3-2020

Determining Virtual Practicality from Physical Stereo Vision Images and GPS

Bradley S. French

DETERMINING VIRTUAL PRACTICALITY FROM PHYSICAL STEREO VISION IMAGES AND GPS

THESIS

Bradley S French, 2d Lt, USAF

AFIT-ENG-MS-20-M-020

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-20-M-020

DETERMINING VIRTUAL PRACTICALITY FROM PHYSICAL STEREO VISION IMAGES AND GPS

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

Bradley S French, B.S.C.S.

2d Lt, USAF

March 19, 2020

AFIT-ENG-MS-20-M-020

DETERMINING VIRTUAL PRACTICALITY FROM PHYSICAL STEREO VISION IMAGES AND GPS

THESIS

Bradley S French, B.S.C.S.
2d Lt, USAF

Committee Membership:

Dr. Scott L. Nykl, Ph.D
Chair

Dr. Douglas D. Hodson,, Ph.D
Member

Dr. Clark N. Taylor,, Ph.D
Member

AFIT-ENG-MS-20-M-020

Abstract

Current research efforts for Automated Aerial Refueling (AAR) at The Air Force Institute of Technology (AFIT) utilize Stereo Computer Vision to compute a relative pose between a tanker and receiver aircraft. Due to costs, time, and availability, it can be onerous to test these algorithms using actual Air Force (AF) aircraft. Our solution to this problem consists of using a 3D Graphics Engine to simulate AAR endeavors. However, the question then arises, "Does the virtual world accurately represent the physical world?" This can be explored by comparing a set of truth data to a similar set of virtual data. First, a set of truth data is collected using physical aircraft. Next, using the same flight path as that of the truth data, a set of virtual data is collected. Finally, a comparison of the physical and virtual data can provide information regarding how well the virtual world accurately represents the physical world, and if so, to within what margin of error? The results show that the virtual world roughly approximates the physical world but performs 2-6x better on average.

Table of Contents

Page

List of Figures

List of Tables

DETERMINING VIRTUAL PRACTICALITY FROM PHYSICAL STEREO VISION IMAGES AND GPS

I. Introduction

Recently, the unmanned aerial vehicle (UAV) has become the focus of the Air Force (AF) regarding aircraft. UAVs complete the same task as a manned aircraft; however, there is not a pilot in the cockpit. The UAV is controlled by a remote pilot placed in a remote location. Although UAVs operate similarly to manned aircraft, UAVs do not have the capability of being refueling in the air. The AF is well known to utilize aerial refueling capabilities during operations. However, due to the inability to aerially refuel UAVs, the AF has to limit the use of them. Reasonably, it would benefit the AF to find a way to provide aerial refueling to UAVs.

The limitation of UAV aerial refueling originates from the precision it requires. UAVs encounter a communication latency between the aircraft and the operator during flight movements, which currently induces too much latency to allow safe aerial refueling techniques for UAVs. Although, if a safer method is discovered, it would significantly benefit the AF. Assuming UAV aerial refueling ability and pilot availability, UAVs could theoretically fly indefinitely. They would not need to land unless maintenance or a system failure occurred.

Current research efforts aim to fulfill UAV aerial refueling requirements through usage of Automated Aerial Refueling (AAR) capabilities. The fundamental principle of AAR aims to automate the process of aerial refueling between the tanker and receiver aircraft. AAR has been attempted in many ways. Nearly all research currently focuses the position and orientation difference between the tanker and receiver, also

known as the relative pose between the aircraft. Assuming the relative pose can be correctly computed, the UAV can be correctly placed during aerial refueling with a low enough latency to refuel the UAVs in the air.

In March 2019, a flight test was conducted at Test Pilot School (TPS) at Edwards Air Force Base (AFB). Truth data was collected using an Inertial measurement units (IMU) unit to record orientation and Global Positioning System (GPS) units to record positioning. Additionally, stereo vision cameras were placed to capture images for usage during AAR algorithms. Using the collected truth data, these same tests can be re-run inside the virtual world. The resulting pose estimations from the physical world and virtual world can provide comparisons between the two domains. If proven comparable, the virtual world can then be utilized as a low-cost, rapid indicator for performance in the physical domain.

1.1 Problem Statement

Initial AAR research efforts aim to correctly compute the relative pose between the two aircraft involved. Due to costing millions of dollars, preparation exceeding six months, and short availability of resources, physical flight tests are not ideal for AAR experiments. An optimal solution would utilize 3D virtual simulation to allow realistic, rapid experiments of algorithms. However, the virtual world in discussion would need to be quantified as a viable and optimal tool, predictive of the real world. The virtual world would need to accurately predict the real world's corresponding behavior. The first problem is obtaining truth data in order to feed actual flight profiles and stereo imagery into the virtual world. Second, a set of equivalent virtual imagery corresponding to the actual flight profiles would need to be obtained. Third, relative pose estimations would need to be calculated for both sets of data. Finally, we perform a quantitative analysis testing to quantify the correspondence between

the physical world and the virtual world.

1.2 Assumptions

This research involves specific assumptions to determine how well the relative pose estimation functions. First, the relative pose estimations perform differently at varying distances, thus the refueling point is essential knowledge. This research defines the refueling point at 30 meters from the cameras position. Figure 1 displays refueling point compared to the camera position which is displayed as a blue circle. Next, it can be assumed that the receiver and tanker will have a relatively similar look direction, also known as orientation. This information becomes relevant when determining how to place the filtering model referenced in Section 3.6. Finally, we assume there will be more points toward the front of the receiver in the point clouds. The stereo vision sensor will collect more data at the front of the nose, wings, and fuselage than any other part.

Figure 1: Aerial Refueling via Boom Method [1]

1.3 Contributions

This research most impacts the utilization of the 3D graphics engine at The Air Force Institute of Technology (AFIT) to simulate AAR problems. Computer simulations are a growing field that allow experiments to run in a timely, automated manner. This research would allow researchers to focus on furthering AAR solutions without the hassle of preparing physical flights, and it continues to refine and develop the current techniques of AFIT's AAR stereo vision algorithms. Ultimately, this research aims to find and correct any potential errors related to the truth data. Any errors related to the physical world can then be accounted for in the virtual world to increase fidelity.

1.4 Overview

This Thesis has five chapters: The Introduction, Background, Methodology, Results and Analysis, and Conclusion. The Introduction, this chapter, introduces the research problem. The Background chapter discusses previous work completed in the AAR field as well as terminology necessary to understand this thesis in full. The Methodology provides an in-depth exploration of new topics, why they were researched, and how they were tested. The Results and Analysis chapter expresses all of the results from the flight data using the methods discussed in chapter three and discusses a more extensive search into the physical compared to virtual resulting pose estimations. Lastly, the Conclusion chapter explores the affect of the new data and discusses prospective future work.

II. Background and Literature Review

2.1 Automated Aerial Refueling

The United States Air Force (USAF) leads the world in air superiority. USAF aircraft are known to be some of the most powerful in the world. Although these aircraft are robust, they all require fuel to maintain operations. Two techniques have proven dominant for aerial refueling: the flying boom method and the probe-and-drogue method [7]. [8] explores the switch over to the boom aerial refueling method in the 1950s due to the priority of refueling bombers at 6000lbs per minute compared to the 2000lbs per minute that the probe-and-drogue method refuels. Regardless of the Air Force (AF)'s choice, neither refueling method supports aerial refueling for unmanned aerial vehicle (UAV)s.

UAV operators pilot the UAVs from a remote location instead of a cockpit. Understandably, there exists a time latency between the time the pilot sends a message and the time the UAV receives the message [9]. [9] discusses that the latency ranges from 100 to 160 milliseconds. For aerial refueling purposes, delays of 100 milliseconds significantly deteriorate performance. Additionally, delays of 250 to 300 milliseconds demonstrate inoperable conditions for aerial refueling. Thus, current latencies for UAV do not support aerial refueling for UAVs [10].

In hopes of utilizing aerial refueling for operations involving UAVs, Automated Aerial Refueling (AAR) research efforts aim to aerially refuel all types of aircraft. The initial problem of AAR consists of locating the relative position and orientation, also known as the relative pose, of the receiver aircraft in relation to the relative pose of the tanker aircraft. Figure 2 illustrates the idea of calculating the relative pose estimation between the tanker and receiver aircraft. The red line represents the relative pose from the stereo vision cameras mounted on the tanker to that of

the receivers position. This image illustrates the gray truth tanker (left), gray truth receiver (right), and the red sensed model. The red model represents the calculated relative pose estimation from the position of the stereo vision cameras mounted on the tanker. The value between the truth receiver pose and the red sensed model pose estimation represents the estimation error between the models. The following sections in this chapter discuss the properties and methods to calculate the relative pose estimation and the estimation errors. Nonetheless, once the relative pose is found, one could potentially automate the process of moving the boom and refueling the aircraft. Presently, there is not a complete and reliable, known solution that calculates relative pose with sufficient accurracy in a passive, non-deniable way.

Current research efforts for AAR calculate the relative pose between two aircraft using a technique known as stereo vision, which utilizes a pair of cameras to take images, match features between the images, reproject to a sensed 3D point cloud, and use a registration algorithm known as Iterative Closest Point (ICP) to calculate a relative pose estimation using the sensed 3D point cloud. Ideally, this calculation should be without error. However, due to a series of related error sources prior to the pose estimation, the final calculation will be inaccurate. This research aims to reduce these inaccuracies. Finally, a valid comparison between the physical and virtual world can be obtained.

Figure 2: Demonstration of AAR Relative Pose Estimation Problem

2.2 Stereo Computer Vision

Stereo computer vision consumes a pair of 2D images and outputs a 3D point cloud. Similar to the human eyes, it utilizes a pair of lenses to formulate the x or *depth* component of the 3D point cloud [11]. Stereo vision cameras require at least two cameras since the depth component cannot be obtained from a single camera. Stereo computer vision can be utilized to solve a variety of problems [12]. For this research, stereo vision will provide 3D point cloud estimates. Additionally, the resolution of the cameras used determin+es the density and depth resolution of the point cloud. Thus, the better the resolution of the cameras, the better the relative pose estimations. The sections below describe the process of stereo vision and how it relates to AAR.

