3-2022

Real Time Evaluation of Boom and Drogue Occlusion with AAR

Xiaoyang Wu

Real Time Evaluation of Boom and Drogue Occlusion with AAR

THESIS

Xiaoyang Wu, 2d Lt, USAF

AFIT-ENG-MS-22-M-077

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-22-M-077

Real Time Evaluation of Boom and Drogue Occlusion with AAR

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Xiaoyang Wu, B.S.C.S.

2d Lt, USAF

March 25, 2022

AFIT-ENG-MS-22-M-077

Real Time Evaluation of Boom and Drogue Occlusion with AAR

THESIS

Xiaoyang Wu, B.S.C.S.
2d Lt, USAF

Committee Membership:

Scott L. Nykl, Ph.D
Chair

Clark N. Taylor, Ph.D
Member

Robert C. Lieshman, Ph.D
Member

AFIT-ENG-MS-22-M-077

Abstract

In recent years, Unmanned Aerial Vehicles (UAV) have seen a rise in popularity. Various navigational algorithms have been developed as a solution to estimate a UAV's pose relative to the refueler aircraft. The result can be used to safely automate aerial refueling (AAR) to improve UAVs' time-on-station and ensure the success of military operations. This research aims to reach real-time performance using a GPU accelerated approach. It also conducts various experiments to quantify the effects of refueling boom/drogue occlusion and image exposure on the pose estimation pipeline in a lab setting.

Acknowledgements

First, I would like to thank my advisor for his unweaving support. I would also like to thank my committee members for their contributions to this thesis. Next, I would like to thank my fellow AAR members, as our work goes hand in hand; I could not have accomplished much without their guidance and assistance. Finally, I would like to thank my mom and girlfriend for keeping me on the right path. This work goes to all of you.

Xiaoyang Wu

Table of Contents

Page

List of Figures

Figure Page

Figure Page

Real Time Evaluation of Boom and Drogue Occlusion with AAR

I. Introduction

1.1 Problem Background

In recent years, Unmanned Aerial Vehicles (UAV) have seen a rise in popularity in military operations. Navigational algorithms combined with Global Positioning Systems (GPS), Mono and Stereo Vision Sensors, or LIDAR based equipment have been developed to estimate a UAV's relative pose to the refueler aircraft. The end goal is to safely guide the UAV to conduct automated aerial refueling (AAR), improving its time-on-station and ensuring the success of military operations.

Previous works have laid the foundation of the AAR Vision Pipeline (AARVP), which involves Stereo Block Matching (SBM) in conjunction with the Iterative Closest Point (ICP) Algorithm. One research has proven the use of virtual simulation to study the effectiveness of the vision pipeline.[1]. Bownes has established a framework leveraging augmented reality with motion capture to test AARVP with real-life imagery [2]. This research serves as a follow-on experiment to analyze USAF Boom and Navy Drogue occlusion. As boom and drogue occlusions and lighting changes can occur during real refueling approaches, the experiments aim to quantify the effects of aforementioned factors on the pose estimation pipeline in a lab setting. It also looks to improve the performance of AARVP to reach real-time performance using a GPU accelerated approach.

1.2 Research Objectives

The questions this research aims to answer are:

- How do refueling boom and drogue occlusions affect the results of the vision pipeline in a realistic approach?
- How do different image exposures affect the results of the vision pipeline?
- How can the vision pipeline be improved to run in real-time?

1.3 Document Overview

Chapter II presents background knowledge of the vision pipeline as well as the latest advancements in the AAR field. Chapter III describes the improvement to the AARVP and testing methodology in which the data is collected. Chapter IV shows the testing data which demonstrates the effectiveness of the vision pipeline under occlusion. To conclude, chapter V summarizes what this thesis has accomplished and potential future follow-on research topics for AAR-based work.

II. Background and Literature Review

This chapter provides the necessary background information to understand the experimental methodology and the results that follow. The first section will provide a high-level overview of the AAR Vision Pipeline (AARVP). This includes the fundamentals of a stereo vision camera system and pose estimation through the Iterative Closest Point algorithm. The following section will discuss the benefits of using GPU-based code to accelerate computer vision processing. Finally, the chapter will cover previous research conducted in the AAR domain and provide motivation that serves as a basis for this research.

2.1 AAR Vision Pipeline

This section gives a high-level overview of the AARVP, including the pinhole camera model, camera calibration, and Epipolar geometry required to project features from a pair of 2D images into 3D space. The section will also go over the Iterative Closest Point Algorithm and its relevance to pose estimation.

The pinhole camera model is used to explain the relationship between a 3D coordinate and its respective 2D coordinate on an image plane [3]. A plane containing a single pinhole is placed in front of the image plane. Light is emitted or reflected from distant objects placed in front of the pinhole plane, but only a single ray of light enters the pinhole from any particular point in the object world. The point is projected onto the image plane and is always in focus, and the distance from the pinhole to the image plane is known as the focal length.

A variation of the pinhole camera model reverses the order of the pinhole plane and the image plane in which the objects now appear right side up. The pinhole point is now re-defined as the center of projection. Each 3D Point (X, Y, Z) on an object

emits a ray that heads towards the center of reprojection. The point of intersection with the image plane produces a point defined as (X, Y, f), with the X and Y being the 2D coordinates on the image plane and f being the focal point. The pinhole camera model can be visualized by Figure 1.

The point at the intersection between the image plane and the center of projection is known as the principal point. Ideally, the principal point is equivalent to the center of the image plane but is not always guaranteed due to manufacturing imperfections. A camera calibration produces a camera intrinsic matrix that contains both the principal point and focal length of the calibrated camera.

Another motivation for performing camera calibration is to calculate the distortion coefficients of camera lenses [3]. Two prominent lens distortions are radial and tangential. A radial distortion, which takes the form of barrel, pincushion,or mustache, occurs when the rays farther from the center of a lens are bent too much compared to rays that are closer to the center of the lens. The distortion can be represented by the first three terms of the Taylor series. The corrected X and Y location with radial

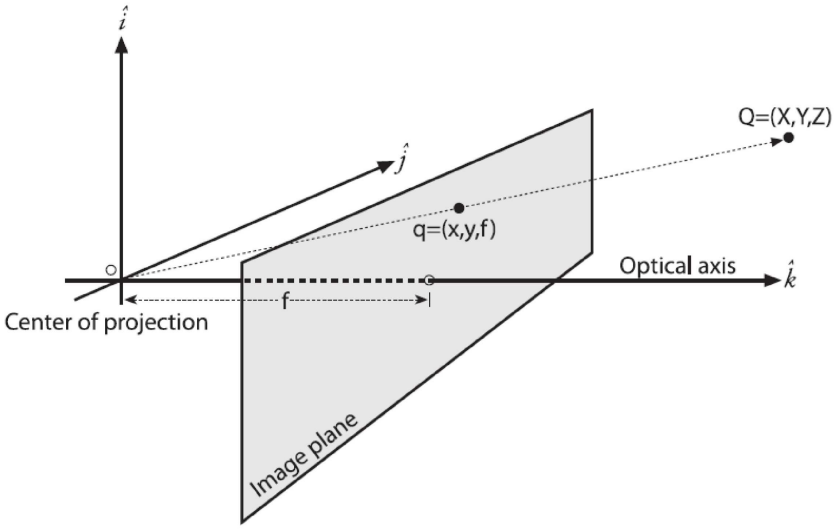

Figure 1: The Pinhole Camera Model

distortion factored in can be represented by the following equations

$$X_{Corrected} = x * (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{1}$$

$$Y_{Corrected} = y * (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2}$$

A tangential distortion occurs when the lens is not fully parallel to the image plane. The corrected X and Y location with tangential distortion factored in can be represented by the following equations

$$X_{Corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \tag{3}$$

$$Y_{Corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \tag{4}$$

The technique for camera calibration employed by OpenCV is described in [4]. A chessboard is captured from multiple orientations by either adjusting the board or the camera. The chessboard corners are then detected as features and used to estimate the intrinsic and extrinsic parameters using a closed form solution. Figure 2 shows an example of a checkerboard pattern that has been successfully detected by OpenCV. Then the distortion coefficients, a 5x1 matrix which consists of $[k_1, k_2, p_1, p_2, k_3]$ are calculated and used to fine-tune the final values by minimizing the following equation:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} ||m_{ij} - \breve{m}(A, k_1, k_2, R_i, t_i, M_j)||^2 \tag{5}$$

A is the intrincic calibration matrix which contains the focal length and the principal point location. k_1 and k_2 are radical distortion coefficients. R and t are extrincic calibration matrix for an image required to convert from world coordinate frame into the camera coordinate frame. M_j is a chessboard corner and $\breve{m}(A, k_1, k_2, R_i, t_i, M_j)$, the projection of M_j into a 2D image coordinate is subtracted

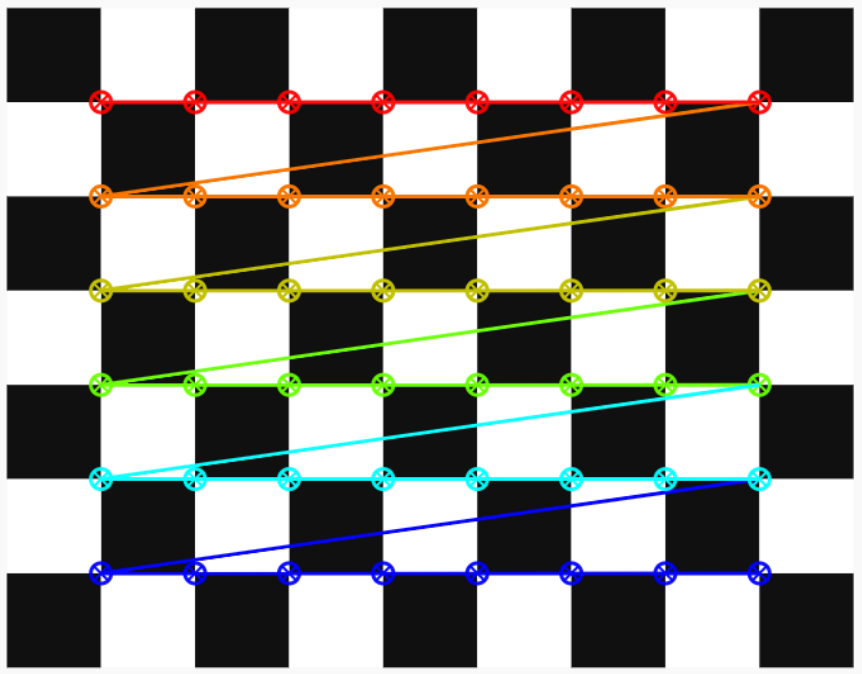

Figure 2: Checkerboard Corner Detection Using OpenCV

from m_ij, the observed location of the chessboard corner. This is performed for every corner in every image, and the iterative minimization of the difference between m_ij and $\breve{m}(A, k_1, k_2, R_i, t_i, M_j)$ further refines the all the parameters.

A common problem for planar projection from 3D space to 2D image space is the loss of depth. With stereo vision, depth can be inferred via triangulation and Epipolar geometry. A pair of cameras, $C1$ and $C2$ is intrinsically calibrated and looking at a 3D point X, with $X1$ and $X2$ being the reprojection 2D coordinate of X on their respective image plane. Two rays $R1$ and $R2$ are extended from $C1$ and $C2$ and passes through both $X1$ and $X2$ before intersecting at X, which can be seen in Figure 3.

Epipoles $E1$ and $E2$ are defined as the projection of $C1$ and $C2$'s center point in the opposite camera frame. A baseline can be found as the intersection of both Epipoles. With the baseline established, the construction of an Epipolar plane in which the point $X1$ can be determined to be on the Epipolar plane as $X2$ limits the

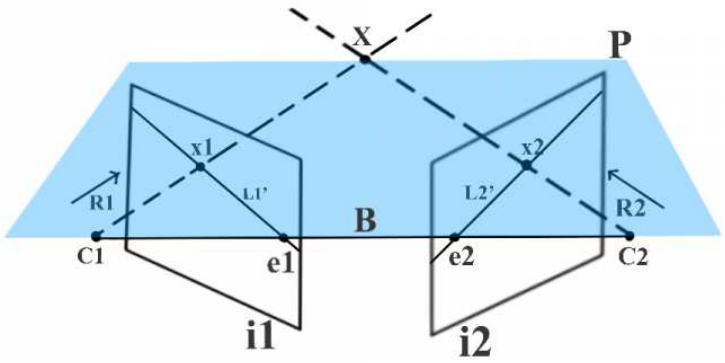

Figure 3: Epipolar Geometry

corresponding feature matching from a 2D search to a 1D search. This is also known as Epipolar constraints.

For epipolar constraint to correspond with horizontal lines in stereo imaging, the stereo vision sensors will need be parallel to each other. Through stereo rectification, OpenCV realigns the image planes to align the rows of the image while intersecting their optical axis at infinity. Figure 5 shows an example of a pair of images before and post rectified using OpenCV's algorithm.

To perform triangulation, Stereo Block Matching was first used to detect and store corresponding features which exist in both images. Following that, computing the disparity map of a corresponding stereo image pair is performed. Figure 4 showcases an example of a disparity map. Disparity refers to the pixel difference of a corresponding feature being detected in the image pair. Disparities further away from the cameras will have a smaller value, while disparities closer to the cameras will have a larger value. One can store the different intensities of every disparity in a disparity map, and using the following formula:

$$Z = B * f/disparity \tag{6}$$

where B is the baseline in meters and f is the focal length of the stereo vision system in pixels, to compute the depth Z in meters of every detected feature. OpenCV allows for the reprojection of a 2D coordinate to a 3D coordinate using the camera calibration and the disparity map through the $reprojectImageTo3D$ function.

To obtain a 6 DoF estimate for an object with stereo vision, a 3D Point Cloud Registration algorithm called Iterative Closest Point is utilized. The algorithm takes in the reprojected 3D coordinates of the disparity map as the sensed point cloud generated by the stereo vision cameras. It then uses a predetermined truth model and attempts to fit the truth point cloud model onto the sensed point cloud model using a point-to-point approach. The algorithm, using the Euclidean distance metric, searches for each point in the truth model a corresponding nearest neighbor point in the sensed points. By iteratively minimizing the Euclidean distance metric, a rotational matrix and a translation vector can be extracted, and this is used as the 6 DoF estimate for the object.

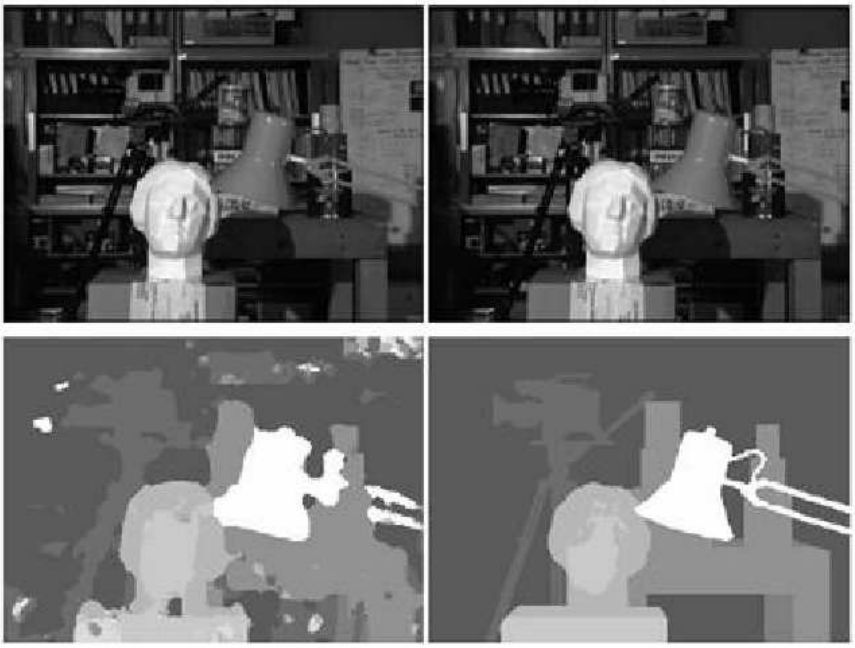

Figure 4: An Exmaple of Disparity Map

Figure 5: Image Rectification

2.2 GPU Parallelization

This section explores the motivation behind using Graphical Processing Unit (GPU) to perform computations. Two areas of GPU applications for computer vision and image processing are provided as examples demonstrating the effectiveness of GPU accelerated workflow.

Modern Computer Processing Units (CPU) contains cores with higher clock speeds, allowing for the speedy processing of various tasks using a wide range of instructions available to the processor. Due to the CPU's form factor, the number of cores limits the concurrency of running tasks. Thus, researchers turned to GPU as a viable solution for parallel processing. Modern GPUs contain thousands of processors. Combined with access to high bandwidth memory, they allow for the concurrent execution of commands over all the GPU's cores, making it suitable for processing large data. Figure 6 shows a comparison of the architecture of CPU and GPU.

