL REFUEING: RELATIVE NAVIGATION WITH STRUCTURE FROM MOTION

THESIS

Kevin W Colson Jr, B.S.E.E.
Capt, USAF

Committee Membership:

Maj Brian Woolley, PhD
Chair

Dr. John Raquet
Member

Dr. Scott Nykl
Member

Abstract

The Unisted States Air Force (USAF) use of Unmanned Aeriel Systems (UAS) has expanded from reconnaissance to hunter/killer missions. As the UAS mission further expands into aerial combat, better performance and larger payloads will have a negative correlation with range and loiter times. Additionally, the Air Force Future Operating Concept calls for "formations of uninhabited refueling aircraft...[that] enable refueling operations partway inside threat areas." However, a lack of accurate relative positioning information prevents the ability to safely maintain close formation flight and contact between a tanker and a UAS. The inclusion of cutting edge vision systems on present and future refueling platforms may provide the information necessary to support an Automated Aerel Refueling (AAR) mission by estimating the position of a trailing aircraft to provide inputs to a UAS controller capable of maintaining a given position. This research examines the ability of Structure from Motion (SfM) to generate this relative navigation information. Previous AAR research efforts involved the use of differential Global Positioning System (GPS), Light Detection and Ranging (LiDAR), and vision systems. This research aims to leverage current imaging technology to compliment these solutions. The algorithm used in this thesis generates a point cloud by determining three dimensional (3D) structure from a sequence of two dimensional (2D) images. It then utilizes Principal Component Analysis (PCA) to register the point cloud to a reference model. The algorithm was tested in a real world environment using a 1:7 scale F-15 model. Additionally, this thesis conducts a study of common 3D rigid registration algorithms in an effort characterize their performance within the AAR domain. Three algorithms are generated and tested for runtime and registration accuracy on four point cloud data sets.

Acknowledgements

Thank you God for providing me with the knowledge, drive, and stubbornness necessary to successfully generate the following document. Thank you to my wife and my daughter for trading the warm beaches of the Gulf Coast for the frozen tundra of Ohio winters & enduring the single-parent days, nights, & weekends. Thank you Mom, Dad, and Sis for setting the bar high, lifelong support, a constant example of humbly achieving success, and providing a holistic perspective enabling me to aim high, achieve my goals, & enjoy life at every opportunity. Thank you to my in-laws for providing the emotional support and warming reminders of home as our young family simultaneously moved out into the world, learned to be parents, & began this highly stressful undertaking. Thank you Maj Woolley, my advisor, for the general support and guidance necessary to produce the following document. Thank you engineers, past, present, and future, who have and will successfully generate a similar product in an effort to expand the boundaries of human knowledge. Finally, thank you reader. If you are reading this line than you have read at least a single page of my work, ensuring the hours spent in the pursuit of this effort were not in vain.

"If his works do not work, he is damned. That is the phantasmagoria that haunts his nights and dogs his days. He comes from the job at the end of the day resolved to calculate it again. He wakes in the night in a cold sweat and puts something on paper that looks silly in the morning. All day he shivers at the thought of the bugs which will inevitably appear to jolt its smooth consummation." –Herbert Hoover

Kevin W Colson Jr

Contents

Page

List of Figures

List of Tables

List of Abbreviations

2D two dimensional

3D three dimensional

4D four dimensional

AAR Automated Aerel Refueling

AFRL Air Force Research Laboratory

AFIT Air Force Institute of Technology

BA bundle adjustment

DCM Direction Cosine Matrix

DGPS differential Global Positioning System

DOF degree of freedom

dSfM dynamic Structure from Motion

GPS Global Positioning System

Hz Hertz

ICP Iterative Closest Points

INS Inertial Navigation System

LED light emitting diodes

LiDAR Light Detection and Ranging

LM Lavenberg-Marquadt

m meters

NASA National Aeronautics and Space Administration

pBA partial bundle adjustment

PCA Principal Component Analysis

RANSAC Random Sample Consensus

RMS root-mean-square

RVS Rockwell Collins Remote Vision System

SEP Spherical Error Probable

SfM Structure from Motion

SIFT Scale Invariant Feature Transform

SVD singular value decomposition

UAS Unmanned Aeriel Systems

US United States

USAF Unisted States Air Force

VisNav Vision-Based Sensor and Navigation System

vSfM visual Structure from Motion

TOWARD AUTOMATED AERIAL REFUEING: RELATIVE NAVIGATION WITH STRUCTURE FROM MOTION

I. Introduction

The use of UASs in support of military operations began during the Civil War when balloons loaded with explosives were used in an effort to destroy supply and ammunition depots [40]. However, technology did not permit effective use until Vietnam when drones were first utilized for simple day reconnaissance with conventional cameras. Successful operations quickly led to night photo operations, signals intelligence, leaflet dropping and surface-to-air missile radar detection. UASs have since become critical to many military operations driven by a dramatic increase in sensor, guidance, and navigation technologies. These advances enabled operations to extend past the limits of human endurance to that of the platform itself. The first modern UAS systems (the RQ-4 Global Hawk & the MQ-1 Predator) were developed for reconnaissance with a focus on long loiter times. Soon after, the Predator's mission was expanded to include armed reconnaissance. The past decade has seen continuous expansion of UAS missions with their quick evolution leading to many hunter/killer systems such as the MQ-4 Reaper. As this trend continues, UASs will become more capable and complex while also playing a larger role on the battlefield [56].

As the UAS mission further expands into aerial combat, designs will focus on better performance and larger, more complicated payloads. Such upgrades will have a negative correlation to the range and loiter times that have become expected. The requirement for a robust, AAR capability will become critical in order to efficiently and effectively use these new systems to project force around the world, directly effecting

the overall mission success of the USAF [6, 35, 56]. Further, the Air Force's 2035 operating concept calls for the use of "formations of uninhabited refueling aircraft...[that] enable refueling operations partway inside threat areas"[23].

Aerial refueling itself is a capability that came from the desire to maximize productivity. On Oct. 5, 1922 a world endurance record for a single flight was set with a flight time of 35 hours, 18 minutes, 30 seconds. The flight did not continue because of the limited amount of gasoline the Fokker T-2 could carry. Less than nine months later the first successful mid-air refueling took place as gasoline was passed between two DH-4B aircraft with a hose. Within another four months, the endurance record, as well as 15 other world records, were broken by a flight using 16 refueling contacts [1]. These monumental flights removed fuel capacity as a limiting factor, extending mission times to that of the physical endurance of the crew.

Since then, the use of aerial refueling has become a critical piece of the United States (US) Military's strategic plan for projecting power around the world. Two aerial refueling approaches are currently used by the military (hose-and-drogue and flying boom). The hose-and-drogue system (Figure 1a) is reminiscent of the original hose implementation discussed above. This method employs a flexible hose with a drogue (i.e. a small windsock for stabilization) that trails the tanker. The receiving platform inserts a probe into the hose making the connection required to begin the transfer of fuel.

In the 1950s, the Cold War led to a rise in long-range bombing missions requiring the need of a refueling system that could transfer larger amounts of fuel at a higher rate. A second approach was developed that implemented a flying boom based system (Figure 1b). Its rigid, telescoping tube uses an operator located on the tanker to extend and insert the fuel nozzle into a receptacle on the refueling aircraft. The refueling aircraft's pilot focuses primarily on station keeping. While the hose-and-

(a) Hose-and-drogue **(b)** Boom

Figure 1. Current US military aeriel refueling systems. All Marine Corps aircraft, Naval aircraft and Air Force helicopters use the hose-and-drogue method for aerial refueling. 96% of Air Force aircraft utilize the boom method for refueling.

drogue method can refuel multiple aircraft, it is only capable of transferring 1,500-2,000 pounds of fuel per minute per hose. The boom system is capable of transferring 6,000 pounds of fuel per minute. Currently, AF helicopters and all Navy and Marine Corps aircraft refuel using the hose-and-drogue, while 96% of Air Force aircraft are refuelable with the flying boom [4]. The development of the KC-46 Pegasus refueling platform has cemented the flying boom as the primary aerial refueling system for Air Force aircraft.

In addition to underscoring the importance of the flying boom system, the KC-46 has also led to a paradigm shift in AAR research with the integration of state-of-the-art visual sensors on the tanker platform. Rockwell Collins Remote Vision System (RVS) includes sensors, graphics subsystems and 3D displays to provide operators the depth cues necessary to safely and accurately control the boom during refueling operations [31]. While the new systems provide the foundation for more advanced refueling techniques, they do not currently provide relative positioning information about the trailing aircraft. The AAR mission requires this information to facilitate precise and continuous station keeping as at least two aircraft will make a mid-air connection while others fly in close proximity. The benefits of safe close formation flight extend beyond aerial refueling and may play a critical role in autonomous aerial

combat operations currently conducted by manned systems [57, 27, 58]. Thus, the aim of this thesis is to explore what capabilities may now be available with the inclusion of the RVS and determine how close we are to having the position information needed to facilitate a safe, robust AAR solution.

1.1 Problem Statement

The AAR problem requires a UAS to achieve a position proximate to a tanker (manned or unmanned) and safely maintain close formation flight. This effort investigates the ability of an electro-optical imaging system and SfM based algorithm to provide relative positioning information between a tanker and a receiver aircraft. Additionally, this thesis aims to characterize the performance of well known 3D rigid registration algorithms to provide empirical evidence supporting future AAR research and implementations.

1.2 Overview

This thesis is organized as follows: Chapter II introduces the mathematical notation used throughout this thesis and defines the relevant reference frames. It also provides an overview of subject areas necessary to support the algorithm as well as previous relative positioning and AAR research efforts. Chapter III presents preliminary experimentation used to characterize the performance of SfM in the AAR domain. Chapter IV introduces the methodology implemented in thesis by defining a mathematical representation of the problem, detailing the role of each stage pf the algorithm in solving the problem and presenting the experimental environment and data collection techniques used to test the hypothesis. Chapters V and VI present the relative navigation experiment and 3D rigid registration study conducted. Each chapter states the objectives, discusses the necessary assumptions and limitations, details

the experimental process and presents and analyzes the results. Finally Chapter VII concludes the thesis by drawing conclusions from both experiments and discussing their impacts on AAR research. This chapter also suggests areas for future work that leverage and build on the results of this thesis.

II. Background

This chapter provides an overview of the areas of study critical to the proposed solution as well as previous research focusing within the AAR domain. First, mathematical notation is developed for use throughout the document. Next, necessary reference frames are introduced followed by the mathematcial background required to develop the algorithms in Chapters IV and V. Finally, an overview of past AAR research is presented with a focus on relevent capabilities and performance.

2.1 Mathematical Notation

The following notation is used to represent mathematical concepts in this thesis:

Position: Positioning is the 6 degree of freedom (DOF) attitude of a rigid body in 3D space, including location (x, y, z) and orientation (roll, pitch, and yaw).

Scalars: Scalars are represented by an italicized lowercase letter (a).

Points: Points are represented by italicized, lowercase letters with a dot ($\dot{p}$).

Sets: Sets are represented by italicized, uppercase letters where set A=$\{A_1, A_2, \ldots, A_n\}$.

Mean: Means are represented by lowercase, italicized letters with a bar ($\bar{p}$).

Vectors: Vectors are represented by an italicized lowercase letter with an arrow above ($\vec{v}$). The translation vector is denoted $\vec{t}$.

Direction Cosine Matrix (DCM): DCMs are represented by the matrix R. A DCM that converts from a world orientation to a body orientation is denoted R^{body}_{world}.

Quaternions: Quaternions are denoted by a lowercase italicized q. A quaternion that converts a world frame to a body frame is denoted q^{body}_{world}.

Transformations Transformations consist of a rotation and a translation. While homogenius equations are not used in this thesis, transformations will be represented by Δ where $\Delta^y_x = [R|\vec{t}]^y_x = [R^y_x|\vec{t}^y_x]$.

2.2 Reference Frames

Reference frames are critical in expressing locations and orientation with reference to different objects. For this document the following references frames are utilized:

Tanker Frame: The tanker frame is an orthonormal basis in three dimensions, with its origin located at the centroid of the tanker aircraft with a nose, right wing, down orientation.

Receiver Frame: The receiver body frame is an orthonormal basis in three dimensions with its origin at the centroid of the airframe with a nose, right wing, down orientation.

Camera Frame: The camera frame is an orthogonal basis in three dimensions with its origin located at the camera origin with a nose, right wing, down orientation. It has a known translation from the tanker body frame with an approximate relative orientation of -35° pitch and 180° yaw.

Image Frame: The image frame is an orthogonal basis in three dimensions with a nose right wing, down orientation. It is equivalent to the camera frame at a specific moment in time.

Refueling Frame: The refueling frame is an orthonormal basis in three dimensions with a nose, right wing, down orientation. It has the same orientation and a known translation from the tanker frame.

Virtual Model Frame: The virtual model frame is an orthonormal basis in three dimensions with its origin at the centroid of the point cloud with a nose, right wing down orientation. Its origin is colocated with the camera frame with with the same orientation as the tanker frame.

SfM World Frame: The SfM frame is an orthonormal basis in threes dimensions with an arbitrary origin and orientation.

Vicon World Frame: The Vicon world frame is an orthonormal basis in three

dimensions with its origin on the floor at the center of the Vicon space with a north, east, down orientation.

2.3 3D Rigid Transformations

A coordinate transformation is the mapping of the positioning of an object in one frame to another. Rigid transformations ensure that the relative distances between a set of points are maintained after the transformation has been applied [12]. A rigid transformation can be modeled by the by the formula

$$o' = \vec{t} + Ro \tag{1}$$

where o is an object (point, vector, etc) being transformed to o' by a rotation, R, and a translation, $\vec{t}$. $\vec{t}$ represents a shift in the location or x, y and z directions. For a 3D translation this requires that Equation 1 be expanded to

$$\begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + R \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix}. \tag{2}$$

Given that a rigid transformation maintains the distances between all points, it also preserves the angles between segments ensuring the structure of the object is maintained throughout the operation which can be proven by the cosine law [12].

The rotation matrix, R, is a set of rotations around each axis that transform an object into the desired orientation, or roll, pitch and yaw. DCMs representing these rotations are determined according to Equation 3, inline with [61]. It should be noted that this operation is not commutative.

$$R = R_z R_y R_x \tag{3}$$

where

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\Theta_x & sin\Theta_x \\ 0 & -sin\Theta_x & cos\Theta_x \end{bmatrix} \quad R_y = \begin{bmatrix} cos\Theta_y & 0 & -sin\Theta_y \\ 0 & 1 & 0 \\ sin\Theta_y & 0 & cos\Theta_y \end{bmatrix} \quad R_z = \begin{bmatrix} cos\Theta_z & sin\Theta_z & 0 \\ -sin\Theta_z & cos\Theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Tranformations combine the R and $\vec{t}$ components in such a way that Equation 1 can be denoted $\Delta_o^{o'} = [R|\vec{t}]_o^{o'}$ [12]. Transformations can be combined by multiplying the rotations and adding the translation vectors after accounting for the new orientation as shown in Equation 4.

$$\Delta_o^{o''} = \Delta_{o'}^{o''} \Delta_o^{o'} = [R_{o'}^{o''} R_o^{o'} | \vec{t}_{o'}^{o''} + R_{o'}^{o''} \vec{t}_o^{o'}] \tag{4}$$

Additionally, the inverse transformation can be used to move the object back to the original frame by applying the transpose (or inverse due to orthogonality) of the rotation matrix and the negative translation as shown in Equation 5.

$$\Delta_o^{o''-1} = \Delta_{o''}^{o} = [R_o^{o''-1} | -\vec{t}_o^{o''}] = [R_{o''}^{o} | \vec{t}_{o''}^{o}] \tag{5}$$

Rotations can also be represented as a quaternion, the quotient of two directed lines in 3D space [30]. As defined in [29] a quaternion is a 4×1 vector consisting of a scalar part, s, and a vector part, $\vec{v}$. Unit quaternions simplify many rotation operations when compared to rotation matrices (or DCMs). The inverse of a quaternion, q^{-1}, reverses its transformation, converting back to the original orientation such that if $q = q_a^b$, then $q^{-1} = q_b^a$. Assuming unit quaternions, the quaternion inverse is identical to the quaternions conjugate q^* where

$$q^* = \begin{bmatrix} q_s \\ -q_x \\ -q_y \\ -q_z \end{bmatrix}.$$

Quaternions can be combined such that $q_a^b \otimes q_b^c = q_a^c$. However quaternion combinations are more complex than combining DCMs requiring the graßman exterior product defined as

$$q_1 \otimes q_2 = q_3 = \begin{bmatrix} q_{1q} \cdot q_{2s} - q_{1\vec{v}} \circ q_{2\vec{v}} \\ q_{1s} \cdot q_{2\vec{v}} + q_{2s} \cdot q_{1\vec{v}} + q_{1\vec{v}} \times q_{2\vec{v}} \end{bmatrix} \tag{6}$$

where $\circ$ represents a vector dot product and $\times$ represents a vector cross product [28, 15].

While DCMs and quaternions both transform between reference frame orientations they require fundamentally different operations. In order to conduct rigid transformations on 3D points with quaternions requires

$$\begin{bmatrix} 0 \\ \vec{v'} \end{bmatrix} = q \otimes \begin{bmatrix} 0 \\ \vec{v} \end{bmatrix} \otimes q^{-1} = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} \otimes \begin{bmatrix} 0 \\ \vec{v}_x \\ \vec{v}_y \\ \vec{v}_z \end{bmatrix} \otimes \begin{bmatrix} q_s \\ -q_x \\ -q_y \\ -q_z \end{bmatrix} \tag{7}$$

Note that the transformation is not completed by simpy using $q * \begin{bmatrix} 0 \\ \vec{v} \end{bmatrix}$ as when using a DCM in Equation 1. This is due to quaternions operating in a four dimensional (4D) space in which the vector part, $\vec{v}$, consists of imaginary numbers. In order to completely apply the transform, the vector must be converted back to the original 3D space. Multiplying by an additional q^{-1} does this. Each of the imaginary numbers

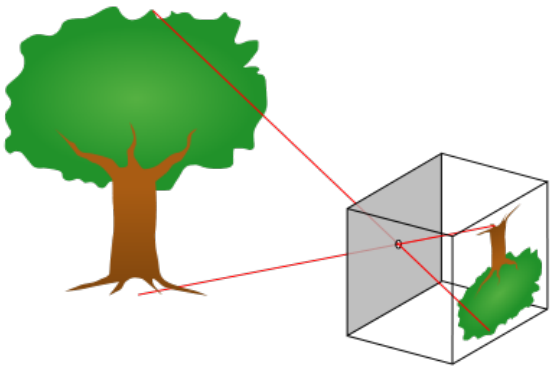


Figure 2. The pinhole camera model. The actual image is an inverse representation of the scene capture by a sensor. The focal length is the distance between the sensor and the lens, modeled as a pinhole. Reprinted as an element of the public domain [2].

are multiplied by a second imaginary number forcing the non-real values to become 1 because $i^2 = -1$. This also takes the scalar part of the quaternion back to zero ensuring the transformed point lies back in a 3D space.

2.4 Computer Vision

Computer vision is a field of study that aims to make useful decisions about real objects and scenes based on images [66]. Digital imagery has enabled every scene to be captured as a matrix, or possibly several matrices. Pixels in an image provide two main types of data, direction and color. Each pixel is assigned a set values describing its red, green and blue properties. Additionally, pixels are identified by their 2D location in the camera's vision plane. Each pixel is assigned a 2D vector identifying its location from the top left corner of the image, the image origin ($\dot{p}_i = \begin{bmatrix} x_p \\ y_p \end{bmatrix}$). A pinhole model, shown in Figure 2, is commonly used for camera systems where the actual image is an inverted representation of the scene being captured. Image inversions can be accounted for by modeling an equivalent image plane in front of the camera where the focal length, f, is the distance from the lens, modeled by the pinhole, to the sensor that physically captures the image. Once the image plane has

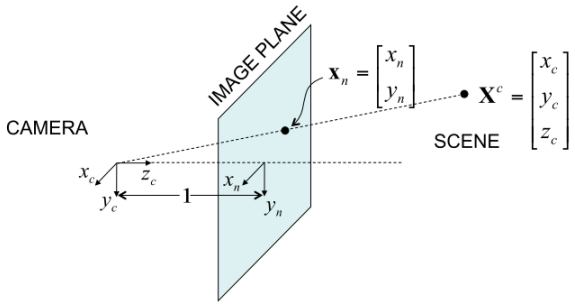


Figure 3. The image plane is a 2D model of the scene. The image plane provides 2D image coordinates for every direction in the physical scene.

been derived, each point in the scene can be assigned a corresponding direction in the camera frame (Figure 3). In many applications the focal length is unknown. Setting it to 1 allows the direction to be represented as a normalized vector, $\vec{p_i}=\begin{bmatrix} x_n \\ y_n \end{bmatrix}$, where the origin is the center of the image plane. The relationship between the normalized direction coordinates and the actual three dimensional location of the point in the camera frame can be modeled by

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_c/z_c \\ y_c/z_c \end{bmatrix} \tag{8}$$

where $\begin{bmatrix} x \\ y \\ z \end{bmatrix}_c$ is the 3D location in the camera frame. The relationship between the pixel and normalized coordinates of a point in the image (Figure 4) is defined as

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} f_c & 0 \\ 0 & f_c \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} n_x/2 \\ n_y/2 \end{bmatrix} \tag{9}$$

where f_c is the focal length and n_i is the number of pixels in the i dimension. Note, that if the focal length is set to 1, the relationship is the offset between the origins of

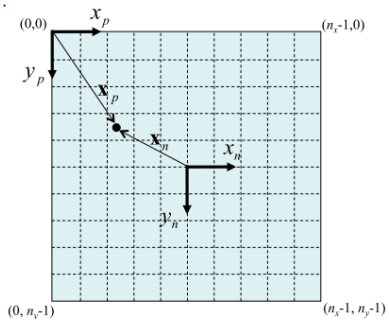


Figure 4. The relationship between pixel and normalized image coordinates. The pixel coordinate plane has its origin at the top left corner of the image plane while the normalized coordinate plane has its origin in the center of the image plane.

each coordinate plane. In order to fully determine the 3D location of an object the camera calibration matrix, K, must be known. K models the intrinsic parameters of the camera, such as distortion. Given K, the pixel location of an object is related to the actual three dimensional location of the object in the camera frame by

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K[I_{3x3}|0_{3x1}] \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{10}$$

where $[I_{3x3}|0_{3x1}]$ is a 3x3 identity matrix combined with a 3x1 0 matrix, $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.
If the position of the camera and the object are known in some arbitrary world frame (such as in Figure 5), then a DCM can be used to identify the location of the object in the camera frame with Equation 14. Therefore, the relation between the pixel

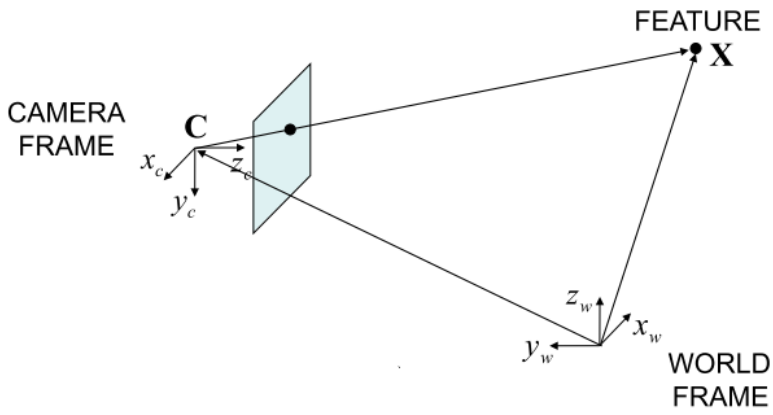


Figure 5. The positioning of an object in the camera and world frames. Given the positions of the camera and a feature in the world frame, the position of the feature in the camera frame can be determined.

location of an object and its arbitrary world location is represented by

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = K[R_w^c | - R_w^c \vec{c}_w] \begin{bmatrix} obj_x \\ obj_y \\ obj_z \end{bmatrix}_w . \tag{11}$$

In this equation $\vec{c}_w$ is the location of the camera in the world frame while $-R_w^c \vec{c}_w$ is the origin of the world frame expressed in the camera frame, or a translation vector $\vec{t}$. $K\left[R|\vec{t}\right]$ represents the camera matrix, P, that will translate any three dimensional point in the scene into its corresponding pixel location. All of the calculations considered above can be derived from the information provided by a single image. However, scene ambiguity remains due to the fact that a single image can not directly measure scale.