2.2.1 Pinhole Camera Model

The stereo vision process begins with collecting a pair of images, which are also referred to as a set of pixels. These pixels must then be transformed into a 3D environment in order to calculate pose estimations. The pinhole camera model provides the framework to translate the properties of a camera into mathematical properties. These mathematical properties are then utilized to locate a pixel's location within a 3D space. Figure 3 illustrates the transformation process. To accomplish the task of transforming a pixel into a 3D space, the pinhole camera model is constructed by a tiny pinhole that determines the light that passes through the aperture. Then, the focal point, F_c as illustrated in Figure 3, is placed in the center and establishes the optical axis, Z_c, which passes through the principal point (c_x, c_y). The light that was refracted from a point, point P in Figure 3, in 3D space travels through the point (u, v). The point (u, v) is the pixel from an image that would describe point P in 3D space.

2.2.2 Epipolar Geometry

Epipolar geometry is the geometrical concept to translate a pair of pixels obtained from a pair of 2D images into a single location within a 3D virtual environment. Provided a point X, as indicated in Figure 4, an epipolar plane, the green plane in Figure 4, can be drawn with the points X, O_R, and O_L where O_R and O_L are the focal points of the right and left camera, respectively. The plane will then intersect the points X_L, e_L, X_R, and e_R. Since these point intersections are now known, the problem is determining what point X_R corresponds to X_L. To determine the points, epipolar lines are drawn to intersect each camera image plane. For example, the red line that passes through X_R and e_R is an epipolar line. Next, an epipolar baseline is then created to connect the focal points of the camera. The line that passes through

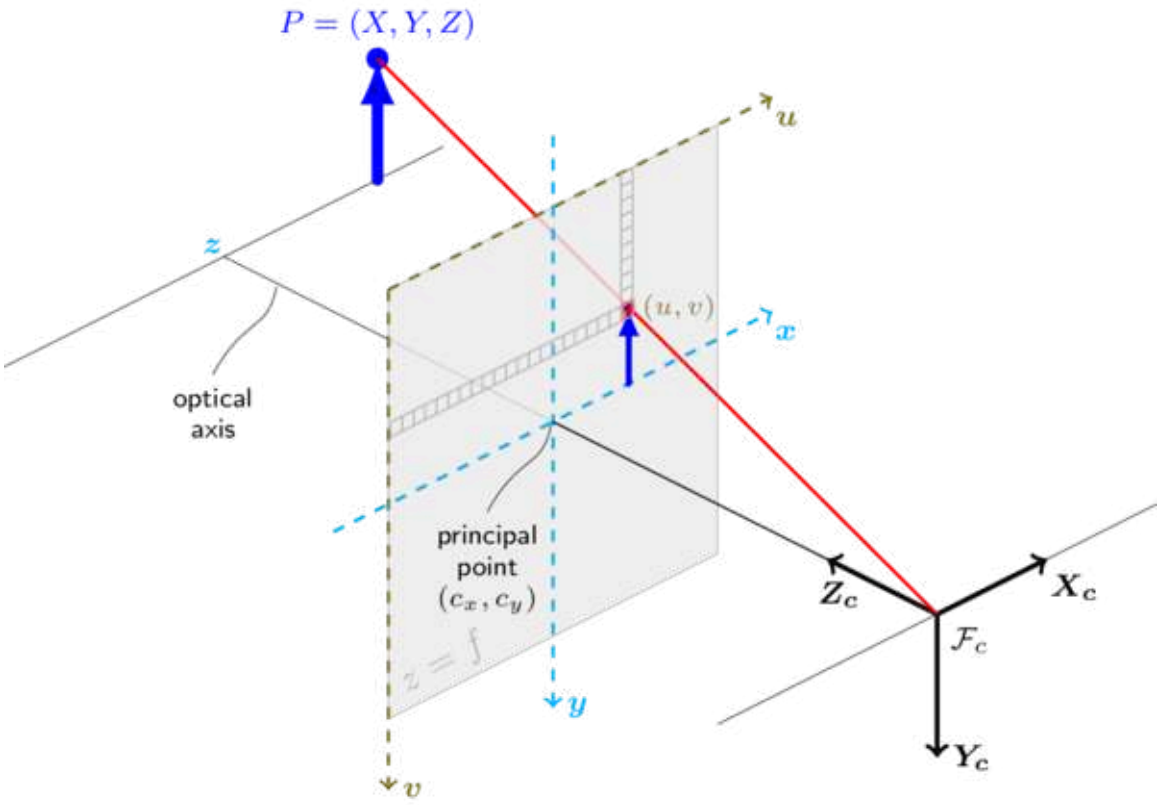


Figure 3: Pinhole Camera Model [2]

O_L and O_R is the epipolar baseline in Figure 4. Assuming the epipolar baseline, camera focal lengths, and the depth of the points are known, the point X_R can be identified.

2.2.3 Camera Calibration

Camera calibration determines the necessary values for epipolar geometry, such as the epipolar baseline and camera focal lengths. Camera calibrations for stereo vision cameras consist of intrinsic and extrinsic properties. The intrinsic properties are the values for each camera such as the focal length and principal point location. The extrinsic properties are the values associated with both cameras such as the translation and rotation difference between the cameras, also known as the epipolar baseline [13, 14, 15]. The process of calibrating cameras begins with capturing imagery

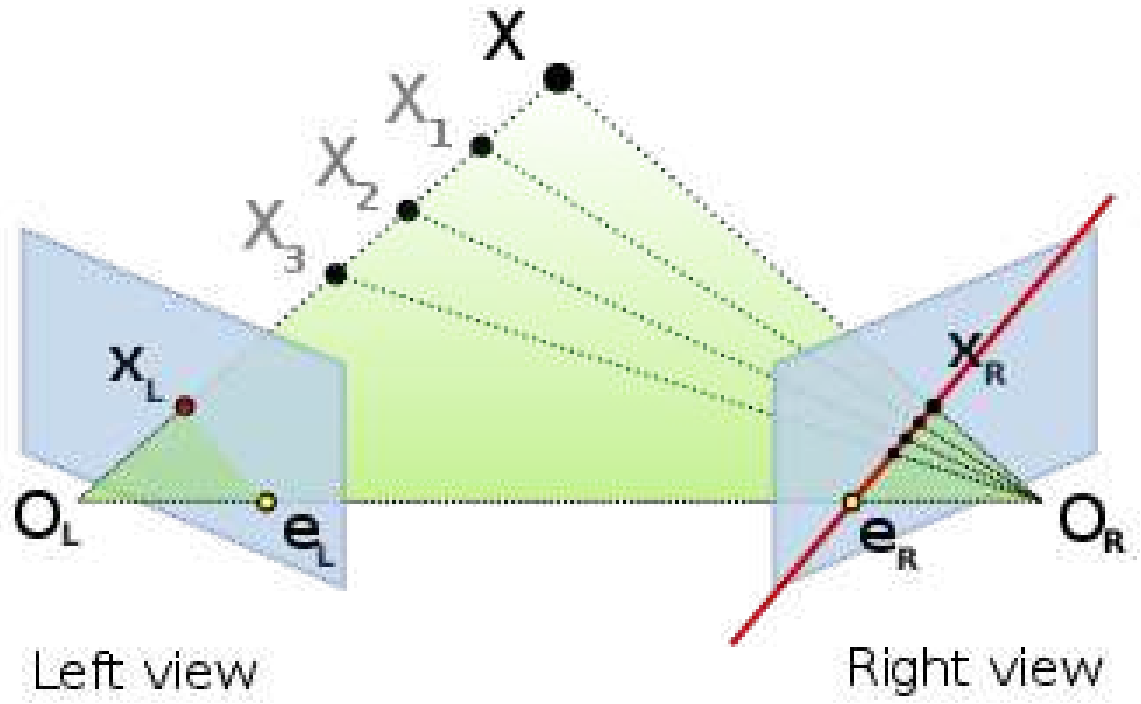


Figure 4: Epipolar Geometry Visualization [3]

of patterns with known dimensions, such as a checkerboard [16]. Both the entire checkerboard dimensions can be calculated as well as each individual square on the board. For best calibration results, multiple images need to be taken with varying orientations. This method reduces the error when calibrating the cameras by creating more variation in the depth of the individual squares. Figure 5 displays sample checkerboard captures for camera calibrations.

2.2.4 Image Rectification

Once the cameras are calibrated and the epipolar lines are drawn onto the set of pixels, the epipolar lines must then be parallel in order to match features along the same epipolar lines. Image rectification rotates each set of pixels such that the epipolar lines are then parallel from one image to the other. An example of the image rectification process, as well as the math behind the scenes, can be found in [17].

Figure 5: Stereo Images from a Left and Right Infrared Camera [4]

Once both sets of pixels are aligned parallel the epipolar lines, a stereo block matching function finds matching features between the epipolar lines and creates a disparity map. The disparity map is a grayscale image that illustrates the depth of features by indicating the white pixels were feature matched and calculated to be closer in the 3D environment than that of the darker pixels. Provided a disparity map with depth indications, the disparity map can be transformed into a 3D point cloud.

2.2.5 Registration

The final step in the stereo vision process is registration. Registration is the process of reprojecting the disparity map from the image rectification stage which results in a series of 3D position vectors x,y,z and then utilizing error minimization methods to place a known model to the sensed points, or point cloud. The final placement of the point cloud is the final relative pose estimation for that pair of images.

One error minimization method for registration is using an ICP algorithm. The algorithm works by matching a known model to the sensed points to minimize the

amount of error between the known model's pose and the sensed point cloud's pose. To calculate these values, each point in the known model finds the closest point in the sensed model. Referring back to Figure 2, for each point in the yellow 3D point cloud, the closest point in the known, red model is found. Next, the translation and rotation is found between each red point and the closest yellow point. For each of these values, the Root Mean Squared Error (RMSE) is calculated. This process is repeated until the RMSE can no longer be reduced. However, the ICP algorithm may not find the optimal solution, only a local optima. Examples of various ICP algorithms can be found in [18, 19, 20, 21].