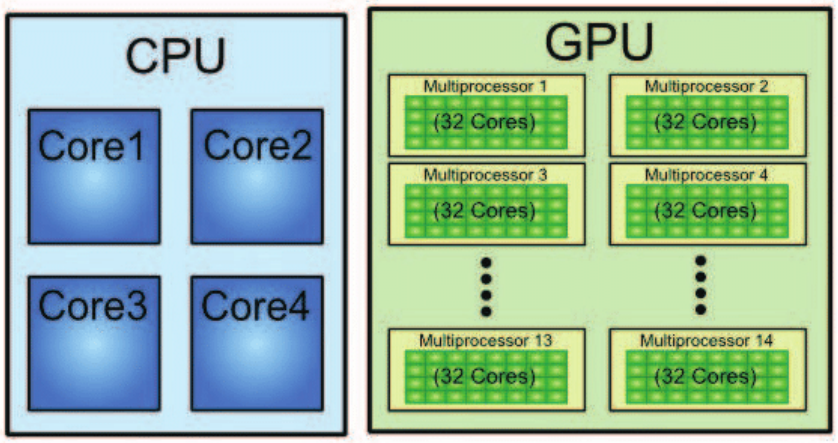

Figure 6: Comparison of CPU and GPU Architecture

GPUs are often used by the industry and academia to tackle time-consuming computer vision problems. Computer vision is the intake of an image from a camera source and interpreting each pixel into numerical form, enabling additional processing like feature matching. The ever-growing camera sensor size led to larger image resolutions, denoted by the number of pixels contained in the image. GPU allow vectorized operations on each pixels simultaneously, resulting in lower overall computation times. GPU manufacturers such as NVIDIA have enabled programmers to interface with the GPU using languages like CUDA– extensions of the C language. Shams et al used CUDA to adapt traditional sequential algorithms like histogram computations to be parallelized and accelerate the registration of medical images over the GPU, thus resulting in a 50 fold decrease in processing time [5].

Researchers also saw the benefit of GPU parallelization towards AI and deep learning for image processing. The bottleneck of a convolutional neural network, or CNN, is the convolution operation of a $n * n$ matrix using a $k * k$ kernel. A GPU can parallelize the process while allowing for the acceleration of subsampling/pool layers [6]. Numerous libraries have been developed to take advantage of GPU for CNN-based work to include PyTorch, Tensorflow, and Caffe. [7] used a cuDNN-

HYCNN approach to classify hyperspectral Images, achieving a 75 fold speedup while maintaining a comparable result against the HYCNN CPU-based algorithm.

2.3 GPU Parallelization of AARVP Algorithms

This section discusses some of the CUDA implementations of the algorithms used by the AARVP. The following two examples showcase some of the advancements in CUDA Stereo Block Matching and GPU-driven point registration algorithms.

Ivanavi et al have designed a CUDA implementation of Stereo Block Matching aimed towards embedded systems [8]. They tested two separate approaches to handle the passing of data to the GPU. The first method involves two different GPU kernels invoked by the CPU to calculate the absolute differences between pixels, calculate new weights, and perform minimization. Due to the second operation taking longer to complete, both threads are kept separate to avoid halting. The second approach uploads images to the GPU global memory. A collective shared memory block is allocated for each of the images' pixels, then the algorithm is performed over all the pixels in the block in sequence. The second algorithm is comparatively faster than the first algorithm due to data accessed from the shared memory of the GPU as opposed to the global memory, but it comes at the cost of image quality of the disparity map generated. By taking advantage of the CUDA architecture, the researchers achieved high framerate processing, from over 70 FPS on 750x500 sized images and up to 23 FPS on 1500x1000 on an NVIDIA Jetson TX1.

Previous research conducted by Mourning et al discussed the viability of using GPU to accelerate the point registration process [9]. The bottleneck of Robust Point Matching (RPM) is the mathematical operations of N * N matrices, with N representing the number of pixels contained in the image. The suggested algorithm took advantage of the parallelism of a GPU to tackle large image datasets by assigned ma-

trix operations like multiplication, inversion, and QR decomposition to be performed in unison on the GPU using CUDA. The implementation was compared to the CPU iterative closest point algorithm and the MATLAB optimized version suggested by Chui et all. The GPU accelerated ICP performed 20 fold faster than the CPU version and 25 percent faster than the Matlab, showing the potency of GPU acceleration for other point registration methods like Iterative Closest Point. As the listed research showed great promises towards the speed up of AARVP algorithms, this paper hopes to leverage GPU parallelization to test the speedup of GPU accelerated AARVP against the previous CPU-based AARVP.

2.4 Related work on AAR

Previous research on AAR involves the use of Gobal Positioning System to obtain the pose of the receiver [10, 11, 12, 13]. Since GPS's capabilities can be degraded or denied via jamming on the battlefield, other complementary sensors like Inertial Measurement Units (IMU) are incorporated and tested to compensate for the shortcomings of standalone GPS solutions [11]. Researchers have also tested electro-optical (EO) and infrared solutions like LIDAR as a standalone package for pose estimation of the receiver [14, 15, 16].

Other researchers have also made strides towards the area of probe and drogue refueling. There are studies done to develop docking controls and close loop motion characteristics for the probe and drogue [17, 18]. Drogue detection is also under examination. Several researchers used a convolutional neural network in conjunction with Cascade AdaBoost Classifiers and You Only Look Once (YOLO) to speed up the detection of the refueling basket [19, 20]. Another study uses monocular vision with ellipse detection to estimate the 5 DoF of the drogue basket under various distances and weather conditions [21].

2.5 AftrBurner vs. Real World

Flight testing for AAR-based work is expensive, time-consuming, and poses additional risks to the aircraft and the operator. Therefore, researchers turned to virtual computer simulations as a solution for testing vision algorithms. Previous studies [22, 23, 2] at AFIT utilize the AftrBurner engine, a high fidelity 3-dimensional virtual world (3DVW) [24], as the foundation platform for the research. The researchers used the truth data from a motion capture system (MCS) with a 1/7th scaled F-15 model. The tests used simulated refueling approaches driven by truth data from a physical approach in the motion capture chamber (MCC). The conclusion from these studies has found the positional and rotational errors from the virtual environment to be similar to real-world errors, thus hinting at the viability of the virtual environment as a testing platform of vision algorithms for receiver pose estimation. This research will also utilize AftrBurner as a testing platform for the latest iterations of AARVP with GPU acceleration for boom and drogue occlusion testing.

Figure 7: Virtual Approach

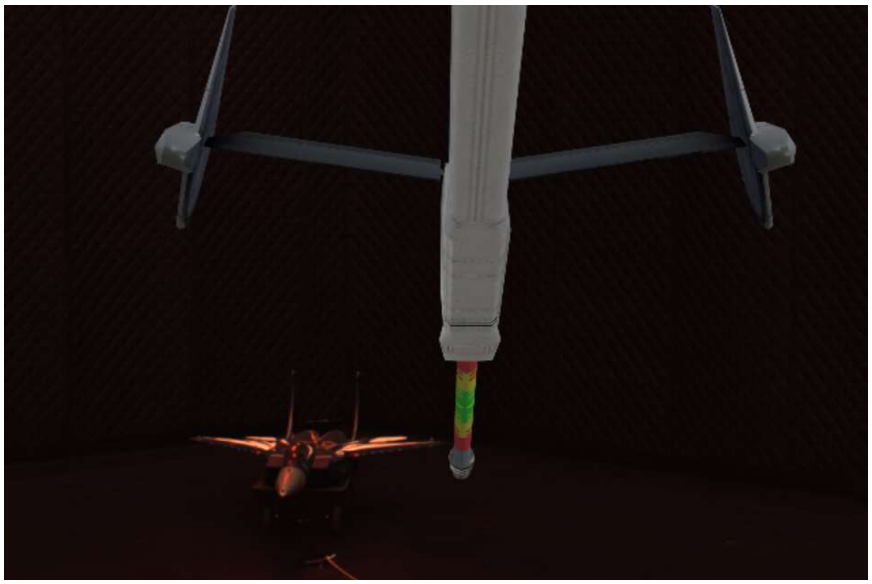

Figure 8: Real Approach

2.6 Motivation for Research

AAR requires centimeter accuracy for the receiver to make contact with the refueling boom/drogue. For the AAR vision pipeline to produce pose estimate results, it has to conform to real-time standards. According to the IEEE, a real-time system is a computing system whose correct behavior depends not only on the value of the computation but also on the time at which outputs are produced. Previous research [2] has produced good pose estimation at centimeter accuracy with a roll, pitch, and yaw error under 2 degrees. However, the CPU implementation of the AARVP failed to achieve 30 Hertz when processing 4K stereo images. This research aims to improve the previous setup by using CUDA to accelerate the workflow to achieve the aforementioned standards of real-time computing.

A critical element of the Iterative Closest Point algorithm is the input sensed point cloud. An ideal sensed point cloud should be identical to the truth model, thus ensuring a perfect convergence in fit and the generation of accurate pose estimate matrices. The receiver conducting an AAR approach will be occluded by the refueling

boom or drogue. The generated sensed point cloud with such occlusions will alter the sensed point cloud model and produces outlier points which result in an inaccurately estimated pose. Previous research [2] briefly explored boom occlusion by measuring the error of the receiver with a virtualized boom inserted into the approach. However, the boom remained stationary while the receiver approach was adjusted. This research aims to conduct experiments to study the effects of boom and drogue in various poses to examine potential approaches which allow for an accurate pose to be generated by the algorithm.

Chang et al have determined that the exposure of images can have a significant impact on stereo block matching [25]. As AAR can occur under various conditions, this proposes a challenge for the AARVP. [2] obtained excellent results with perfectly exposed images but did not test poorly exposed images as part of the study. This research performs two separate experiments to simulate extreme daytime and nighttime refueling operations. The research aims to quantify the effects underexposure and overexposure have on Stereo Block Matching and pose estimation while testing various image filters to see if they can mitigate the shortcomings of improperly exposed images acquired by the stereo vision system.

2.7 Differences between Boom and Drogue refueling

During aerial refueling, a plane can take on fuel via a refueling boom or a probe-and-drogue system. Modern USAF refueling platforms come configured with a refueling boom. During a boom refueling scenario, the receiver plane first approaches the contact position. The boom operator manipulates the boom and inserts the refueling boom into the open receptacle, and then the fuel is transferred (Figure 9). The U.S. Navy aircraft and helicopters come equipped with a refueling probe, which requires a probe-and-drogue setup. A probe-and-drogue system involves a drogue basket at-

tached to a hose that can be extended or retracted. During refueling, the receiver will align and insert the probe into the drogue basket to receive fuel (Figure 10).

Figure 9: Boom Refueling

Figure 10: Drogue Refueling

III. Methodology

3.1 Preamble

The primary goals of this work are:

- Examine the effects of refueling boom and drogue occlusions on AARVP pose estimation
- Examine the effects of real world image exposure on AARVP pose estimation
- Create and evaluate a GPU accelerated implementation of the AARVP

The paper will highlight two distinct testing scenarios to test boom and drogue occlusions. The refueling drogue scenario will utilize the virtual AftrBurner engine, capable of reproducing a high fidelity simulated environment for testing. The refueling boom scenario will leverage an augmented reality scenario that incorporates virtual and real-world imagery using AftrBurner and a motion capture chamber (MCC). The truth data provided by the motion capture system (MCS) have demonstrated sufficient accuracy required to conduct AAR and deemed the scenario suitable for boom occlusion testing [2]. This chapter provides a detailed description of the testing methodology for this paper. It begins by describing the virtualized setup for drogue occlusion testing. Following that, the chapter highlights the augmented reality scenario used for boom occlusion testing [2]. The chapter will also discuss the GPU accelerated AARVP scenario with potential sources of errors and solutions to correct them. The chapter will conclude with different testing parameters for the boom and drogue scenarios with the real-world image exposure test setup.

3.2 Drogue Scenario

The virtual drogue scenario contains both a refueling aircraft and a receiver aircraft. The refueling aircraft comes equipped with a refueling pod located on the right wing. The scenario mounted a pair of cameras on the refueling pod with a baseline of 0.4m. Each camera contains a 2K sensor with a resolution of 2048x1080, a horizontal field of view of 55.6 degrees, a near plane of 0.25m, and a far plane of 500m. The pod also contains a refueling hose capable of extending to 14m. A refueling drogue basket is attached at the end of the refueling hose. Figure 11 provides a visualization of the virtual refueler setup, while Figure 12 provides an example output of the 2K stereo vision sensor.

The virtual receiver aircraft is a modified F-15E with a refueling probe mounted to the aircraft's right side (Figure 13). The accurate recreation of the refueler and the receiver allows for a realistic refueling approach with the probe-and-drogue setup.

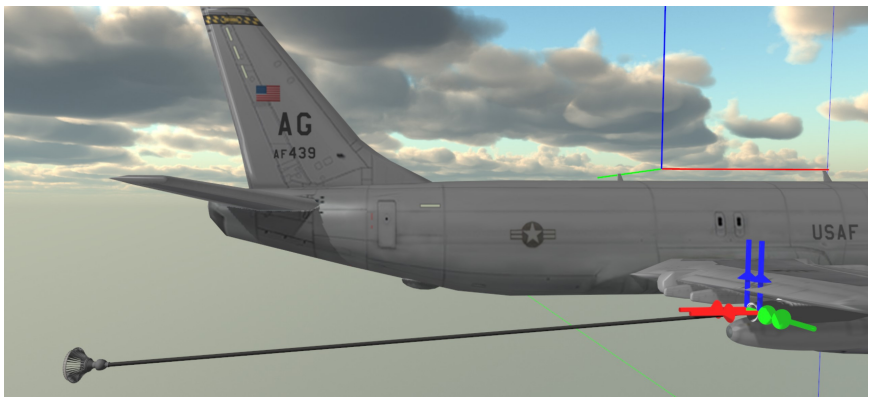

Figure 11: Virtual Refueler

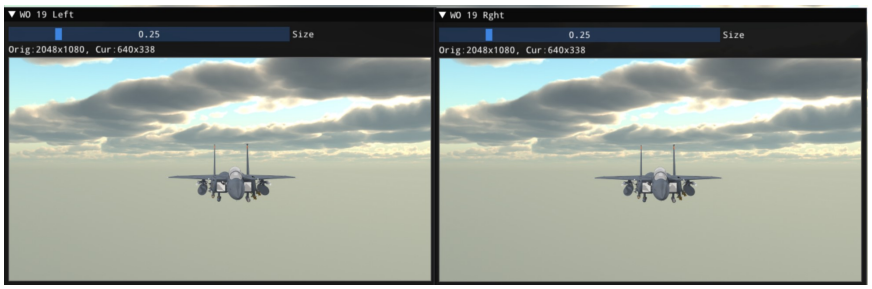

Figure 12: 2K Stereo Camera

Figure 13: Virtual Receiver

3.3 Boom Scenario with Augmented Reality

The boom scenario allows for the testing of both virtual and real-world imagery. This section will first cover the lab setup. Following it will be a discussion of the virtual recreation of the aforementioned environment. The section finally will consider the augmented reality (AR) setup used for boom occlusion testing.

The experiment utilized a motion capture chamber equipped with 60 IR sensors as the basis of real-world approaches. The motion capture system can detect IR

reflective markers within the room and provide millimeter accurate positional and rotational truth data at 75 Hz. A pair of 4K cameras with a resolution of 4096x3000 are placed 8 meters above the motion capture chamber with a downward orientation to mimic the stereo vision sensors located on a refueling tanker (Figure 14). A 1/7th scaled replica F-15E model is placed on a cart and serves as the test receiver in the real-world approach (Figure 15). IR markers are attached to the receiver, allowing it to be tracked by the motion capture system.

The virtual testing environment contains a recreated virtual model of the MCC textured to mimic its real-world counterpart(Figure **??**). A full sized virtual F-15 is inserted into the room with a 1/7th scaling applied, making the dimensions equivalent to the real-world F-15 model (Figure 17). The motion capture chamber also contains a pair of virtual 4K cameras. As Stereo Block Matching with an inaccurate camera rotation as slight as 1 degree can produce reprojection errors at large distances, a Gauss-Newton optimization is used as part of the camera calibration process to align the physical cameras accurately in the MCS coordinate frame [2].