In the same way humans use depth perception, epipolar geometry can be used on a set of two offset images to solve for scale. If two images of the same object are taken from different view points the 3D location of the object can be derived with the same principles discussed above using the epipolar constraint. The epipolar constraint is the relationship between a point correspondence, or the location of a single object in

the two separate images , x_n and x'_n, modeled by

$$x'_n T R x_n = 0 \tag{12}$$

T contains the translation values in matrix form such that $T = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$ from the translation vector $\vec{t}$. TR is known as the essential matrix, E. By setting the camera matrix of the first camera to $P = K[I|0]$ the matrix of the second camera becomes $P' = K'[R|\vec{t}]$. Given the relative locations of the two cameras ($\vec{t_w}$), the bottom line of the triangle in Figure 6 is known as well as the angles from each camera origin to a matched point, X. The 3D location of the point in each of the camera frames can then be determined using basic trigonometric principles. While E utilizes the camera coordinates, or extrinsic parameters R and $\vec{t}$, the fundamental matrix, F, accomplishes the same tasks using image (or pixel) coordinates by utilizing intrinsic camera parameters. F tells how pixels (points) in one image are related to the epipolar lines in the other image. While the essential matrix applies the epipolar constraint as shown in Equation 19, the fundamental matrix applies the epipolar constraint such that $x'_p F x_p = 0$. The relationship between E and F is simply K such that

$$E = K'^T F K \tag{13}$$

2.5 Feature Detection

In computer vision, feature detection is the process of identifying an image location that is distinct in some manner. While no single definition exists for a feature some important characteristics include uniqueness from other features in an image,

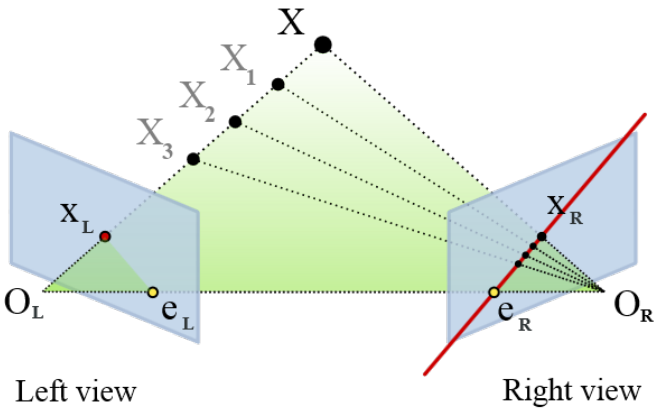


Figure 6. The foundation of epipolar geometry. A feature identified in one image corresponds to a location contained on a single line in a second image of the same scene. Reprinted as an element of the public domain [53].

appearing in multiple images and corresponding to a stationary object in the real world. These characteristics ensure the feature is identifiable in many images of the same scene. A good feature is able to be identified based on salient characteristics that are consistent through multiple images. While the feature itself should be robust to different views its descriptor, the way it is represented and stored would ideally be deduced in such a way that is scale, rotation and affine invariant. Determined by the needs of the particular application, many types of features are used including corners, lines, edges, faces, bar codes and point features.

One of the most prominent point features is known as Scale Invariant Feature Transform (SIFT) [47]. While wide ranging research has led to the development of many point features, SIFT is considered one of the more robust approaches. Using a blurring technique SIFT detects locations that can be repeatedly identified from multiple views of the same scene. With a difference-of-Gaussian approach, extrema are found that represent major differences between two nearby scales of blurring. Once locations are identified they are filtered to ensure chosen keypoints have a high contrast with neighboring pixels and are not poorly localized along an edge. In order to be able to continually identify the keypoints in subsequent images, SIFT uses a

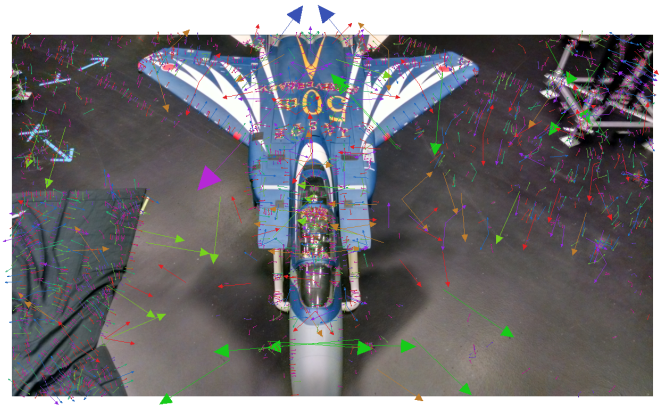


Figure 7. Examples of keypoints identified in an image. Each arrow represents a keypoint in the image. The keypoints are uniquely charaterized by the direction and length of their corresponding arrow.

descriptor that is based on the orientation of local image gradient directions. The differences in pixel coloring surrounding each keypoint are quantified by the direction of color change (direction of the arrow) and then weighted by the magnitude of the color change (the length of the arrow), as shown in Figure 7.

The descriptor used to store and identify each keypoint is derived by creating a vector containing the value of the 128 orientations surrounding it. This creates a descriptor that is invariant to image location, scale and orientation. Normalizing the vector will provide illumination invariance by removing the effects of a change in variable that is constant across the set of pixels. Finally, in order to reduce the risk of non-linear illumination changes that have different effects on 3D surfaces the influence of large gradient magnitudes is thresholded to a small value. This puts a larger emphasis on the distribution of orientations rather than matching magnitudes of large gradients. Figure 8 illustrates an image of a model F-15 and its corresponding SIFT features. Using robust features, such as SIFT, ensure applications that require feature matches, or correspondences, throughout a set of images are able to continuously provide stable information.

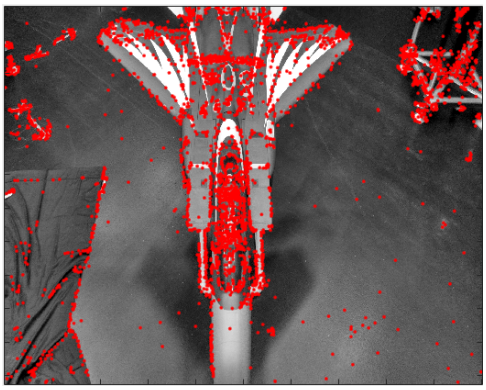


Figure 8. SIFT fetures determined from the keypoints in Figure 7. Each red dot is a SIFT feature that was identified as a robust feature that can be recognized from many different perspectives.

2.6 Structure from Motion

SfM is the estimation of 3D structure from 2D information. The SfM concept was first presented in [76] where Ullman's Theorem states:"Given three distinct orthographic projections of four non-coplanar points in a rigid configuration, the structure and motion compatible with the three views are uniquely determined up to a reflection about the image plane", assuming correspondences between projections have been established. Modern SfM implementations are based on the combination of three major areas; feature detection and matching, multiview geometry, and bundle adjustment. Similar to the approach discussed in Section 2.7, feature detection and matching is used to generate a set of correspondences between multiple images. It is common to filter the correspondences to reduce the likelihood of false correspondences that have a negative effect on the accuracy of the model. The most prominent point filtering algorithm is Random Sample Consensus (RANSAC) which is capable of filtering data containing a significant percentage of errors [21]. Once point correspondences have been established, the epipolar geometry techniques previously discussed can be extended to estimate object locations in more than two images using multiview geometry.

While two view epipolar geometry relies on the fundamental and essential matrices, three views require the trifocal tensor (τ_i^{jk}), a 3x3x3 array of numbers relating correspondences [34]. Like the fundamental matrix, the trifocal tensor is determined by the three camera matrices. The relationship between three views uses a correspondence between two lines and a point, $\dot{x} \leftrightarrow l' \leftrightarrow l''$. A point, $\dot{x}$, in one of the images corresponds to points,$\dot{x'}$ and $\dot{x''}$, in the other two images that reside on single lines, l' and l'', in those images. The relationship between between the feature coordinates in these three images can then be modeled by:

$$\sum_{ijk} x^i l'_j l''_k \tau_i^{jk} = 0 \tag{14}$$

While this relationship provides a single linear relationship between the 27 elements of a tensor, point correspondences, $x \leftrightarrow x' \leftrightarrow x''$, provide more equations. This is due to the uncertainty that is removed by knowing the exact locations of the feature in all three images instead of only knowing its exact location in a single image. It is only necessary to generate seven point correspondences to derive the trifocal tensor linearly, while a minimum of six can be used with non-linear methods [34]. Three-view reconstruction enables the use of lines as well as point correspondences to compute the projective reconstruction, avoiding unstable configurations possible in two-view reconstruction. Given four views, a quadrifocal tensor can be computed, though it is much more difficult to implement. Other techniques are possible in specific circumstances or given different assumptions, such as the use of an affine camera model. However, reconstruction of greater than three views are typically completed piece by piece on a sequence of images with two- and three-view techniques.

When implementing a piece by piece approach, a common methodology is bundle adjustment (BA) [68]. BA is an iterative method that refines a reconstruction to produce jointly optimal 3D structure and image parameter estimates. As new in-

formation (images) is added to the sequence, BA continually adjusts all feature and camera positions under consideration. An initial sparse reconstruction is established using two or three images as discussed above. Another image is added to the sequence and its intrinsic and extrinsic parameters are estimated from those of previous images. Features present in the new image, matching those in previous images are compared by pixel location. The new image's parameters are then refined by minimizing the re-projection error between the estimated pixel locations and those observed in the image itself. This process is repeated iteratively for each new image added to the sequence [68]. The reprojection error between pixel locations can be modeled by

$$\sum_{a=1}^{y}\sum_{b=1}^{z} = x_{a,b}||q_{a,b} - P(\theta_a, p_b)|| \tag{15}$$

where θ is the camera model defining the estimated position and focal length of the new image, p is the 3D position of each feature within a image, and $P(\theta, p)$ is the predicted pixel location of the features. q is the observed pixel location of each feature. y is the total number of images under consideration in the BA and z is the number of features in the new image. If feature b is contained in image a, $x_{a,b}$ is one, otherwise it is zero. BA aims to minimize the right side of Equation 15 by refining the camera models as well as feature locations.

This minimization is a non-linear least squares problem typically solved with the Lavenberg-Marquadt (LM) algorithm. Using the gradient of the error equation for the current estimate, LM iteratively adjusts parameters by a certain delta. While it only finds local minimums, if only one minimum exists it can be found with poor initialization [26].

The SfM steps discussed above and much of the practical implementations of SfM focus around projective reconstruction. This method utilizes uncalibrated cameras and therefore must estimate camera parameters such as focal length, distortion

coefficients and the geometric center of the images. This causes distortion in the reconstruction which appear differently than the objects captured in a scene of the Euclidean world viewed by humans. If fully calibrated cameras are used, meaning the intrinsic parameters are known, ambiguities can be removed and a Euclidean reconstruction can be accomplished [34].

The ability to accurately estimate camera parameters in order to establish relative positioning between an object and an image makes SfM a valuable tool for navigation as well as reconstruction. Much work has focused on the use of SfM for this purpose including the use of calibrated images for the navigation of mobile vehicles [32, 33], the use of uncalibrated images in batches [50], and the use of uncalibrated images by sequential update [7, 72].

As SfM has matured, its use has spread to other applications outside of 3D reconstruction, such as the navigation implementations discussed above. However, a major drawback to SfM is its time complexity which is commonly accepted to be $O(n^4)$ for n images [16]. While a combination of discrete and continuous optimization has been shown to reduce the complexity to $O(n^3)$, the use of SfM in real-time applications such as AAR requires a much more efficient implementation [16]. Wu suggest that the efficiency of SfM can be dramatically increased by optimizing each piece of the algorithm [82]. Feature matching is one of the most expensive steps in SfM, requiring $O(n^2)$ time for n images to complete a full pairwise matching. However, for large data sets a full pairwise matching may not be necessary. Rather, a subset of feature matches could reduce the time complexity to $O(n)$. An approach to achieve further speed-up is to use preemptive feature matching in an effort to robustly and efficiently identify only good matches, shown to be as little as 2-25% in large datasets [82]. This is accomplished by sorting the SIFT features of each image into decreasing scale order, generating a subset of images that need to be matched and then checking image

compatibility. This check consist of matching the first h, parameter for subset size, features of two images. If the number of matches is below a defined threshold the two images are not considered together. If the number of matches exceeds the threshold regular matching and geometry estimation are done. This approach can reduce the pairs of image matches by up to 95% while still recovering sufficient good matches. In an effort to minimize the amount of costly BA operations, a geometric sequence is used. A full BA is only performed when the size of a model increases relatively by a certain ratio, r, causing the full BA cost to become $O(\frac{n}{r})$ as derived by

$$\sum_{t}^{\infty} T_{BA}\left(\frac{n}{(1+r)^t}\right) = O\left(\sum_{t}^{\infty} \frac{n}{(1+r)^t}\right) = O(\frac{n}{r}). \tag{16}$$

As the model grows, full BAs are conducted less, however, accuracy does not suffer because full BAs always improve more for the parts with large errors. Accumulated errors are minimized by consistently running partial bundle adjustment (pBA) on a set number of added images, requiring only $O(1)$ time each [82]. The total time spent on full and partial BAs is reduced to $O(n)$.

A third optimization step is re-triangulation. Drift occurs in point and image position estimates due to the accumulated loss of correct correspondences. Additional correspondences are identified by increasing the threshold for reprojection errors between image pairs that produce few feature matches. Using a similar approach to the one modeled in Equation 16, re-triangulation is achieved for $r = 25$, or when the number of images is increased by 25%. The re-triangulation is then followed by a full BA, still maintaining $O(n)$ time. This approach was used in the development of the visual Structure from Motion (vSfM) application, an open source tool and the application utilized for SfM in this thesis. Figure 9 illustrates a 3D reconstruction completed by vSfM.

The points outside of the camera circle are features that were identified and

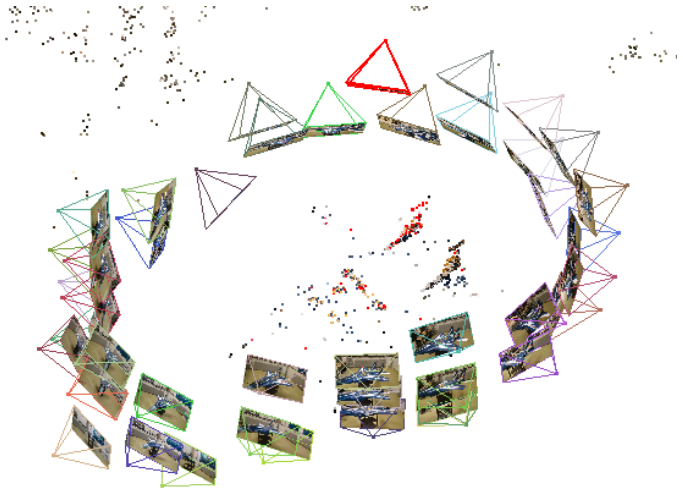
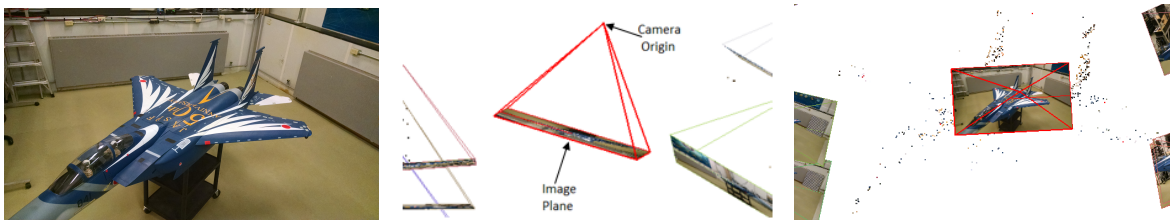


Figure 9. A 3D reconstruction generated by vSfM. A SfM 3D reconstruction showing the rigid objects in a scene and the estimate position in of the images used to generate the reconstruction.



(a) Original Image **(b)** SfM Camera Model **(c)** Reconstruction Location

Figure 10. An image, its SfM representation, and its estimated positioning. The vSfM representation of an image includes the estimated camera origin with an image plane containing the original image. The camera representation is placed at its estimated position relative to the scene.

matched in the background of each image showing the 3D reconstruction of the entire lab. The red points represent features identified in the red image. Figure 9 also clearly illustrates the ability of SfM to produce relative positioning estimates for each image. When operating vSfM the images are represented as four sided pyramids where the base is the image plane and the point is the estimated camera origin (Figure 10).

A major limitation of SfM is the assumption that the scene being reconstructed is static with only the camera being in motion. Additional research has produced several attempts at solving the dynamic Structure from Motion (dSfM) problem where multiple structures in the scene have different motions relative to the camera [64, 74,

22]. While such techniques may be necessary for the AAR domain, they are beyond the scope of this work.

The following terminology is used when discussing SfM in this thesis. The SfM model is the underlying mathematical formula developed to estimate3 the location of point correspondences and image positioning. Reconstructions are the visible points generated that represent the estimated locations of correspondences in the arbitrary SfM frame. Image position estimates are the location and orientation of images in the arbitrary SfM frame. The term initial is used to reference the first model,reconstruction and image position estimates produced by SfM on a set of images. The term quality is used to describe a model, reconstruction, and position estimates that appear to accurately represent the scene and relative image locations.

2.7 Principal Component Analysis

Originally proposed by Pearson [59] and independently presented by Hotelling [36, 37], PCA is a concept regularly applied in statistical data analysis for classification and compression. It is based on the fact that the direction of largest variance corresponds to the largest eigenvector of a set of data [8]. The principle components identify the variables of greatest variance within the set of data [67]. The principal components are determined by first computing a covariance matrix, C, which captures the correlation between dimensions. While variance typically refers to the spread in of data in a single dimension the covariance models described how two dimensions vary together. The covariance between two dimensions, x and y, can be modeled as

$$\sigma_{xy}^2 = (x_i y_i). \tag{17}$$

This equation provides two important pieces of data. If $\sigma^2_{xy} = 0$, then x and y are entirely uncorrelated. If $\sigma^2_{xy} = \sigma^2_{xx}$, then x equals y. If the two sets of dimensional data are converted into row vectors, such that $\vec{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$, Equation (17) can be represented as the dot product matrix computation

$$\sigma^2_{xy} = \frac{\vec{x}\vec{y}^T}{n-1}. \tag{18}$$

This can be generalized to include additional dimensions by using matrix P to represent a set of data with n variables such that

$$P = \begin{bmatrix} \vec{x} \\ \vec{y} \\ \vec{v_3} \\ \vdots \\ \vec{v_n} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \\ v_{31} & v_{32} & \dots & v_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nn} \end{bmatrix}$$

where each row corresponds to the data of a particular variable while each column corresponds to a particular measurement. C can now be generated for data in three dimensions by

$$C_P = \frac{PP^T}{n-1} = \begin{bmatrix} \sigma^2_{11} & \sigma^2_{12} & \sigma^2_{13} \\ \sigma^2_{12} & \sigma^2_{22} & \sigma^2_{23} \\ \sigma^2_{13} & \sigma^2_{23} & \sigma^2_{33} \end{bmatrix}. \tag{19}$$

The diagonal terms of C are the variance within a particular variable while the off-diagonal terms are the covariance between two variables. Therefore, C describes all relationships between pairs of variables within a set of data. C is a special matrix that is square, real and symmetric.

It should be noted why an $n-1$ term is used to compute the covariance matrix. Known as Bessel's correction, in statistical applications this increases the mean squared error of the estimate by attempting to correct a bias in the population vari-

ance [60]. The true population variance and covariance information is unknown, thus the information derived from the data set is used to compute the residual values. Due to the fact that a model, instead of the true value, is used to estimate the residual data a degree of freedom must be removed from the residual vector to account for any bias.

The eigenvectors of C can be derived using singular value decomposition (SVD) producing

$$C = U\Sigma V^T. \tag{20}$$

U is an orthogonal matrix composed of the eigenvectors of CC^T. V is also an orthogonal matrix but is composed of the eigenvectors of C^TC. Σ contains the eigenvalues corresponding to the eigenvectors of both CC^T and C^TC. Due to C being square, real and symmetric U=V producing

$$C = V\Sigma V^T. \tag{21}$$

In this way eigenvectors contained in V represent the principle components (i.e. axes of greatest variance) of P. While the eigenvectors and eigenvalues of a set of data provide much useful information, for the context of this thesis, their values are the only area of interests.

2.8 SVD Registration Algorithm

Many algorithms have been developed to solve the 3D rigid registration problem. This work investigates the performance of three approaches, including SVD, PCA, and Iterative Closest Points (ICP). The SVD approach attempts to minimize the sum of Euclidean distances between known correspondences in two point clouds using the cross-covariance matrix. ICP is an iterative SVD approach that utilizes

the same cross-covariance matrix, though it estimates correspondences prior to each iteration enabling it to disregard outliers and improve upon previous estimates [8]. PCA regstrations are based on the underlying geometry of the individual point clouds providing a faster solution as it does not require the identification of correspondences.