2.3 Previous and Related Work

There has been many difficult problems, as well as potential solutions, while trying to solve the problem of AAR. Many solutions have been created in an attempt to solve the current problem: pose estimation between the tanker and receiver. Possible solutions include radar, GPS, vision navigation, Electro-Optical (EO) systems, and systems that utilize more than a single approach [22]. While many of these solutions fulfill a portion of AAR's problems, no single solution has completely solved the entire problem. This section further discusses AAR research that has been conducted up to this point and any work related to this specific research.

2.3.1 Stereo Vision

Similar to this research, prior research also utilized stereo vision to tackle AAR problems. Stuart developed an algorithm using stereo vision to calculate relative pose estimation [23]. Paulson attempted to mitigate boom occlusion using stereo vision for the boom aerial refueling method [24]. Similarly, Parsons added a shelled reference model with ICP to more accurately calculate pose estimations using stereo vision [5].

Additionally, Dallman utilized stereo vision to develop physical test flights to test and capture truth data. Dallman also ran virtual experiments to calculate and quantify relative pose estimation accuracy [4]. This research employs Dallman's AAR research methods to compare physical and virtual data.

2.3.2 Truth Flight Data

In March 2019, several flight tests were conducted at Test Pilot School (TPS) located at Edwards Air Force Base (AFB). The flight tests contain numerous measuring components to record associated flight truth data [4]. To begin with, EO and Infrared (IR) cameras were placed under the tanker facing aft toward the receiver. Images from these cameras were used to calculate pose estimations between the tanker and receiver. Next, a series of truth data was recorded. Inertial measurement units (IMU) and Global Positioning System (GPS) units were placed in both aircraft. The IMU units record the orientation of the aircraft while the GPS units allow for Differential GPS (DGPS) computations. The DGPS calculations provide information regarding the $\{x, y, z\}$ distances between the aircraft. Additionally, the indicated refueling point must be known to appropriately calculate relative pose estimations. For this research, the refueling distance measures at 30 meters [25].

2.3.3 Sources of Error

This research depends upon correctly measuring initial data sets to appropriately compute relative pose estimations. However, observed measurements are rarely perfect [26]. A major portion of this research involves correctly accounting for any sources of bias and error involved in the initial measurements. Once the errors are corrected for the physical data, the virtual pose estimations can be appropriately compared to the physical estimations. The sources of error we found to be most important include

camera calibrations, hardware mounting errors, and noisy data points. We discuss each of these in the following sections.

2.3.3.1 Camera Calibrations

To begin with, camera calibrations are requisite for correct relative pose estimations. They contribute heavily to how many points are computed during feature matching, where the points will be located in the 3D domain, and the performance of the pose estimates. Therefore, any inaccuracy in camera calibrations will result in large errors. Provided the flight data in March 2019, there are sets of camera calibrations measured and recorded before each flight. As expected from [26], there will be errors associated with each flight. These errors will result in an incorrect translation or rotation in the final relative pose estimations.

2.3.3.2 Hardware Mounting Errors

Second, one of the biggest sources of error originates from the lever arm offsets between the cameras, IMUs, and GPS units placed on the aircraft. The assumed orientation of the IMUs point directly through the nose of the aircraft. Additionally, a measurement is recorded between the distance of the GPS and IMU units. If either of the two measurements are even slightly incorrect, there will be a translation or rotation error in the pose estimation. These errors must be accounted for to calculate comparisons between the physical and virtual pose estimations.

2.3.3.3 Noisy Data

Noisy data exists as a severe threat to the pose estimation calculations. Noisy data stems from incorrect errors in calibration, sensing, feature matching, and lever arm errors and must be accounted for to obtain better pose estimations. For example, an

object in the images may be perceived as a feature on a receiver and thus registered as an erroneous 3D point in the virtual world. During the registration phase, the outlier points would then be included and add error to the relative pose estimations. The resulting pose will then be skewed to an incorrect orientation or location. Research exists to remove outliers from a set of data [27]; however, the question then becomes, which data points are outliers in terms of AAR research within this specific virtual environment?

2.3.4 Simulation Environments

AAR experiments require many resources, such as time, money, pilots, and aircraft just to name a few. Therefore, The Air Force Institute of Technology (AFIT) relies on virtual experiments more often than physical. Prior to this research, many have focused on creating a simulation environment for general AAR experiments [28, 29]. Other research focused on specific AAR tasks. For example, Shuai An focuses on UAV position holding while refueling [30]. Fezans researches simulating aerial refueling with the probe-and-drogue method [31]. Further, Mammarella simulates using GPS and machine vision for refueling UAVs [32]. Lastly, Seydel attempts to compare simulated imagery and real world imagery for the AAR problem [33]. These simulation environments, however, have not been validated using a comparison with physical data sets and known truth data.

2.3.5 Discussion on Previous Works

Although various research methods exist for the AAR pose estimation problem, this research utilizes stereo vision EO and IR cameras to calculate the pose between the two aircraft. Additionally, research has been done using both the boom and probe-and-drogue aerial refueling methods; however, since the AF uses the boom

method, this research focuses on the boom aerial refueling method. Finally, many researchers utilize various simulation environments to test different aspects of the AAR problem. This research utilizes a 3D virtual world created at AFIT to compare AAR algorithms in both the virtual and physical world by using a set of truth data collected at Edwards AFB.

III. Methodology

Previous Automated Aerial Refueling (AAR) research efforts have attempted to determine how well a specific algorithm can accurately calculate the relative pose estimation between a tanker and receiver aircraft. This current research aims to quantify how well the virtual world can accurately simulate actual AAR approaches from the March 2019 test flights. To appropriately determine the viability of utilizing a virtual world to simulate the physical world, a series of equivalent tests must be conducted. The first step is to gather the control variables that will be constant in all testing. For this AAR research, these variables are a set of truth data that consists of Differential GPS (DGPS) and Inertial measurement units (IMU) measurements and associated timestamps. While these observations are made, a series of cameras on the bottom of the tanker facing aft toward the receiver captures images of the receiver. These images will be used to calculate the relative pose estimation. This research utilizes both Electro-Optical (EO) and Infrared (IR) cameras to take imagery. A virtual world is set up that visually recreates a tanker and receiver via the truth data so that there is a basis to test the accuracy of the final relative pose estimations. The images from the physical flight test are ran through a pipeline, producing a point cloud, which is then compiled through an Iterative Closest Point (ICP) algorithm, essentially calculating a relative pose estimation. This relative pose estimation, created solely through the imagery, is then compared to the original location of the tanker and receiver, resulting in a 6 Degrees of Freedom (DOF) vector quantifying the error between the sensed pose and truth pose.

With the provided physical world calculations, the same exact tests are then executed with images derived purely from the virtual world. The same virtual tanker and receiver equipment can be used since they are control variables, but the images that are used to calculate the relative pose estimations are no longer coming from

the physical world. Instead, a series of virtual cameras are placed on the virtual tanker facing aft toward the receiver. The same tests are executed from above, but with an intermediary step of capturing virtual stereo images. Once the relative pose estimation is calculated, it is compared to the results from relative pose estimation calculated with physical world images. Through statistical analysis, comparisons can aim to explain within what margin of error the virtual world approximates the physical world.

Relative pose estimations between the physical and virtual worlds would not be comparable without first correcting a set of physical world related errors. To start with, the set of errors include camera calibration correction, hardware pose correction, and filtering invalid data points. The virtual world does not have these errors because the virtual world represents a perfect situation. For instance, the virtual world has nearly perfect camera calibrations resulting in nearly all valid data points. Additionally, the truth data is collected in the physical world and not the virtual world. Therefore, the virtual world would have much better camera calibrations, it wouldn't need to account for hardware mounting errors, and no filtering is necessary since there will be negligible points to filter. Thus, only the physical world data need to be adjusted correctly.

First and foremost, camera calibration coefficients designate the position of the point clouds resulting in the integrity of the pose estimations. If camera calibrations are incorrect, the resulting point cloud could have less points, be incorrectly translated, or be incorrectly rotated resulting in an inaccurate relative pose estimation. Correct camera calibrations are the fundamental building blocks to The Air Force Institute of Technology (AFIT)'s AAR relative pose estimation computations. Additionally, errors related to assumed hardware placement and look direction affect calculations. As noted in Section 2.3.3.2 and Figure 24, each hardware unit has a

specific set of assumed positions and rotations. A slight variation in any related hardware may have significant, unfavorable outcomes on pose calculations. Thus, a translation and rotation vector must be added prior to the final pose estimation. Finally, noisy data may also adversely affect relative pose estimation calculations. Noisy data stems from erroneous feature matching during the stereo vision process. A series of steps must be implemented to counter invalid data points. These steps include creating an aircraft composed of ellipsoids that encompass the model aircraft, implementing point inclusion for said model, and filtering points prior to relative pose estimation calculations. Finally, after correcting camera calibrations, adjusting for hardware offsets, and including only valid data points, more accurate relative pose estimations can be calculated.

3.1 The Virtual World

This research utilizes a 3D virtual engine that is capable of simulating AAR interactions. The engine is based off of the AFTRBurner Engine [34]. This 3D virtual engine enables a graphical illustration of AAR simulations [4, 5, 24, 23, 35].

To successfully comprehend the disparity between the physical and virtual worlds during experiments, a complete understanding of the virtual world must be established. The virtual world acts as both a way to visualize relative pose estimation capabilities for physical and virtual imagery and as a way to simulate the act of taking imagery necessary for virtual world calculations. Figures 6 and 7 demonstrate the physical world and virtual world experiments, respectively.