As it is not feasible to install a full-sized refueling boom into the motion capture chamber, the work employed augmented reality as a solution to generate boom occlusion. Augmented reality is the depiction of elements from both the virtual and the real world. In AftrBurner, we created two quad objects that serve as the green screen in a traditional AR setup. The quads use imagery from the virtualized and real-world motion capture chamber and render them with 1:1 resolution and fidelity (Figure 18). Two additional AR virtual cameras are placed in front of the quads with the same orientation as the real-world counterpart (Figure 18). The virtual cameras capture the quads' textures in their viewing frustum, creating replica images mirroring the original input imagery. The setup finally inserted two virtual refueling booms in front of each AR camera (Figure 18, 19). The disparity map generated by

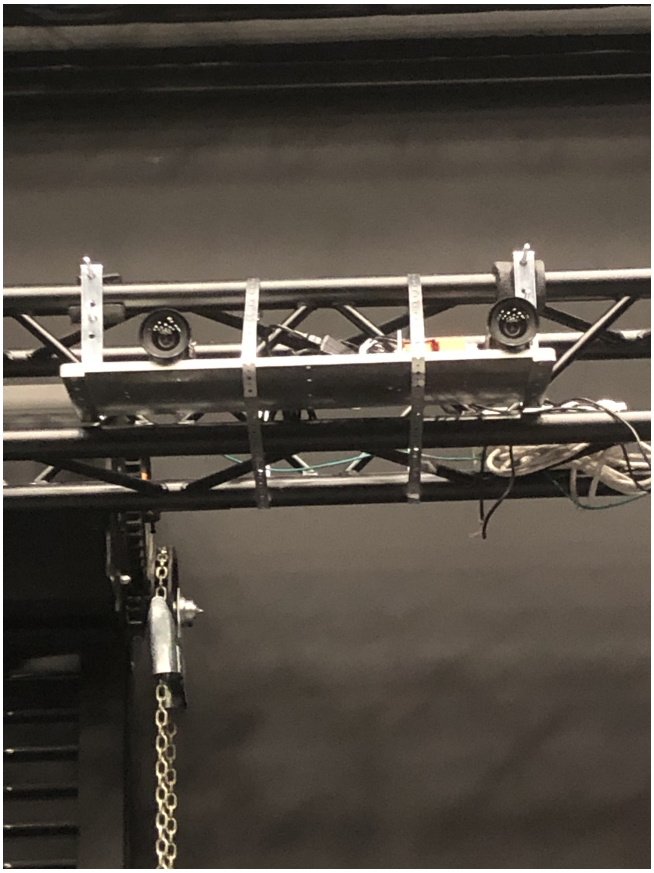

Figure 14: Real Camera

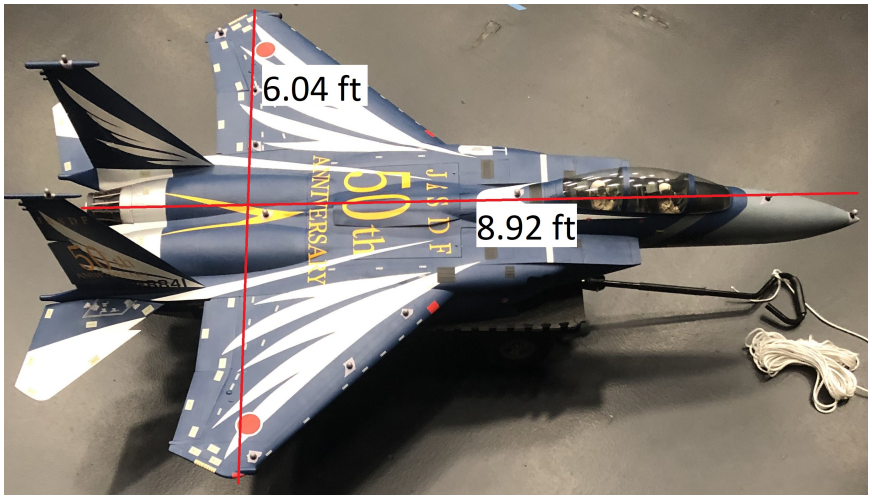

Figure 15: Real F-15 Model

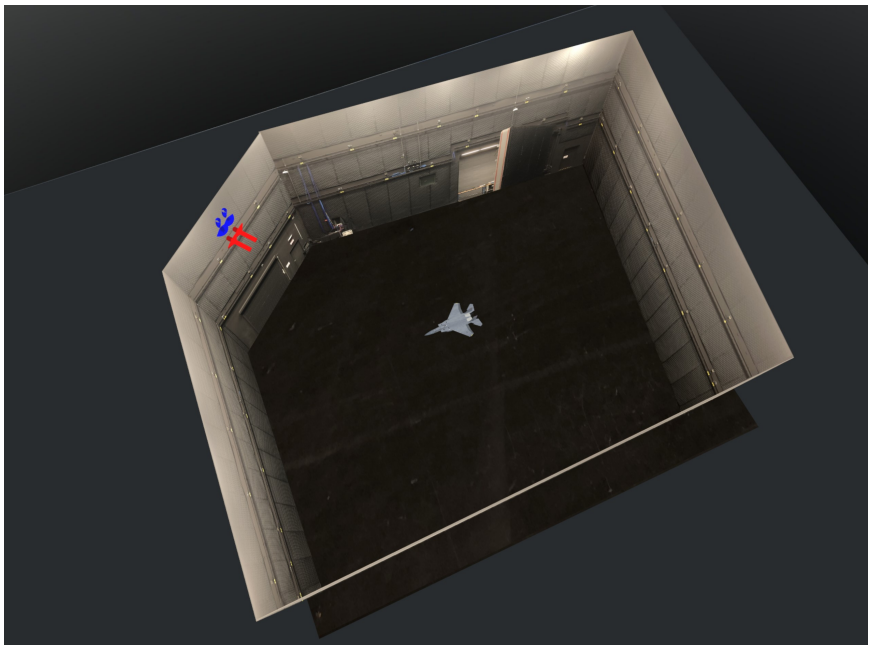

Figure 16: Virtual Model of MCC

Stereo Block Matching on the AR cameras will include scenery from the quad and the virtual boom. The virtual refueling boom occludes the same set of pixels as a real boom on images. Thus the virtual augmented reality provide an accurate alternative to installing a physical boom replica in the MCC.

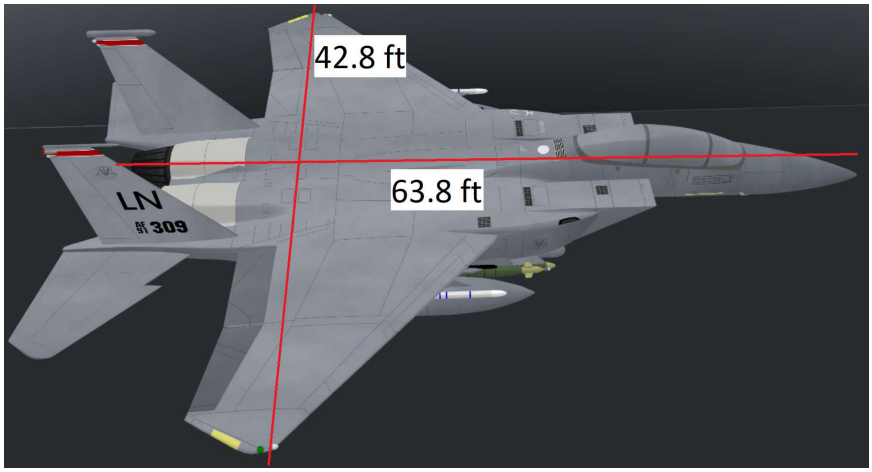

Figure 17: Virtual F-15 Model pre-scale

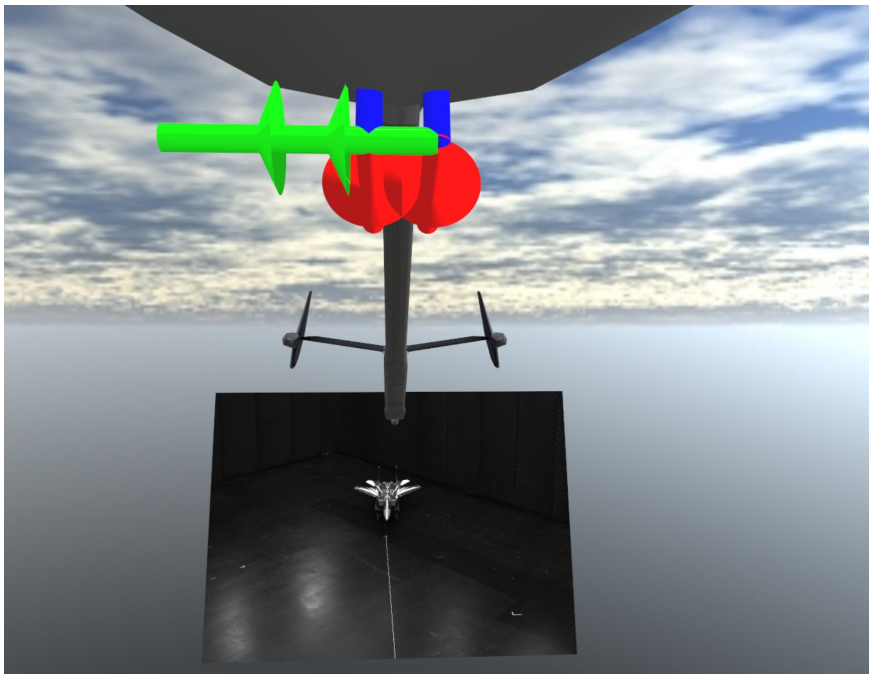

Figure 18: Quad Setup with Virtual Boom

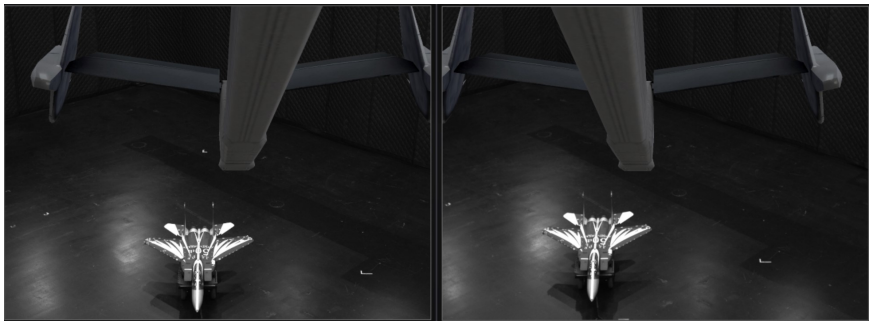

Figure 19: New imagery generated using augmented reality

3.4 GPU Accelerated AARVP

This section will elaborate on the GPU implementations of AARVP. It will describe some of the additional improvements on top of the CPU-based AARVP to speed up run time.

3.4.1 GPU Accelerated Stereo Block Matching

This paper utilized OpenCV's CUDA Stereo Class to accelerate the generation of disparity maps. Calling OpenCV CUDA Stereo Block Match algorithm uploads the stereo images onto the GPU's memory as a cv::cuda::GpuMat. The GPU performs Stereo Block Matching and returns the disparity map also as a cv::cuda::GpuMat. cv::cuda::reprojectImageTo3D() function uses the generated disparity and the Q matrix obtained from stereo rectify to reproject all the pixels of the 2D image into 3D space coordinates.

The coordinates are then transformed from the local frame into the camera frame and inserted into a std::vector using the following formula:

$$StereoCam_{DCM} \cdot Vector(X, Y, Z) \tag{7}$$

The end-user can specify an optional filter to eliminate outliers. Finally, the AftrBurner engine visualizes the reprojected points as the yellow point cloud (Figure 20) as the sensed points generated from the stereo image pair.

Previous iterations of AARVP processed the reprojected points by iterating through each pixel of the 3D reprojection via two for loops indexing the width and height of the cv::Mat. A 4K disparity would cause a significant slowdown as the operation is single-threaded. This paper uses the cv::foreach function to distribute all available CPU threads to process the pixels in parallel. The new process led to a significant

Figure 20: Sensed Points after Reprojection

decrease in the time required to complete point insertion into the vector.

3.4.2 GPU Accelerated ICP

This paper uses a GPU accelerated version of the Iterative Closest Point algorithm described in [1][26]. One of the most time-intensive operations of the algorithm is the nearest neighbor search, intended to match every sensed point to the closest correlating point in the truth model (Figure 21). The GPU accelerated ICP algorithm takes advantage of the parallel nature of the operation by distributing each point to available GPU threads to perform the nearest neighbor matching.

A traditional ICP algorithm would use a tree-based approach like a K-D tree to partition the data using the median value of each sequential axis. However, this

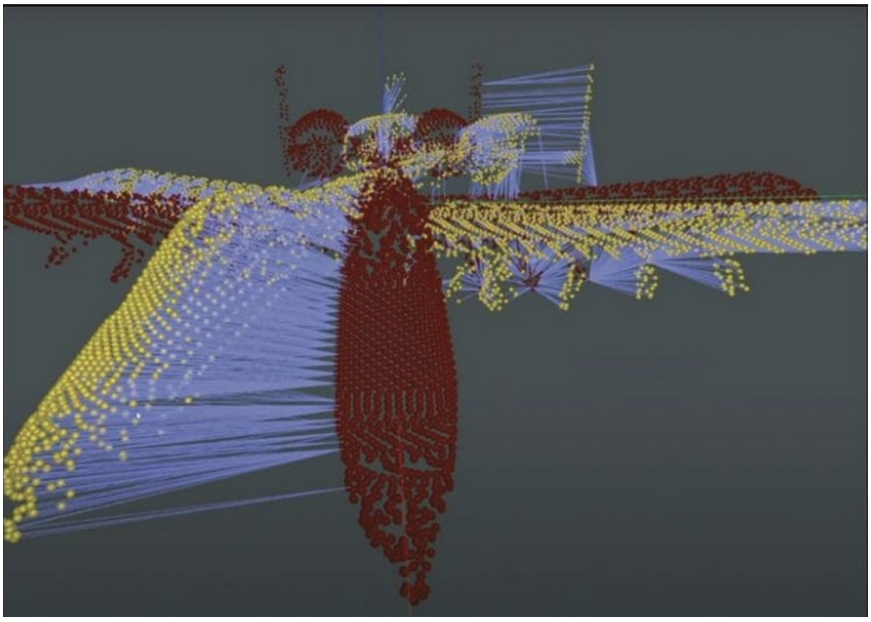

Figure 21: Nearest Neighbor Correspondence between Sensed Model and Turth Model

approach requires the search to always start from the root node of the tree. This GPU-based ICP algorithm further sped up the process by implementing a Delaunay triangulation [26]. This Delaunay structure enables a query point to find its closest neighbor node with a minimal graph walk. By traversing a Delaunay structure, the search walks along edges in the graph until there are no other points closer to the current candidate point. The new search structure allows a query to be conducted at any node, resulting in only 1-2 walks required for each ICP iteration and effectively reducing ICP's overall runtime.

3.5 Sources of Errors

An ideal sensed point cloud model maintains a 1:1 correspondence with the truth model producing a optimal pose. However, Stereo Block Matching and reprojection often result in noisy points, leading to higher rotational and translation errors. This section will discuss some of the challenges associated with CUDA Stereo Block Matching and mitigating steps to achieve good reprojection results.

3.5.1 Probe and Drogue Scenario

The point cloud generated by CPU-based Stereo Block Matching fits tightly around the receiver aircraft (Figure 22). There are minimal outlier points in the sensed point cloud model, which does not drastically affect ICP results.

Using OpenCV's CUDA Stereo Block Matching produces undesired reprojections. Despite the disparity map generated by both algorithms being similar (Figure 23), the reprojected sensed point cloud generated by CUDA is noisier, with points often being reprojected past their intended depth. As the ICP algorithm requires a close correspondence of points between the sensed point cloud and the truth point cloud, the additional noisy points will lead to a poor correspondence pairing and higher translational and rotational errors (Figure 24).