The SVD approach is a solution to the classical least-squares problem of fitting two 3D point sets [5]. Given two corresponding 3D point sets P and P' where $\dot{p}_i$ and $\dot{p'}_i$ are points represented as 3x1 column matrices then

$$\dot{p'}_i = R\dot{p}_i + \vec{t} + N_i \tag{22}$$

where R is the rotation matrix, $\vec{t}$ is the translation vector and N_i is a noise vector. The solution is found by minimizing

$$\Sigma^2 = \sum_{i=1}^{N} ||\dot{p'}_i - (R\dot{p}_i + \vec{t})||^2. \tag{23}$$

It can be shown that if $\hat{R}$ and $\hat{t}$ are the solutions to Equation 22 then they can be decoupled reducing Equation 23 to

$$\Sigma^2 = \sum_{i=1}^{N} ||\dot{q}_i - (R\dot{q}_i)||^2. \tag{24}$$

where $\dot{q}_i = \dot{p}_i - \bar{p}$ for $\bar{p} = \frac{1}{N}\sum_{i=1}^{N} \dot{p}_i$ [38].

The problem can then be solved in two parts. First, determine the R that minimizes the sum squared error (Equation 24) and then compute the translation by

$$\vec{t} = p - Rp. \tag{25}$$

Once Q and Q' have been determined the 3x3 cross-covariance matrix can be calcu-

lated by

$$M_x = \sum_{i=1}^{N} QQ'^T. \tag{26}$$

The SVD of M_x will produce

$$M_x = U\Lambda V^T \tag{27}$$

where

$$R = VU^T. \tag{28}$$

If $det(R) = +1$, then R will minimize Equation 24. However, if $det(R) = -1$, the solution has failed due to the point clouds being a reflection of each other. In this case, the desired rotation can then be found by

$$R = V'U^T \tag{29}$$

where $V' = \Big[v_1, v_2, -v_3\Big]$.

2.9 Iterative Closest Points Algorithm

Like SVD, ICP also aims to minimize sum squared error (Equation 23) by analyzing the cross-covariance between two sets of corresponding points [10]. However, ICP uses a more elaborate approach that utilizes quaternions. It is assumed that point correspondences between two point clouds are not known and must be estimated prior to each minimization. Therefore an iterative approach is utilized to continually improve alignment.

Each iteration of ICP begins with a nearest neighbor search between the two point clouds, P_1 and P_2. The closest P_2 point is then stored in a the point cloud, P_{match}, where its index is the same as its corresponding point in P_1. The centroid, or mean,

of each point cloud is then determined by

$$\bar{p} = \frac{1}{N}\sum_{i=1}^{N} \dot{p}_i \tag{30}$$

Next the ICP algorithm determines the cross-covariance of the two point clouds by

$$\Sigma_{P_1 P_{\text{match}}} = \frac{1}{N}\sum_{i=1}^{N} (P_1 P_{\text{match}}^T) - \bar{p}_{\text{match}} \bar{p_1}^T \tag{31}$$

where $\Sigma_{P_1 P_{\text{match}}}$ is the cross-covariance matrix. Next a Q matrix is made from $\Sigma_{P_1 P_{\text{match}}}$, using auxiliary matrix A and auxiliary vector $\vec{\delta}$ where $A = \Sigma_{P_1 P_{\text{match}}} - \Sigma_{P_1 P_{\text{match}}}^T$ and $\vec{\delta} = \begin{bmatrix} A_{23} \\ A_{31} \\ A_{12} \end{bmatrix}$. Given these two relations Q can be defined as

$$Q(\Sigma_{P_1 P_{\text{match}}}) = \begin{bmatrix} trace(\Sigma_{P_1 P_{\text{match}}}) & \vec{\delta}^T \\ \vec{\delta} & \Sigma_{P_1 P_{\text{match}}} + \Sigma_{P_1 P_{\text{match}}}^T - trace(\Sigma_{P_1 P_{\text{match}}}) I \end{bmatrix} \tag{32}$$

The optimal quaternion rotation, q, is the eigenvector corresponding to the largest eigenvalue of Q. The optimal translation can then be calculated as the difference between the means of the sets corrected for the rotation as shown in Equation (33).

$$\vec{t} = \bar{p} - R(q)\bar{p}_m \tag{33}$$

The rotation and translation can then be applied to P using Equation (25). This transformed set is then used to start a new iteration of the algorithm. The algorithm will continue until a pre-determined parameter is achieved. Typically this parameter is either a minimum change in distance between the two point clouds, a maximum number of iterations or a maximum amount of time.

2.10 Previous AAR Efforts

2.10.1 GPS.

GPS technology has led to many advances in positioning and navigation capabilities. Its success has enabled it to support most critical areas of the USAF mission and, consequently, has led to a heavy reliance on its avaiability and reliability. Given GPS's widespread integration, specifically in military weapon systems, it became the primary focus in initial AAR efforts.

In 1994, GPS was used in a flight test investigating relative positioning of two aircraft [44]. The approach placed two single frequency GPS receivers on each platform, one on a 5 meter tail boom that was subjected to buffeting and a second mounted in a more stable position on the fuselage. Using a robust carrier smoothing approach, the system was able to achieve submeter accuracy Spherical Error Probable (SEP) with a distance of 1 kilometer between aircraft. The use of GPS for autonomous station keeping during close proximity flight was first demonstrated in 2002 by the National Aeronautics and Space Administration (NASA)[3]. Two F-18 Hornets conducted a concept flight using semi-autonomous techniques. However, the aircraft were never closer than 56 feet throughout the flight.

In 2006 research was conducted as a joint effort between Air Force Research Laboratory (AFRL), the Air Force Test Piot school, and Air Force Institute of Technology (AFIT) that utilized a more precise form of GPS known as carrier-phase differential Global Positioning System (DGPS) to determine relative positioning in real-time [71]. Using a wireless data-link, the receiver platform used the tanker position as a local truth reference. The approach developed a new ambiguity resolution technique based on special characteristics of the carrier-phase residuals. The system successfully demonstrated automated close formation flight, culminating in more than 11 hours of flight with a mean radial spherical error of 3.3 centimeters. Simul-

taneously, additional research developed a flight controller that maintained relative position [62]. This system successfully demonstrated GPS as a valid approach for accurate relative positioning during close formation flight.

While GPS has proven to be a successful approach for relative positioning in close formation flight, it is not able to meet the stringent fault-free integrity and accuracy requirement of piloted aerial refueling. TO thie end research has been conducted on the effect of sky-blockage caused by a large tanker flying above the receiving aircraft during refueling missions [41], but with the emerging need for operations in GPS denied environments, the AFRL is investigating alternative techniques for AAR including inertial navigation, LiDAR, and optical solutions [79].

2.10.2 LiDAR.

LiDAR has historically aided navigation by determining surrounding environments [65]. It is useful in providing information in unpredictable environments such as object tracking and plane detection [69]. However, this approach assumes a predictable environment free of objects in close proximity to the subject. While AAR requires at least one other plane to fly in close proximity, the environment is much more predictable because all planes involved are known before hand. This enables the LiDAR system to search for specific objects and facilitate a more accurate position solution. In 2012, research was completed at AFIT on the use of LiDAR for AAR position estimation [17]. During flight testing, a LiDAR system was located in the nose cone of the receiver aircraft, providing a view fo the bottom of the tanker. Using a high quality Inertial Navigation System (INS) to provide the attitude of each aircraft, a modified ICP algorithm was used to fit the LiDAR measurements to a model of the tanker aircraft. A second algorithm predicts LiDAR scans and compares them to actual measurements while perturbing the estimated location of the

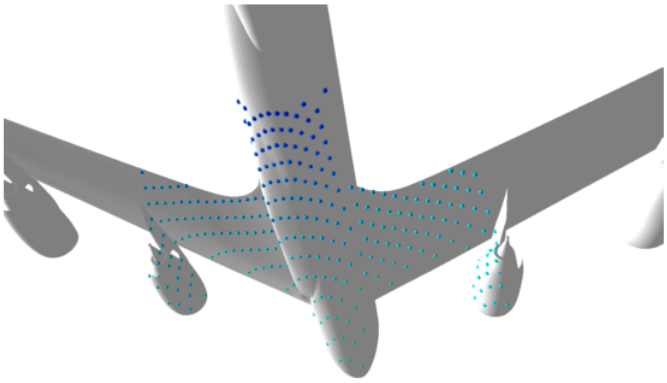


Figure 11. A Custom LiDAR scan on the tanker. Previous AAR research utilized a LiDAR system located on the receiver aircraft. The blue shperes are measurement points returned bu the simulated LiDAR. Reprinted as an element o the public domain [17].

tanker. This solution was able to produce attitude and position estimates resulting in root-mean-square (RMS) of 52 centimeters with Euler under 25 milliradians. However the implementation could not perform in real-time averaging 12 seconds per position estimate for both algorithms.

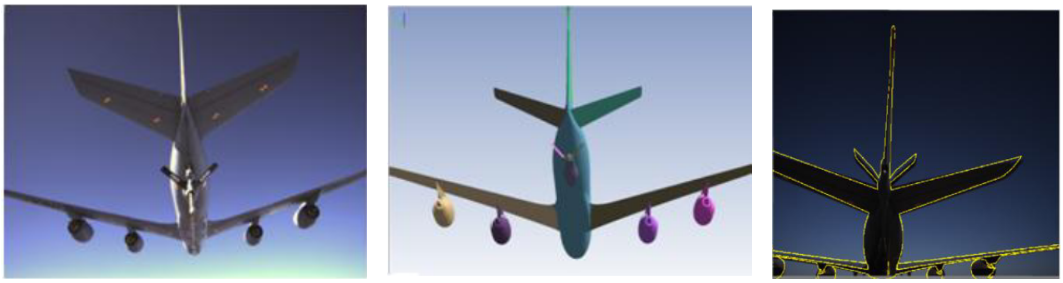
2.10.3 Computer Vision.

Multiple sensor fusion approaches for vision aided GPS are proposed in [24, 48, 49]. The machine vision portion of the solutions is divided into three distinct sections (feature extraction, detection and labeling, and pose estimation) with multiple methods for each being discussed. Position and orientation estimates are determined by using corner detectors to find features in a 2D image of the tanker. Next, known feature locations are projected into the 2D image plane and aligned to the detected features to determine the relative position. An extended kalman filter is used to combine the machine vision estimate with the GPS data.

A popular technique in vision navigation is tracking specific known features through frames. Such features correlate to known positions on a stored model of the aircraft. The sensed location of these features provide an estimated relative position and orien-

tation of the aircraft when compared to their locations on the model. This approach was implemented using corner detection in [70]. After extracting the features, both structural and painted corners, 3D pointing vectors were calculated to their locations on the model. The vectors were then passed to a navigation algorithm that utilized a Kalman filter and data-linked INS to determine the relative location and orientation. Flight testing of the system showed that each frame could produce at least 12 useful measurements with a detection error of 2.7 pixels for a 12.5 millimeter lens and 1.95 pixels for a 25 millimeter lens. While the accuracy of the system was heavily dependent on accurate navigation updates, the dominant source of error was the estimation of features on the tanker. A similar approach applied red markers to the tanker platform that were identified and matched to known locations by the vision system located on the receiving aircraft [25].

A slightly different vision based tracking system for aerial refueling is the Vision-Based Sensor and Navigation System (VisNav) [77]. Designed for a hose-and-drogue system, multiple infra-red light emitting diodes (LED) were fixed to the drogue chute with a known orientation. The vision system, located on the receiving aircraft, established a data-link based connection with electronics mounted on the drogue in order to activate each LED in a precise, predetermined sequence. The sensor on the receiving aircraft could then detect and distinguish each LED, determining its relative distance and orientation. While this approach is seemingly robust, it relies on the ability to establish a reliable communication link and would require retrofitting every aircraft with a refueling capability as well as adding the LED system to every tanker. Another technique for vision navigation, that does not utilize feature tracking, instead generates rendered or pseudo-images of the tanker aircraft [80]. This approach determines the relative positions of the aircraft by perturbing the relative attitude and position states until a maximum correspondence with actual sensor imagery is

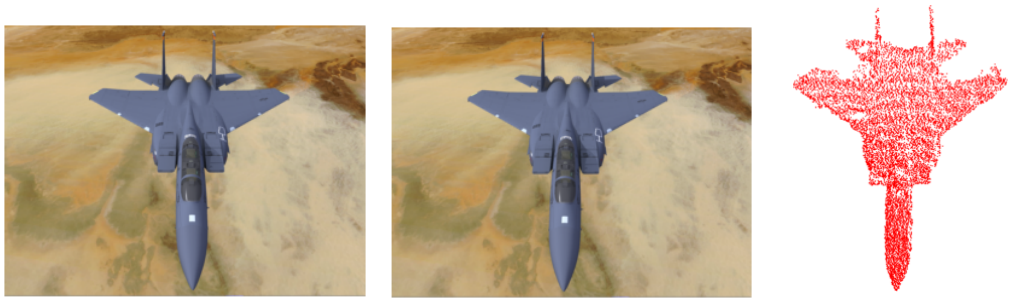


(a) EO Sensor Image **(b)** Rendered Image **(c)** Model Overlay

Figure 12. Flight test imagery from previous monocular AAR effort. Previous monocular research determined relative positioning state between a tanker and receiver by perturbing the relative position and attitude state and generating rendered images until maximum correspondence with actual sensor imagery is achieve. Reprinted as an element o the public domain [11].

achieved (Figure 13). This image rendering technique has been extensively studied for multiple applications including machine vision [13], medical image registration [63], object detection and pose estimation [46, 55]. Flight testing of this approach showed it to be a viable approach to precision navigation for close formation flight [11]. 95% relative navigation accuracies were achieved on the order of 35 centimeter within a 20meters (m) range. However, the periodic divergences on the order of 50 centimeter were observed possibly due to algorithm robustness issues where correspondence minimization locked onto local minimums.

The vision navigation research discussed above all implement a system based on a single vision sensor (i.e. monocular vision). In an effort to leverage the new vision systems organic to the KC-46A platform, research was conducted using a stereo vision approach [81]. Similar to the way humans perceive depth, stereo vision systems obtain structural information about the scene from the geometric relationship of corresponding features seen by both cameras. By calculating the position of the trailing aircraft with respect to where it is expected to be, a small amount of information can be passed to the receiver describing its positioning. An autopilot can then maneuver the platform into the desired refueling position. After generating a point cloud from



(a) Left Image **(b)** Right Image **(c)** Point Cloud

Figure 13. A set of stereo images and the point cloud they generate. Previous computer vision research generated point clouds by projecting pixels along there epipolar lines in a set of stereo images. Reprinted as an element of the public domain [81].

the stereo images the algorithm fits it to a virtual model by applying the same ICP algorithm used in [17]. Experimental results showed that the accuracy increased significantly as the trailing aircraft approached the tanker. Within 17m, an accuracy of +/- 10 centimeter was reached. The algorithm, however, was not able to support real-time operations. Generation of the point cloud alone requires more than 1.3 seconds. The ICP process, after being adjusted by a constant offset, took 5.22 minutes to reach the minimum movement threshold (running 63 iterations) on a single set of images. Speed-up was achieved by reducing the size of the point cloud, however this had a negative effect on the accuracy.

III. Experimental Environment & Preliminary Experimentation

This chapter discusses the experimental environment developed for this thesis and the initial experimentation used to better understand the performance of SfM in the AAR domain. First the development of a 1:7 scale AAR environment is presented. Next, the test data collection process is discussed followed by the techniques for acquiring truth data. Finally, five preliminary experiments are presented and important findings within the context of the AAR domain are discussed.

3.1 Experimental Environment and Data Collection

The use of a 1:7 scale modeled environment facilitated the collection of realistic test data as well as accurate truth data. This section describes the environment and outlines the data collection techniques and system used to generate truth data.

3.1.1 1:7 Scale Experimental Environment.

For testing purposes, a 1:7 scale environment served as the world frame. Within this environment, a 1:7 scale F-15 model was used as the receiving aircraft. A pair of cameras with fixed focal lengths were used to model the imaging system. The camera frames were separated in the y direction by 1:7 of the distance believed to be used on current tanker imaging systems. The cameras were connected to the same network via a switch using an Ethernet chord. A central computer, on the same network, was used to control image capture and store the images.

3.1.2 Data Collection.

The data collection consisted of 15 runs, each modeling the receiver's approach to the rear of the tanker. While the aircraft model remained stationary, the imaging

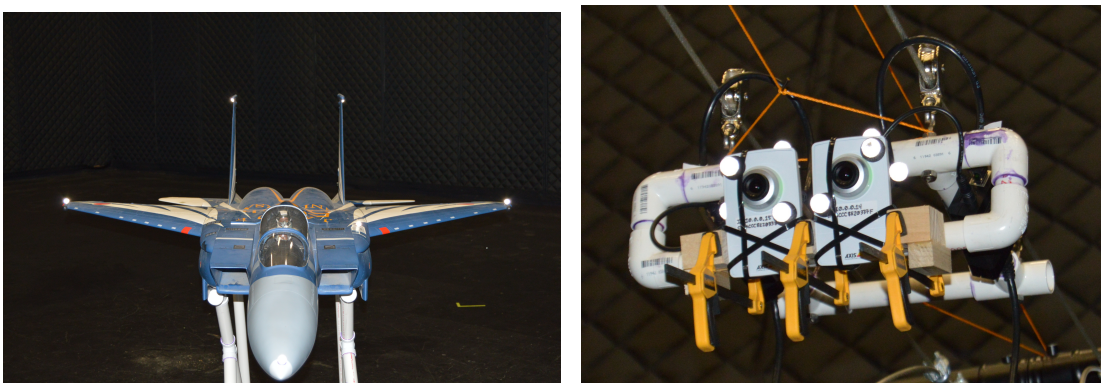
(a) Side View **(b)** Imaging System View

Figure 14. The experimental environment. A zipline connected to two pieces of scaffolding was used to model the appraoch of a receiver to the tanker's imaging system.

system was connected to a pair of zip-lines. The zip-lines were tied to two support platforms and increased at $\sim 12^{\circ}$ away from the aircraft model. Figure 14 shows the experimental environment and the perspective of the imaging system. The real world movement between the aircraft and cameras is irrelevant because the imaging system will only capture information based on the relative motion between the tanker and the receiver, exactly like the cameras in our modeled environment. In order to facilitate smooth image capturing, the cameras started next to the aircraft model and were pulled back (up and away) from it. While the actual order of the images models the receiving aircraft backing away from the tanker, simply reversing the order models an approach. Each run lasted approximately 20 seconds and produced about 700 useful images, modeling an image capture rate of about 35Hz.

3.1.3 Truth Data.

The data collection was conducted in a large Vicon chamber in order to simultaneously record the true relative positions of each camera and the aircraft model. Vicon systems act like an indoor GPS by recording the instantaneous position and orientation of objects of interest at a rate of 150hz. The system utilized for this



(a) Reflective Nodes on Receiver Model **(b)** Reflective Nodes on Imaging System

Figure 15. The Vicon nodes located on the receiver model and the imaging system. Each object was outfitted with reflective nodes to enable the Vicon system to recognize them and determine their positioning within its world frame at a rate of 150Hz.

experiment included 60 infrared cameras mounted around a hangar. Each camera emits an infrared signal and records the directions from which they return. Each object is covered in reflective nodes with its orientation declared prior to recording position data. This ensures that only objects of interest are recorded and objects can be identified based on the pattern of reflection their nodes produce regardless of location or orientation within the Vicon frame. Figure 15 shows the reflective nodes on the aircraft model and the imaging system.

The Vicon system was located on the same network as the imaging system. The network also included a GPS antenna as a standard, reliable timing source for time-stamping the images and truth data. Utilizing the same timing source and using a lower image capture rate then truth capture rate (35Hz vs 150Hz) ensured accurate truth data for each image. An issue to note about the network was the difference in time-stamping between the imaging and Vicon systems. While the cameras stamped the images when they were taken (at the cameras) the Vicon data was stamped when it was recorded (at the truth server). This means that the position estimates may

include error due to the same timestamps representing truth data that was collected sightly later than the image was taken.

The Vicon system provided x,y and z locations in meters, ϕ, θ, and ψ orientations in radians and $q_{\text{Vicon}}^{\text{Object}}$ for the two cameras and the aircraft model in its reference frame. This data was used to determine the true relative positioning between the aircraft model and each image. First, each image was matched with its corresponding truth data based on timestamps. Then the truth data for the location of the camera corresponding to each image and the aircraft model were filtered out. The location of the camera in the aircraft model frame, $\vec{t}_{\text{model}}^{\text{camera}}$, was determined by

$$\begin{bmatrix} x_{\text{camera}} \\ y_{\text{camera}} \\ z_{\text{camera}} \end{bmatrix}_{rec} = q_{\text{Vicon}}^{\text{model}} \begin{bmatrix} x_{\text{camera}} - x_{\text{model}} \\ y_{\text{camera}} - y_{\text{model}} \\ z_{\text{camera}} - z_{\text{model}} \end{bmatrix}_{\text{Vicon}} q_{\text{Vicon}}^{\text{model}-1}. \tag{34}$$

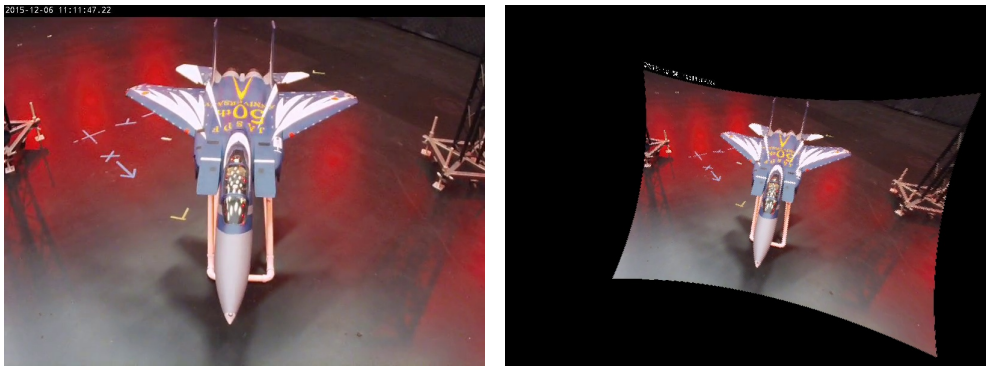
The orientation of the camera in the aircraft model frame was determined by

$$q_{\text{model}}^{\text{camera}} = q_{\text{Vicon}}^{\text{camera}} q_{\text{model}}^{\text{Vicon}-1} \tag{35}$$

Due to the information now being known in the aircraft model frame the derived orientation and location of the camera can be used to determine the true $\Delta_{\text{receiver}_t}^{C_x}$ being estimated by the approach described in the previous section.