The only difference between the physical and virtual world comparisons lie within the captured imagery. The physical imagery is collected during the flight tests, while the virtual imagery is collected after all the truth data is obtained and replayed within the 3D virtual world. Referring to the physical world process, Figure 6, the images

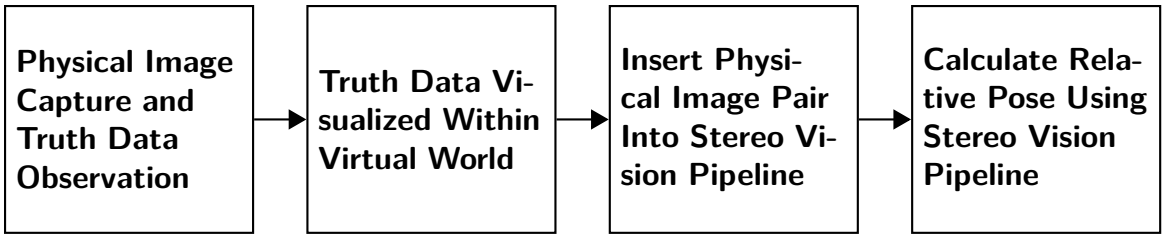


Figure 6: Process for Relative Pose Estimations for Physical World Imagery

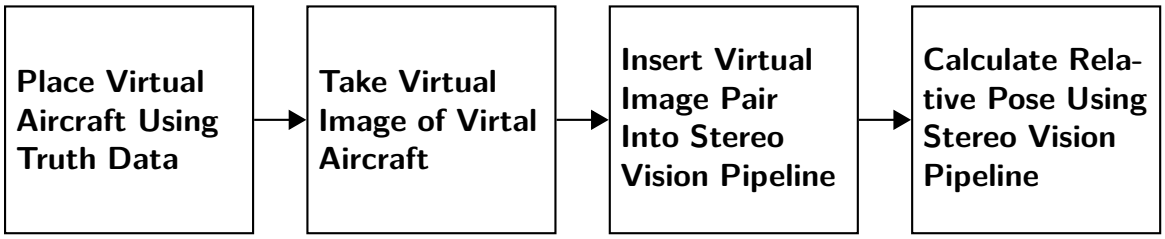


Figure 7: Process for Relative Pose Estimations for Virtual World Imagery

are collected via flight tests. As noted in Figure 6, the truth data is simultaneously collected. Thus, the data collection results in both physical images, DGPS, IMU, and timestamps. They can then be placed within the virtual world to calculate relative pose estimations. At this point, however, an intermediary step can be used to insert the virtual process. Figure 8 demonstrates how both the physical and virtual relative pose estimation processes can run in parallel.

Referring to Figure 8, the red boxes indicate the relative pose estimation process for physical imagery while the blue boxes display the relative pose estimation process for the virtual imagery. As indicated, the processes can be run in parallel. Once the truth data is collected and placed in the virtual world, the physical images can be added. Since the truth data is already present, virtual images can be taken. Relative pose estimation calculations are then made for both the physical and virtual worlds. Specific details relating to the truth data, physical world imagery, virtual world imagery, and how to calculate their respective pose estimations are discussed in the following sections.

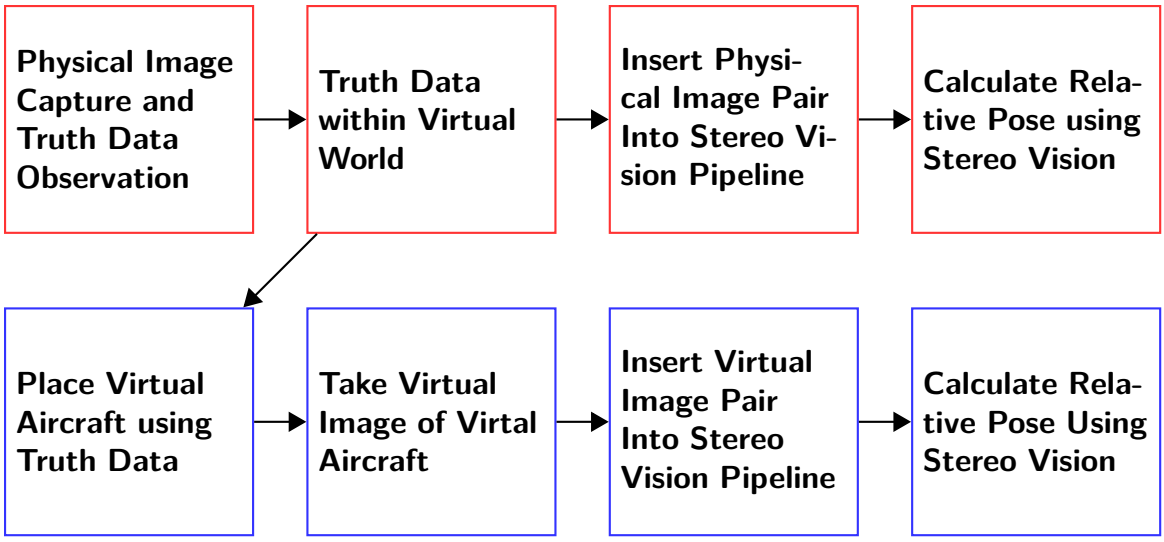


Figure 8: Physical World and Virtual World Relative Pose Estimation Processes Executing in Parallel

3.2 Initial Flight Truth Data

A series of truth data needs to be recorded to allow for equivalent comparisons between the virtual and physical world. A single set of truth data consists of correlated DGPS, IMU, and an associated timestamp. In March 2019, AFIT collected a series of truth data for a set of flights: flights 1, 2, 3.1, 3.2, 3.3, 4, 5.1, 5.2, 5.3, and 6. Flights 3 and 5 break down into a smaller, specific set of flights. Section 3.2.1 further discusses the flights and their specifications. Once the truth data is captured, it is used to place a simulated tanker and receiver in the virtual world. Figure 9 portrays a simulated tanker and receiver placed using a subset of the truth data.

Figure 9 illustrates the basic setup of the truth data placed within the virtual world. First, the virtual tanker's location is determined by the Global Positioning System (GPS) unit embedded on the physical tanker during flight tests. Similarly, the virtual tanker orientation is sensed by the physical tanker's IMU. The virtual tanker is indicated by the aircraft below the yellow text "Tanker". Second, receiver's location and orientation are set using a second set of GPS and IMU units from the

Figure 9: Truth Data Sample

flight tests. The virtual receiver is indicated by the yellow text "Receiver". Then, the DGPS measurements are calculated by subtracting the GPS units of the receiver from those of the tanker. The DGPS calculation is then compared to the relative pose estimations calculated as discussed below. Additionally, Figure 9 illustrates the physical and virtual imagery; the physical world in the top corners and the virtual in the bottom corners. Further discussions of those images are provided in Sections 3.2.1 and 3.7.

3.2.1 Physical World Data

After receiving and inputting truth data, imagery used for stereo vision calculations must be captured. To collect said images during flight tests, a set of cameras

were placed under the tanker facing aft toward the receiver aircraft. Figure 10 shows the setup of cameras that were placed on the tanker. Instead of just a left and right camera, there are four cameras total. There is a left and right EO camera as well as a left and right IR camera. Details on the images related to these cameras are further discussed in Sections 3.2.2 and 3.2.3.

Capturing physical world data for each flight is impacted by the cameras' exposure settings. Various aperture settings allow for different lighting situations to be tested. For example, at a specified time during flight 3, the camera's exposure setting was altered. Lighting that deviates within the captured imagery could provide differing results because of the feature matching technique utilized by the stereo vision process.

For AFIT's AAR system to be as robust as possible, various background environments were also captured during flight tests. For example, backgrounds of land, water, and clouds were taken throughout testing. Figure 11 displays a set of timestamped images that show different lighting settings and background environments.

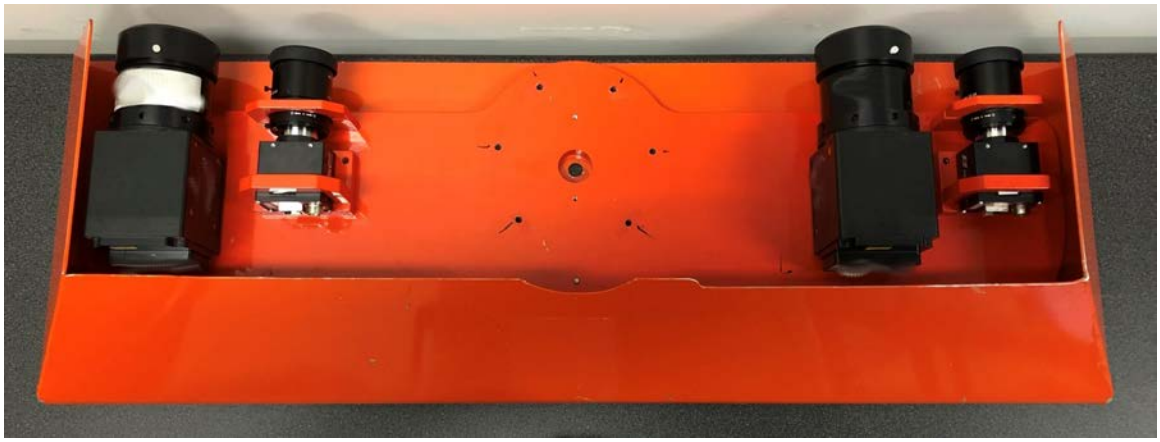


Figure 10: Camera Setup Used for Flight Tests [4]

Figure 10 illustrates the camera setup. The larger cameras, the first and third, are the IR cameras. The smaller cameras, the second and fourth, are the EO cameras. This bracket is mounted under the backside of the tanker facing aft toward the receiver.

Figure 11: Pairs of Flight Images Obtained at Edwards Air Force Base (AFB)

3.2.2 Electro-Optical Cameras

The EO cameras produce images similar to that of the human eye, a normal color image. Before an EO image is taken, its exposure is adjusted according to the

surrounding environmental lighting. This can affect the colors of the image. With several different flights, each with multiple photos, this produces various lighting and color results. Additionally, the exposure settings of the cameras were altered mid-flight to create photos with varying settings. All EO images are recorded with a 1280x960 resolution. Each EO image is recorded with a bitmap (bmp) format. Recording as a bitmap, instead of a traditional Portable Network Graphic (PNG) or other traditional graphical formats, allows for lossless image storage and replay [36]. The better quality images allow for a deterministic replay and pose estimation. The top four images in Figure 11 illustrate EO imagery at various exposure settings.