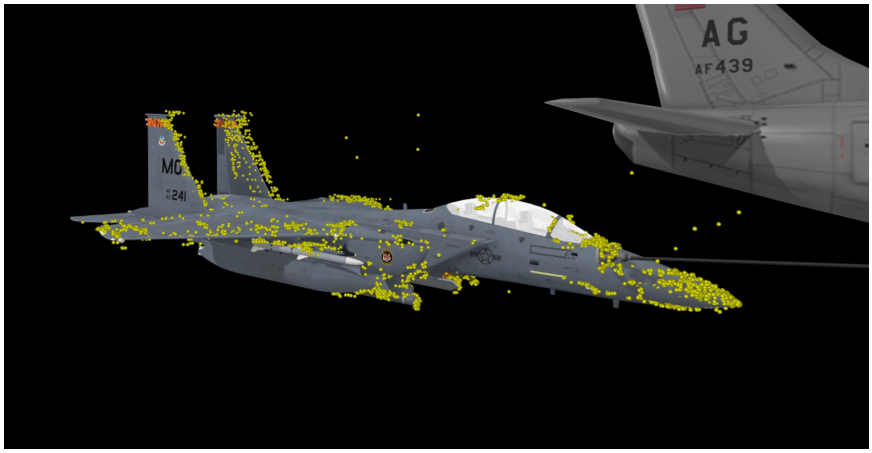

Figure 22: CPU SBM Reprojection

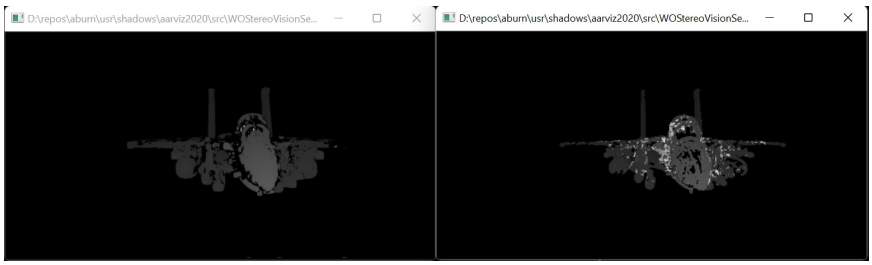

Figure 23: Disparity Map Comparison, with CPU on the left and CUDA on the right

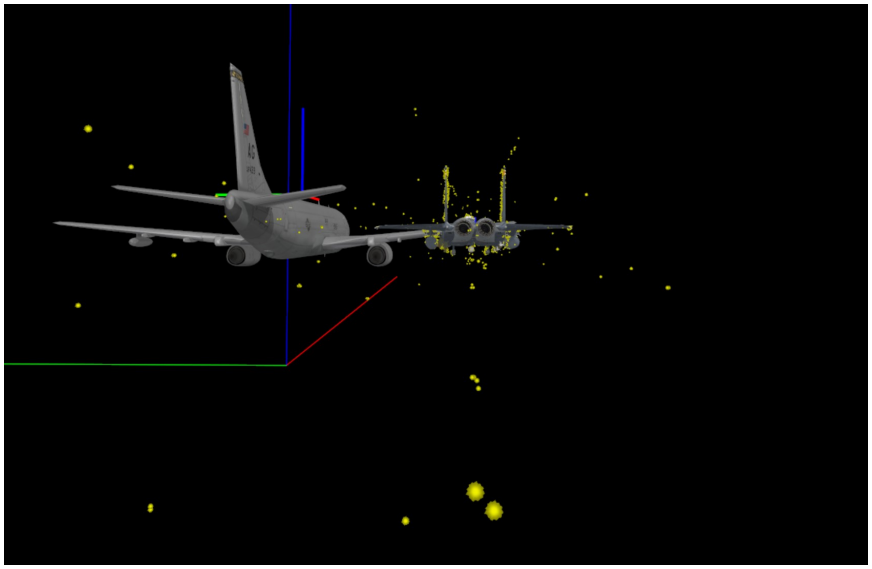

Figure 24: CUDA SBM Reprojection

This paper employs a near/far plane filter as a part of the reprojection process to eliminate outlier points generated by CUDA Stereo Block Matching. When the receiver aircraft is at contact point, the near plane filter is the displacement along the x-axis from the camera position to the nose of the receiver aircraft and any points closer to the filter are eliminated. The far plane filter is the displacement along the x-axis from the camera position to the back of the receiver aircraft. The precise measurement for the near plane at the contact point is 10 meters, while the far plane is 30 meters. The experiment subtracted and added a few meters to the near and far plane filters to generate outlier points and match the behavior of CPU-based Stereo Block Matching reprojection.

The new filter parameters for a F-15 receiver are:

- Near plane filter - 9 meters
- Far plane filter - 40 meters

The filter will discard any points with a displacement along the x-axis less than the near plane distance or greater than the far plane distance and results in a valid

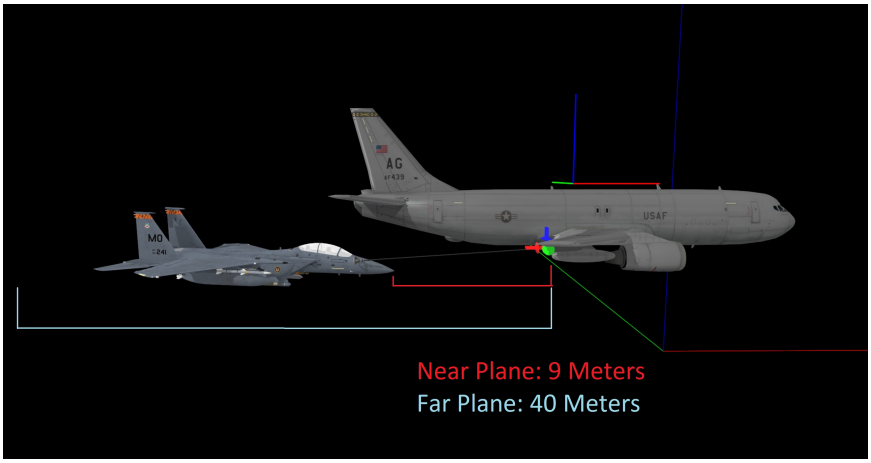

Figure 25: Near/Far Plane Filter

ICP convergence of the truth point cloud onto the sensed point cloud.

3.5.2 Boom scenario

A limitation of CUDA Stereo Block Matching is the numDisparity range. The max numDisparity available to CUDA is 256, whereas CPU-based Stereo Block Matching algorithms have no such constraints. The boom scenario utilizes stereo vision sensors with a resolution of 4K. Stereo Block Matching takes unique features in the left image, shifts the search area by the numDisparity, and finds correlations in the right camera frame to produce a valid disparity map. The increased horizontal pixel of the image width and the limitation of the numDisparity causes CUDA Stereo Block Matching to fail in correlating features, thus resulting in the lack of disparity produced in the disparity map and the reprojected points as the receiver comes closer to the cameras and disparities increases. (Figure 26).

As a solution to the previous issue, the experiment applied downsampling to the stereo 4K images. The resolution of the 4K stereo vision sensors is halved, from 4096x3000 to 2048x1500. The downsampling allows the CUDA Stereo Block Matching to search the entirety of the horizontal image pixels when performing feature

correlation and generating valid reprojection points from the disparity map.

Another requirement for downsampling to work is the modification of the existing camera calibration. Using the idea by Miller[27], the downsampling camera calibration can help fix this issue. The intrinsic matrix, the rectification matrix, and the Q matrix, generated from performing camera calibration, is used to reproject points from 2D space to 3D space. As the stereo camera sensors are modified, a new calibration is needed. Instead of calibrating the new sensors using chessboards, the approach took the previous 4K camera calibration and made modifications to the following values shown in Table 1.

After downsampling and adjusting the camera calibration, CUDA Stereo Block

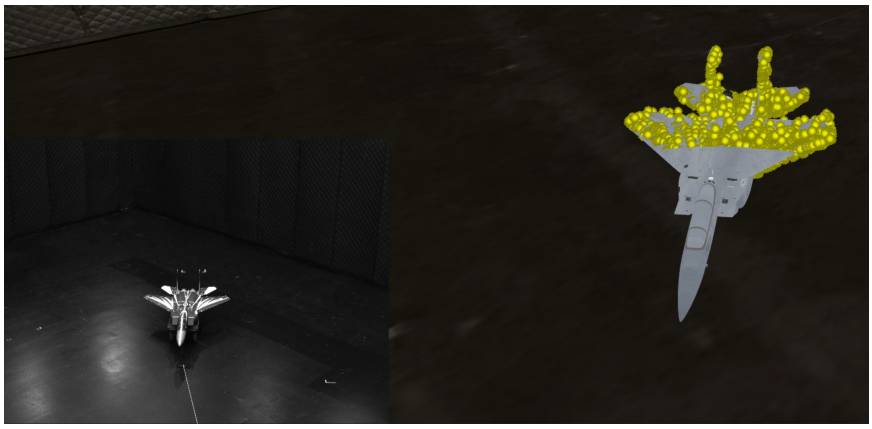

Figure 26: CUDA SBM with 4K Imagery and 256 numDisparity

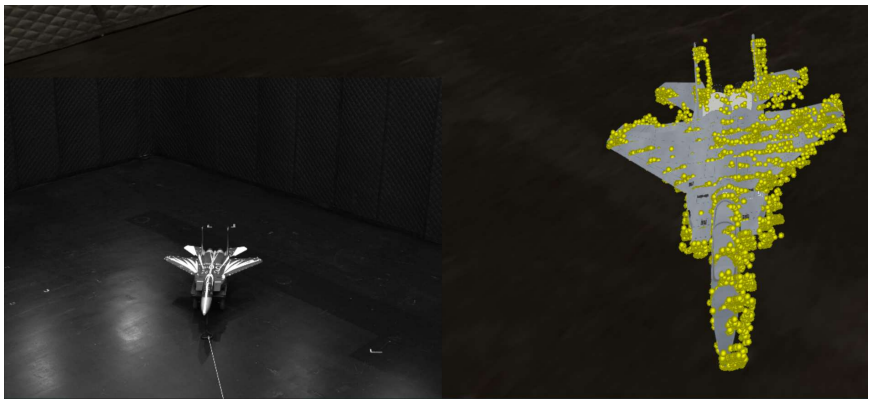

Figure 27: CUDA SBM with Downsampled 4K-2K Imagery and 256 numDisparity

Table 1: Calibration Adjustment

Left Cam/Right Cam Intrinsic Matrix	Stereo Cam Rectification Matrix	Stereo Cam Q Matrix
fx = fx/2	fx = fx/2	-cx = -cx/2
fy = fy/2	fy = fy/2	-cy = -cy/2
x0 = x0/2	x0 = x0/2	f = f/2
y0 = y0/2	y0 = y0/2	

Matching can generate a valid sensed point cloud from disparity map reprojections (Figure 27).

Many factors can affect the reprojection of disparities to the 3D virtual world, because the boom testing scenario utilizes real-world imagery from the Motion Capture Chamber. Figure 28 serves as an example of reprojection without a filter in the MCC. The walls, floor, and cart located in the MCC generate additional outlier points post-reprojection. In a real-world refueling scenario, there will be nothing near the receiver aircraft to cause outliers. As the motion capture system provides and updates the pose of the receiver aircraft, a stricter filtering technique can be applied. The virtual receiver model consists of a bounding box (Figure 29) that encapsulates the receiver within its boundaries. Using the bounding box and the known position of the virtual receiver, the filter eliminates any outliers located outside the bounding box, resulting in a similar point cloud generated with the near plane/far plane filter in the Probe and Drogue scenario.

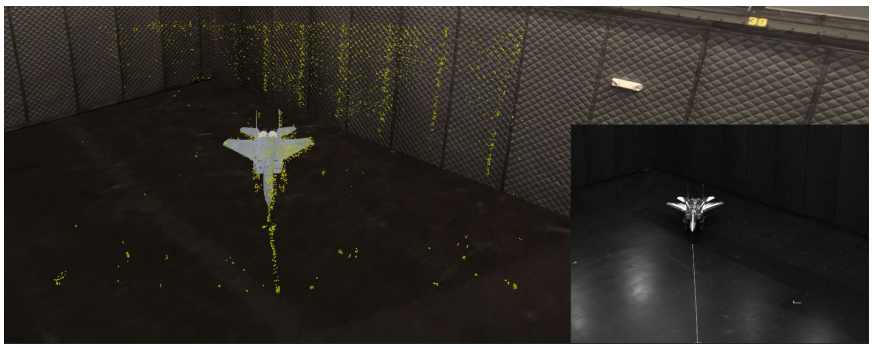

Figure 28: CUDA SBM with No Filter

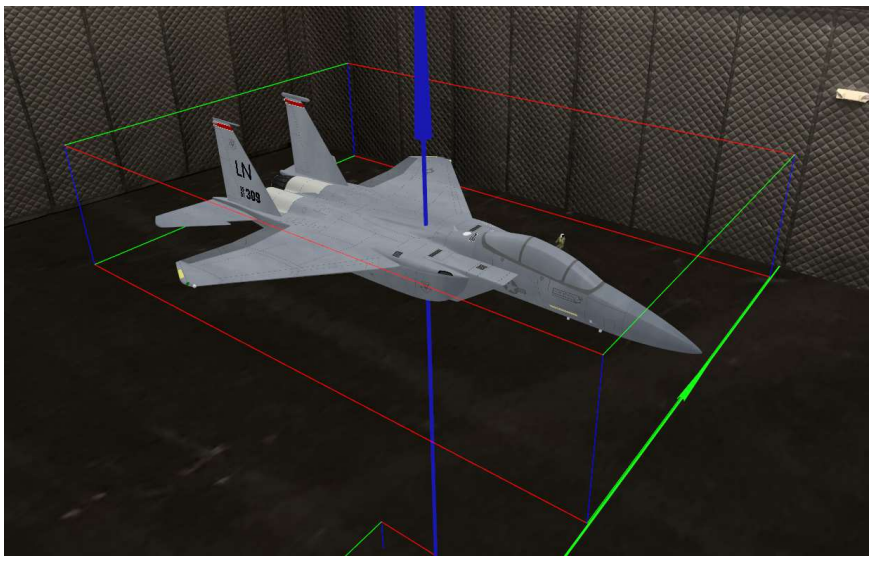

Figure 29: F-15 Bounding Box

3.6 Experiment Parameters

This section covers the metrics and testing scenario necessary to evaluate the newly implemented GPU accelerated AARVP against the CPU-based AARVP. It will then detail various scenario-specific experiments conducted to study boom and drogue occlusion and their effects on pose estimation. Finally, this section elaborates on the physical setup and camera parameters necessary for real-world image exposure testing. It will conclude with an explanation of the filtering techniques aiming to improve the solution.

3.6.1 Comparison Metrics and Testing scenario

Two criteria are necessary to judge the efficiency of the GPU accelerated AARVP against the CPU-based AARVP:

- Quantify the runtime of CPU and CUDA based Stereo Block Matching and Iterative Closest Point algorithms.
- Quantify the pose estimation results of both AARVP algorithms without occlu-

sion.

The experiments propose conducting two separate approaches for both the refueling drogue and boom scenarios:

3.6.1.1 Drogue Scenario

The drogue scenario had the receiver aircraft start at a precontact position 41 meters away from the 2K stereo vision sensors. The receiver moved slowly towards the camera, stopping at the contact position 10 meters away. The experiment analyzed 12,900 pairs of images during the approach.

3.6.1.2 Boom Scenario

The boom scenario had the scaled receiver aircraft start at a precontact position 22 meters away from the 4K camera located in the Motion Capture Room. The receiver aircraft was then pulled slowly in a straight line towards the camera until stopped at the contact position 15 meters away. The virtual receiver was updated using the Motion Capture Chamber truth data and mirrored the movement of the real-life receiver model. The experiment analyzed 490 pairs of images during the approach.

3.6.2 Collecting Results

The experiment recorded the runtime of both iterations of AARVP for different scenarios. The runtime data includes the timing data of individual components of AARVP and the total runtime of the AARVP under each implementation. It also recorded the pose estimation generated by ICP for each approach generated from the cameras' local coordinate frame specific to scenarios as visualized by Figure 30 and 31. For the boom scenario, the experiment generated a runtime with the CPU-based

AARVP and 4K native sensor resolution to serve as a baseline from the previous research. It compares the results with the downsampled 2K sensor results of CPU/GPU implementations.