3.2 Preliminary Experimentation

This thesis is the first time SfM has been introduced to the AAR domain. The AAR problem consists of a well defined mission set with many known parameters (i.e. receiver approach and stereo systems). Additionally, the SfM implementation utilized in this thesis is known. Five preliminary experiments were conducted to



(a) Nonrectified Image **(b)** Rectified Image

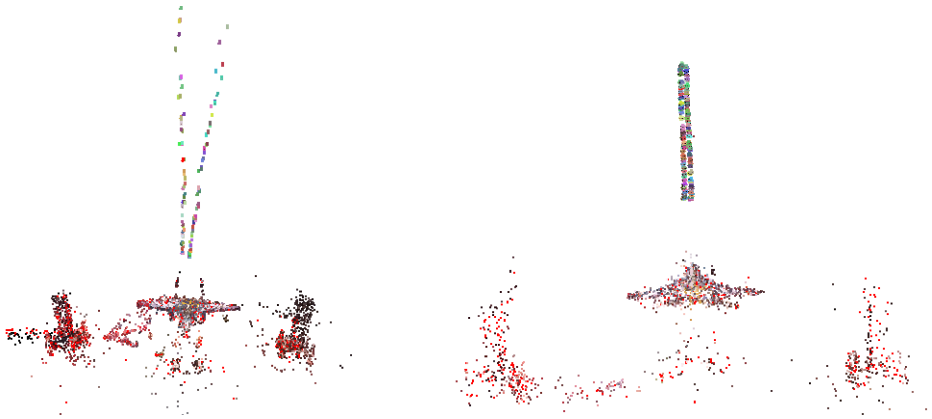
Figure 16. The rectification of flight imagery.

gain an understanding of how vSfM operates within the constraints of this thesis and minimize the number of variables considered in the methodology. The conclusions from this section were based on inspection and not quantitative analysis.

3.2.1 Rectified vs Non-Rectified Images.

In theory, SfM can operate with unrectified images to produce a projective reconstruction. However projective reconstructions can produce point clouds that are not visibly equivalent to the scene SfM is attempting to reconstruct. An experiment was conducted to determine the effect of rectified and non-rectified images on the point clouds generated by the particular SfM application used in this thesis. Rectified images were generated from the non-rectified images taken by the imaging system (Figure 16).

A sample of 116 images from a single run were used to generate a 3D reconstruction. The reconstruction was run in batch mode where the application had access to all images and performed a full bundle adjustment to accomplish the reconstruction. While this is different than the sequential mode applied in later experiments, it creates a best case reconstruction as a reference for what is possible. It is assumed that batch reconstructions are necessarily more accurate than sequential reconstruction



(a) Reconstruction of Nonrectified Images **(b)** Reconstruction of Rectified Images

Figure 17. Reconstructions from rectified and nonrectified imagery. A reconsttruction based on non-rectified images and a reconstruction based on rectified images. Distortion can be seen in the non-rectified image where the support post lean out and the rear stabilizers bend in.

because they have access to all imagery from the start. The resulting reconstructions are shown in Figure 17.

While the non-rectified reconstruction includes more points, which is good for model fitting, it is clear that the rectified image set produced a superior underlying model and therefore a superior reconstruction. The effect of image distortion seen in the non-rectified reconstruction is surprising in that SfM is typically robust to the distortion in uncalibrated imagery, i.e. it simultaneously estimates camera calibration matrices for individual cameras. The scaffolding (slanted outward) and the rear stabilizers (slanted inward) should be vertical. Such errors are obviously wrong and can be expected to cause problems when trying to align the derived point clouds to the reference model. Looking at the estimated camera positions as they approach the model it is clear that unrectified images are not producing an accurate scene reconstruction. In contrast rectified images preserved the slight offset between cameras representative of the known offset in the imaging system and also yielded the expected vertical properties in the scaffolding and rear stabilizers.

While rectified images facilitate superior reconstructions, a full stereo system is

not necessary for SfM. Any single calibrated camera should produce similar results. A reconstruction completed with standard camera phone images is shown in Figure 9.

3.2.2 Monocular vs Multiple Camera.

Initial experiments focused on using a monocular imaging system to evaluate if SfM could conduct a reconstruction from the motion of a single camera. However the availability of a second camera may provide additional perspective that may prove useful given the offset between cameras. Therefore, a second experiment was conducted to investigate the trade-off between monocular and multiple camera implementations. The term multiple camera is used to illustrate that the system does not have to be a stereo system with complete knowledge of both cameras. Such a multi camera system can use any number of cameras in various locations. A single run of rectified images from a stereo camera imaging system was used to complete the reconstructions. In one reconstruction all images from both cameras were used while in the other all the images from a single camera were used. When images from multiple cameras were used vSfM produced an initial reconstruction from the first two images, while the monocular sequence required 22 images to generate its initial reconstruction. The ability of the multiple camera system to generate a useful model based on the first set of images, which it could then adjust with future image sets, is attributable to the translational perspective difference provided by the stereo configuration. In contrast, the lack of parallax in the monocular image set resulted in a much poorer reconstruction, evaluated by the camera position estimates (Figure 18a). When the images are taken in real-time there is very little relative motion between images. Therefore the monocular approach must wait until some minimum movement is achieved while the multiple camera approach leverages the existing physical cam-

(a) Monocular Position Estimates **(b)** Stereo Position Estimates

Figure 18. Monocular and multiple camera position estimates during reconstruction. The monocular image position estimates should be in a single line. The multiple camera position accurately model the offset contained in the imaging system.

era offset to generate enough parallax to create an initial reconstruction, which then resulted in what appears to be a more accurate estimation of camera pose (Figure 18b).

Figure 18 shows the estimated image locations after each approach processes 22 images. Note that the multiple camera system has access to the 22 images well before the monocular system. While the monocular images should be in a straight line, as they are the sequential images from the left camera, they exhibit a large degree of positional uncertainty that fails to capture the linear motion of the camera toward the model. Furthermore, it was also noticed that the multiple camera reconstruction was more representative of the actual scene throughout the entire image sequences. While the monocular image set does eventually produce a quality model, the initial model is essentially based on a single perspective of the receiving aircraft. Thus all remaining experiments are conducted with a multiple camera system as to leverage the benefits provided by the physical offset of the stereo camera setup.

3.2.3 Image Capture Rates.

The superior performance of the multiple camera system showed that the perspective of images is just as important, if not more important, than the visible representation of the scene in the image. It appears that more perspectives of a scene will generate better reconstructions than the same amount of images from the same or a only a few perspectives. This theory is underscored by the quality reproduction shown in Figure 9 which utilized 43 different perspectives to generate highly accurate image position estimates. Given the minimal motion between sets of real-time images, another experiment was conducted to explore if images captured at a slower rate could provide the same or better performance. Thus a set of 213 rectified images from a single multiple camera run was down-sampled to represent a 15 Hertz (Hz) image capture rate (107 images) from the initial 30 Hz rate. In this way, a third set of images, the alternating set, was generated that alternated image capturing between cameras. While this set still met the 15 Hz rate, it utilized less than half of the images (52) in the other down-sampled set. The alternating set was able to generate a quality model with 9 images, equating to 9 image capture iterations. The 15Hz set was able to generate a quality model with 17 images. It takes 2 images per iteration and therefore would still require 9 image capture iterations resulting in the same estimation performance with more image processing. The 30Hz set was not able to produce a quality model with the images captured in its first 10 image capture iterations thus implying that physical perspective changes are critical to the effectiveness of the SfM reconstruction. This observation further underscores the theory that the number of perspectives is more important than the number of images. This is logical because images with little offset provide less new information than overlapping images with large offsets. Astoundingly, the alternating set produced more accurate image position estimates faster than the other sets. From these results the remainder

of experiments will be conducted with images captured at 15 Hz from alternating cameras.

3.2.4 Feature Matching Operations.

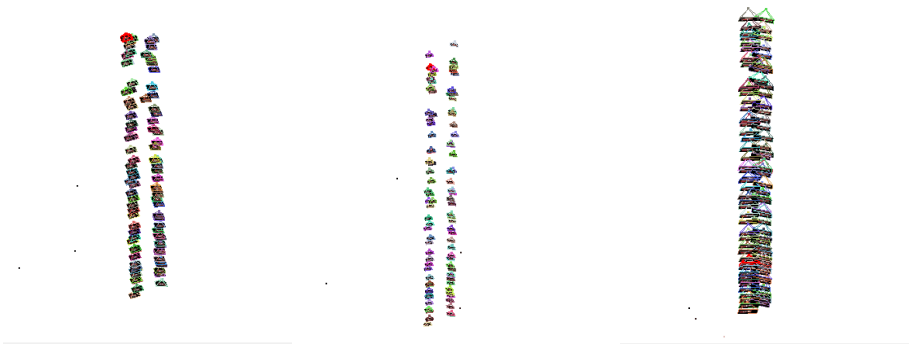
Typical SfM computation assumes unordered image sets, and thus completes a full pairwise matching between all images. In the AAR domain, this means that every new image that is captured must go through the feature matching process with every previous image despite the sequential nature of the problem. When image capturing lasts for extended periods of of time, a 15Hz capture rate produces thousands of images, wherein a full pairwise feature matching event will not support a real-time requirement. While Wu suggests ways to minimize matching efforts by filtering out poor quality image pairs, another approach may be to use a sliding window to only match the current image with a set number of past images [82].

This preliminary experiment investigates the effect of conducting feature matching on a subset of images for each new image introduced. Three matching schemes were used: full pairwise, sliding window and, seeded-sliding window. The sliding window approach matches each new image to the previous 20 images. The seeded-sliding window matches each new image to a 20 image subset of past images, s, determined by

$$s = \{I_x, I_{2x}, I_{3x}, \ldots, I_{20x}\} \tag{36}$$

where $x = \frac{n}{20}$ and n is the number of past images.

As expected, the reduced matching schemes take much less time than the full pairwise matching scheme. To conduct image matching on the entire set the pairwise scheme required 22 seconds (0.24 seconds/image)— a number that will increase as the size of the image set grows. The sliding window and seeded sliding window approaches required 8 seconds (.09 seconds/image), which should remain constant. While all



(a) Full Pairwise **(b)** Sliding Window **(c)** Seeded Window

Figure 19. Position estimates provided by the feature matching schemes. Each of the feature matching schemes generate similar position estimates however, the realtive positioning between cameras and the focal lengths are different for each.

three approaches provide image position estimates that maintain the known camera offset (Figure 19), the seeded sliding window scheme produces the best estimates in that they maintain a constant forward movement throughout the sequence and clearly maintain the alternating laft/right pattern of the two cameras operating at 15 Hz. This observation leads to the idea that efficiencies are gained from matching with older images while avoiding the need to match to all previous images.

3.2.5 Occlusion Robustness.

Knowing that SfM incorporates past information in the reconstruction fornewly introduced images leads to interesting possibilities such as providing position estimations for features occluded in the new image. This experiment tests the ability of SfM to develop useful point clouds for an object that is not fully visible in the scene. In the AAR domain, aircraft occlusion is expected (though not addressed in this thesis) given the fact that the boom is between the imaging system and the receiving aircraft. Additionally, as the receiving aircraft moves in and out of the system's field of view, occlusions may be significant as large portions of the receiver are beyond the camera's field of view (Figure 20a). Such occlusions cannot result in the inability to produce

(a) Image with Occlusion **(b)** Occluded Image Point Cloud

Figure 20. Point cloud generated for image with mass occlusion. An image with nearly half of the plane occluded generated a full point cloud using the information available from previous images. The position of the nose points in the image frame are accurately estimated even though they are not included in the image.

accurate estimates. Approaches that generate point clouds for each image [17], or sets of images [81], as independent measurements risk loosing information when part of the receiving aircraft is not included in the image. Figure 20b shows the point cloud generated for an image in which the receiving aircraft is partially occluded by utilizing information about the occluded features from previous images.

In this way SfM clearly demonstrates an ability to use information from multiple images to generate accurate point clouds, from which relative positioning can be estimated, even when much of the receiver is not visible in the image.

IV. SfM Relative Navigation Experiment

This chapter presents an approach that determines relative position with SfM. The chapter is organized as follows. First, the proposed algorithm is presented and discussed in detail. Next, the experimental design is presented by detailing the objective, assumptions & limitations, and the experimental process. Next, the results are presented and analyzed. Finally, the chapter concludes with a discussion of lessons learned and their impact in the AAR domain.

4.1 SfM Relative Positioning Algorithm

The proposed algorithm consist of six main stages: 1) image capture, 2) image rectification, segmentation and masking, 3) sequential SfM, 4) SfM to camera frame transform , 5) PCA registration and 6) relative position estimation (Figure 21). Each of the stages supports the computation of the solution to the overarching problem of relative positioning for AAR problem defined as:

> Given a set of tanker cameras, C, recording a receiving aircraft, determine the relative positioning between the receiving aircraft and the optimal refueling position at time t such that
>
> $$\Delta^{\text{optimal}}_{\text{receiver}_t} = \Delta^{\text{optimal}}_{\text{tanker}} \Delta^{\text{tanker}}_{C_x} \Delta^{C_x}_{\text{receiver}_t} \tag{37}$$
>
> where $\Delta^{\text{optimal}}_{\text{tanker}}$ is the known rigid offset between the origin of the tanker frame and the optimal refueling position and $\Delta^{\text{tanker}}_{C_x}$ is the known rigid offset between the tanker frame and the frame of a particular camera, C_x.

While the AAR problem is solved by determining $\Delta^{\text{optimal}}_{\text{receiver}_t}$ the true focus of the proposed algorithm is estimating $\Delta^{C_x}_{\text{receiver}_t}$, the relative position of the receiving air-

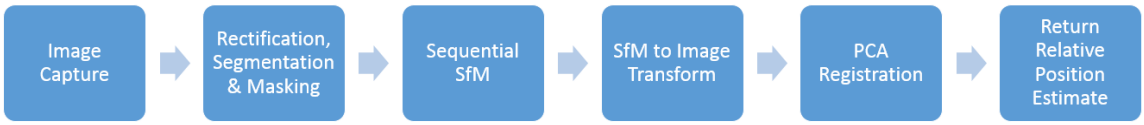


Figure 21. The proposed algorithm flowchart. The algorithm includes 6 stages responsible for estimating the relative position of the receiver from a sequence of images.

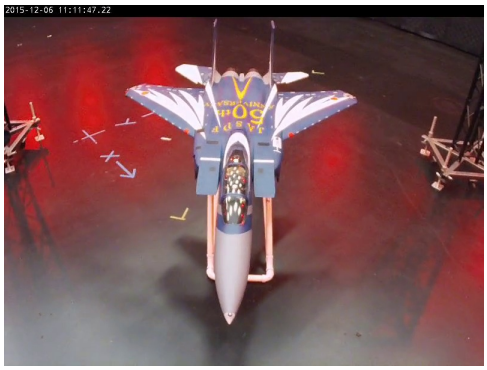
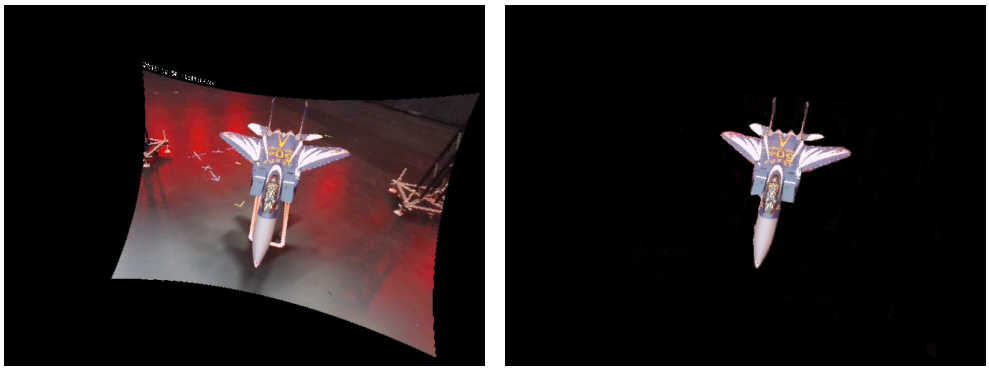


Figure 22. Raw image captured by the imaging system. A sample image, $i_{t,x}$ output by the image capture stage. It is associated with a particular camera, C_x, and a particular time, t.

craft and a camera at time t. Each of the algorithm stages and how they contribute to determining $\Delta^{C_x}_{\text{receiver}_t}$ are discussed in turn.

4.1.1 Image Capture.

The first stage is responsible for capturing an image ,i, and associating it with the specific camera, C_x, and time, t. SfM is a robust system with respect to image capture. It can operate as a monocular, stereo, or many camera system. SfM derives all information about a scene directly from the images, therefore any number and type of cameras can be used in the same reconstruction. The set of cameras that contain the receiving aircraft in their field of view, $C = \{C_1, C_2, \ldots, C_n\}$, will be utilized to capture images and contribute to the reconstruction. Such an approach provides the maximum amount of perspectives, shown in Section 3.2 to be positively correlated with point cloud quality.. Figure 22 shows a sample image as initially captured in stage 1.



(a) Rectified Image **(b)** Masked Image

Figure 23. The captured image after each step of processing. Examples of rectified and masked images. The masked image represents a useful image that is added to the image set I.

4.1.2 Rectification, Segmentation & Masking.

The second stage processes the images captured in stage 1, i, to prepare then for use in the reconstruction conducted in stage 3. This stage first rectifies the image such that the pixel information can support an accurate reconstruction. Additionally, this stage considers the problem of dSfM [64, 74, 22]. While other aircraft can be kept out of the field of view of the imaging system there will still be relative motion in most cases, most commonly the ground moving in the opposite direction as the receiving aircraft. In order to account for this an image masking technique is proposed. Common segmentation techniques are capable of identifying objects in an image. Once the receiving aircraft has been identified a solid color mask can be used to cover all pixels not associated with it. Any features identified in the next stage will be part of the rigid airframe and therefore all identified points will have the same relative motion. Once i has been adequately processed and is suitable for SfM it will be added to the image set, I, in chronological order where $I_{t,x}$ was captured at the current timestep. Figure 23a shows a rectified version of the image, which is then masked in Figure 23b to isolate the receiver.

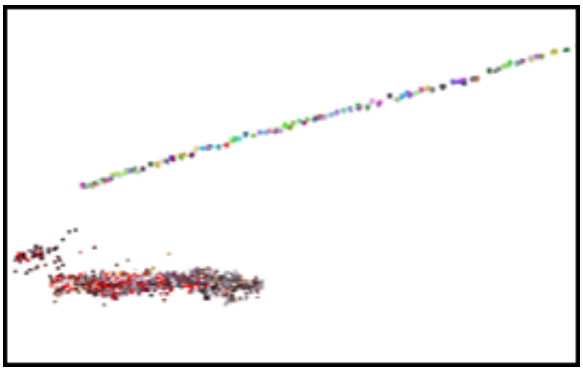


Figure 24. The image position estimates of an entire run. Each dot in the diagonal line represents the estimated position of an image during the a run. While each image is added individually the final image will have knowledge of the positioning of all previous images.

4.1.3 Sequential SfM.

The third stage provides the relative positioning between the reconstruction of the receiver and $I_{t,x}$ in the arbitrary SfM reference frame. Using I for inputs, SfM will sequentially consider each $I_{t,x}$ as discussed in Section 2.5. First, the SIFT algorithm, discussed in Section 2.3, will be used to identify features. Then $I_{t,x}$ will be matched to a representative subset of past images in I as discussed in Section 3.4. If a quality initial model has not been generated a complete BA will be conducted to generate a new model based on the additional information provided by $I_{t,x}$. If a quality model has already been established, $I_{t,x}$ will be added to the reconstruction based on the current model and a pBA will be completed. Once $I_{t,x}$ has been added to the reconstruction, its estimated position in the arbitrary SfM world frame, $\Delta_{\text{SfM}}^{I_{t,x}}$, is known. The locations of each point in the 3D reconstruction are also known in the SfM world frame providing the point set P_{SfM}. Figure 24 shows the 3D reconstruction of an entire approach. Each colored dot in the straight line represents the position of an image as well as the position of the a camera at the time each image was taken.

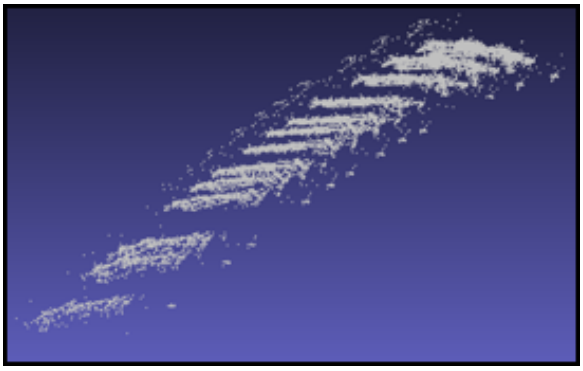


Figure 25. Receiver position estimates after reference frame transform. The relative positioning of the receiver for an entire run after the reference frame transform from the SfM frame to the image frame.

4.1.4 SfM to Image Frame Transform.

The fourth stage moves the reconstructed point cloud, P_{SfM}, into the image frame producing the point cloud representation of the receiver aircraft, P_{receiver}. First, a reference space transform is completed to move P_{SfM} into $I_{t,x}$'s reference frame. This can be accomplished by using the relation between the SfM frame and $I_{t,x}$'s frame derived by sequential SfM.

$$\dot{p}_{I_{t,x}} = \Delta_{SfM}^{I_{t,x}} * \begin{bmatrix} \dot{p}_{\text{SfM},x} - I_x \\ \dot{p}_{\text{SfM},y} - I_y \\ \dot{p}_{\text{SfM},z} - I_z \end{bmatrix}_{SfM} \tag{38}$$

Figure 25 shows the approach from Figure 24 after the reference frame has been transformed from the SfM frame to the camera frame. During operations each point cloud will be considered independently, however Figure 25 clearly illustrates the effect of transforming the point clouds into the camera frame where the camera is stationary and the aircraft is rising and moving forward on approach.

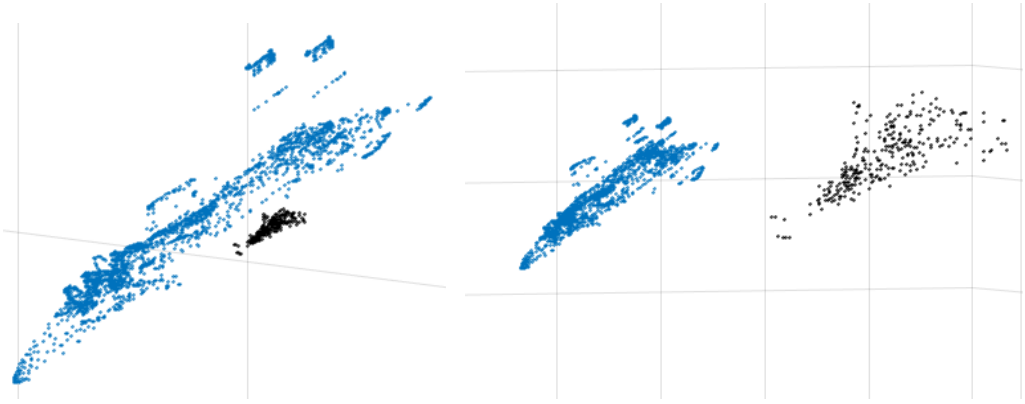
After the point cloud has been transformed in the image frame it must be scaled to reflect the real world size of the receiver. SfM has no knowledge of the scene

(a) Virtual Model Points for Scaling **(b)** Point Cloud Points for Scaling

Figure 26. The points used to scale the SfM point cloud. The camera maintains a clear view of the y dimension (wings) of the receiver. Scaling the based on the wingtip to wingtip distance transforms the point locations into real world space.

outside of the images it considers, including scale. While it can relatively determine where points are in the camera's frame, it can not determine the actual distance between points in the scene. Therefore, SfM establishes an arbitrary scale to be used for its reconstruction and camera position estimates. In order for the true relative position of the point cloud to be accurately represented for registration, the point cloud must be scaled to its correct real world size. Due to the straight forward approach of the receiver in the AAR scenario, which only includes movement in the x and z directions, the wingspan (y-axis of the receiver frame) is clearly visible and parallel to the camera's y-axis. The receiver's wingspan is known and therefore provides a reference point for size. A scaling factor can be computed by determining the maximum and minimum y values of the point cloud and scaling it to match the wingtip to wingtip distance of the virtual model, P_{model}, a real world sized point cloud used for registration in the next stage. Figure 26 shows the maximum and minimum y values in the virtual model and a SfM generated point cloud.