3.2.3 Infrared Cameras

The IR cameras selected for the flight tests were long-wave infrared (long-wave infrared) cameras. The IR images produced by these type of cameras are a heat map that is based on temperature rather than an image based on visible light. These heat maps have two useful properties. First, the IR images permit a second viewpoint for which to compare pose estimation calculations during daylight hours. Second, since IR imagery does not rely on visible light, it permits image capture in the dark. This has potential for nighttime aerial refueling [4]. The IR images were recorded in a Portable Gray Map Image (pgm) format because the pgm format is lossless. The bottom pair of images in Figure 11 display a sample of IR images collected during flight tests.

3.3 Initial Relative Pose Estimations

Following the capturing of images during the flight tests, the imagery can then be fed to AFIT's AAR algorithm to produce a point cloud. A point cloud is obtained by the reprojection of the disparity map which yields a 3D point cloud. The disparity

map comes from the initial captured images, the rectification of those images, which are processed to create the disparity map. From a point cloud, a registration algorithm known as ICP minimizes the error between the sensed points and a known model. Referencing Figure 12, the sensed points are the yellow points and the known model is comprised of the red points. The final position and orientation of the red model is known as the relative pose estimation. Figure 12 illustrates the calculated relative pose estimations from the sensed pose (red) to the truth pose (gray). Notice, the majority of the red model overlaps the truth model, resulting in little error between the truth data and the calculated pose estimation.

Figure 12: Calculated Relative Pose Estimations

Currently, however, the relative pose estimations are not returning sufficient quality results due to various errors. These errors stem from poor camera calibrations,

hardware mounting errors during the flight tests, and erroneous points that cause the sensed model to rotate and translate incorrectly. Figure 13 demonstrates the current results of the pose estimations. Compared to Figure 12, Figure 13 does not perform near as well. The red sensed model is much further away from the gray truth model.

Figure 13: A Set of Relative Pose Estimations Using Original Calibrations Without Correcting for Errors

It is clear, when comparing the sensed, red point cloud pose estimation, and the truth gray aircraft truth pose, the algorithm does not seem to perform as well as required. While this may look like an algorithm issue, the problem seems to arise from other various errors that need to be corrected. To test this theory, a series of virtual tests were ran using perfect camera calibrations, a perfect virtual hardware scheme, and a series of virtual images taken within the virtual world. These results can be referenced in Figure 14 from experiments run by [5]. Since the algorithm worked correctly with a set of perfect data near the refueling envelope, the error must be located within the physical flight data. Therefore, to correctly be able to compare physical relative pose estimation to virtual relative pose estimation, the physical estimations need to be corrected as best as possible by minimizing error. To correctly account for these errors, 1) each camera calibration needs to be altered to obtain the maximum amount of points at the correct location from the camera's positions, 2) hardware rotation and translation needs to be corrected, and 3) a method to filter

out noisy data. Once these issues are fixed, the relative pose estimations for physical imagery can be appropriately compared to the relative pose estimations calculated in the virtual world.

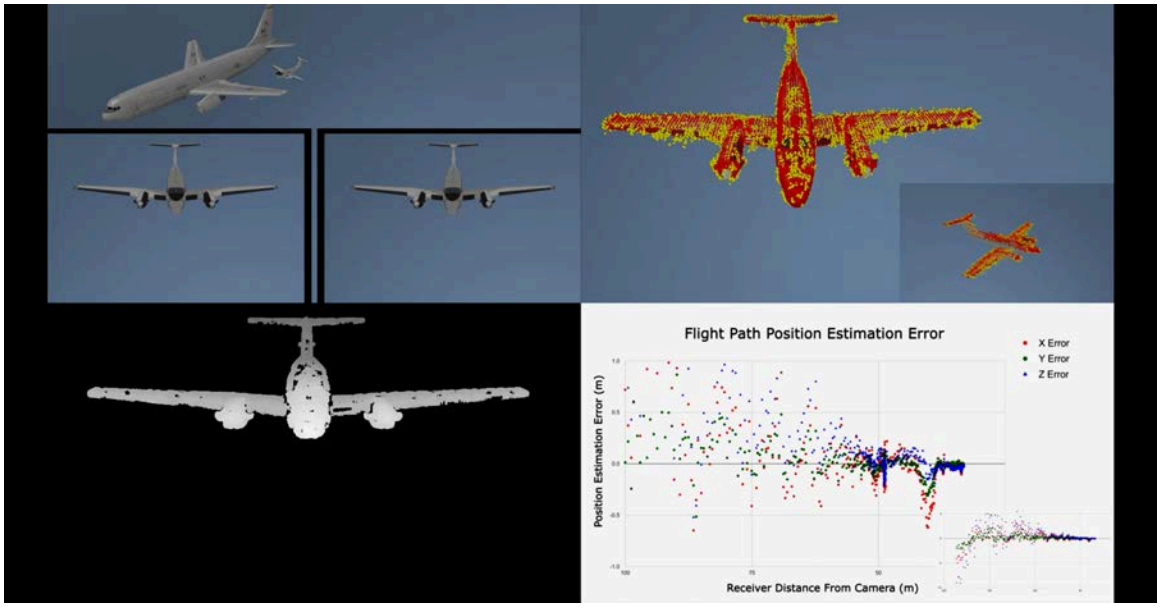


Figure 14: An Experiment Within the Virtual World Using a Virtual Tanker, Virtual Receiver, and Near-Perfect Calibrations. Additionally, a Graph to Illustrate the Amount of Error During an Approach Towards the Tanker [5]

3.4 Correct Calibrations

Correct camera calibrations are paramount when representing a pair of 2D images within a 3D domain. As noted in 2.3.3.1, camera calibrations are represented through a set of matrices. Each of the coefficients in the matrices affect either the intrinsic, extrinsic, or distortion factors. This research focuses on improving only the intrinsic parameters of the camera to better correct relative pose estimations because the extrinsic variables are not associated with the center of the camera's content, but on the camera's coordinate system (the center of the camera) [14]. Additionally, the distortion variables focus on content closer to the edge of the cameras and this

research expects the aircraft to be relatively in the center of the camera. For this research, intrinsic camera calibration can be defined as:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

The intrinsic properties of a camera are combined from four coefficients: the focal length for the X and Y axis, f_x and f_y, and the optical center point for the X and Y axis, c_x and c_y. The values of the focal length, f_x and f_y, alter the placement of the individual points while the values of the optical center, c_x and c_y, alter the placement (left/right/up/down) of the point cloud generated during the stereo vision process. In other words, changing the focal length changes how well the features are matched during the stereo vision process resulting in the number of points in the point cloud and the optical center changes the location of the entire point cloud relative to the camera. Figure 15 illustrates the translation of the aircraft by modification of the optical center camera calibrations. The best camera calibrations will provide the most points and be at the correct center of the camera. The focal length affects how far away an object is in relation the camera [13]. Editing the focal length alters the distance at which the feature points are calculated. Resulting in an object that is closer or further from before. But, the problem of correcting calibrations for stereo vision requires two sets of camera calibrations. Consequently, instead of having to alter and correct four coefficients, eight intertwined values must be improved. Due to the association between all eight calibrations, there exists more than one set of calibrations that could correctly satisfy the problem. Initial relative pose estimations are illustrated in Figure 17 using the intrinsic calibration values displayed in Figure

16. This research utilizes two procedures to identify local, optimal calibrations to better correct camera calibration constraints.

(a) Initial Optical Center Calibrations Illustrated Initial Sensed Red Model Placement

(b) Modified Optical Center Calibrations Illustrating Altered Sensed Red Model Placement

$$M1: \begin{bmatrix} 953.834 & 0.000 & 522.624 \\ 0.000 & 949.215 & 380.459 \\ 0.00 & 0.000 & 1.000 \end{bmatrix}$$

$$M2: \begin{bmatrix} 953.818 & 0.000 & 527.134 \\ 0.000 & 948.694 & 379.361 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$$

(c) Optical Center Calibrations for Figure 15a

$$M1: \begin{bmatrix} 953.834 & 0.000 & 582.624 \\ 0.000 & 949.215 & 380.459 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$$

$$M2: \begin{bmatrix} 953.818 & 0.000 & 587.134 \\ 0.000 & 948.694 & 379.361 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$$

(d) Optical Center Calibrations for Figure 15b

Figure 15: Modifying Optical Center Camera Calibrations Translates (left/right/up/down) the Aircraft

Initial efforts to correct camera calibrations employed an intelligent brute force method. As shown in Figure 18, this strategy utilized three virtual cameras placed at different angles. The goal of this methodology aimed to use human observations to determine the optimal camera calibrations at a specific image pair. For the experiment, each camera was orientated to face toward the receiver aircraft and a single image pair was used for relative pose estimation for this entire experiment. If different image pairs were tested, the calibrations would not be comparable. Only the focal

$$K_1 = \begin{bmatrix} 1231.46451339497708 & 0 & 663.98551411169683 \\ 0 & 1228.24807722601008 & 522.65405740523227 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 1222.36726601806345 & 0 & 713.78104221064871 \\ 0 & 1220.00586008914524 & 518.71722455329063 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 16: Original Calibrations Measured at the Flight Test at Edwards AFB for Flight 3.3

Figure 17: Sample Relative Pose Estimation Result from Calibrations in Figure 16

lengths, f_x and f_y, were adjusted. The modification of the optical center point came later in testing.

The initial calibrations for each flight were recorded prior to the flight. The initial

Figure 18: Intrinsic Values: [1231, 1232, 1230, 1232]

calibrations in Figure 16 related to flight 3.3. For simplicity purposes, the initial calibration values were rounded to $[1220, 1220, 1220, 1220]$. A set of calibration values must be selected to test against. To decrease the amount of testing, the beginning value was selected at ± 25 for each f_x and f_y. The initial and final calibrations for this experiment can be identified in Figure 19.