3.6.3 OpenCV Settings

The experiment will use the following OpenCV settings for Stereo Block Matching:

The experiment also used uniform sampling to decimate the number of points generated by Stereo Block Matching to be equivalent to the number of points located

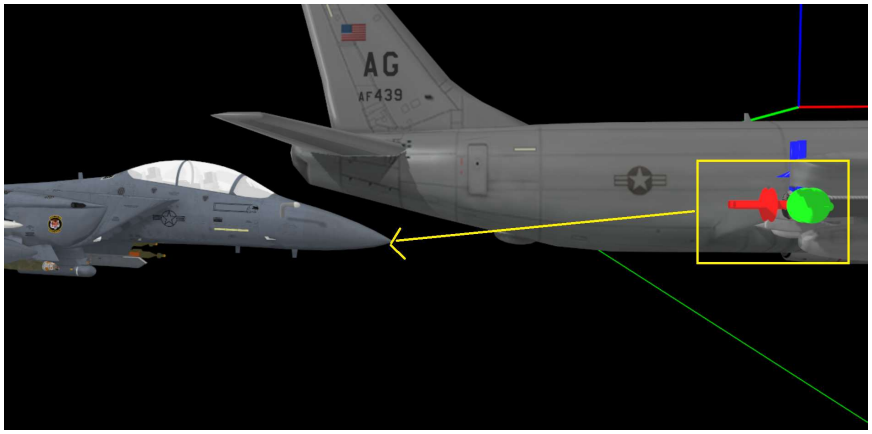

Figure 30: A visualization of the cameras' local coordinate frame in drogue scenario

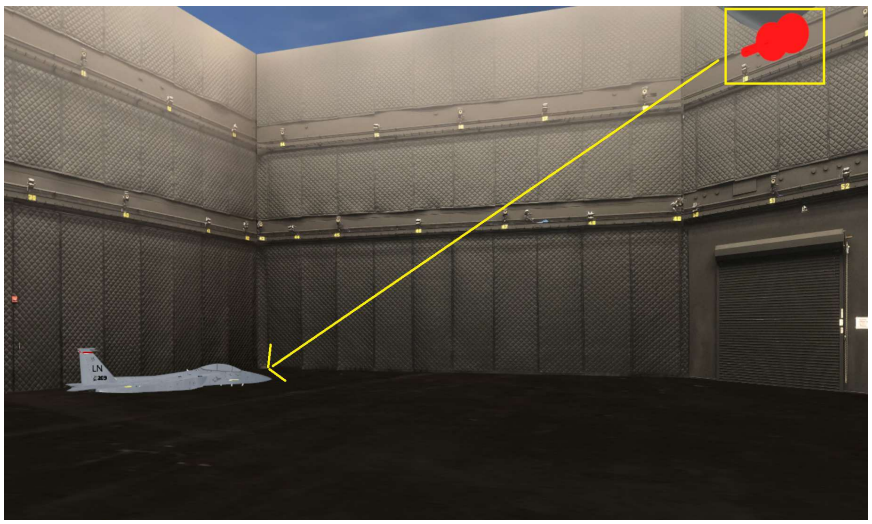

Figure 31: A visualization of the cameras' local coordinate frame in boom scenario

Table 2: OpenCV Stereo Block Match Settings

Settings
blockSize = 21
preFilterType = XSOBEL
preFilterSize = 9
preFilterCap = 31
minDisparity = 0
numDisparity = 256 for 2K Sensor Size / 512 for 4K Sensor Size
textureThreshold = 10
uniquenessRatio = 500
speckleRange = 0
speckleWindowSize = 0

in the truth model.

3.6.4 Computer Hardware

The experiment used the following Computer Hardware for timing tests:

Table 3: Computer Hardware

Parts
CPU = AMD Ryzen 9 5900x
RAM = 32GB DDR4 3600Mhz
GPU = NVIDIA Geforce RTX 3080Ti 12GB

3.7 Boom and Drogue Occlusion

This section goes over in-depth the scenario-specific experiments that generated boom and drogue occlusions. While using the scenario receiver approach patterns from above, the experiment added the following guidelines to the testing scenarios:

3.7.1 Drogue Scenario

In the drogue scenario, the experiment fully extends the hose and the basket in a rigid state for the receiver aircraft to establish contact with its refueling probe. After

making the connection, the receiver will undergo a simulation that mimics turbulence experienced during aerial refueling. The experiment obtained approximately 20 hose positions. AftrBurner, using a third-degree polynomial fit, rendered the hose in motion with the basket attached at the end of the hose. The AARVP pipeline obtains the pose estimation results from the cameras' local coordinate frame and records them as part of the experiment.

3.7.2 Boom Scenario

In the boom scenario, the location of the physical receiver poses a unique challenge. The receiver aircraft cannot be elevated above the ground, thus incapable of presenting itself in different image regions. The experiment will instead manipulate the pitch and yaw state of the boom to generate occlusions. During a receiver approach, the refueling boom will sweep through the entire refueling envelope for each image, occluding different regions of the receiver aircraft. The experiment will record the number of features pre and post-occlusion. It will use that data to quantify the effects occlusions have on pose estimation.

3.8 Real-world Imagery Exposure Testing

Real-world aerial refueling can occur at all times of day, from sunny conditions at noon all the way to pitch black at midnight. The images generated from these extreme conditions can be overexposed and underexposed. If the stereo vision sensor lacks auto exposure and gain compensation, AARVP must account for these conditions to generate a pose estimation. This section will cover the physical setup for real-world image exposure testing and explore image filters to see if they can improve the AARVP solution in edge cases.

The experiment uses the physical 4K cameras located in the MCC as the input

source. A series of ceiling lights illuminate the MCC but aren't bright enough to cause image overexposures. With the lights turned off, MCC appears dark to the human eye. However, the floor coating reflects the IR beams emitted from the Motion Capture Systems into the camera, causing the image to be more exposed than intended. The paper instead chooses to leverage camera settings to simulate overexposure and underexposure. Each camera contains settings to automate exposure and gain which is turned off to allow for manual control. We adjusted the exposure time of the cameras to allow for the camera shutters to close at the specified interval. The individual performing the experiment inspected the image capture to ensure the result mimics possible real-world exposure cases during refueling.

3.8.1 Nighttime Approach

For the nighttime approach, the experiment placed a 500W floodlight in front of the receiver approach path and turned off the lights in the MCC. The floodlight gives off a similar illumination generated by the refueling light located on the refueler tanker. The receiver at the precontact position is invisible to the human eye. As the receiver approaches the contact position, the front of the plane can be visible while the back remains hidden (Figure 32). The camera settings are:

Table 4: Camera Exposure Settings

Settings
Exposure = 2000
Gain = 0.1

3.8.2 Daytime Approach

For the daytime approach, the experiment kept the same scenario as the nighttime approach. With the exposure time set at 141000 and the gain set to 9, the wings

Figure 32: Underexposed Receiver at Contact

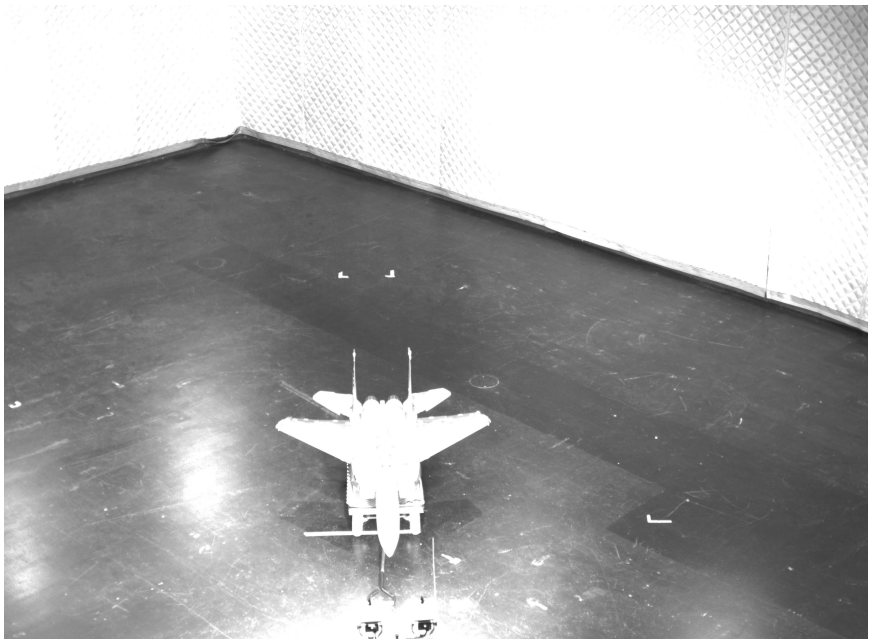

Figure 33: Overexposed Receiver at Contact

of the receiver appeared washed out, but the aircraft managed to retain its overall outline 33.

The experiment compares the number of features and the pose estimation gener-

ated by the edge cases with a previous collection with proper exposure settings for analysis.

3.8.3 Image Filtering

A possible method to improve the number of features in poorly exposed imagery is through histogram equalization. A histogram is a graphical representation of the tonal values of brightness in an image, with the left side of the histogram being 0 percent brightness and the right side being 100 percent. It contains components such as blacks, shadows, mid-tones, highlights, and whites. Histogram values in an underexposed image will be on the left, with possible data being clipped and lost. An overexposed image will shift the histogram values to the right and produce clippings of the highlights. Figures 34 and 35 are the histograms of the underexposed and overexposed imagery used above.

Figure 34: Underexposed Histogram of Figure 32

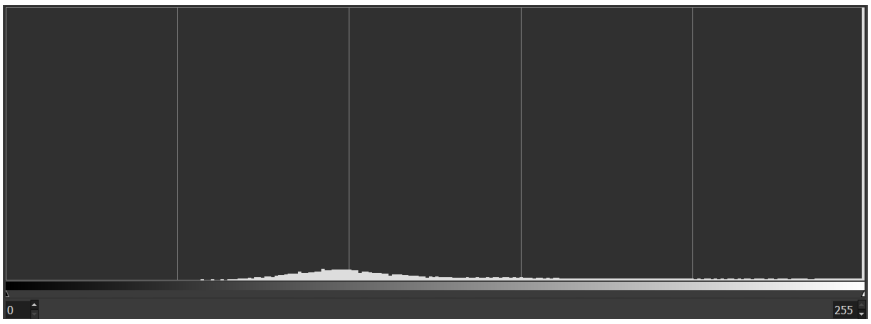

Figure 35: Overexposed Histogram of Figure 33

Histogram equalization will attempt to equalize the histogram's value and distribute it across the histogram spectrum (Figure 36). Previous studies [28] have demonstrated an improvement in feature generation using histogram equalization with disparity map computation. This experiment will explore two histogram equalization filters:

- Global Histogram Equalization
- Contrast Limited Adaptive Histogram Equalization (CLAHE)

The main difference between global histogram equalization and CLAHE is the operating range of pixels in the image. Global histogram equalization operates on the entire image's pixel all at once. CLAHE instead performs operations on tiles or small regions, iterating until completion. The experiment will evaluate the results of filter applications and compare them with non-filtered results obtained earlier.

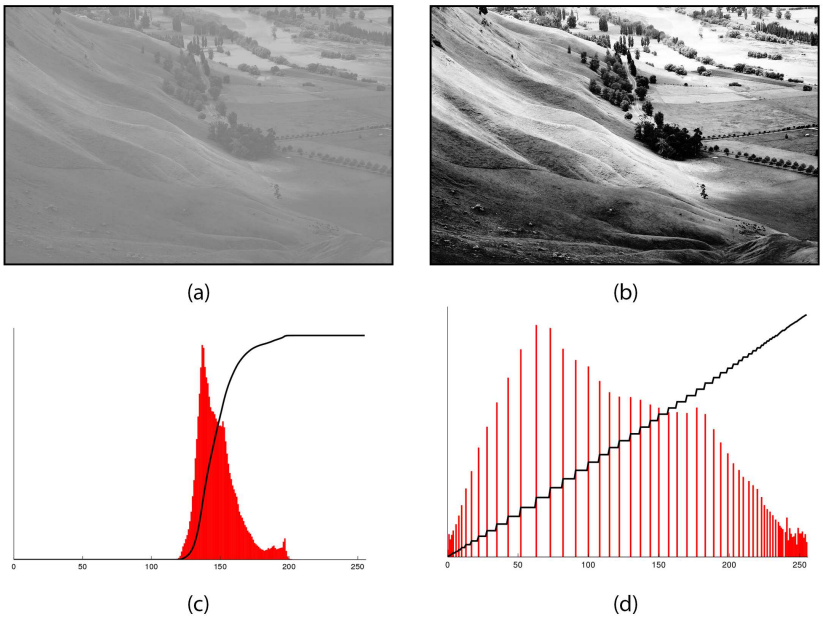

Figure 36: An Example of Histogram Equalization
Subfigure A and C shows an image and its histogram pre histogram equalization. Subfigure B and D shows the image and its histogram post histogram equalization.

IV. Results and Analysis

4.1 Preamble

This chapter presents the results from multiple experiments annotated in the methodology section. The chapter will provide and analyze the runtime data of both CPU and GPU-based AARVP using different stereo sensor parameters. The chapter will also highlight the translational and rotational errors of both AARVP algorithms for each framework and scenario. Finally, the chapter will discuss the results of image exposure experiments.

4.2 Probe and Drogue Framework

4.2.1 Straight in Approach Without Occlusion

4.2.1.1 Algorithm Runtimes

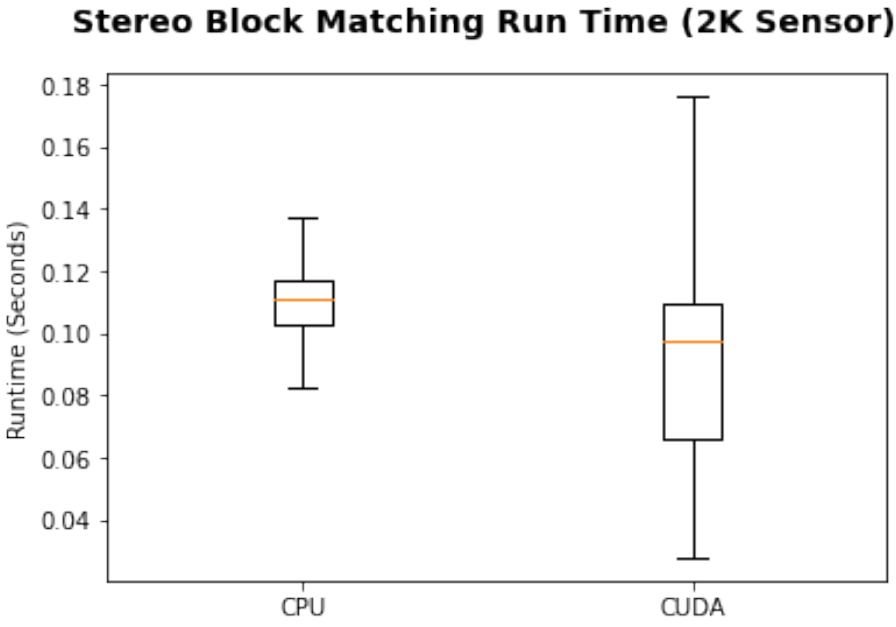

Figure 37: AARVP SBM Runtime (2K Sensor/ Virtual Images)

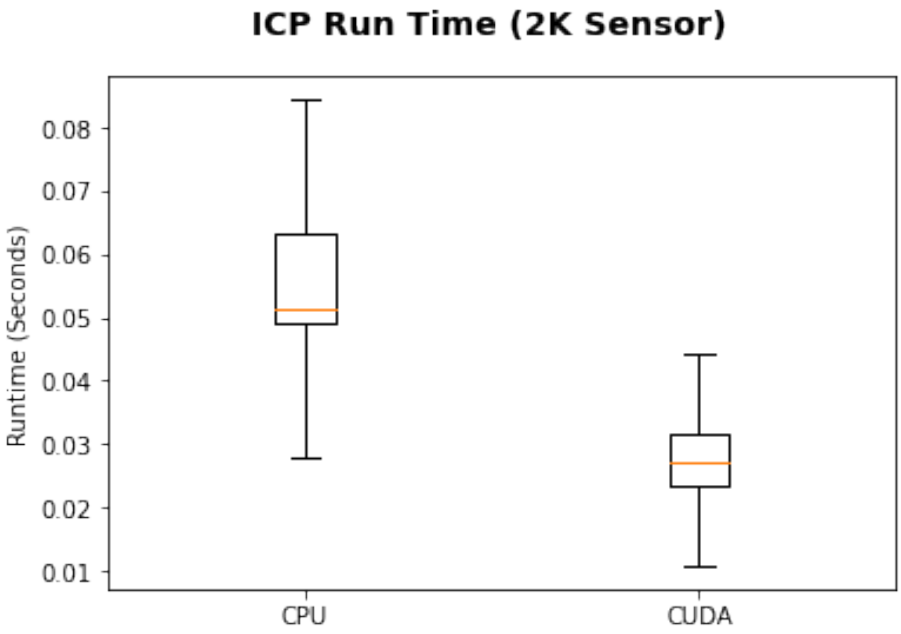

Figure 38: AARVP ICP Runtime (2K Sensor/ Virtual Images)

With a 2K stereo vision sensor resolution, CUDA-based SBM does outperform CPU-based SBM in some instances. However, the CUDA-based algorithm does encounter a much larger spread in execution time over the entirety of the data. The CPU-based algorithm has a tighter bound, making it more deterministic for runtime computation. The CUDA-based ICP algorithm outperforms the CPU-based version with a faster run time and a tighter bound, making it an ideal replacement solution in the AARVP.