Once the maximum and minimum y-values of the point cloud have been deter-



(a) Size & location in SfM frame **(b)** True size & location after scaling

Figure 27. The initial point cloud generated by SfM is of arbitrary size. Scaling the point cloud transforms it into the correct size and realtie location.

mined the scaling factor, γ, can be determined by Equation 39.

$$\gamma = \frac{P_{\text{model},y-max} - P_{\text{model},y-min}}{P_{I_{t,x},y-max} - P_{I_{t,x},y-min}} \tag{39}$$

The actual location of the points from the receiver can then be determined by simply multiplying $P_{I_{t,x}}$ by the scaling factor as shown in Equation 40. Once scaled, the points are now at the exact location of their features relative to the camera at time, t and therefore are considered to be an accurate model of the receiver, receiver_t.

$$\text{receiver}_t = \gamma P_{I_{t,x}} \tag{40}$$

Figure 27 illustrates the effect of this scaling on the point cloud shown in Figure 26.

4.1.5 PCA Registration.

While the location of each point in the point cloud is known in the camera frame individually, the actual position of the receiver is still unknown. In order to determine the $\Delta^{C_x}_{\text{receiver}_t}$ needed to solve Equation 37, a virtual point cloud, P_m, is used. P_m is

an exact point cloud model of the receiver with a known relative orientation in the camera frame. Therefore $\Delta^{C_x}_{\text{receiver}_t}$ can be defined as

$$\Delta^{C_x}_{\text{receiver}_t} = \Delta^{C_x}_{P_m} \Delta^{P_m}_{\text{receiver}_t} \tag{41}$$

where $\Delta^{C_x}_{P_m}$ is known and $\Delta^{P_m}_{\text{receiver}_t}$ is determined by the following 3D rigid registration.

First, receiver$_t$ is centered on the origin by subtracting its centroid, $\bar{r}_t$, from each point. This also co-locates receiver$_t$ with P_m. Using the PCA computations discussed in Section 2.6, the principal components of the centered receiver$_t$, V_{receiver_t}, are determined. The R contained in $\Delta^{P_m}_{\text{receiver}_t}$ solves the problem

$$R^{P_m}_{\text{receiver}_t} V_{\text{receiver}_t} = V_{P_m}. \tag{42}$$

Therefore $R^{P_m}_{\text{receiver}_t}$ is computed by

$$R^{P_m}_{\text{receiver}_t} = V_{P_m} V^T_{\text{receiver}_t} \tag{43}$$

Equation 43 will generate correct values for the rotation matrix, however, in three dimensions there are 4 possible axis orientations. It is possible that the R derived in Equation 43 models an incorrect orientation of receiver$_t$ with respect to P_m. A simple check is used to determine the correct orientation. Given the orientation provided in R, the other three orientations can be found by manipulating the diagonal values to produce the following four matrices.

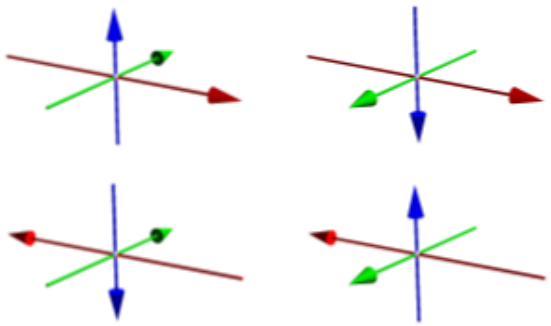


Figure 28. The four possible axis orientations in 3D space. In 3D space there are four axis orientation, assuming the right hand rule. It is critical to verify the estimated rotation matrix conforms to the correct orientation.

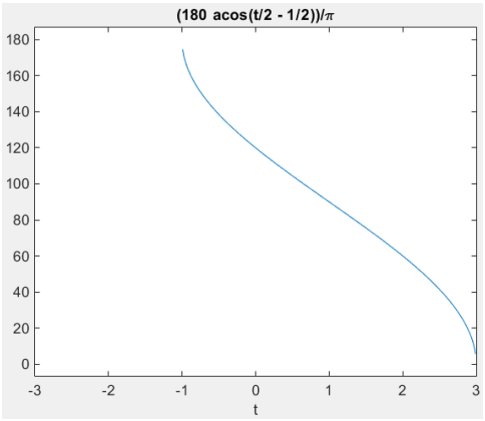


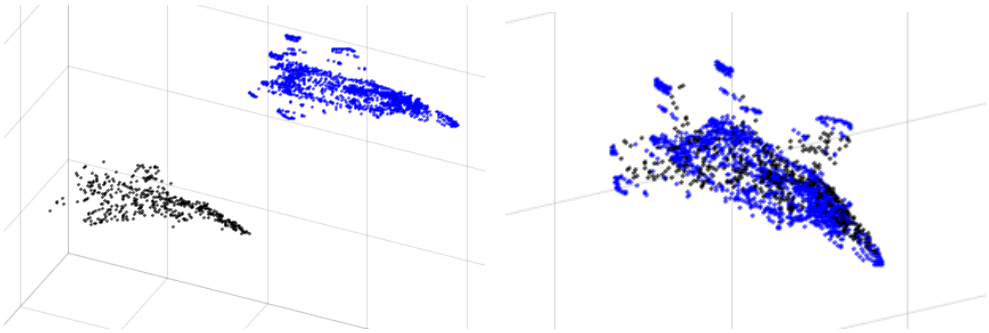
Figure 29. The relationship between the trace and angle of rotation corresponding to a DCM. The trace and angle of rotation are negatively correlated. Therefore, the DCM that contains the highest trace is associated with the minimum rotation.

$$\begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix} \begin{bmatrix} -R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & -R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix} \begin{bmatrix} -R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & -R_{3,3} \end{bmatrix} \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & -R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & -R_{3,3} \end{bmatrix}$$

Of these, the matrix corresponding to the smallest angle of rotation has the correct orientation. Using the trace of a matrix, the rotation angle can be calculated by:

$$\Theta = \frac{180 * cos^{-1}(\frac{trace(R)-1}{2})}{\pi} \tag{44}$$

Due to R being orthonormal, all diagonal values are between -1 and 1. Therefore the



(a) Relative Positioning **(b)** Alignment after 3D Rigid Registration

Figure 30. The 3D rigid registration of a generated point cloud and a the virtual model. The virtual model is aligned to a point cloud located in its position estimated by SfM. The better the alignment, the more accurate the relative positioning estimate.

sum of the 3 values on the diagonal must be contained in [-3,3]. Figure 29 shows that given these constraints there is a strictly negative correlation between the trace and the angle of rotation of the matrix. Therefore the R with the largest trace must be the true rotation matrix that aligns receiver_t to P_m. Once $R^{P_m}_{\text{receiver}_t}$ has been verified, the translation, $\vec{t}^{P_m}_{\text{receiver}_t}$, associated with $\Delta^{P_m}_{\text{receiver}_t}$ can be computed with Equation 33. Figure 30 shows the relative positioning of receiver_t and P_m before and after a 3D rigid registration.

4.1.6 Relative Positioning Estimation.

The relative positioning problem can now be solved using Equations 37 and 41 as follows.

$$\Delta^{C_x}_{\text{receiver}_t} = \Delta^{C_x}_{P_m} \Delta^{P_m}_{\text{receiver}_t}$$
$$\Delta^{\text{optimal}}_{\text{receiver}_t} = \Delta^{\text{optimal}}_{\text{tanker}} \Delta^{\text{tanker}}_{C_x} \Delta^{C_x}_{\text{receiver}_t}$$

The estimated $\Delta^{\text{optimal}}_{\text{receiver}_t}$ is then communicated to the receiver. Standard autopilots can use this information to correct the receiver's current position to achieve the optimal refueling position.

4.2 Experimental Objective

This experiment aims to test the feasability of the SfM relative navigation aglorithm in the AAR domain. Specifically, the algorithms ability to estimate $\Delta^{C_x}_{\text{receiver}_t}$ will be tested for accuracy in each of the six degrees of freedom.

4.3 Assumptions & Limitations

The performance of the algorithm was quantitatively tested in a modeled environment. The environment aims to accurately represent the real world, however, some assumptions were necessary.

Boom Occlusion: During an actual refueling the rigid boom would be in the field of view of both cameras occluding sections of the receiving aircraft. This would cause a dynamic environment and cause problems for a basic SfM implementation. For this thesis a boom was not included in the imagery allowing the experiments to focus solely on the ability to estimate the relative positioning between the cameras and the model aircraft.

Imaging System Configuration: The imaging system consists of a pair of electro-optic cameras that are assumed to have the same, but scaled down, relative positioning and view to those included on a tanker vision system. The position and orientation of the imaging system with respect to the tanker is known. The left and right cameras are separated along the y-axis by a known baseline and are aligned along the x- and z-axes. The cameras share an overlapping field of view.

Camera Model: The cameras used in this thesis are assumed to follow the pinhole camera model, with a fixed focal length and field of view.

Rigid Body: The F-15 model is a rigid body which does not flex during imaging. Therefore, it is assumed that the receiver is rigid as well, with no changes to its form (wing deflection, flaps, ailerons, landing gear, etc.)

System and Error are Scalable: The 1:7 scale imaging system and F-15 model represent a full-scale system. The error reported by the 1:7 scale system can be linearly scaled.

Image Masking vs Manual Point Removal: The experimental environment was not a dynamic scene and therefore did not require image segmentation and masking prior to SfM. However, the reconstructions included points for objects other than the aircraft model which were manually removed from the point clouds. It is assumed that the same images with masked backgrounds would produce the same point clouds.

4.4 Process

Rectified images from the data collection were down-sampled as described in Section 3.3 and used as inputs to vSfM. The application did not allow for matching to be conducted between a new image and past images already included in a reconstruction. Therefore, all images were input into the application and feature identification and matching were conducted simultaneously and stored. Each image was then added sequentially to develop the reconstruction as described for the proposed solution. The reconstructions for each image were then saved. A separate program was used to conduct the reference space transform to generate the point clouds in the camera frame. MeshLab was then used to remove any points assumed to not be associated with the aircraft model. Finally, MATLAB was used to scale the generated point cloud and conduct the 3D rigid registration. The estimated $\Delta^{C_x}_{\text{receiver}_t}$ for each point cloud was saved for analysis. A total of 121 point clouds from three runs were generated and utilized for this test.

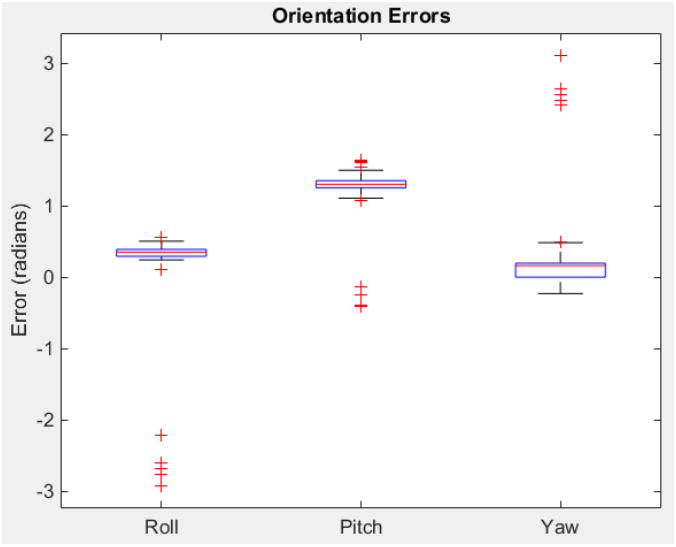


Figure 31. The orientation estimate errors. The central marks are the medians, the edge of the boxes are the 25^{th} and 75^{th} percentiles, the whiskers extend to the most extreme data points not considered to be outliers and the outliers are plotted individually as red + marks. The orientation estimates had a high error rate specifically with respect to pitch.

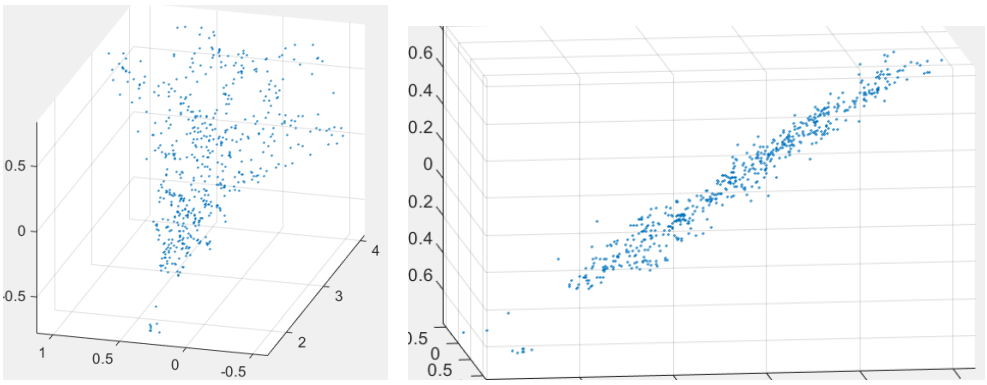
Table 1. Orientation Errors by Run

Run	Roll (radians)	Pitch (radians)	Yaw (radians)
Run 1	0.1828(±0.7856)	1.1295(±0.4217)	0.5431(±1.1085)
Run 2	0.3234(±0.041)	1.2885(±0.0638)	0.1234(±0.0961)
Run 3	0.0223(±0.8673)	1.1555(±0.5861)	0.6031(±1.374)
Total	0.2305(±0.6027)	1.2365(±0.3685)	0.3328(±0.6027)

4.5 Results & Analysis

4.5.1 Orientation Accuracy.

The estimates derived for the relative orientation between the receiving aircraft and the cameras, $R^{C_x}_{\text{receiver}_t}$, are presented in Figure 31. The central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered to be outliers, and the outliers are plotted



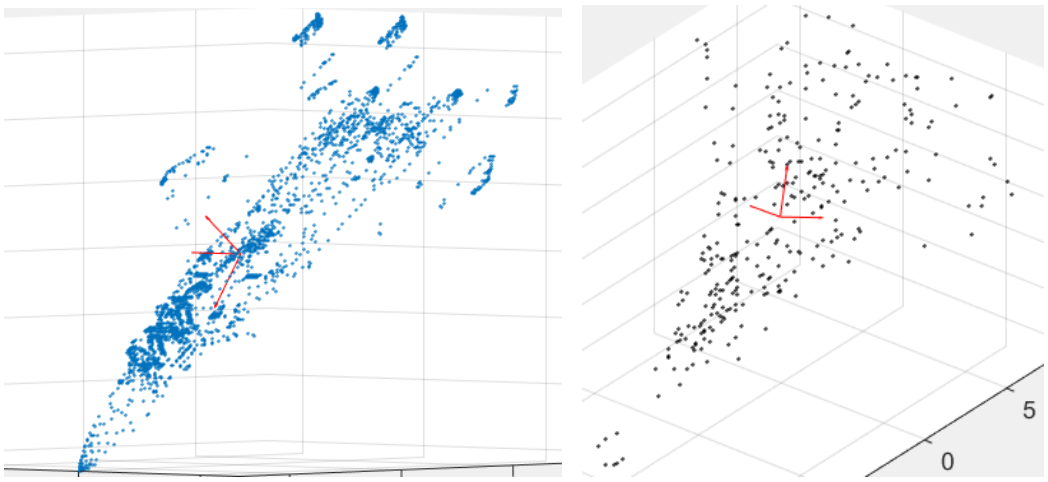
(a) Camera Perspective **(b)** Side Perspective

Figure 32. A sample point cloud geneated by SfM. The shape of the receiver can be recognized in the camera's perspective. The side view shows the lack of rear stabilizers and some outlier points in the front.

individually as red + marks. Table 1 presents the mean and standard deviations of the orientation errors for each run.

The primary cause for the high errors is the inability of SfM to generate representative point clouds of the receiving aircraft. Figure 32 shows multiple views of a point cloud generated when considering 76 images. While the general shape of the receiving aircraft can be recognized from the camera's perspective, the side view clearly shows a lack of rear stabilizers and what may be an abundance of noise in front of the aircraft. These reconstruction errors are the result of weaknesses in SfM when dealing with minimal perspectives.

The preliminary experimentation, Sections 3.2 and 3.3, showed that it is important to have many different perspectives of the reconstructed scene. The relative motion in this experiment, and in AAR, is mostly forward motion. This is essentially the expansion of a single perspective as the receiving aircraft approaches the tanker. The particular perspective, looking straight at the front of the receiving aircraft, minimizes the visible portion of the rear stabilizers and makes it hard to identify correspondences between images. Therefore, the rear stabilizers are not reconstructed and are not represented in the generated point clouds.

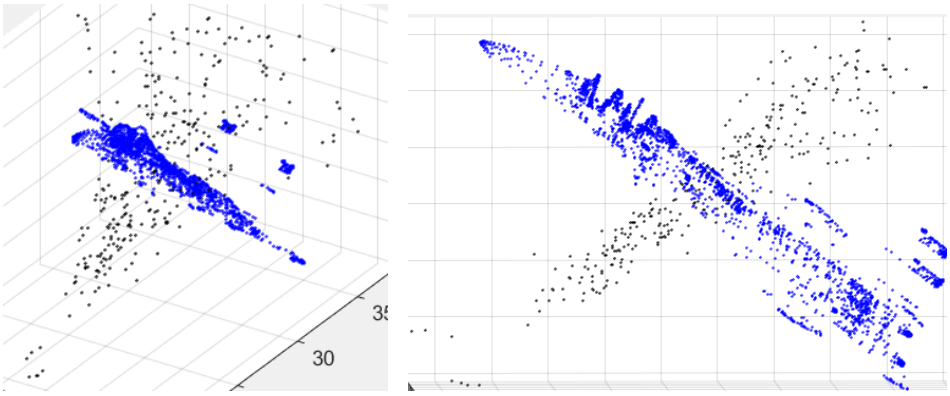


(a) Virtual Model Principal Components **(b)** Point Cloud Principle Components

Figure 33. The principal components of the virtual mode and a SfM generated point cloud. The difference in principal components between the virtual model (left) and a generated point cloud.

The poor quality of the generated point clouds directly prevented the PCA based registration from being able to accurately register the virtual model. As discussed in Chapter 4, PCA registrations have low operation times because the alignment method is based on the underlying geometry of the point clouds and not a least-squares fitting of point correspondences determined by an expensive nearest neighbor search. It works best when the point clouds are normally distributed, geometrically similar and contain no outliers [9]. Therefore, when the underlying geometry of the generated point clouds is fundamentally different than that of the virtual model the principal components will not align as expected. The principal components of the virtual model and a SfM generated point cloud are shown in Figure 33.

While SfM was able to consistently produce point clouds with similar principal components, the differences from the virtual model make it impossible for PCA to accurately register the model and therefore produce accurate $R^{C_x}_{\text{receiver}_t}$ estimates. Figure 34 illustrates a typical registration from this experiment. The blue virtual model was registered to the black point cloud in the experiment. The different principal



(a) PCA Registration **(b)** PCA Registration Side Perspective

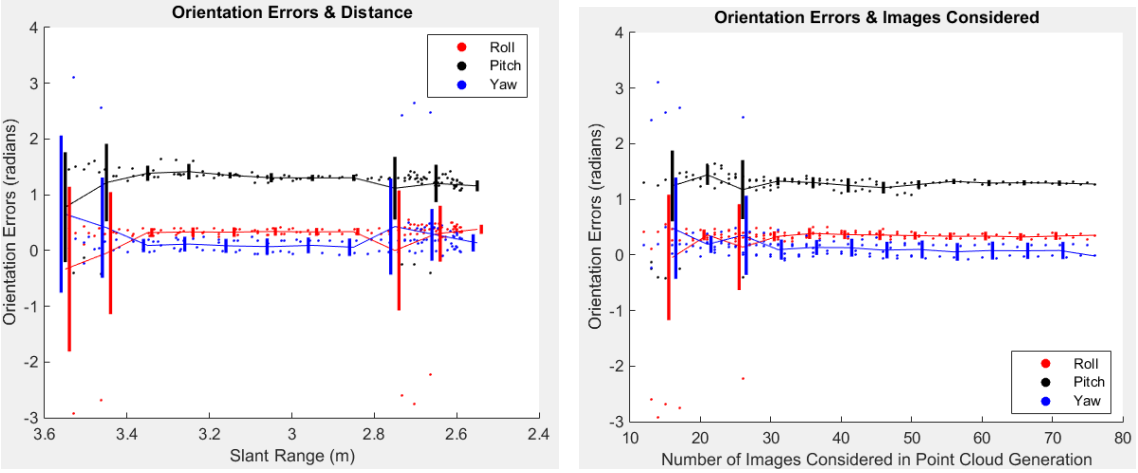
Figure 34. The PCA registration of the virtual model to a point cloud. The virtual model, blue, after being registered to a SfM point cloud using a PCA registration. The different principal components forced the registration to align the wrong axes.

components forced the PCA registration to align the wrong axes leading to the large error in pitch estimation. This result is reflected in the high pitch errors shown in Figure 31.

The primary cause of orientation error in this experiment is the inability of SfM to generate representative point clouds due to limited perspectives of the receiving aircraft. The PCA registration algorithm performed exactly as expected by determining the correct principal components of the point clouds under consideration and aligning the virtual model to the point clouds. If SfM generated representative point clouds of the receiving aircraft, then the PCA registration would have been able to determine accurate estimates for $R^{P_m}_{\text{receiver}_t}$.

Additionally, the AAR scenario is concerned with the effect of distance on errors. It has been shown that error rates are negatively correlated with the distance between the tanker and the receiving aircraft [81]. Figure 35a models the relationship between orientation error and distance. The errors remain constant with the exception of two sets of outiers clustered around two distances, approximately 3.5m and 2.7m. These

distances correspond to the spot in each run where the initial point clouds were generated by SfM (2.7418m for Run 1, 3.5263m for Run 2, 3.5040m for Run 3).



(a) Orientation Error vs Distance **(b)** Orientation Error vs Number of Images

Figure 35. The orientation errors presented against distance and the number of images considered while generating the point cloud. The charts present individual data points and the mean and standard deviations of 0.1 meter or 5 image interval sets of data. The outliers The distance between the receiver and cameras has no effect on the accuracy of the orientation estimates. The accuracy errors reach a near constant value after 30 images were considered in each run. The initial point clouds were generated with the information fewer images causing erratic errors.