However, this strategy of iterating through the focal lengths has a downfall; the time complexity of this algorithm is $O(n^4)$. Generously speaking, each image would take around 0.1secs to compute the respective relative pose estimation. Thus, if the algorithm iterates from $1245 - 1195 = 50$, then $50^4 = 6250000$ computations will be made with a final time of 7.23 days. After all of these images were captured, a human observer now has to inspect 6,250,000 images in an attempt to find the optimal camera calibrations. Each image consumes about 3.5 Megabyte (MB)s on average. Therefore, if all the 6,250,000 were saved for inspection, that would require right around 22 Terabyte (TB)s. For the above reasons, an improvement to the algorithm is required.

$$K_1 = \begin{bmatrix} 1195 & 0 & 665.31230214507218 \\ 0 & 1195 & 499.18293270488499 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 1195 & 0 & 712.03827622918266 \\ 0 & 1195 & 496.74734662364131 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K_1 = \begin{bmatrix} 1244 & 0 & 665.31230214507218 \\ 0 & 1244 & 499.18293270488499 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 1244 & 0 & 712.03827622918266 \\ 0 & 1244 & 496.74734662364131 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 19: First and Final Calibrations for Intrinsic Value Improvement Testing From Top to Bottom

To decrease the amount of calibrations the brute force algorithm searched through, a heuristic was applied to the algorithm. Due to the synergistic nature of the intrinsic properties of camera calibrations, the values could only differ within set values before the relative pose estimations worsened. Therefore, to maintain minimal spacing between the intrinsic values, the algorithm would limit the width difference to 5. For example, the first batch would be as signified in Figure 20.

$$[1195, 1195, 1195, 1195], [1195, 1195, 1195, 1196], [1195, 1195, 1195, 1197],$$
$$[1195, 1195, 1195, 1198], [1195, 1195, 1195, 1199], [1195, 1195, 1196, 1195],$$
$$\ldots$$
$$[1195, 1195, 1199, 1199], [1195, 1196, 1195, 1195], [1195, 1196, 1195, 1196],$$
$$\ldots$$
$$[1199, 1199, 1199, 1197], [1199, 1199, 1199, 1198], [1199, 1199, 1199, 1199]$$

Figure 20: The Initial Batch of Calibrations for Intrinsic Value Experimentation

After the initial batch of testing, the same sequence of batch tests will happen for calibrations measuring up to [1245, 1245, 1245, 1245] from the original [1195, 1195, 1195, 1195]. A time and spacing calculation can determine how well the heuristic improves the algorithm. Instead of 6,250,000 iterations, there will be $5^4 \times \frac{1245-1195}{5} = 6250$ iterations. The 5^4 portion comes from the number of iterations per batch and the $\frac{1245-1195}{5}$ stems from the number of batches ran. That is a 1000× decrease in time. In terms of spacing, using 3.5MB as the average image size, that computes to $3.5 \times 6250 = 0.022$TB. The spacing also equates to about a 1000× improvement. Finally, a series of pose estimations can be computed to decide the best calibrations within the specified domains.

(a) Intrinsic Values: [1231, 1232, 1230, 1232]

(b) Intrinsic Values: [1232, 1230, 1230, 1232]

(c) Intrinsic Values: [1232, 1231, 1230, 1232]

(d) Intrinsic Values: [1234, 1234, 1230, 1232]

Figure 21: Four Different Camera Calibrations Using a 3-Camera Setup to Compare Each Camera Calibration

Figure 21 illustrates some of the best and worst choices using the brute force method. Notice how Figure 21d sensed red model lies behind the nose of the gray

truth model. Figure 21a does not have as many points on the tail compared to that of the others. And, between Figures 21b and 21c, Figure 21c looks slightly less skewed to the left. Thus, Figure 21c and the respective intrinsic camera calibrations became the new camera calibrations for flight 3.3.

The intelligent brute force method only modified the focal lengths. The goal was to increase the number of points for a point cloud to get a better reading and correctly position the point cloud to reduce error. However, the optical center point, c_x and c_y, must be altered as well to better correct the (left/right/up/down) position of a point cloud and result in a more accurate relative pose estimation. As recalled from above, though, each of the intrinsic coefficients are related to one another. Editing both sets of the focal lengths and both sets of the center points, though, requires repetitive adjustments across all eight variables, which, considering the ± 25 range, also requires a great deal of time. Assuming the current set of calibrations are not optimal, adding in both sets of center point values, two from each camera, would exponentially increase the time to discover a better set of calibrations. In turn, a guess-and-check method is utilized to further discover sets of more correct calibrations.

The guess-and-check method manually modifies all eight variables, consisting of the cameras' focal lengths and center points, repetitively in an attempt to find the most correct camera calibrations. To manually find an optimal set of calibrations, a table recorded the set of camera calibrations and the associative relative pose estimation calculation. Instead of following the previous calibration testing of using a single image pair for all testing, a series of image pairs were used. The Mean Absolute Error (MAE) of the set of relative pose estimations was calculated to determine if a set of calibrations were considered better than another set. To begin the testing, the optimal calibrations found in the intelligent brute force method are selected for the start point. For this method of camera calibration correction, flight 5.2 will be

used, instead of the 3.3 mentioned in the last method. But, the amount of time spent during guess-and-check is arbitrary for each flight. Thus, this method will need to be applied to each flight individually and will take varying amounts of effort. The initial calibrations are indicated in Figure 22.

$$3.3_1 = \begin{bmatrix} 1219.69122843467449 & 0 & 665.31230214507218 \\ 0 & 1220.61664458947257 & 499.18293270488499 \\ 0 & 0 & 1 \end{bmatrix}$$

$$3.3_2 = \begin{bmatrix} 1217.90259830721971 & 0 & 712.03827622918266 \\ 0 & 1220.07619854918539 & 496.74734662364131 \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 22: Initial Calibrations for Guess-and-Check Camera Calibration Testing for Flight 5.2

Table 1 references camera calibrations corrections for flight 5.2 instead of the aforementioned flight 3.3. The camera calibrations chosen for flight 3.3 referred to in Figure 21c will not illustrate the most amount of change since flight 3.3 was decently calibrated during the flight tests. To better illustrate the magnitude of altering camera calibrations for the guess-and-check method, flight 5.2 has been selected. First, the initial test calibrations will be tested against. Then, gradual, random checks will determine if the calibrations are going in the correct direction. To determine what is considered the "right direction", the error associated with each axis must be reduced. In Table 1, the errors and their associated spread values can be referred to in the last six columns. The test number can be identified in column 1. The camera calibrations associated to each test can be referenced in column 2. The final observed calibrations that best fit flight 5.2 can be found in the last row of the table.

Table 1 displays the steps from the initial calibrations in Figure 22 to the final calibrations observed to reduce the amount of error. To determine the least amount of

Table 1: Camera Calibrations Flight 5.2 Pulses 5132 - 5714

Test #	Calibration Coefficients	XMAE	XStdDev	YMAE	YStdDev	ZMAE	ZStdDev	Error Magnitude
1 (Initial)	1232, 1231, 1230, 1232, 665, 449, 712, 496	0.5166	0.3837	1.6165	0.42295	0.4504	0.2786	1.7557926
2	1232, 1231, 1230, 1232, 650, 449, 697, 496	0.5231	0.3542	1.9982	0.4309	0.4603	0.2712	2.1162024
3	1300, 1231, 1300, 1232, 650, 449, 697, 496	1.4868	0.4667	2.0155	0.4133	0.2716	0.2327	2.5192421
4	1300, 1180, 1300, 1180, 650, 449, 697, 496	1.5103	0.4934	2.0097	0.4174	0.2617	0.2161	2.5275258
5	1310, 1200, 1310, 1200, 650, 449, 698, 496	0.5603	0.2884	1.8359	0.4088	0.6139	0.2646	2.0152761
6	1310, 1200, 1310, 1190, 650, 449, 698, 498	0.7746	0.5229	1.8589	0.4343	0.5694	0.2674	2.0927806
7	1232, 1231, 1230, 1232, 665, 496, 712, 496	0.5234	0.4676	1.6169	0.4427	0.3562	0.2623	1.7364307
8	1280, 1231, 1280, 1232, 665, 496, 712, 496	0.6733	0.5378	1.603	0.4614	0.2742	0.2148	1.7601498
9	1280, 1231, 1280, 1225, 665, 496, 712, 496	0.6478	0.5446	1.5645	0.4614	0.2677	0.2087	1.7143419

error, the MAE was calculated for each relative pose. Following the error calculation, the translation error was recorded for each axis: x, y, and z. Because the spread of the error is pertinent to help determine the precision of the MAE values, the standard deviation (StdDev) was also calculated for each axis. The specific values in order of f_{x1}, f_{y1}, f_{x2}, f_{y2}, c_{x1}, c_{y1}, c_{x2}, and c_{y2} are recorded in the "Intrinsic Calibration Coefficients" column. Following the intrinsic values, the final observed calibrations that best fit flight 5.2 for the specified pulses can be found in the last row. The associated relative pose errors for each axis can be observed in the following columns on the final row. The final calibrations for flights 5.2 can be applied to reduce the errors for future testing. Figure 23 shows a relative pose estimation after the corrections made for camera calibrations.

Figure 23: Relative Pose Estimation for Flight 3.3 After Camera Calibration Adjustment

Figure 23 depicts the improvements for flights 3.3 and 5.2 from their respective camera calibration improvement methods. While these flights have been corrected and their data points can be utilized for comparisons with the virtual world, other data sets associated with physical flight tests must be corrected as well. Furthermore, there is still too much error between the sensed red model and the gray truth model. Additional methods of error reduction are discussed in Sections 3.5 and 3.6 and Figures 27 and 37. Additional results of other physical data sets after camera calibration correction can be referred to in Appendix A.

3.5 Correct Hardware Translation and Rotation Errors

Following the correction of camera calibrations, another set of errors must be appropriately dealt with. As noted in Section 2.3.3.2, underlying assumptions for relative pose calculation stem from the DGPS and IMU placements and orientations. These hardware assumptions prove unreliable and result in a significant amount of error. Surprisingly, even minor errors related to the hardware can result in significantly inaccurate relative pose estimations. Figure 24 portrays the assumed positions of the hardware units. Thus, if the hardware is incorrectly positioned, a set of vectors must be calculated to correct for the translation and rotations assumption flaws.