4.2.1.2 AARVP Results

The follow results are generated with the coordinate frame from Figure 30:

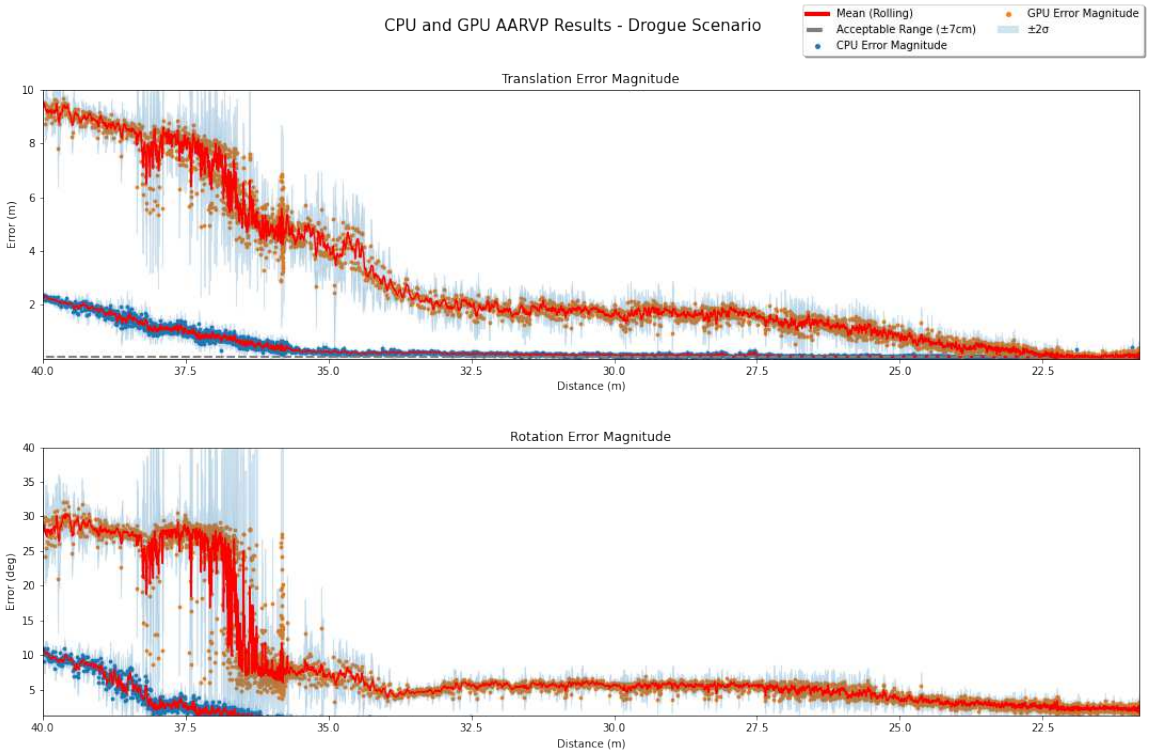

Figure 39: AARVP Errors (2K Sensor/ Virtual Images) (No Drogue)

Figure 39 indicated that CPU-based AARVP showcased lower translational and rotational errors compared to the GPU-based AARVP. The result stemmed from the convergence of the truth point cloud with the much cleaner point cloud produced by CPU-based SBM.

4.2.2 Straight in Approach with Drogue Occlusion

4.2.2.1 AARVP Results

With the drogue basket and hose present, CUDA-based AARVP produced lower translation and rotational errors than the CPU-based AARVP (Figure 40). At 27.5 meters away from the camera, the results from both algorithms converged with minimal differences.

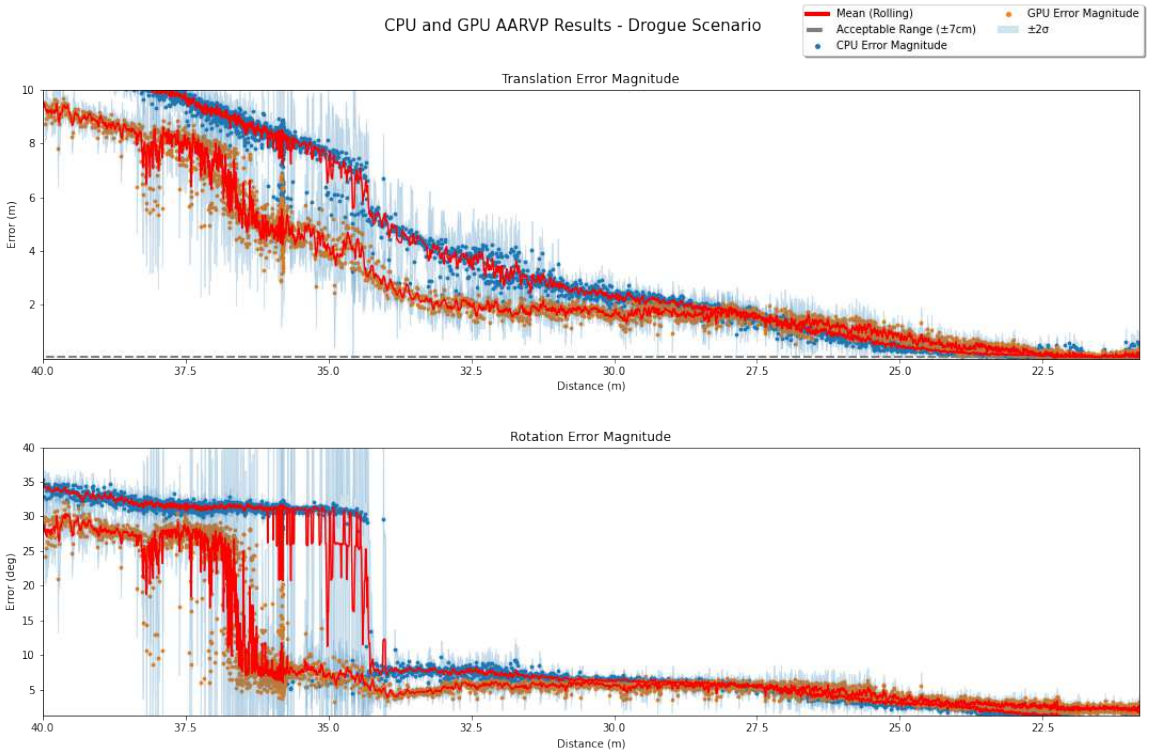

Figure 40: AARVP Errors (2K Sensor/ Virtual Images) (With Drogue)

4.2.3 At Contact With Turbulence

4.2.3.1 AARVP Results

At contact with turbulence, CPU-based SBM with the cleaner sensed point cloud produced lower translation and rotational errors against the CUDA-based SBM (Figure 41,42). Combined with the previous experiments in the drogue scenario, a combination of the CPU-based SBM along with the CUDA accelerated ICP brings forth a good balance between speed and accuracy.

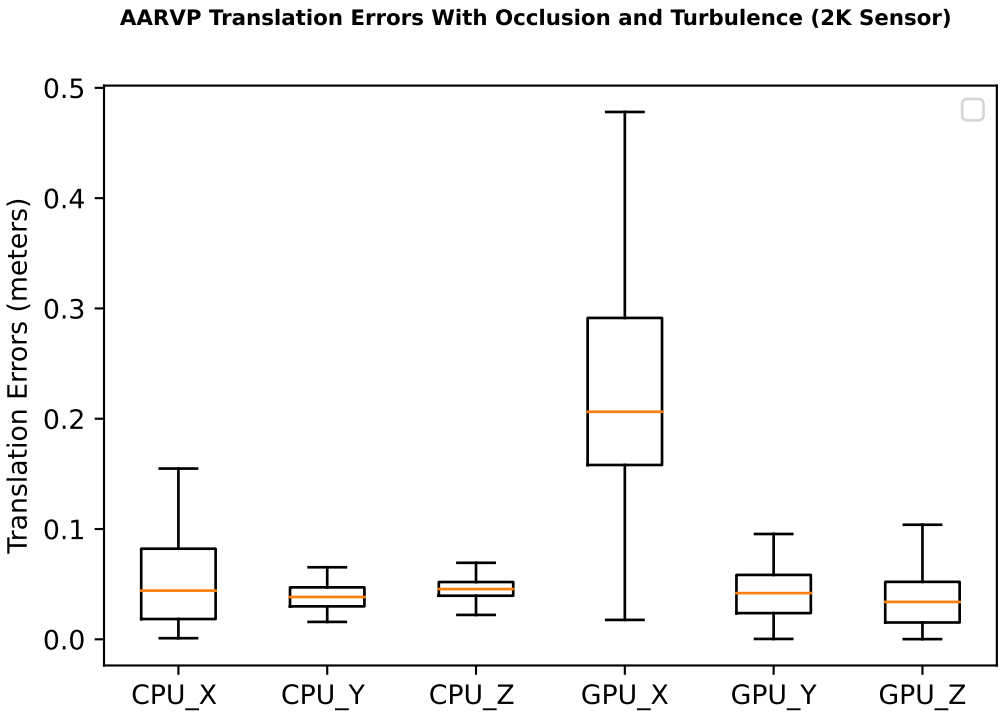

Figure 41: AARVP Translation Errors (2K Sensor/ Virtual Images)

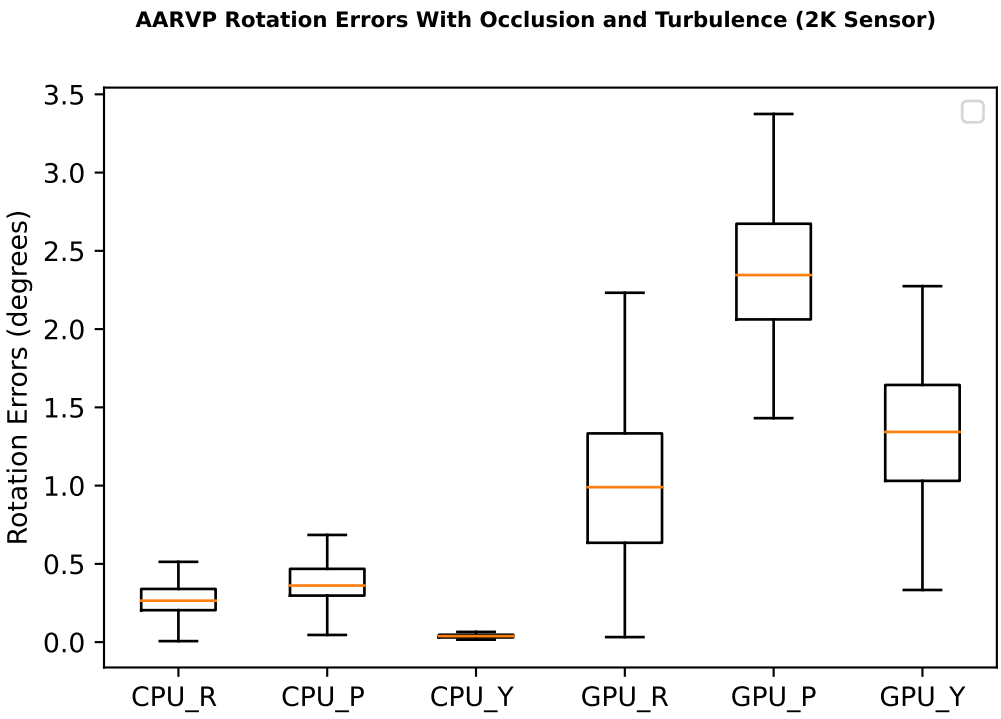

Figure 42: AARVP Rotation Errors (2K Sensor/ Virtual Images)

4.3 Boom Framework

4.3.1 Straight in Approach Without Occlusion

4.3.1.1 Algorithm Runtimes

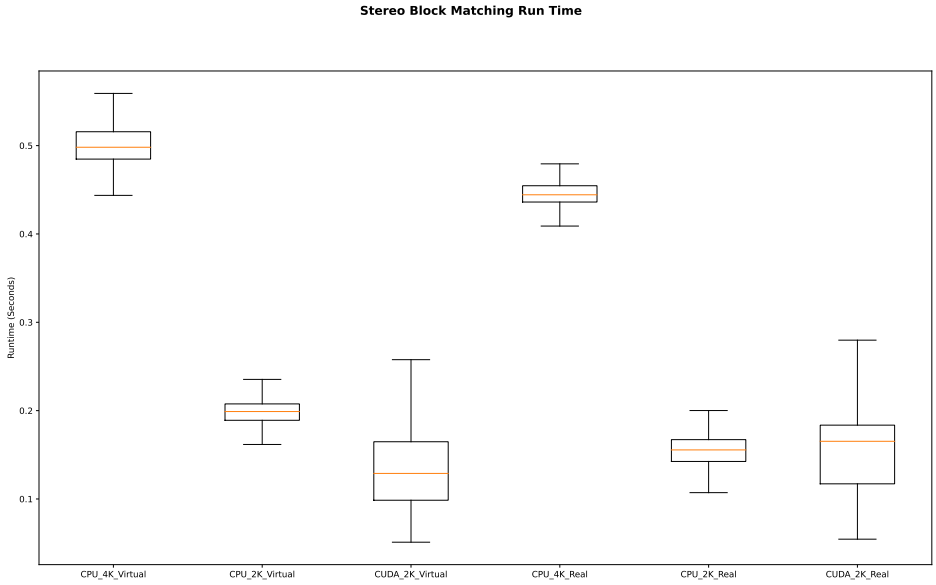

Figure 43: AARVP SBM Runtime (4K Sensor)

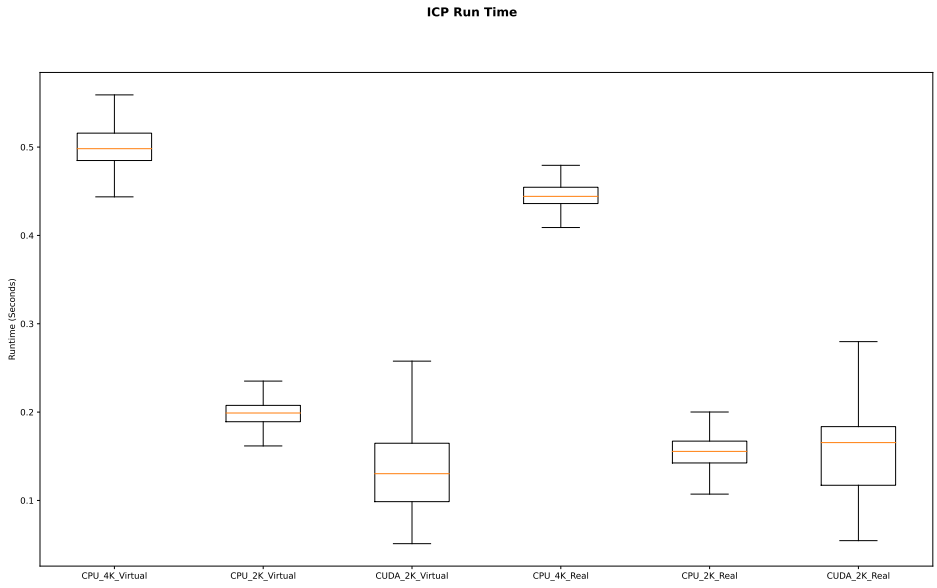

Figure 44: AARVP ICP Runtime (4K Sensor)

In Figures 43 and 44, the native 4K resolution images took the longest to process by SBM and ICP. Using Miller's camera adjustment techniques [27], the source images are resized from 4K to 2K resolution. CPU-based AARVP experienced a significant speedup with the 2K resolution images compared to the 4K native resolution images. CUDA-based AARVP outperformed both CPU-based solutions in certain cases but does suffer from the larger time spread as seen in the drogue scenario. The real-world imagery times followed closely with virtual images, which serves as a baseline for future work using virtual imagery alone.