Figure 35b models the relationship between orientation error and the number of images considered during the point cloud generation stage. The initial point clouds in a run are generated from a model based on a minimal amount of images. As more images are added to the dataset, the model is refined by partial bundle adjustments and the reconstruction, and therefore point clouds, become more representative of the actual scene. While the estimates based on few images are noisy, the estimates based on more than 30 images produce steady errors. The relationship between number of images considered and distance is obviously negatively correlated for each run, however different runs start and end in different relative locations. Therefore, the error rates are clearly dependent on the number of images considered and independent of distance.

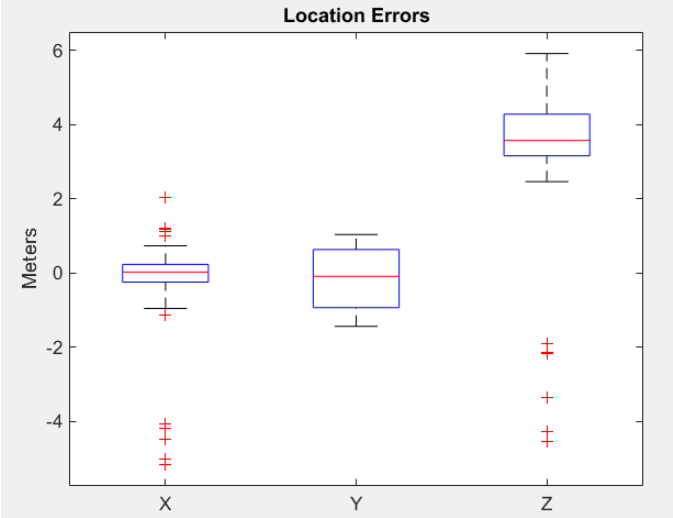


Figure 36. The location estimate errors. The location estimate errors are very high due to the use of the orietnation estiamtes in their computation.

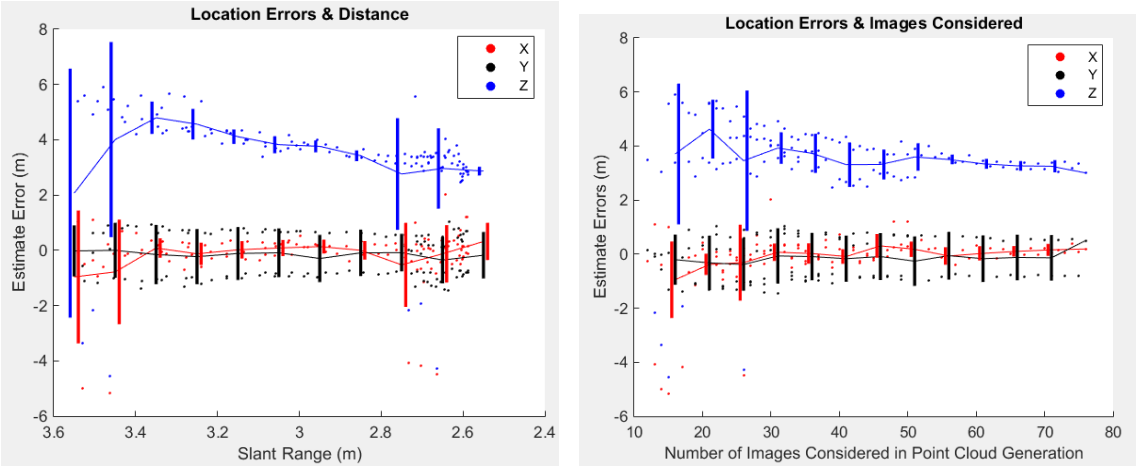
4.5.2 Location Accuracy.

Table 2. Location Errors by Run

Run	X (meters)	Y (meters)	Z (meters)
Run 1	-0.293(±1.2925)	-0.2904(±0.755)	2.7605(±1.7091)
Run 2	0.0616(±0.2834)	-0.1155(±0.8201)	3.9555(±0.6852)
Run 3	-0.4217(±1.4207)	-0.148(±0.833)	3.557(±2.6733)
Total	-0.1606(±1.0212)	-0.1854(±0.8004)	3.4827(±1.712)

The estimates derived for the relative location between the receiving aircraft and the cameras, $\vec{t}^{C_x}_{\text{receiver}_t}$, are presented in Figure 31. The central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points no considered to be outliers, and the outliers are plotted individually as red + marks. Table 2 presents the mean and standard deviations of the orientation errors for each run.

Figure 37 shows the relations between location error and distance and number of



(a) Location Error vs Distance

(b) Location Error vs Number of Images

Figure 37. The location errors plotted against distance and the number of images considered suring point cloud generation. Like the orientation errors, the location errors have no correlation with slant distance and appear to reach a steady state when 50 or more images were used to generate the point clouds.

images considered during point cloud generation. Like the orientation errors discussed previously, the errors are closely related to the number of images considered and independent of distance.

The errors are again very large throughout the experiment. Similar to the orientation errors, the location errors start noisy but eventually become very consistent after a certain amount of images are considered. However, once the errors become constant they are still very large. This is caused by two problems, primarily the reliance on relative orientation to determine location and forward motion.

The PCA registration process, described in Section 4.X, first aligns the virtual model to the generated point cloud to determine $R_{P_m}^{\text{receiver}_t}$ and $\vec{t}_{P_m}^{\text{receiver}_t}$. These values are then used to determine $\vec{t}_{\text{receiver}_t}^{C_x}$ by

$$\vec{t}_{\text{receiver}_t}^{C_x} = R_{P_m}^{C_x}(\bar{p}_m - R_{P_m}^{\text{receiver}_t{}^T}\overline{\text{receiver}}_t) \tag{45}$$

where $R_{P_m}^{C_x}$ is the known rotation between the camera frame and the virtual model

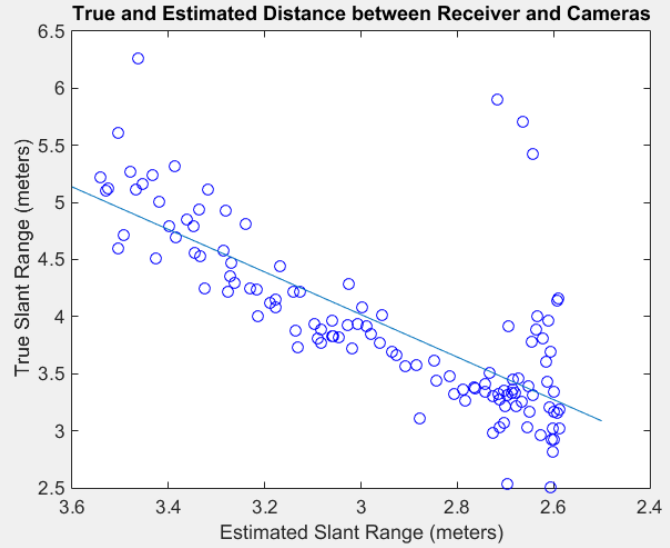
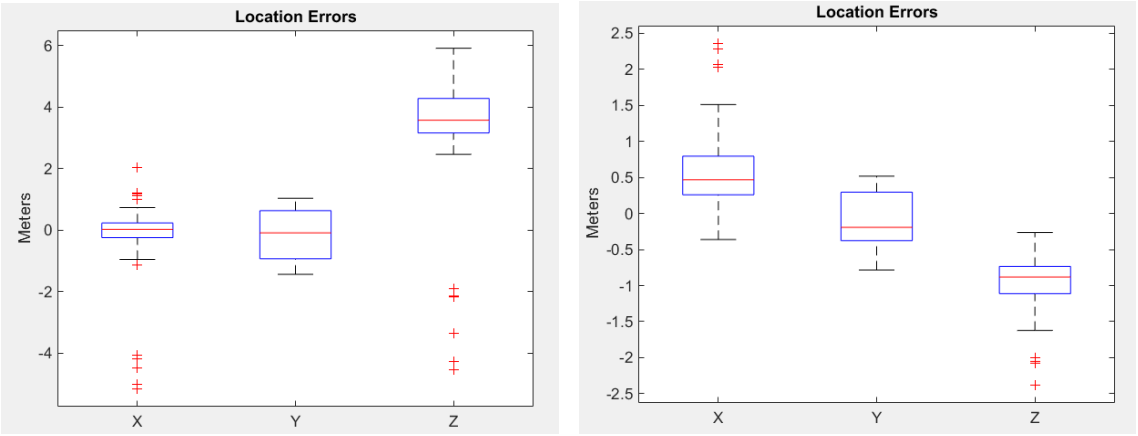


Figure 38. The true slant range plotted against the estimated slant range. Focusing solely on slant range isolates the location estimates from the orientation estimates. The algorithm was generally able to determine the direction of relative motion as the cameras approached the receiver.

frame, both located at the camera origin. $R_{P_m}^{\text{receiver}_t}$ is the estimate determined by aligning principal components, discussed in the previous sections. Due to the estimated orientation being used to derive the estimated location, high orientation errors directly cause high errors in location estimates. This can be shown by ignoring orientation and focusing solely on the slant ranges. While the estimated and true dimensional distances between the receiver origin and the camera origin, shown in Figure 37, are very different due to poor orientation estimation, the slant ranges should be similar because the Euclidean distance between the receiver and camera are independent of orientation. Figure 38 shows the relationship between the estimated and true distances between the receiver and cameras.

The trend line shows a positive correlation between the estimated and true slant ranges. As the cameras get closer to the receiver, the estimates generally reflect the relative motion. To show the effect of orientation estimate accuracy of the accuracy of location estimates, the location estimates were re-computed using the true relative



(a) Estimated Orientation based Location Errors **(b)** True Orientation based Location Errors

Figure 39. The location errors computed by using the true orientations. The location errors greatly improve when the true relative orientations are used to in the computations.

orientations. Figure 39 shows updated location errors when the true orientations are used for calculations.

As expected, the location errors are improved when accurate orientations are used. While the mean error in the x direction increased from -0.1606 to .5735, the mean error in the y dimension decreased from -0.1854 to -0.076 and the z dimension decreased from 3.4827 to -0.9486. Therefore, much of the location estimate errors can be attributed to the unrepresentative point clouds that lead to high orientation estimate errors. However, once this problem is accounted for large errors still exist. This is believed to be the result of the forward motion problem, a known problem in SfM. The error data supports this assumption based on the fact that the receiver approaches the cameras at a constant positive x and negative z direction (rising and moving forward). The error caused from this direction of motion would then be expected to be in the x and z dimensional estimates. Figure 39 shows that this is the case. The mean of the error in the y direction is approximately 3 inches while the means in the x and z directions are much larger at 22.57 inches and 37.34 inches respectively.

While the forward motion problem is rarely seen in most commercial applications of SfM, the recent push for autonomous driving has increased its visibility [78]. Forward motion causes scale change in the domain of the image that can cause significant deformations even while restricting attention to regions that remain visible between images [75]. This leads to the presence of a large number of local minima in the least-squares error of the reprojection [54] [14]. Imposing a bound on the depth of the reconstructed scene has been shown to be a theoretical solution to the forward movement problem [78]. This technique enables many of the local minima to be avoided by showing that they do not really exist. The forward motion problem was practically addressed in a humanoid system by integrating visual and inertial measurements into a single coherent ego-motion state [75]. Both sensor types are passive and have complementary failure modes allowing the inertial sensor to account for the forward motion problem of the visual sensors.

4.6 Discussion

The SfM based approach utilized in this experiment did not facilitate accurate relative positioning estimates. It is believed that the SfM application was able to generate accurate position estimates in its arbitrary world frame. However, the reconstruction was not good enough to generate representative point clouds for registration. Therefore, the positioning accuracy in the SfM frame could not be leveraged to provide accurate positioning estimates in the receiver frame. The poor reconstructions are attributed to the lack of perspectives used to generate the reconstruction. The straight on view of the rear stabilizers make it hard for correspondences to be identified and therefore they don't show up in the reconstruction or point clouds. The loss of this important information has a major effect on the variance contained in the generated point clouds. The lack of variance in the $-z$ direction prevents PCA

from identifying the correct principle components leading to the misalignment of the point clouds and incorrect orientation estimates. Due to the use of the orientation estimates to determine the location estimates, the unrepresentative point clouds also directly reduce the accuracy of the relative location estimates.

Additionally, forward motion, a well known and documented problem with SfM, causes additional error in the location estimates. The straight approach is similar to the expansion of a single perspective and therefore does not add much new information with each image. Figure 38 clearly shows that SfM can recognize the forward movement between images. However the high x and z directional errors, the positional changes on approach, indicate the inability of SfM to determine the exact amount of movement in each direction.

In this experiment all of the error can be contributed to the SfM phase. However, it should be noted that all SfM computations were conducted using the single vSfM application. This application was primarily designed for the purpose of generating 3D reconstructions with very large image sets. Its focus was not on highly accurate position estimations and no additional effort was included to account for forward motion. In AAR, the problem of having minimal perspectives available for reconstruction can be solved by using cameras located on other locations of the tanker, such as out on the wings, farther forward or on the boom. The forward motion problem is also well understood and focus on addressing it has increased due to its effects on autonomous driving. The combined visual and inertial approach discussed above is particularly suited to AAR given the inertial data readily available on both the tanker and receiver platforms.

Nearly all of the error can be attributed to the problems in SfM. The other phases of the proposed solution worked as expected. The high error rates identified in this experiment are not a reflection of the PCA registration's capabilities. PCA identified

and aligned the principal components of each point cloud as expected. However, the major weakness of PCA was underscored in this experiment. Given unrepresentative point clouds, or point clouds with outliers, a PCA based registration can not provide accurate position estimates. This was understood and accepted in order to achieve a faster registration time. If representative and normally distributed point clouds without outliers can not be generated then a slower, but more robust registration algorithm should be used, such as ICP or SVD.

V. 3D Rigid Registration Study

This chapter presents a trade space study focused on the performance of common 3D rigid registration algorithms in the AAR domain. First, the objectives of the study are presented. Next, the relevant assumptions and limitations are stated. Next, each of the registration algorithms are presented and described. Next, the experimental process presented. Next, the results are presented and analyzed. Finally, the chapter is concluded by a discussion on the results in the context of AAR.

5.1 Objectives

It has been shown that there is a trade-off between the accuracy and runtime of 3D rigid registration algorithms [9]. Understanding the performance of each is critical to developing a robust solution for AAR. Additionally, the technique used to generate point clouds can have major effects on the solution's ability to estimate $\Delta^{C_x}_{\text{receiver}_t}$. This study will compare the performance of the three main 3D rigid registration algorithms, ICP, SVD and PCA. To gain a better understanding of each algorithm's performance throughout the AAR space multiple sets of point clouds will be utilized, each generated by a different process. The accuracy and runtime will be recorded to facilitate discussion of the relative performance between algorithms and data sets.

5.2 Assumptions and Limitations

RMS as an error measurement: Truth data was not available for two of the point cloud data sets used in this study. RMS is used to quantitatively measure the accuracy of alignments. It is assumed that the accuracy of a 6DOF position estimate is a function of the alignment accuracy and that a smaller RMS value suggests the ability to generate more accurate position estimates.

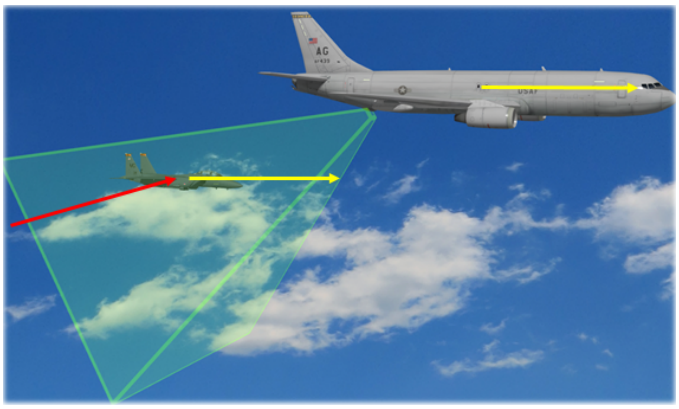


Figure 40. The receiver approaches the tanker with a known orientation. The receiver approaches the tanker in the x and $-z$ directions while maintaining the same relative orientation, modeled by the red arrow. Once the optimal refueling position is achieve the receiver maintains a constant relative positioning during contact, modeled by the yellow arrows.

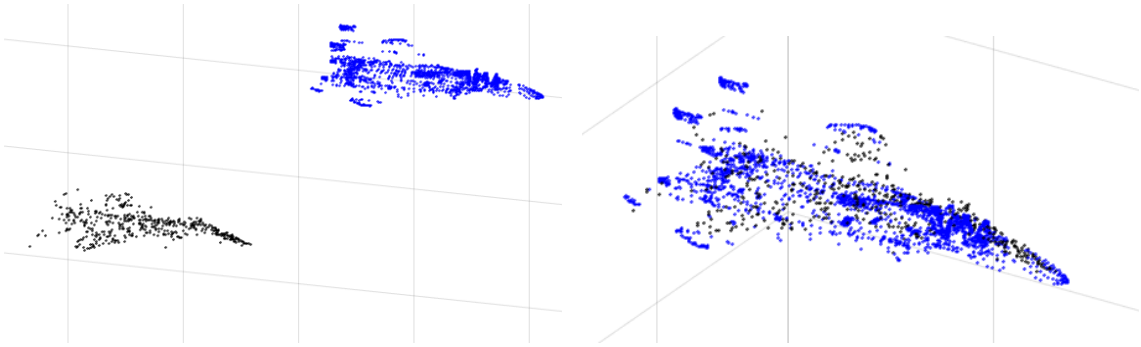
Receiver maintains a known, constant orientation: It is assumed that an AAR mission consists of standard, controlled maneuvers where the receiver aircraft maintains a constant orientation relative to the tanker. On approach, the receiver will utilize a constant translation in the x and $-z$ directions while maintaining its heading (orientation). During contact there will only be nominal relative motion between the receiver and tanker.

5.3 Algorithms

In order to minimize the variables effecting the results of this study, each of the three algorithms were modeled in MATLAB with identical code outside of the individual registration techniques. The PCA registration algorithm utilized in this study is identical to implementation in the relative navigation experiment. This section discusses the specific SVD and ICP implementations used in this study.

5.3.1 Modified SVD algorithm.

The core of the SVD algorithm used in this study is based on the original paper discussed in Section 2.8 and was retrieved from [20]. However, a modification is



(a) Virtual Model (blue) & Point Cloud **(b)** Centroid based Co-location

Figure 41. Generated point cloud in realtive position and co-located with the virtual model. The generated point cloud models the aprraoch of the receiver at a certain time. The same point cloud and virtual model co-located using a centroid based alginment.

necessary to account for the fact that correspondences are not known a priori. There is enough information known about the scenario to estimate point correspondences with reasonable accuracy. The receiver approaches the tanker with a constant translation in the x and z directions. Throughout the mission the receiver will maintain a nearly identical orientation to that of the tanker. Figure 40 illustrates the approach path and relative orientation of the tanker and receiver during AAR.

Due to the constant relative orientation between the tanker and the receiver a virtual model can be used for alignment that exists in the same relative orientation. Simply co-locating the virtual model and generated point cloud by aligning their centroids provides enough of an alignment to generate accurate estimated correspondences. Figure 41 shows the relative position of the virtual model and a generated point cloud of a receiver on approach and the same two point clouds after a centroid based alignment.

It is believed that once the virtual model and generated point cloud are co-located a simple nearest neighbor search will provide estimated correspondences capable of supporting a SVD based registration. The registration will then determine the exact

relative positioning to account for the orientation changes caused by both expected and unexpected changes in the mission environment.

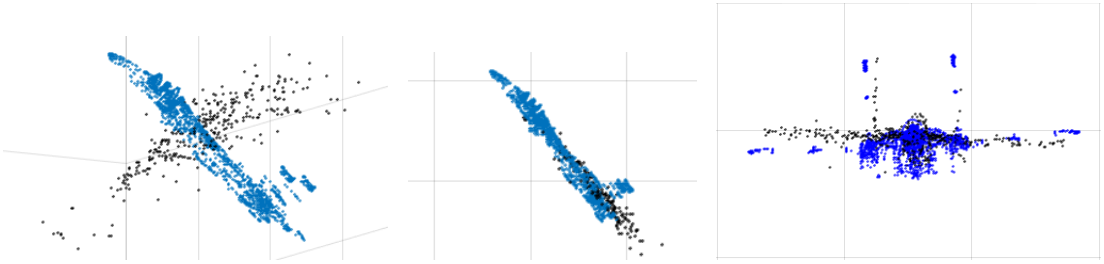
5.3.2 ICP algorithm.

The ICP implementation utilized in this study was originally developed for use in [42] and was retrieved from [43]. Due to ICP being an iterative approach, multiple runs were completed in an effort to establish a relationship between number of iterations and runtime and accuracy.

5.4 Process

In an effort to span the AAR research space, four sets of generated point clouds are utilized. The SfM point clouds from the SfM relative navigation experiment are the first data set used for testing. The second data set is comprised of point clouds generated by a block matching stereo reprojection technique [19]. These point clouds are based on images collected in the same experimental environment used in this thesis. Additionally, stereo reprojection point clouds generated from simulated images, utilized in [81], were used. While these point clouds were generated from data collected in a different environment, they are more representative of the receiving aircraft and provide information on performance in an ideal environment. Finally, a fourth set of point clouds was generated by applying random transformations to the virtual model itself. This provides an ideal data set where the point clouds are exactly representative of the virtual model and provides a best case scenario. The same 5001 point virtual model is used for alignment with each data set.

Due to the lack of true positional data for two of the data sets, accuracy is measured by the RMS distance between the transformed virtual model and the generated point cloud. RMS is a statistical measure defined as the square root of the arithmetic



(a) RMS=4.1299 meters **(b)** RMS=1.265 meters **(c)** RMS= 0.44708 meters

Figure 42. Examples of registrations producing different levels of accuracy. The first registration has an RMS of 4.1299 meters, the second has an RMS of 1.265 meters and the third has an RMS of 0.44708 meters

mean of the squares of the distance between points. RMS values are calculated for each registration by

$$RMS = \sqrt{\frac{\sum_{1}^{n} ||\dot{\text{receiver}}_i - \dot{P}_{m,i}||^2}{n}} \tag{46}$$

where n is the number of points in the generated point cloud, $\dot{\text{receiver}}_i$ is a point in the generated point cloud and $\dot{P}_{m,i}$ is the nearest point in the virtual model to $\dot{\text{receiver}}_i$. This metric is chosen because it is independent of the number of points contained in the point clouds. Lower RMS values represent a better registration. Figure 42 illustrates registrations with the associated RMS values.