Figure 24 displays the assumed positions and orientations of the physical hardware during the flight tests. First, the GPS units obtains position data from the satellite. Next, the IMU records the orientation data. The IMU hardware has an assumed orientation facing directly through the nose. Thus, any slight initial orientation error will result in a relative orientation error. Similarly, the distance between the GPS units and the IMU units are recorded inside the aircraft prior to flight tests. If the distance is recorded wrong, there will be a translation error for every pose estimate as well. Each of the variables noted above are then used to calculate relative distance

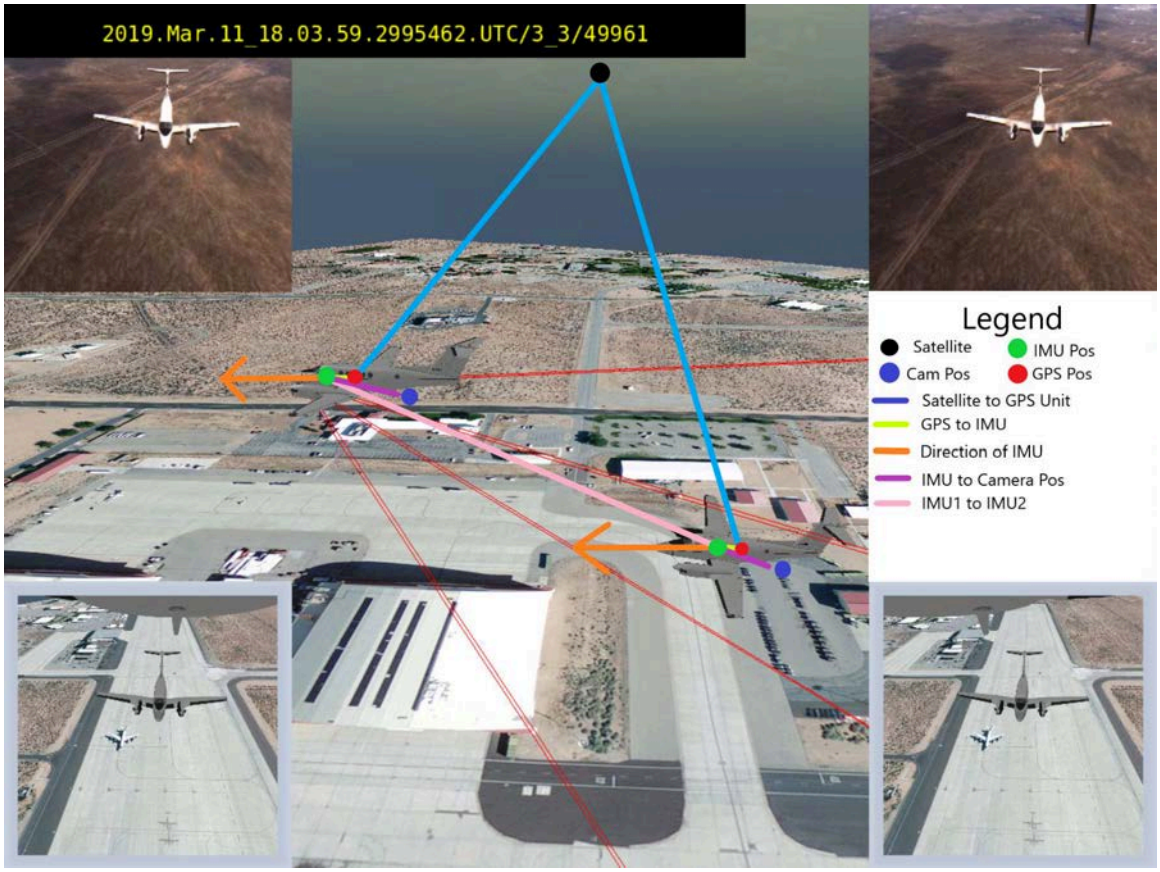


Figure 24: Figure Illustrating the Assumed Positions and Orientations of the DGPS and IMU Units

and orientation between the tanker and receiver. Thus, even if the measurements are slightly off, the final relative pose estimation can be significantly off. Figures 23 and 27 illustrate the difference between not accounting for hardware mounting errors and then correcting for them.

As discussed in [6], Anderson provides a solution to "visually validate the correctness of the truth data during test flights" of "error in the camera's parameters such as the position and rotation," [6, p. 6-7]. To correct for the hardware mounting errors, Anderson first extracts an image at a specific pulse taken from a physical flight test and inserts it into the virtual world on a large quad behind the tanker facing the virtual cameras. The gray, truth aircraft is then placed by using the truth data

from that pulse. Figure 25 illustrates the image at that pulse placed behind the gray truth model. The hardware mounting error is indicated by the difference of the white aircraft in the physical imagery and the truth model. To account for the errors, the cameras are then rotated and translated such that truth model overlays the aircraft in the physical image. Figure 26 depicts the final pose of the cameras after aligning the aircraft. The gray model now better aligns with the white aircraft in the images.

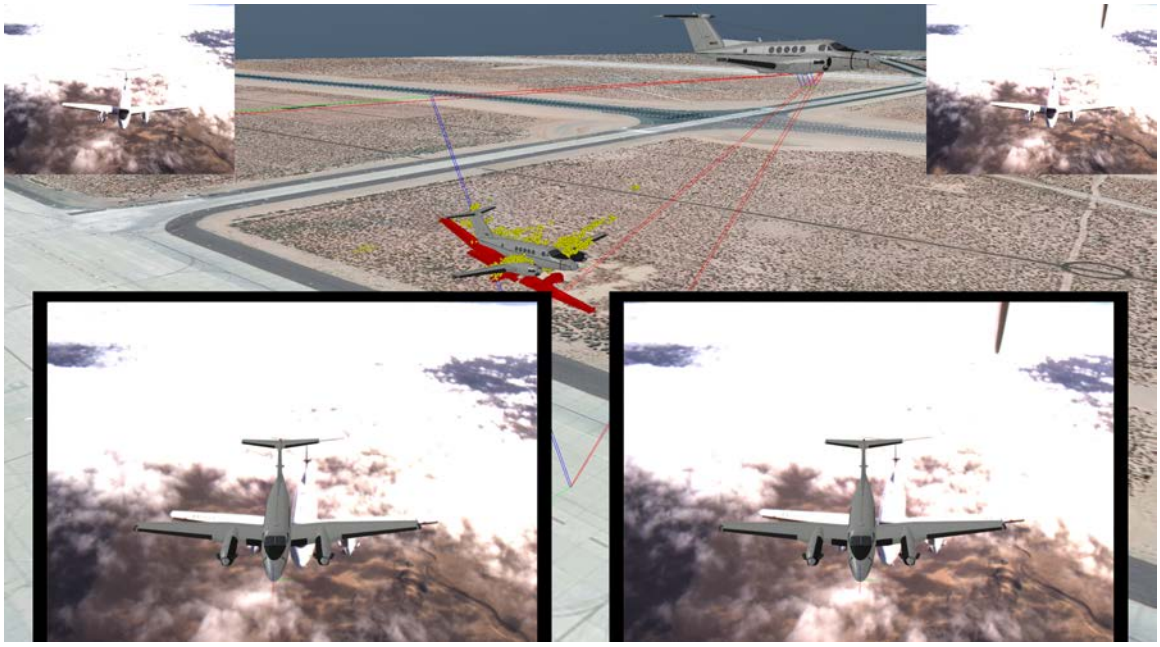


Figure 25: Camera Alignment Before Pose Movement

Although adjusting the camera alignment eliminates the hardware mounting errors, the actual calculations for each error is not calculated. The individual errors are not accounted for, only a general solution to account for all of the associated errors.

While Figure **??** illustrates the setup to calculate the hardware mounting errors, Figure 27 portray a relative pose estimation after the adjustment of camera calibrations and the correction for rotation and translation from the hardware mounting errors. Comparing to Figure 23, the relative pose estimation is significantly improved. Finally, most of the errors are accounted for. However, there are still outlier points

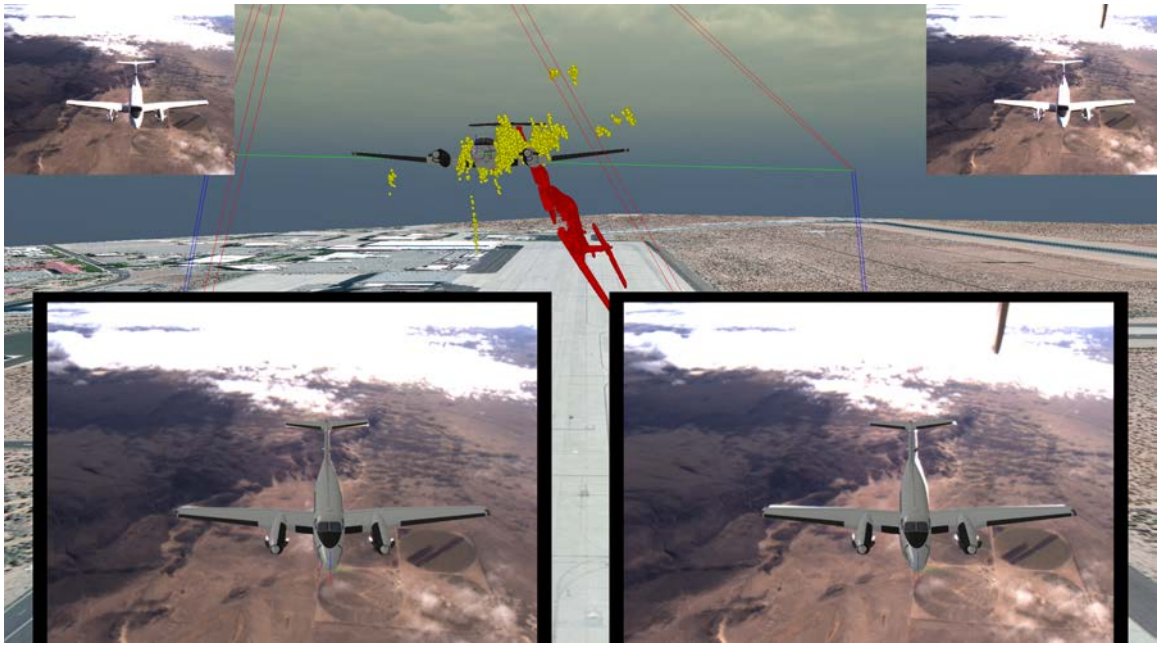


Figure 26: Camera Alignment After Pose Movement

Figure 27: A Sample Relative Pose Estimation for Flight 3.3 Corrected for Camera Calibrations and Initial Hardware Errors

from the feature matching during the stereo vision process that needs correction. Section 3.6 removes the outlier points to finalize the stability and correction of the physical flight data.