4.3.1.2 AARVP Results

The follow results are generated with the coordinate frame from Figure 31:

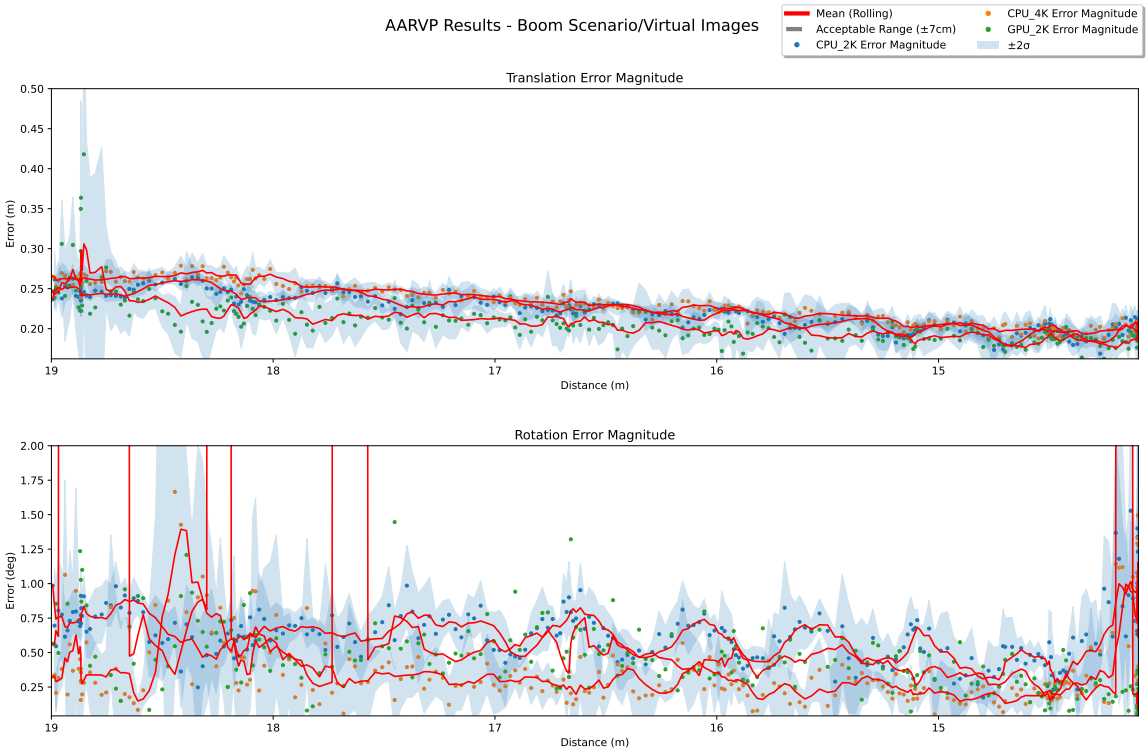

Figure 45: AARVP Results (No Boom/ Virtual Images)

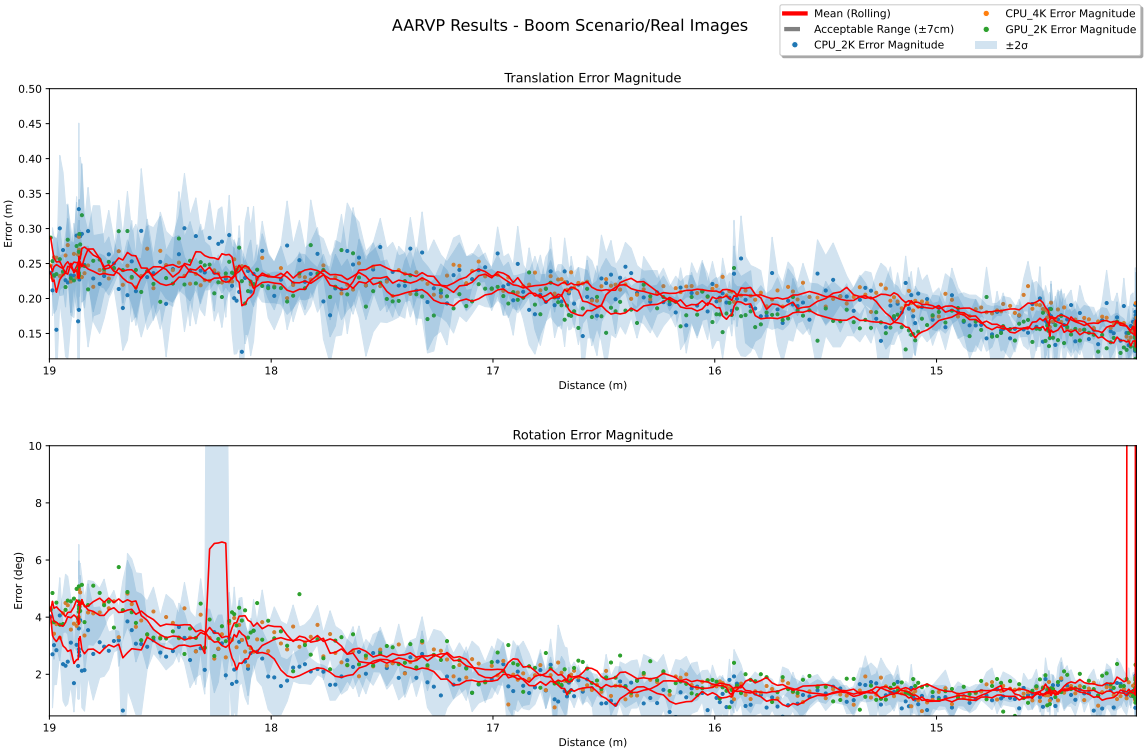

Figure 46: AARVP Results (No Boom/ Real Images)

All three SBM implementations generated results that trend closely to each other (Figure 45, 46). At the contact point, the differences in translational and rotation errors between the three implementations are minimal, suggesting an interchangeable approach of SBM processing to be used with the bounding box filter.

4.3.2 Boom Occlusion

The experiment places the receiver aircraft near the contact point 16 m away from the camera. The boom pitch is set to 35 degrees, 38 degrees, and 42 degrees while the boom yaw is alternated between -20 degrees and 20 degrees. The experiment configuration is within the safe refueling envelope of the KC-46 and ensures that all parts of the boom geometry can occlude the receiver aircraft.

4.3.2.1 AARVP Results

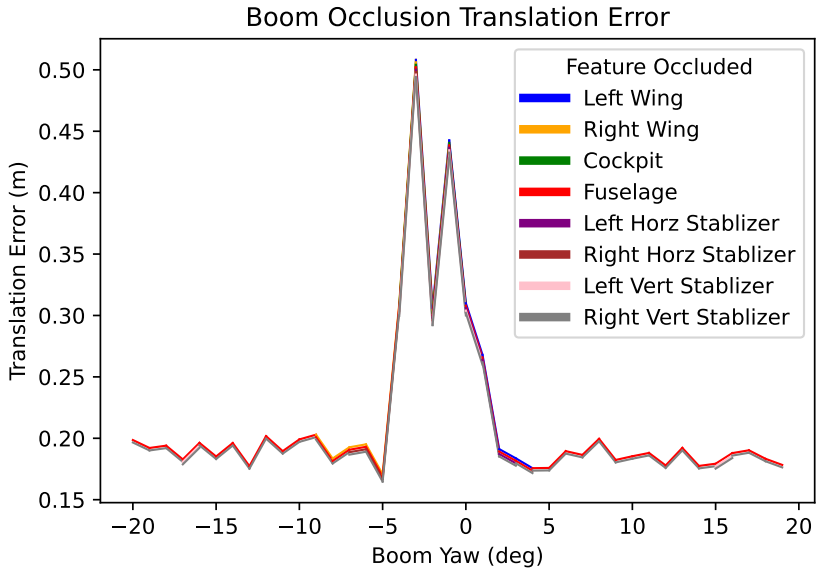

Figure 47: AARVP CPU Translation Error (Pitch Set to 35 Degrees)

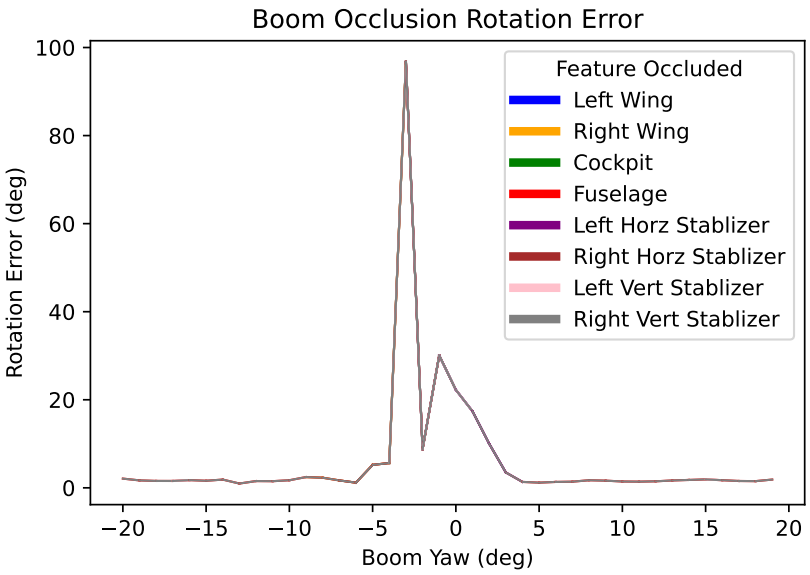

Figure 48: AARVP CPU Rotation Error (Pitch Set to 35 Degrees)

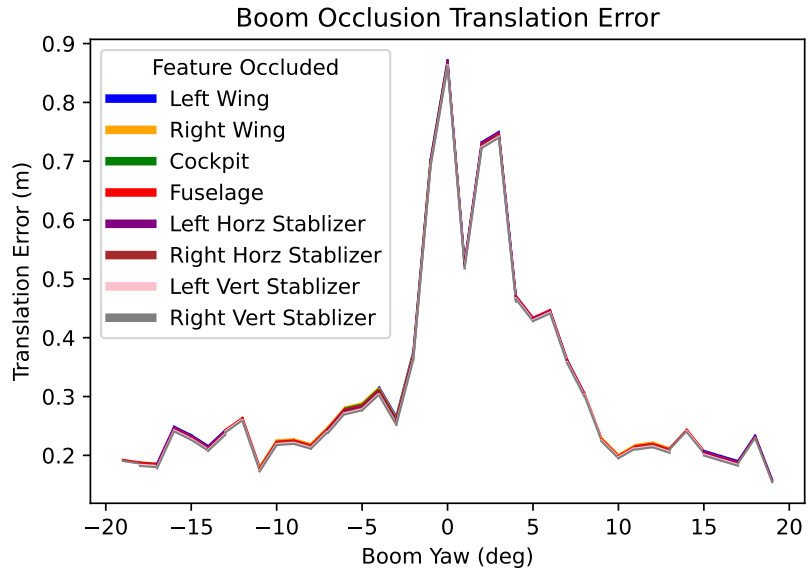

Figure 49: AARVP CPU Translation Error (Pitch Set to 38 Degrees)

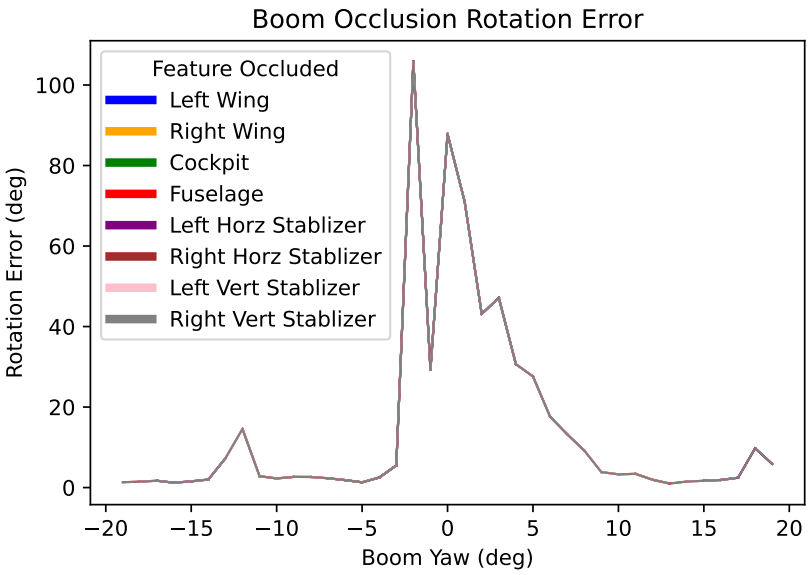

Figure 50: AARVP CPU Rotation Error (Pitch Set to 38 Degrees)

With the boom pitch set to 35 and 38 degrees, the boom's outer and inner tubes generated occlusion with the receiver aircraft. With the refueling boom set to 0 degrees in yaw, it predominately obstructed the cockpit, parts of the fuselage, and both the vertical stabilizers. It is during this boom position that ICP reported the highest translation and rotation errors. This suggests a challenge with refueling aircraft with centerline-based refueling receptacles.

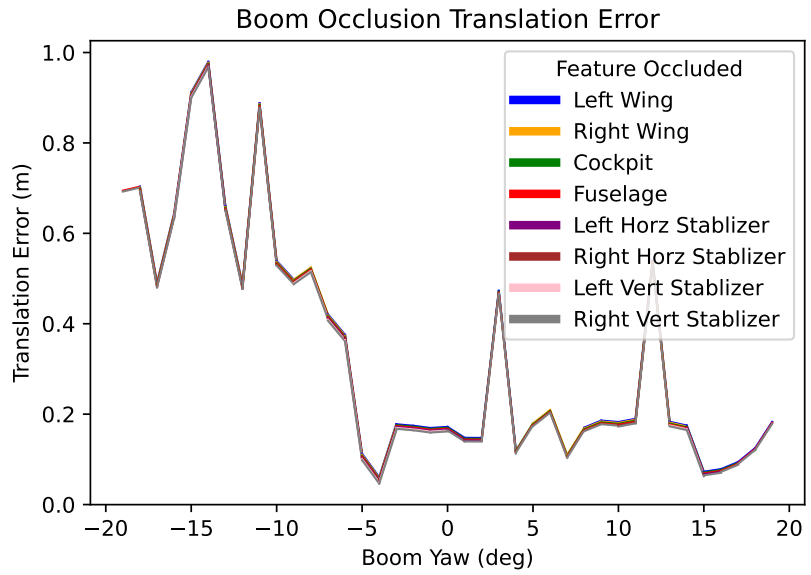

Figure 51: AARVP CPU Translation Error (Pitch Set to 42 Degrees)

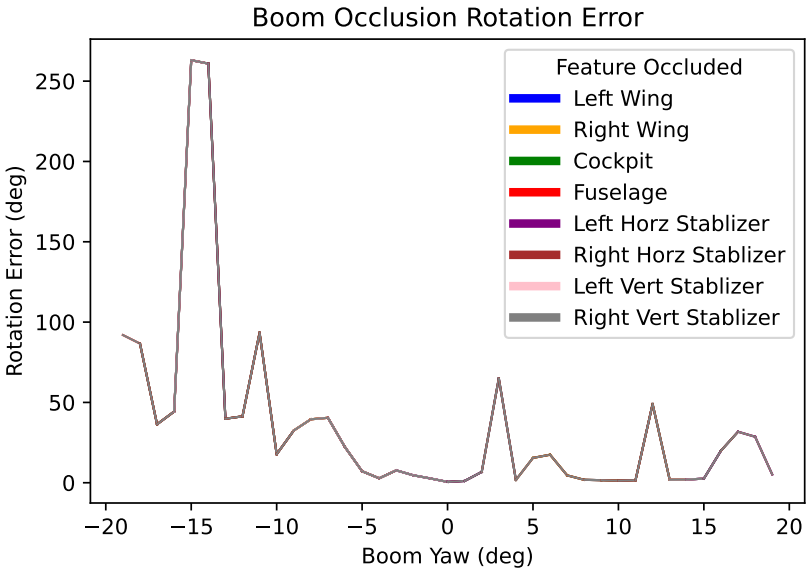

Figure 52: AARVP CPU Rotation Error (Pitch Set to 42 Degrees)

When the boom pitch is at 42 degrees, the refueling boom airfoil fins cause further obstruction of the receiver aircraft, resulting in different translation and rotation errors from previous observations. It can also be observed that both the rotation and

translation errors closely mirror each other with boom movement.

4.4 Exposure Testing

4.4.1 Underexposed Images

4.4.1.1 Feature Testing

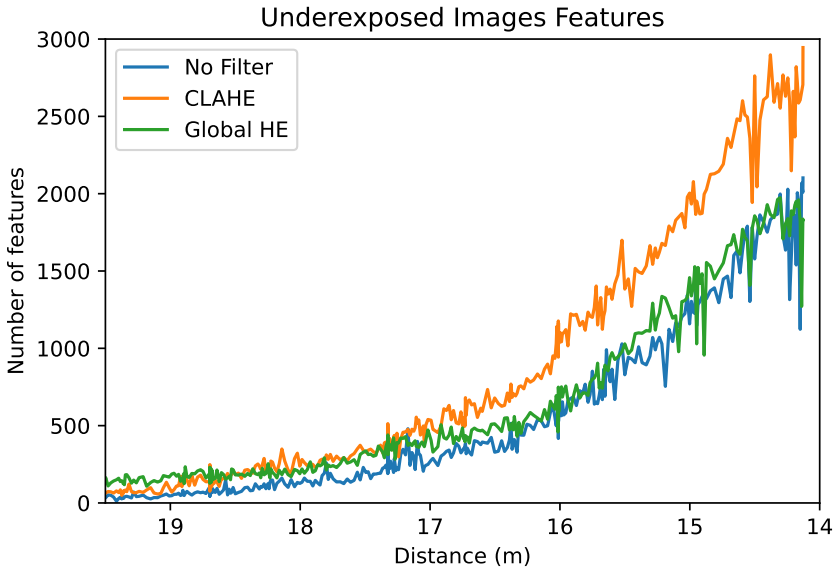

Figure 53: Number of Features Produced by SBM (Underexposed 4K Real World Images)

Using no filter and global histogram equalizer filter has a similar feature count with 2000 features at contact point. CLAHE produces a higher feature count with 3000 features at contact point.