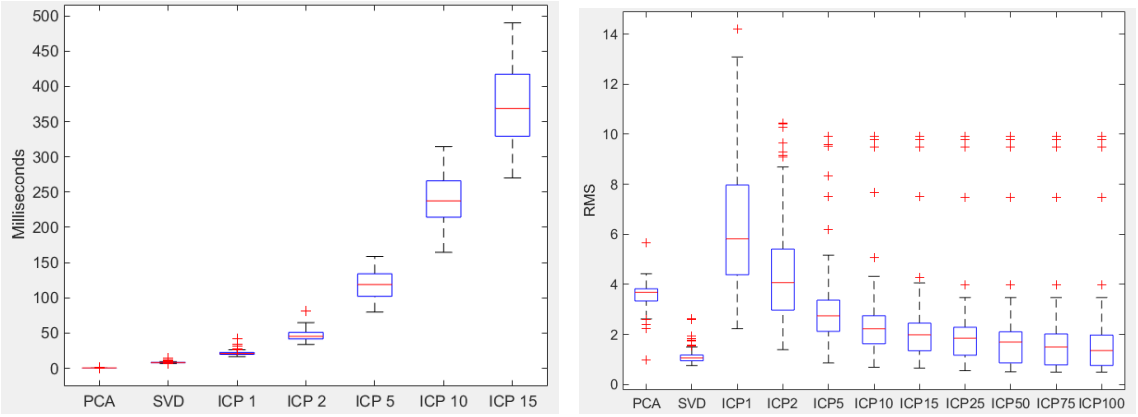
After the registration algorithms estimate $\Delta_{P_m}^{\text{receiver}_t}$, the transformation is applied to the virtual model. The MATLAB functions knnsearch() and rms() are then used to determine the nearest neighbors and RMS values. The run times are determined at the microsecond level using the MATLAB tic() and toc() functions.

5.5 Results & Analysis

5.5.1 SfM Point Cloud Data Set.

The same point clouds used in the proposed approach experiment were reused in this section in an effort to leverage the conclusions drawn about the PCA based

registration algorithm in the experiment. These point clouds do not include rear stabilizers and contain many outliers. An example of the point clouds included in this data set can be seen in Figure 32. A total of 120 SfM generated point clouds were used. They contained an average of 344 points with a maximum of 607 points and a minimum of 155 points. The runtimes and accuracy of each algorithm are presented in Figure 43. The mean and standard deviation of the run times and accuracies are shown in Table 1. The ICP naming convention identifies the number of iterations used such that ICP 15 indicates the ICP algorithm was used for 15 iterations. ICP 25, 50, 75 and 100 were left off of the run time chart to ensure the differences between the PCA, SVD and ICP 1 run times remain visible.



(a) Registration Algorithm Runtimes **(b)** Registration Accuracy

Figure 43. The runtime and accuracy of registration algorithms using SfM generated point clouds. The PCA algorithm was the fastest followed by the SVD algorithm. However, the SVD algorithm was much more accurate than both PCA and ICP. ICP could not achieve the accuracy of SVD after 100 iterations.

The run time results are as expected with PCA and SVD being the fastest and ICP increasing at a constant rate as the amount of iterations increase. While PCA should be by far the fastest, it was expected that the SVD and ICP 1 algorithms would be much closer because they are both a least squares fitting of estimated nearest neighbors. The large difference in mean run time (8.1698ms vs 21.4354ms) is

Table 3. Run time and accuracy of registration algorithms with SfM point clouds

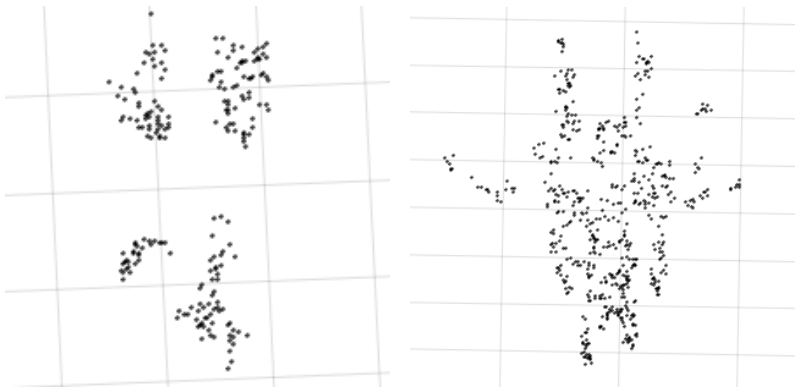
Algorithm	Run Time(ms)	Accuracy(RMS)
PCA	0.4664(±0.1181)	3.5781(±0.4986)
SVD	8.1698(±1.0755)	1.1292(±0.3013)
ICP 1	21.4354(±3.5250)	6.2756(±2.5222)
ICP 2	46.5608(±7.3197)	4.4606(±2.0819)
ICP 5	117.3177(±19.3816)	3.0178(±1.6210)
ICP 10	238.8407(±36.0451)	2.4225(±1.5267)
ICP 15	370.7458(±51.7846)	2.2006(±1.5275)
ICP 25	614.8(±81.9967)	1.9949(±1.5403)
ICP 50	1252.4(±152.6862)	1.8133(±1.5664)
ICP 75	1780.3(±260.6574)	1.7934(±1.7146)
ICP 100	2439.0(±311.6705)	1.7188(±1.5892)

not a function of the approaches but of the specific implementations. It is believed that the MATLAB knnsearch() used as part of the SVD algorithm is more efficient than the nearest neighbor search implemented in the ICP algorithm. Therefore, a more efficient ICP implementation would be much closer to the runtime of the SVD algorithm for a single iteration. The accuracy data is also as expected. The high PCA RMS values reflect the alignment issues identified in the previous experiment. The SVD algorithm performs the best. This is due to the accuracy of the estimated correspondences after the virtual model is collocated with each point cloud. ICP performed the worst, ans was not able to meet the accuracy of SVD after 50 iterations. ICP's poor accuracy is due to the existence of local minima when the generated point cloud is not represetnative of the rear stabilizers. For the SfM data set, SVD is the best approach. While the PCA algorithm is much faster (0.4664ms vs 8.1698ms) its

accuracy is much worse. However, if the generated point clouds were more representative of the receiving aircraft and similar to the virtual model, PCA would have performed much better without an increase in runtime. This conclusion was shown in the proposed approach experiment.

5.5.2 Stereo Reprojection Data Set.

The point clouds contained in this data set were generated for a seperate experiment utilzing images from the same experimental environment used in this thesis [19]. A block matching stereo reprojection technique was used to generate the point clouds. A total of 99 point clouds were used in this data set. They contained an average of 510 points with a maximum of 891 points and a minimum of 213 points. The point clouds in this data set have a wide range of qualtiy with the initial point clouds (cameras farther from the receiver) being very unrepresentative and the later point clouds being very representative. Figure 44 shows the first and 90th point clouds from the camera's perspective.

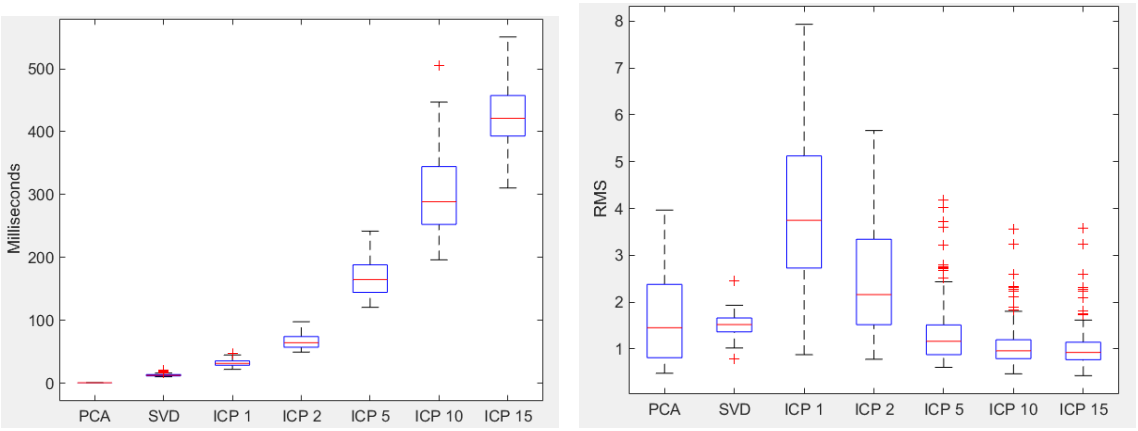


(a) Initial Point Cloud **(b)** Final Point Cloud

Figure 44. Sample point clouds from the stereo reprojection data set. The first point cloud generated by stereo re projection is unrecognizable. The final point cloud in the data set is more representative of the receiver though the wings are still under represented.

The initial point cloud is unrecognizable but is believed to contain points from the

rear stabilizers at the top and some points along the center of the body at the bottom. The 90th point cloud is more representative of the receiver and includes however, the wings are not equally represetnative. Many of the point clouds in this data set are lacking points from the wings. This is expected to induce similar errors as the last data set in for the PCA algorithm. The run times and accuracy of each algorithm are presented in Figure 45. The mean and standard deviation of the run times and accuracies are shown in Table 2. ICP 25 and ICP 50 were not considered with this data set because the ICP algorithm was able to produce more accurate registrations than both PCA and SVD with 5 iterations.



(a) Registration Algorithm Runtimes **(b)** Registration Algorithm Accuracy

Figure 45. The runtime and accuracy of registration algorithms with stereo reprojection generated point clouds. The PCA algorithm was the fastest followed by SVD and then ICP. The SVD algorithm was the most accurate though ICP was able to achieve better accuracy with 5 iterations.

The mean run times all increased on this set of data due to the greater number of points in the point clouds. While the PCA algorithm does not require a nearest neighbor search, its runtime is still dependent on the number of points due to the use of the covariance matrix which multiplies the matrix form of the point cloud. The average number of points increases by 48% (344 to 510), leading to a similar increase in run time for all three algorithms; 52% for PCA, 57% for SVD and 48% for

Table 4. Run time and accuracy of registration algorithms with Stereo Reprojection point clouds

Algorithm	Run Time(ms)	Accuracy(RMS)
PCA	0.7131(±0.1411)	1.6709(±0.9917)
SVD	12.8605(±1.7716)	1.5175(±0.2369)
ICP 1	31.8619(±4.9793)	3.9385(±1.6153)
ICP 2	66.5901(±10.2256)	2.82(±1.2368)
ICP 5	168.1202(±26.9767)	1.3891(±0.7649)
ICP 10	301.8012(±62.9133)	1.229(±0.5441)
ICP 15	423.6173(±50.8338)	1.0869(±0.5479)

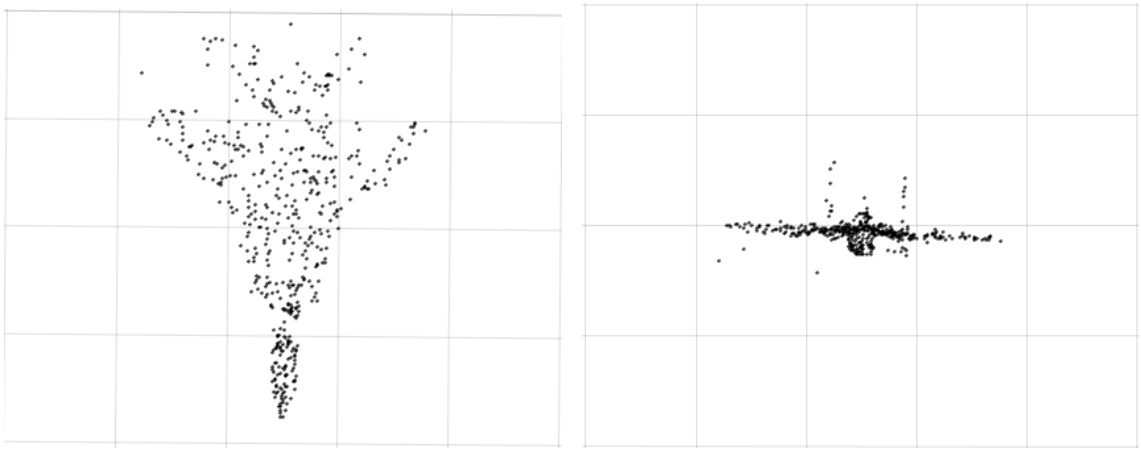
each ICP iteration. Additionally, the ICP algorithm maintains a linear time increase as the number of iterations increase. The small sample sizes in the two data sets prevent these increases from being considered benchmarks, however they provide a useful generalization on the relationship between point cloud size and algorithm run time.

The accuracy generally improved on this data set. The PCA algorithm was much more accurate, reducing its RMS value approximately 46% on average. This suggest that the majority of the stereo reprojection generated point clouds had principal components similar to the virtual model. However, the standard deviation increased by about 98% illustrating the poor quality of the initial point clouds relative to the later point clouds. The SVD algorithm was slightly less accurate on this data set than the SfM generated point clouds. However, it still performed better and was much more consistent than the PCA algorithm on average. This indicates that the correspondence estimation step is very robust and able to accurately handle partial point clouds. The ICP algorithm's natural robustness to outliers an ability to align partial point clouds is shown in this data set. While the first iteration has a high RMS

value, the registration converges quickly (in few iterations) and is able to achieve an RMS value lower than both PCA and SVD in 5 iterations.

5.5.3 Stereo Reprojection PCs from simulated imagery.

The point clouds contained in this data set were originally generated for a previous AAR research effort [81]. The images used were generated by a simulation of an aerial refueling. While a similar block matching stereo reprojection technique to the last data set was used, the point clouds in this data set are much more representative of the actual receiving aircraft. Figure 46 shows a typical point cloud from this data set.

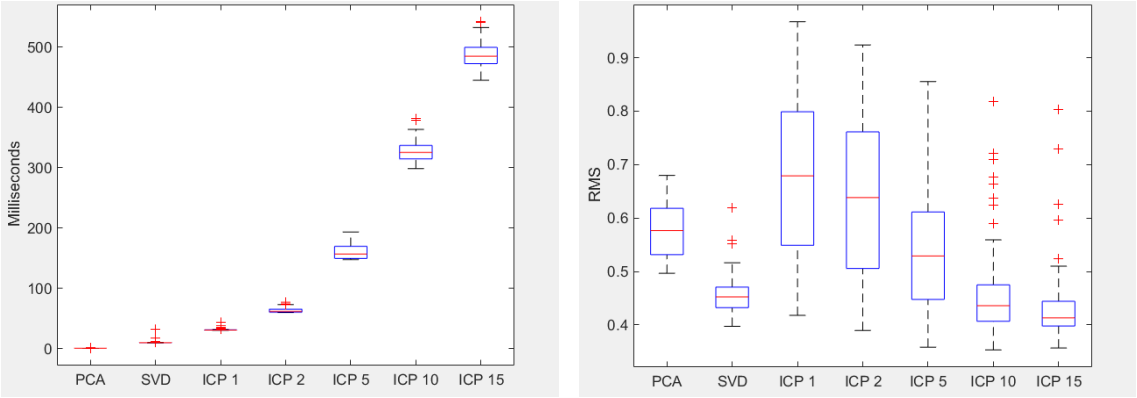


(a) Top Perspective **(b)** Front Perspective

Figure 46. A sample point cloud from the simulated stereo reprojection data set.

The point clouds in this data set are the most representative of the receiver. While there is some noise is many of the point clouds the entire receviing aircraft is represented throughout the entire data set. A total of 304 point clouds were used in this data set. Each point cloud contains exactly 500 points. The run times and accuracy of each algorithm are presented in Figure 47. The mean and standard deviation of the run times and accuracies are shown in Table 3. Again, ICP 25 and

ICP 50 were not considered with this data set because the ICP algorithm was able to produce more accurate registrations than both PCA and SVD with 10 iterations.



(a) Registration Algorithm Runtimes

(b) Registration Algorithm Accuracy

Figure 47. The runtime and accuracy of registration algorithms with stereo rpejection generated point cluods from simulated imagery. PCA was again the fastest algorithm followed by SVD and then ICP. The overall accuracy fo the algorithms was far better on the simulated data set. ICP required 5 iterations to achieve the PCA accuracy adn 10 iterations to achieve the SVD accuracy.

Table 5. Run time and accuracy of registration algorithms with stereo reprojection point clouds from simulated imagery

Algorithm	Run Time(ms)	Accuracy(RMS)
PCA	0.7102(±0.0802)	0.5758(±0.0493)
SVD	10.2941(±2.2173)	0.4553(±0.0351)
ICP 1	31.3663(±1.4525)	0.6831(±0.1382)
ICP 2	63.5284(±3.9771)	0.6434(±0.1349)
ICP 5	160.5530(±11.8782)	0.5407(±0.1056)
ICP 10	326.8919(±16.2147)	0.4575(±0.0794)
ICP 15	485.6474(±20.8700)	0.4325(±0.0648)

The run times of this data set are similar to those of the stereo reprojection data set. This was expected given the similar point clouds size (500 points vs 510 points).

Like the previous two data sets, the ICP algorithm's time increases linearly with the number of iterations. Thea accuracy of the registrations with this data set are by far the best. All of the algorithms produce more accurate registrations with this data set than any one could produce on average in the previous data sets. This result underscores the importance of generating representative point clouds and suggest that the quality of the generated point clouds has a greater effect on accuracy than the registration approach.

5.5.4 Ideal Data Set.

The point clouds contained in this data set were generated by applying random transformations to the virtual model. The generated point clouds represent an ideal data set because each point cloud has an exact correspondence in the virtual model. In the previous data sets, the point clouds consisted of features identified on the surface of the receiver. These were then fit a virtual model that included points located within and under of the receiver. This enabled the algorithms to achieve low RMS values by aligning the generated point clouds with the more dense internal locations in the virtual model. Figure 48 illustrates this concept.

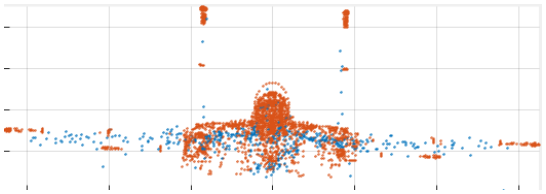
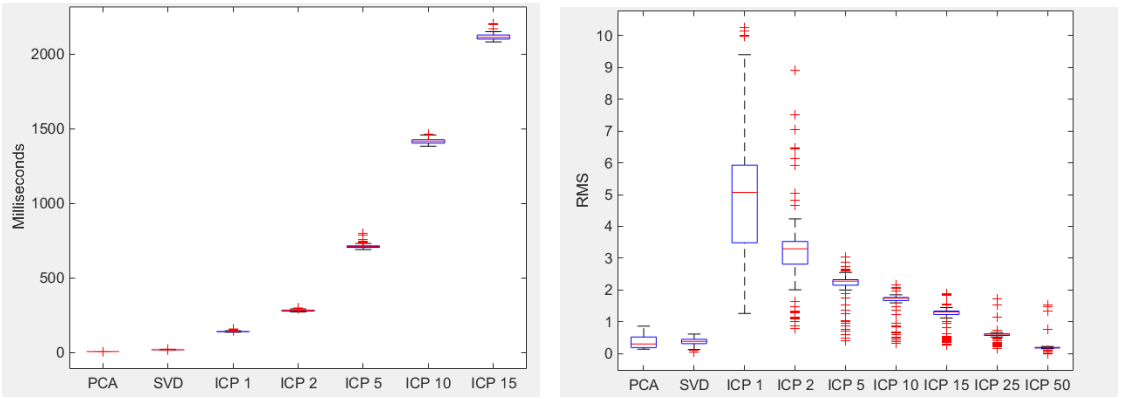


Figure 48. An alignment with artifically low RMS value. The registration algorithm aligned the generated point cloud within the virtual model. The RMS computation based on nearest neighbors will produce a low value, suggesting an accurate alignment. A true registration would align the pint cloud to the top of the virtual model.

The transformations used to generate the point clouds were determined by randomly inducing orientations changes between -5 and 5 degrees about all three axes. Additionally, the translations were generated by randomly choosing offsets in each

direction though an effort was made to keep the relative positioning within the scope of the AAR scenario. With respect to the virtual models reference frame, the random translations were bounded in pixels by [-50 -5] in the x direction, [-5 5] in the y direction and [-5 25] in the z direction. A total of 100 point clouds were generated for this data set. They contained the same 5001 points contained in the virtual model. The run times and accuracy of each algorithm are presented in Figure 49. The mean and standard deviation of the run times and accuracies are shown in Table 4.



(a) Registration Algorithm Runtimes

(b) Registration Algorithm Accuracy

Figure 49. The runtime and accuracy of registration algorithms with ideal point clouds. The relative runtime performance remained the same across all data sets. SVD was slightly more accurate the PCA which is surprising considering the point clouds had identical principal components as the virtual model. ICP required 50 iterations to achieve the accuracy of the PCA and SVD algorithm.

The increase of points contained in the point clouds from approximately 500 to 5000 initially appears to have had a much larger effect on the PCA and ICP algorithms when compared to the SVD algorithm. However, this is believed to be related to the efficiency of the MATLAB knnsearch() function used to estimate correspondences in the SVD implementation. This function appears to scale much better than the nearest neighbor search utilized in the ICP implementation. However, the PCA algorithm is still by far the fastest. The ICP algorithm maintains the linear growth in time

Table 6. Run time and accuracy of registration algorithms with ideal point clouds

Algorithm	Run Time(ms)	Accuracy(RMS)
PCA	4.7284(±0.3053)	0.3777(±0.2050)
SVD	16.720(±1.2284)	0.3774(±0.1125)
ICP 1	139.0(±2.9239)	5.0593(±2.3346)
ICP 2	280.0(±5.2809)	3.2621(±1.359)
ICP 5	711.1(±16.5669)	2.0951(±0.5383)
ICP 10	1415.8(±16.6051)	1.5957(±0.4163)
ICP 15	2116.7(±21.3136)	1.212(±0.3473)
ICP 25	3528.9(±28.2423)	0.6316(±0.3028)
ICP 50	7021.2(±44.0616)	0.2464(±0.3275)

associated with additional iterations, though the runtime of each iteration grew to approximately 140ms.

With respect to accuracy, it surprising to see that none of the algorithms were able to achieve near 0 RMS values given that the point clouds were identical to the virtual model. In the previous experiment is was assumed that if representative point clouds were generated the PCA algorithm would have been able to accurately determine the relative positioning. However, the results from this data set suggest that there may be additional errors resulting form the PCA algorithm. It is also interesting to note the similar accuracies provided by both the SVD and PCA algorithms. This suggest that the registration based on the similarities between the covariance of two point clouds can be just as accurate as rotations based on the cross-covariance between point clouds, without the need for an expensive nearest neighbor search. However, the additional information available to the SVD algorithm may support the smaller standard deviation. This data set also showed that the ICP algorithm can provide

better accuracy than the PCA and SVD algorithms given enough time. The fact that it took between 25 and 50 iterations to provide greater accuracy than the other two algorithms while only taking 5 or 10 in the previous data sets is a reflection on the abilities of SVD and PCA given representative point clouds. This illustrates an important relationship between point cloud quality, accuracy and run time.