4.4.1.2 AARVP Results

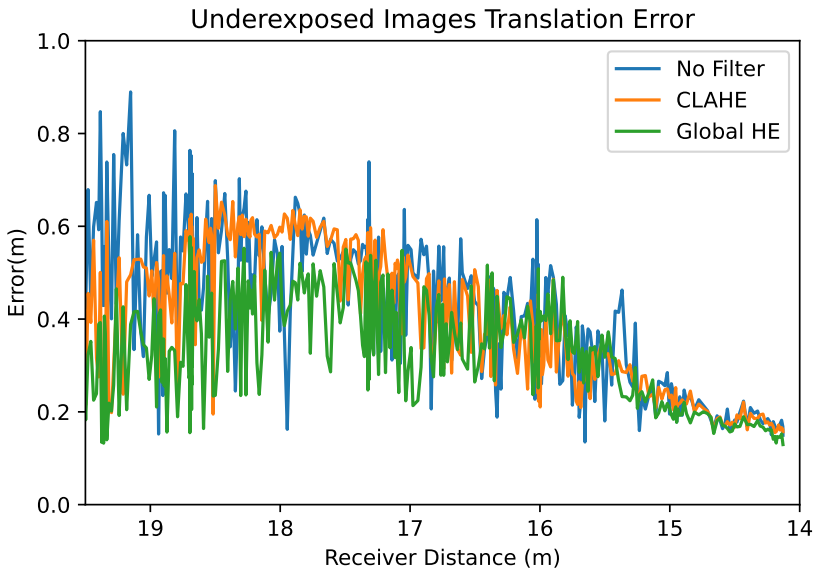

Figure 54: AARVP CPU Translation Error (Underexposed 4K Real World Images)

Figure 55: AARVP CPU Rotation Error (Underexposed 4K Real World Images)

The Global Histogram Equalizer produces the best result when the receiver aircraft is 19 meters away from the floodlight. As the receiver aircraft approaches the contact position, all three results converge to 20 centimeters in translation error magnitude

and 2 degrees in rotation error magnitude.

4.4.2 Overexposed Images

4.4.2.1 Feature Testing

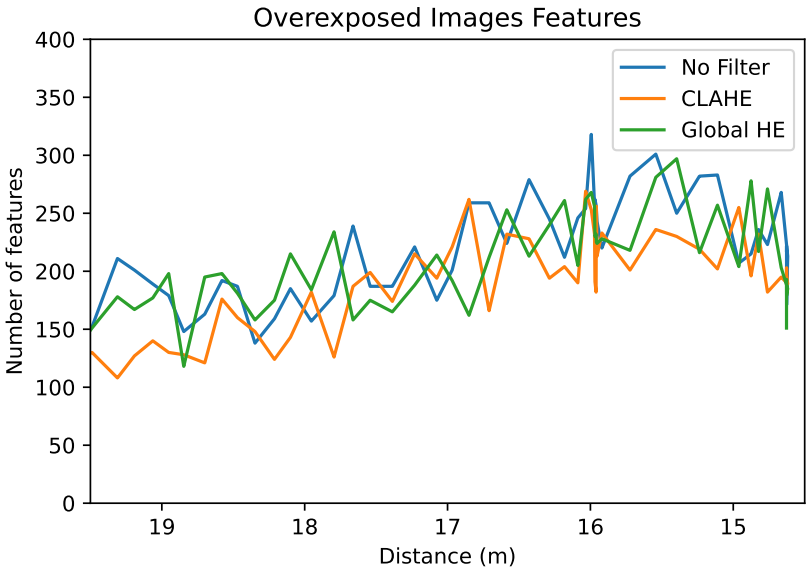

Figure 56: Number of Features Produced by SBM (overexposed 4K Real World Images)

With overexposed images, Stereo Block Matching produced significantly lower features when compared with underexposed images. The result indicates that the Stereo Block Matching algorithm favors the underexposed regions versus the overblown highlights, whereas human vision would prefer the opposite.

4.4.2.2 AARVP Results

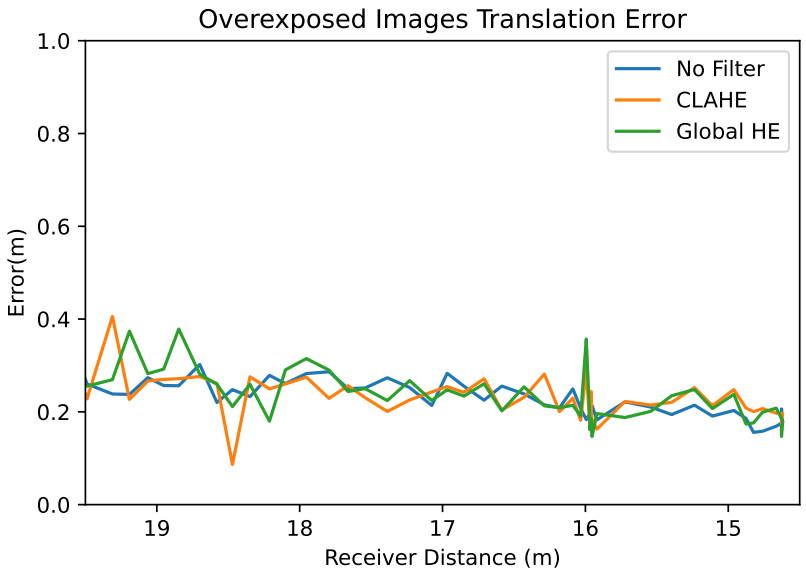

Figure 57: AARVP CPU Translation Error (Overexposed 4K Real World Images)

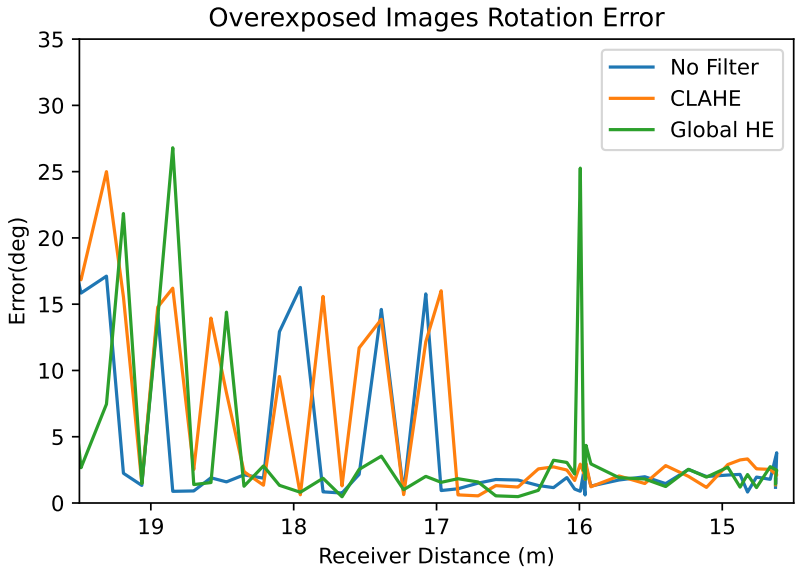

Figure 58: AARVP CPU Rotation Error (Overexposed 4K Real World Images)

Using no filters produce the best translation and rotation error even with fewer features with a translation error magnitude of 20 centimeters and a rotation error magnitude of 2 degrees at the contact position. Even though the wings of the re-

ceiver aircraft were overblown with highlights, Stereo Block Matching still managed to produce features on the edge of the receiver aircraft. The result indicates that the feature shape rather than the number of features can lead to a lower ICP error.

V. Conclusions

This paper discusses a GPU accelerated AARVP using CUDA architecture to generate pose estimation for the use of automated aerial refueling. The GPU-based SBM achieved a faster run time at certain frames when compared with the CPU-based SBM. However, the tighter bounded CPU-based SBM would be a good candidate to be paired up with the GPU-based ICP algorithm. From AARVP results, with a near/far plane filter, CPU-based SBM generated lower translation and rotational errors. With a bounding box filter, all three of the SBM implementations mentioned in III achieved similar results. This suggests that lower errors could be provided by AARVP when combined with an alternative pose estimation algorithm to generate the required bounding box around the receiver aircraft.

The paper also performed various occlusion testing for both the refueling probe and drogue and the refueling boom. The results indicate probe and drogue induced minimal errors to the AARVP pose estimation at all phases of refueling when utilizing CPU-based Stereo Block Matching. The refueling boom does introduce a significant error once the cockpit and the fuselage of the receiver aircraft are occluded. However, since the experiment utilizes a 1/7th scaled receiver model, the errors will likely reduce when the receiver aircraft is fully scaled.

Finally, the paper experimented with different simulated image exposures to test the adaptability of AARVP in real-world refueling environments. The results indicate a degraded tracking performance during both the underexposed and overexposed scenarios. Histogram equalization filters slightly improve the accuracy when the receiver aircraft is approaching from precontact to contact. The receiver aircraft produces good features for ICP to converge properly at the contact point, thus ridding the need for a histogram equalizer filter.

5.1 Future Work

The next steps that build upon this work are as follows:

- Power limits are a good factor for consideration when building AARVP to be deployed on refueling aircraft. Since AARVP demonstrates good performance on powerful desktop machines, the next step would be to explore the deployment on embedded systems. NVIDIA Jetson platforms with CUDA cores support GPU accelerated work. They serve as viable candidates for further testing.

- An augmented reality framework for probe and drogue is required to conduct further testing. It includes the mounting of physical stereo vision sensors with the necessary orientation and baseline specified by the refueling pod. A good camera calibration needs to be performed and tested with real-world imagery to determine the accuracy of AARVP for pose estimation in a real refueling environment.

- The physical camera setup on the refueler tanker can be mutated slightly due to turbulence and wear, leading to higher pose estimation errors. Online camera calibration methods needs to be investigated to correct intrinsic and extrinsic errors during flight operations.

Bibliography

1. James Anderson, Joel Miller, Xiaoyang Wu, Scott Nykl, Clark Taylor, and Warren Watkinson. Real-Time Automated Aerial Refueling with Stereo Vision: Overcoming GNSS-Denied Environments In or Near Combat Areas. *Inside GNSS*, 16(4):32–41, 2021.

2. Vincent J Bownes. Using motion capture and augmented reality to test aar with boom occlusion. Master's thesis, AIR FORCE INST OF TECH, 2021.

3. Adrian Kaehler and Gary Bradski. *Learning OpenCV 3: computer vision in C++ with the OpenCV library.* " O'Reilly Media, Inc.", 2016.

4. Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.

5. Ramtin Shams, Parastoo Sadeghi, Rodney Kennedy, and Richard Hartley. Parallel computation of mutual information on the gpu with application to real-time registration of 3d medical images. *Computer Methods and Programs in Biomedicine*, 99:133–146, 8 2010.

6. Magnus Halvorsen. Hardware acceleration of convolutional neural networks. Master's thesis, NTNU, 2015.

7. Alberto S. Garea, Dora B. Heras, and Francisco Argüello. Caffe cnn-based classification of hyperspectral images on gpu. *Journal of Supercomputing*, 75:1065–1077, 3 2019.

8. Arnas Ivanavičius, Henrikas Simonavičius, Julius Gelšvartas, Andrius Lauraitis, Rytis Maskeliūnas, Piotras Cimmperman, and Paulius Serafinavičius. Real-time

cuda-based stereo matching using cyclops2 algorithm. *EURASIP Journal on Image and Video Processing*, 2018(1):1–15, 2018.

9. Chad Mourning, Scott Nykl, Huihui Xu, David Chelberg, and Jundong Liu. Gpu acceleration of robust point matching. In *International Symposium on Visual Computing*, pages 417–426. Springer, 2010.

10. Ryan Dibley, Michael Allen, and Nassib Nabaa. Autonomous airborne refueling demonstration phase i flight-test results. In *AIAA atmospheric flight mechanics conference and exhibit*, page 6639, 2007.

11. Walton R Williamson, Gregory J Glenn, Vu T Dang, Jason L Speyer, Stephen M Stecko, and John M Takacs. Sensor fusion applied to autonomous aerial refueling. *Journal of Guidance, Control, and Dynamics*, 32(1):262–275, 2009.

12. Samer M Khanafseh and Boris Pervan. Autonomous airborne refueling of unmanned air vehicles using the global positioning system. *Journal of Aircraft*, 44(5):1670–1682, 2007.

13. Marco Mammarella, Giampiero Campa, Marcello R Napolitano, Mario L Fravolini, Yu Gu, and Mario G Perhinschi. Machine vision/gps integration using ekf for the uav aerial refueling problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(6):791–801, 2008.

14. Joseph A Curro II. Automated aerial refueling position estimation using a scanning lidar. Master's thesis, AIR FORCE INST OF TECH, 2012.

15. Daniel T Johnson, Scott L Nykl, and John F Raquet. Combining stereo vision and inertial navigation for automated aerial refueling. *Journal of Guidance, Control, and Dynamics*, 40(9):2250–2259, 2017.

16. William E Dallmann. Infrared and electro-optical stereo vision for automated aerial refueling. Master's thesis, AIR FORCE INST OF TECH, 2019.

17. Jinrui Ren, Xunhua Dai, Quan Quan, Zi-Bo Wei, and Kai-Yuan Cai. Reliable docking control scheme for probe–drogue refueling. *Journal of Guidance, Control, and Dynamics*, 42(11):2511–2520, 2019.

18. Lixin Wang, Haipeng Yin, Youguang Guo, Ting Yue, and Xiaopeng Jia. Closed-loop motion characteristic requirements of receiver aircraft for probe and drogue aerial refueling. *Aerospace Science and Technology*, 93:105293, 2019.

19. Xiaobin Xu, Haibin Duan, Yanjie Guo, and Yimin Deng. A cascade adaboost and cnn algorithm for drogue detection in uav autonomous aerial refueling. *Neurocomputing*, 408:121–134, 2020.

20. Siyang Sun, Yingjie Yin, Xingang Wang, and De Xu. Robust visual detection and tracking strategies for autonomous aerial refueling of uavs. *IEEE Transactions on Instrumentation and Measurement*, 68(12):4640–4652, 2019.

21. Yuebo Ma, Rujin Zhao, Enhai Liu, Zhuang Zhang, and Kun Yan. A novel autonomous aerial refueling drogue detection and pose estimation method based on monocular vision. *Measurement*, 136:132–142, 2019.

22. Nicholas Seydel, Will Dallmann, and Scott Nykl. Visualizing behaviors when using real vs synthetic imagery for computer vision. In *Proceedings of the International Conference on Scientific Computing (CSC)*, pages 3–9. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2018.

23. Bradley D Denby. Towards automated aerial refueling: Real time position estimation with stereo vision. Master's thesis, AIR FORCE INST OF TECH, 2016.

24. Scott Nykl, Chad Mourning, Mitchell Leitch, David Chelberg, Teresa Franklin, and Chang Liu. An overview of the steamie educational game engine. In *2008 38th Annual Frontiers in Education Conference*, pages F3B–21. IEEE, 2008.

25. Yong-Jun Chang, Sojin Kim, and Moongu Jeon. Efficient stereo matching method using elimination of lighting factors under radiometric variation. In *VISIGRAPP (4: VISAPP)*, pages 775–782, 2020.

26. Ryan M Raettig. Accelerating point set registration for automated aerial refueling. Master's thesis, AIR FORCE INST OF TECH, 2021.

27. M. Miller, Joel. Considerations using iterative closest point in presence of occlusions in automated aerial refueling. Master's thesis, AIR FORCE INST OF TECH, 2022.

28. Sebastian Metzner, Rahul Ramesh, and Tino Hausotte. Local histogram equalization and homomorphic filtering to improve wide-angle stereo disparity computation. In *Forum Bildverarbeitung 2018*, page 265. KIT Scientific Publishing, 2018.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
25–03–2022	Master's Thesis	June 2020 — Mar 2022

4. TITLE AND SUBTITLE

Real Time Evaluation of Boom and Drogue Occlusion with AAR

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Xiaoyang Wu

5d. PROJECT NUMBER

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-22-M-077

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

AFRL/RQQC
Dan Schreiter
WPAFB OH 45433-7765
COMM 937-938-7765
Email: dan.schreiter@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQQC

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

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

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In recent years, Unmanned Aerial Vehicles (UAV) have seen a rise in popularity. Various navigational algorithms have been developed as a solution to estimate a UAV's pose relative to the refueler aircraft. The result can be used to safely automate aerial refueling (AAR) to improve UAVs' time-on-station and ensure the success of military operations. This research aims to reach real-time performance using a GPU accelerated approach. It also conducts various experiments to quantify the effects of refueling boom/drogue occlusion and image exposure on the pose estimation pipeline in a lab setting.

15. SUBJECT TERMS

Aerial Refueling, Computer Vision, Motion Capture, Augmented Reality, Boom, Drogue

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Scott L. Nykl, AFIT/ENG
U	U	U	UU	67	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x4395 scott.nykl@afit.edu