5.6 Discussion

This study illuminated many important relationships with respect to 3D rigid registrations within the AAR scenario. With respect to runtime, PCA is the fastest. This is directly related to not requiring an expensive nearest neighbor search. However, its runtime is still related to the size of the generated point clouds do to the calculation of the covariance matrix. SVD is much slower than PCA on all data sets, however its runtime of 8-17ms in this study shows that a slightly more efficient implementation may be able to support the AAR time requirement. The ICP algorithm is the slowest. Each iteration requires a nearest neighbor search where the time complexity grows with the size of the point cloud. The relationship between runtime and iterations is strictly linear allowing for an accurate estimate of the runtime associated with a certain amount of iterations.

An important observation with respect to accuracy is that all algorithms were more accurate on the data sets consisting of quality point clouds. RMS values less than 0.5 were achieved for the ideal and simulated data sets. The data sets derived from real world imagery were much less representative of the receiver and produced RMS values ranging from 1.0869 to 3.57 for the each algorithm. For the data sets considered in this study, SVD was consistently the most accurate. With the exception of the SfM data set, SVD was only slightly better than PCA, however PCA never produced lower mean RMS values. Additionally, SVD was always more consistent

than PCA, producing smaller values for the standard deviation of RMS with every data set. ICP eventually exceeded the accuracy of SVD and PCA on three of the data sets, requiring 5, 10 and 50 iterations. On the SfM data set, ICP could not achieve the accuracy of SVD after 100 iterations. There is a positive correlation between the quality of the generated point clouds and the number of iterations required for ICP to achieve the accuracy of SVD.

Based on the results of this study SVD is the best algorithm for the AAR scenario. While the PCA is much faster, it can not achieve the accuracy and consistency provided by SVD, even with an ideal data set. ICP was able to eventually achieve the accuracy of SVD, however, the additional iterations lead to elevated runtimes unsuitable for AAR. However, it is important to note that as the quality of the point clouds degrades ICP required less iterations to achieve the accuracy of SVD. The success of the SVD algorithm in this study is directly related to the initial relative orientation of the virtual model with respect to the generated point cloud. Due to the specific and constant angle of approach by the receiver in the AAR scenario, the virtual model can be developed to have a similar relative orientation to the camera frame. This generates accurate correspondence estimates after the virtual model is colocated with the generated point cloud.

VI. Conclusion & Future Work

This thesis presents a method for relative positioning estimation for aerial refueling using SfM. In this chapter, conclusions regarding the research and analysis are discussed. Additionally, recommendations for future research are presented to facilitate continued progress toward AAR.

6.1 Conclusion

The SfM based relative positioning solution was not successful in accuratly estimating receiver positions. However, SfM has the potential to contribute to a robust relative navigation solution. It provides unique capabilities and the primary sources of error in this thesis are well understood and documented with available solutions. These solutions combined with the scenario specific performance characterizations identified in this thesis provide a solid foundation for a successful SfM implementation.

Preliminary experimentation highlighted the importance of the number of perspectives. The superior performance of a stereo system over a monocular system showed that even a slight offset produces more useful information. Additionally, the equivalent accuracy of down sampled image sets showed the lack of new useful information provided by images taken from essentially the same perspective, foreshadowing the forward motion problem. An additional optimization of SfM for AAR is the use of a seeded sliding feature matching scheme. Because not much additional information is obtained in successive images, a full pairwise feature matching is not necessary. Matching a new image to a small amount of past images will reduce runtimes while still utilizing previously available information. A capability unique to SfM, is the ability to generate full point clouds for images that have large amounts of occlusion.

Due to its ability to utilize information from past images, if the the receiver moves out of the field of view or is occluded by another object (such as the boom) point cloud quality will not be affected.

The proposed approach experiment highlighted the primary problems with a SfM based solution. The lack of perspectives in the image data prevented an accurate reconstruction and consequently poor quality point clouds. Specifically, the rear stabilizers of the receiver were not reconstructed because the single, straight on perspective prevented correspondences between images. The PCA registration algorithm was not able accurately align the generated point cloud to the virtual model, producing relative orientation estimates with high errors. The relative orientation estimates also had high errors. This was partially caused by using the orientation estimates to calculate the location estimates. When the true orientation data was utilized to computed the location, the location errors decreased. While this correction allowed to the algorithm to generally estimate the direction of movement between images, the estimates still had an abundance of error in the x and z locations but not the y location. This additional error was the result of the forward motion problem, characterized by the receivers constant approach in the x and $-z$ directions. Forward motion is a well documented problem in SfM. The abundance of local minima prevent an accurate estimate of camera locations within the reconstruction. The performance of the PCA registration algorithm was unable to be characterized given the poor quality point clouds generated by SfM. However, the runtime averaged less than $\frac{1}{2}$ms over the entire experiment, well within the ideal time for supporting AAR.

The 3D rigid registration study showed that the modified SVD algorithm was the most efficient when considering runtime and accuracy. The PCA algorithm was by far the fastest throughout the entire study. However, PCA was never more accurate than SVD, even on ideal set of point clouds that were exact replications of the virtual

model. The fact that PCA could not align the principal components of identical point clouds suggest that it was not as accurate as it was assumed to be in the proposed approach experiment. The ICP algorithm was able to achieve better accuracy than PCA and SVD on three of the four data sets, however it took between 5 and 50 iterations. There was also a positive correlation between the quality of the generated point clouds and the number of iterations required by ICP to achieve the accuracy of SVD. The runtime of both SVD and ICP is heavily dependent on the size of the point clouds being considered. However, the ICP runtime increases linearly with the amount of iterations.

6.2 Future Work

This section discusses future work that can leverage this thesis and further AAR research. This thesis marks the first time SfM has been considered in AAR research, therefore there are many different areas of research available for future analysis.

6.2.1 SfM Implementation optimized for AAR.

This thesis relied on the open source vSfM application to generate point clouds. However, this application was developed as a visual 3D reconstruction system for operating on very large sets of images. It was not designed for relative navigation and is therefore not optimized for an AAR scenario. The development of an AAR specific SfM implementation that accounts for dSfM, forward motion, and minimal perspectives may provide far better reconstructions and therefore positioning estimates. Additionally, an AAR specific implementation may account for many of the other stages necessary in the approach proposed in this thesis. The SfM to image frame transformation can easily be included in the reconstruction leading to a more efficient algorithm by minimizing the amount of step necessary to generate estimates.

Some real-time and near real-time SfM algorithms have been successfully implemented for determining the ego-motion of an object [52, 18, 51].

6.2.2 Sensor Fusion.

Of the three major problems identified in this thesis, two have known solutions that can be implemented with the use of additional senors. The problem of a small number of perspectives can be solved with additional vision sensors. A single camera located away from the center line of the tanker will produce an additional perspective enabling a higher quality reconstruction.

The forward motion problem is well understood and has become a major research focus with the rise of the autonomous vehicle effort. A successful solution was implemented in a humanoid system using a cheap inertial sensor to determine ego-motion [75]. Simultaneous use of inertial and imagery data forms a robust system because both of the sensors are passive and have complimentary failure modes. The abundance of inertial data available on military aircraft makes this approach a low hanging fruit that can potentially negate much of the errors in this thesis.

6.2.3 Solving the Dynamic Structure from Motion Problem.

The primary barrier to the successful implementation of SfM based relative positioning is the dSfM problem. While the problem is well understood and documented any proposed solutions are computationally expensive. The proposed image segmentation and masking solution was not implemented in this thesis and therefore is not known to work. Many of the useful features were along the edge of the wings which could be problematic if segmentation is not consistent. Many other solutions have been proposed including optical flow based methods [39, 45] and using epipolar criterion to segment the features and individually reconstructing objects [73].

Bibliography

1. "First Air-to-Air Refueling". World Wide Web Page. Available at *https://www.nationalmuseum.af.mil/factsheets/factsheets*.

2. "Pinhole-camera.svg". World Wide Web Page. Available at *https://commons.wikimedia.org/wiki/File:Pinhole-camera.svg*.

3. *An Overview of Flight Test Results for a Formation Flight Autopilot*. Technical Report Technical Report NASA/TM-2002-210729, Edwards Air Force Base, 2002.

4. *Air Force Aerial Refueling Methods: Flying Boom versus Hose-and-Drogue*. Technical report, Congressional Research Service, 2006.

5. Arun, K.S., Thomas Huang, and Steven Blostein. "Least-Squares Fitting of Two 3-D Point Sets". *IEEE Transactions n Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

6. Barfield, A.F. "An Equivalent Model for UAV Automated Aerial Refueling Research". *AIAA Modeling and Simulation Technologies Conference*, August, 2005.

7. Beardsley, Paul, Andrew Zisserman, and David Murray. "Navigation using affine structure from motion". *Third European Conference on Computer Vision*, 801:85–96, 1994.

8. Bellekens, Ben. "A Benchmark Survey of Rigid 3D Point Cloud Registration Algorithms". *International Journal of Advances in Intelligent Systems*, 118–127, 2015.

9. Bellekens, Spruyt V. Berkvens R., B. and M. Weyn. "A Survey of Rigid 3D Pointcloud Registration Algorithms". *AMBIENT 2014: The Fourth International Conference on Ambient Computing, Applications, Services and Technologies*, 8–13, 2014.

10. Besl, Paul and Neil McKay. "A method for registration of 3-D shapes". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992.

11. Calhoun, S.M. "Flight Test Evaluation of Image Rendering Navigation for Close-Formation Flight". *Proeedings fo the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation*, 826–832, 2012.

12. Casselman, Bill. *Mathematical Illustrations: A manual of Geometry and PostScript*. Cambridge University Press, 2004.

13. Chang, F. "The Hausdorff Distance Template Matching Algorithm Based on Kalman Filter for Target Tracking". *Proceedings of the IEEE International Conference on Automation and Logistics*, 2009.

14. Chiuso, Alessandro, Roger Brockett, and Stefano Soatto. "Optimal Structure from Motion: Local Ambiguities and Global Estimates". *International Journal of Computer Vision*, 39(3):195–228, 2000.

15. Clifford, William. "Applications pf Grassmann's Extensive Algebra". *American Journal of Mathematics*, 1(4):350–358, 1878.

16. Crandall, David, Andrew Owens, Hoah Snavely, and Dan Huttenlocher. "Discrete-continuous optimization for large-scale structure from motion". *Computer Vision and Pattern Recognition*, 3001–3008, 2011.

17. Curro, Joseph. *Automated Aerial Refueling Position Estimation Using A Scanning LiDAR*. Master's thesis, Air Force Institute of Technology, 2012.

18. Davison, Andrew J. "Real-time simultaneous localisation and mapping with a single camera". *Proceedings of the Ninth International Confrerence on Computer Vision*, 1403–1410, 2003.

19. Denby, Bradley. *Toward automated aerial refueling: Real-time position estimation with stereo vision*. Master's thesis, Air Force Institute of Technology, 2016.

20. Evans, Frank. "ROT3DFIT". MATLAB Newsgroup, November 2001.

21. Fischler, Martin and Robert Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartogrophy". *Communications of the Association for Computing Machinery*, 24(6):381–395, 1981.

22. Fitzgibbon, A. and A. Zisserman. "Multibody Structure and Motion: 3-D Reconstruction of Independently Moving Objects". *European Conference on Computer Vision*, 891–906, 2000.

23. Force, United States Air. "Air Force Future Operating Concept: A View of the Air Force in 2035", September 2015.

24. Fravolini, Marco Mammarella Giampiero Campa Marcello Napolitano, Mario and Mario Perhinschi. "Machine Vision Algorithms for Autonomous Aerial Refueling for UAVs Using the USAF Refueling Boom Method". *Inovations in Defence Support Systems*, 95–138, 2010.

25. Fravolini, M.L. "Evaluation of Machine Vision Algorithms for Autonomous Aerial Refueling for Unmanned Aerial Vehicles". *Journal of Aerospace Computing, Information, and Communication*, 4, 2007.

26. Gavin, Henri. "The levnberg-marquardt method for nonlinear lest squares curve-fitting problems". *Department of Civil and Environmental Engineering, Duke University*, 1–15, 2013.

27. Giulietti, F. "Autonomous Formation Flight". *IEEE Control Systems Magazine*, 20(6):34–44, 2000.

28. Graßmann, Hermann. "Die Lineale Ausdehnungslehre - Ein neuer Zweig der Mathematik". 1844.

29. Großekatthöfer, Karsten. "Introduction into quaternions for spacecraft attitude representation". *Department of Astronautics and Aeronautics, Technical University of Berlin*, 2012.

30. Hamilton, William Rowan. *Lectures on Quaternions*. Hodges and Smith, Dublin, Ireland, 1853.

31. Hardinger, Crystal and Katie Shatzer. "Refueling innovation". *Rockwell Collins horizons*, 16(4), 2001.

32. Harris, C.G. "Determination od ego-motion from matched points". *Third Alvey Vision Conference*, 189–192, 1987.

33. Harris, CG and JM Pike. "3D positional integration from image sequences". *Third Alvey Vision Conference*, 233–236, 1987.

34. Hartley, Richard and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, 2003.

35. Hinchman, J. "Automated Aerial Refueling". *Presentation to Aerial Refueling Systems Advisory Group*, April 2007.

36. Hotelling, H. "Analysis of a complex of statistical variables into principle components". *Journal of Educational Psychology*, 24(6):417–441, 1933.

37. Hotelling, H. "Relations between two sets of variates". *Biometrika*, 27:321–377, 1936.

38. Huang, T.S., Steven Blostein, and E.A. Margerum. "Least-squares estimation of motion parameters from 3-D point correspondences". *Proceedings fromIEEE Conference on Computer Vision and Pattern Recognition*, 1986.

39. Irani, M. and P. Anandan. "A unified appraoch to moving object detection in 2D and 3D scenes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):577–589, 1998.

40. J., Garamone. "From US Civil War to Afghanistan: A Short History of UAVs". World Wide Web Page. Available at *https://www.defense.gov/news/newsarticle.aspx?id=44164* April, 2002,.

41. Khanafsesh, Samer. and Boris Pervan. "Autonomous Airborne Refueling of Unmanned Air Vehicles Using the Global Positioning System". *Journal of Aircraft*, 44(5):1670–1682, 2007.

42. Kjer, Hans Martin. *Evaluation of surface registration algorithms for PET motion correction*. Master's thesis, Technical University of Denmark, 2010.

43. Kjer, Hans Martin. "Iterative Cosest Point". Mathworks File Exchange, Jan 2013.

44. Lachapelle, H. Sun M.E. Cannon, Gerard and Gehao Lu. "Precise Aircraft-to-Aircraft Positioning Using a Multiple Receiver Configuration". *Institute of Navigation National Technical Meeting 1994*, 793–799, 1994.

45. Lourakis, M., A. Argyros, and S. Orphanoudakis. "Independent 3D motion detection using residual parallax normal flow fields". *Proceedings of Internation COnference on COmputer Vision*, 1998.

46. Lowe, D. "Fitting Three-Dimensional Models to Images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5), 1991.

47. Lowe, D. "Distinctive image features from scale-invariant keypoints". *International Journal of Computer Vision*, 60:91–110, 2004.

48. Mammarella, Giampiero Campa Marcello Napolitano, Marco and Mario Fravolini. "Comparison of point matching algorithms for the UAV aerial refueling problem". *Machine Vision and Applications*, 21:241–251, 2010.

49. Mammarella, Giampiero Campa Marcello Napolitano Mario Fravolini Yu Gu, Marco and Mario Perhinschi. "Machine Vision/GPS Integration Using EKF for the UAV Aerial Refueling Problem". *IEEE Journal Transactions on Systems, Man, and Cybernetics*, 38(6):791–801, 2008.

50. Mohr, Roger, Francoise Veillon, and Long Quon. "Relative 3D reconstruction using multiple uncalibrated images". *Conference on Computer Vision and Pattern Recognition, Proceedings of*, 543–548, 1993.

51. Molton, Nicholas, Andrew Davison, and Ian Reid. "Locally Planar Patch Features for Real-Time Structure from Motion". *Proceedings of the British Machine Vision Conference*, 2004.

52. Mouragnon, M. Lhuillier M. Dhome F. Dekeyser, E. and P. Sayd. "Generic and real-time structure from motion using local bundle adjustment". *Image and Vision Computing*, 27(8), 2009.

53. Nordmann, Arne. "Epipolar geometry". World Wide Web Page. Available at *https://commons.wikimedia.org/wiki/File:Epipolar_geometry.svg*.

54. Oliensis, John. "The Least-Squares Error for Structure from Infinitesimal Motion". *International Journal of Computer Vision*, 61(3):259–299, 2005.

55. Olson, C. "Probabilistic Self-Localization for Mobile Robots". *IEEE Transactions Robotics and Automation*, 16(1), 2000.

56. Orton, M. "Air Force remains committed to unmanned aircraft systems". World Wide Web Page. Available at *https://www.af.mil/news/story.asp?id=123131324* January, 2009,.

57. Pachter, M. "Tight Formation Flight Control". *Journal of Guidance, Control, and Dynamics*, 24(2):246–254, 2001.

58. Pachter, M., J.J. D'Azzo, and J.L. Dargan. "Automatic Formation Flight Control". *Journal of Guidance, Control, and Dynamics*, 17(6):1380–1383, 1994.

59. Pearson, K. "On Lines and Planes of Closest Fit to systems of Points in Space". *Philosophical Magazine*, 2(11):559–572, 1901.

60. Reichman, W. J. *Usa and Abuse of Statistics*. Pelican, 1970.

61. Rodrigues, Olinde. "Des lois géométriques qui régissent les déplacements d'un systéme solide l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire." *Journal de Mathématques*, 380–440, 1840.

62. Ross, Steven M. *Formation Flight Control for Aerial Refueling*. Master's thesis, Air Force Institute of Technology, 2006.

63. Sarvaiya, J.N. "Image Registration By Template Matching Using Normalized Cross-Correlation". *International Conference on Advances in Computing, Controls & Telecommunication Technologies*, 2009.

64. Schindler, K. and V. Clayton. "Two-view multibody structure-and-motion with outliers through model selection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):983–995, 2006.

65. Schneider, Michael Himmelsbach Thorsten Luettel, Sebastian and Hans-Joachim Wuenshe. "Fusing vision and LIDAR - Synchronization, correction and occlusion reasoning". *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 388–393, 2010.

66. Shapiro, Linda G. *Computer Vision*. Prentice Hall, Upper Saddle River, New Jersey, 2001.

67. Shlens, J. *A Tutorial on Prinicple Component Analysis: Derivation, Discussion and Singular Value Decomposition*. University of California, San Diego, March 2003. Available at *https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf*.

68. Snavely, Noah, Steven Seitz, and Richard Szeliski. "Photo Tourism: exploring photo collections in 3D". *ACM transactions on graphics*, 835–846, 2006.

69. Soloviev, A. and H deHaag. "Three-Dimensional Navigation with Scanning Ladars: Concept & Inital Verification". *Aerospace and Electronic Systems, IEEE Transactions on*, 46(1):14–31, 2010.

70. Spencer, James H. *Optical Tracking for Relative Positioning in Automated Aerial Refueling*. Master's thesis, Air Force Institute of Technology, 2007.

71. Spinelli, Christopher. *Development and Testing of a High-Speed Real-Time Kinematic Precise DGPS Positioning System Between Two Aircraft*. Master's thesis, Air Force Institute of Technology, 2006.

72. Tharp, Justin S. *On the Integration of Medium Wave Infrared Cameras for Vision-Based Navigation*. Master's thesis, Air Force Institute of Technology, 2015.

73. Tola, Engin, Sebastioan Knorr, Evrin Imre, Aydin Alatan, and Thomas Sikora. "Structure from motion in dynamic scenes with multiple motions". *2nd Workshop on Immersive Communication and Broadcast Systems*, 2005.

74. Tron, R. and R. Vidal. "A Benchmark for the Comparison of 3-D Motion Segmentation Algorithms". *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8, 2007.

75. Tsotsos, Alberto Pretto, Konstantine and Stefano Soatto. "Visual-Inertial Ego-Motion Estimation for Humanoid Platforms". *IEEE-RAS INternational COnference on Humanoid Robots*, 704–711, 2012.

76. Ullman, Shimon. *The Interpretation of Visual Motion*. MIT Press, Dublin, Ireland, 1979.

77. Valasek, J. "Vision-Based Sensor and Navigation System for Autonomous Aerial Refueling". *Journal of Guidance, Control, and Dynamics*, 28(5):979–989, 2005.

78. Vedaldi, Andrea, Gregorio Guidi, and Stefano Soatto. "Moving Forward in Structure From Motion". *IEEE Conference on Computer Vision and Pattern Recognition*, 1–7, 2007.

79. Warwick, G. "AFRL Advances Autonomous Aerial Refueling". *Aviation Week*, June 2008.

80. Weaver, Adam D. *Using Predictive Rendering as a Vision-Aided Technique for Autonomous Aerial Refueling*. Master's thesis, Air Force Institute of Technology, 2009.

81. Werner, Kyle P. *Precision Relative Positioning for Automated Aerial Refueling from a Stereo Imaging System*. Master's thesis, Air Force Institute of Technology, 2015.

82. Wu, Changchang. "Towards Linear-Time Incremental Structure from Motion". *Proceedings of the 2013 International Conference on 3D Vision*, 127–134, 2013.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2016	Master's Thesis	Sept 2014 — Mar 2016

4. TITLE AND SUBTITLE

Toward Automated Aerial Refueling:
Relative Navigation with Structure from Motion

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Colson, Jr., Kevin W., Captain, USAF

5d. PROJECT NUMBER

16-359

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-16-M-009

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

Ba T Nguyen
Aerospace Systems Directorate
Air Force Research Laboratory
2210 8^{th} St Wright-Patterson AFB, OH
(937)938-4617 Email: ba.nguyen@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQ

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 subject to copyright protection in the United States.

14. ABSTRACT

The USAF's use of UAS has expanded from reconnaissance to hunter/killer missions. As the UAS mission further expands into aerial combat, better performance and larger payloads will have a negative correlation with range and loiter times. Additionally, the Air Force Future Operating Concept calls for "formations of uninhabited refueling aircraft...[that] enable refueling operations partway inside threat areas." However, a lack of accurate relative positioning information prevents the ability to safely maintain close formation flight and contact between a tanker and a UAS. The inclusion of cutting edge vision systems on present refueling platforms may provide the information necessary to support a AAR mission by estimating the position of a trailing aircraft to provide inputs to a UAS controller capable of maintaining a given position. This research examines the ability of SfM to generate relative navigation information. Previous AAR research efforts involved the use of differential GPS, LiDAR, and vision systems. This research aims to leverage current and future imaging technology to compliment these solutions. The algorithm used in this thesis generates a point cloud by determining 3D structure from a sequence of 2D images. The algorithm then utilizes PCA to register the point cloud to a reference model. The algorithm was tested in a real world environment using a 1:7 scale F-15 model. Additionally, this thesis studies common 3D rigid registration algorithms in an effort characterize their performance in the AAR domain. Three algorithms are tested for runtime and registration accuracy with four data sets.

15. SUBJECT TERMS

Automated Aerial Refueling, Computer Vision, Relative Navigation, Structure from Motion, 3D Rigid Registration, Principle Component Analysis, Vision Navigation

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	113	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4618; brian.woolley@afit.edu