3.6 Filter Points

Lastly, relating to the correction of physical imagery associated errors, there exists outlier points in the point cloud that should not exist. These points can come from bad calibrations or incorrect feature matching during the stereo vision process. Nonetheless, noisy data points hinder relative pose estimation calculations. During the registration process of stereo vision, the red, sensed model is matched to the yellow points. Since, the outlier points are included in the current list, the sensed model is incorrectly placed, as seen in Figure 28. Figure 28 illustrates the sensed model not exactly matching the yellow point cloud. Notice, the red points are not aligned with the yellow points at the wings tips or the tail. The invalid points further behind the model are circled in blue. The outlined points are much further back than the model and come from incorrect feature matching as described in Section 2.2.4. These outlined, outlier points reduce the accuracy of relative pose estimations and must be filtered.

Initial research efforts into identifying relevant data points started with Principal Component Analysis (PCA). PCA is able to identify correlations between data points, condense an n-dimensional data into a 2D illustration, and identify variation and patterns among data sets [37]. To accomplish the transformation to the 2D domain, PCA calculates the principal component for each axis. In any dimension, the first principal component consists of the line of best fit for the data set that goes through the origin. For each subsequent principal component, the principal component is the line of best fit that goes through the origin and is perpendicular to all of the prior

Figure 28: Relative Pose Estimation With Indicated Outliers

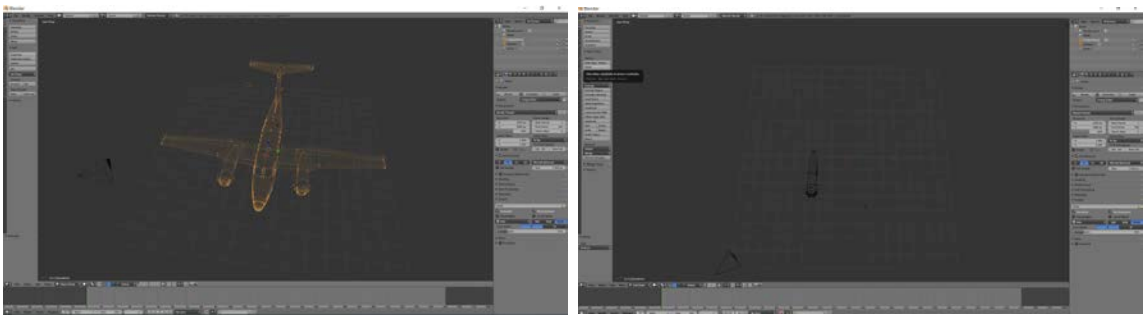
principal components. For example, a 3D data set would have three (3) principal components. Once all the principal components are figured out, you can use the eigenvalues, the sum of squares of the distance to determine the proportion of variation that each principal component accounts for [38]. Next, the top two (2) principal components that account for the largest proportions will be the axes for the 2D graph illustration. The data sets are now projected onto the axes of the two principal components mentioned previously. To correctly find the location of each data point, for each point on the X axis, and for each point on the Y axis, a corresponding point on the graph can be located that conjoins the two. For example, the points $(4, 0), (0, 5)$ would correspond to $(4, 5)$. The corresponding points result in a graph that illustrates the n-dimesnional set of data into a 2D plot that explains variation

and patterns between data sets. In this research, the initial goal aimed to use PCA to find patterns between the data, determine outliers from the patterns, and remove data that does not fall within a certain boundary. To begin with, the axes of the 2D graph would be created the fuselage and wings of the aircraft. From there, a series of checks could determine if each point fell within a certain distance of the axes. If the individual point was within the distance, it would be included for the relative pose estimation, otherwise, it would not be included. However, PCA did not allow for points that would be on the other parts of the aircraft: tail, engines, and vertical stabilizer. Thus, PCA was not viable for filtering points in this AAR domain.

The solution idea takes advantage of the notion that the points should be near the aircraft. Thus, the idea to create an ellipsoidal aircraft that could check point inclusion arose. This methodology involves restraining a point cloud to a specific bounding box similar to that of the original aircraft; however, instead of a bounding box, a bounding aircraft. Therefore, the ellipsoidal bounding aircraft could determine if specific points were within a region or not. If an individual point was in the region of the aircraft, it would be included in the relative pose estimations, otherwise, it would not. This realistic, feasible method utilizes the same 3D virtual engine used for the pose estimations. A series of ellipsoids are be combined to form an aircraft and a single point will be tested against the array of ellipsoids. The only goal of the ellipsoidal aircraft aims to determine point inclusion for each point in a sensed point cloud and include only valid data points for the relative pose estimation calculations.

Prior to the ellipsoidal filtering model, initial efforts constructed a set of virtual cylinders. To determine the size of each cylinder, an example point cloud was placed in Blender, an open source modeling software package [39]. Then, for each component of the point cloud, the fuselage, wings, tail, vertical stabilizer, and engines, selected points were removed to create a point cloud for each part of the aircraft. Figure 29

illustrates the initial point cloud inside Blender as well as the removal of points to only simulate the left engine of the point cloud; however, these steps were iterated for each fragment of the aircraft. From there, each respective portioned point cloud was placed inside the virtual world. Next, a series of virtual cylinders were created to imitate the different point clouds. Separating each part into their own point cloud and cylinder allowed for a more precise cylinder creation. Then, the cylinders were overlaid on their respective point cloud and changed in size to more precisely imitate the size. Once the cylinders were overlaying their respective point clouds, the cylinders were combined to create an aircraft replica. Figure 30 displays the cylinder creation of the initial fuselage and wings cylinder. Figure 31 illustrates the overall cylindrical aircraft replica after each cylinder piece was combined.



(a) Entire Aircraft Point Cloud Within Blender **(b) Engine Point Cloud Selected from Entire Aircraft Within Blender**

Figure 29: Left: Point Cloud Placed in Blender. Right: Points Removed From Point Cloud to Simulate a Specific Portion of an Aircraft

Figure 30: Front and Side View of the Initial Fuselage Cylinder

As a quick overview, the cylindrical point inclusion equation can be broken down into two components which results in a boolean stating if the point lies within the bounds of cylinder. First, the equation will decide if a point, p, resides within the length, l, of the cylinder, Cyl. To calculate if the point resides within the length, a check determines if the point is less than half the distance of the cylinder from the center point, CP, of the cylinder, $p >= \vec{CP} - L/2$ & $p <= \vec{CP} + L/2$. Secondly, if the point resides within the length of the cylinder, a final check determines if the points resides within the radius, R, of the cylinder. To calculate if the point lies within the radius, a vector projection from the endpoint to the point in question takes place. At this point, if the subtraction between the vector projection and the original point is less than the radius, then the point in question resides within the cylinder. Figure 31 encapsulates an equation for cylindrical point inclusion. The entire equation for cylindrical point inclusion can be found in Appendix B.

$$Cyl \in p = ((p <= \vec{CP} - L/2) \ \& \ (p >= \vec{CP} + L/2)) \ and$$
$$|\vec{point} - \vec{pointOnCyl}| \ <= \ R \ where$$
$$\vec{pointOnCyl} = vectorProject((\vec{CP} + L/2) - (\vec{CP} - L/2), \ \vec{p} - (\vec{CP} - L/2))$$

Figure 31: A Brief Equation Exemplifying Cylindrical Point Inclusion

Figure 32: Cylindrical Airplane Replica Used to Remove Noisy Data

While the cylinders mostly replicate the appearance of an aircraft, cylinders are not the most appropriate shape to correctly reduce error. For example, the wings

and the tail are not cylinders; the wings and tail better imitate the shape of an ellipsoid. The most appropriate shape to represent an aircraft would be ellipsoids. To convert the cylinders to ellipsoids, a simple radius modification of the cylinders would return an ellipsoid. Figure 33 illustrates the changes to an ellipsoid from the original cylinders displayed in Figure 32.

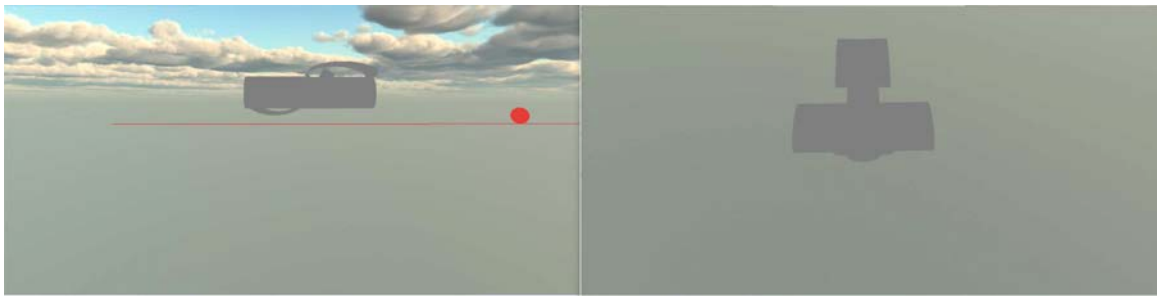


Figure 33: Ellipsoidal Airplane Replica Used to Remove Noisy Data

The ellipsoids ensure a more exact replica of an actual aircraft compared to the cylinders. Figure 32 represents all of the extra space that the cylinders encompass compared to the ellipsoid. Ellipsoids allow each cylinder to be compressed to better replicate their respective parts, especially the wings and tail. The ellipsoids, however, require a different calculation for point inclusion compared to a cylinder. The different equation stems from the ellipsoid having two different radii instead of a single radius. To account for the two radii, a final check to ensure the point in question, p, is within the radii of the ellipsoid is required instead of a check to determine if the point is within the radius of the cylinder. To determine if the point is within the radii of an ellipsoid, the equations in Figure 34 must be satisfied. The elliptical equation, noted as $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$, defines a 2D ellipse where any point inside or on the ellipse will be less than or equal to 1. To apply this formula within a 3D domain, the test point must be projected onto the longest axis of the cylinder, resulting in p_x. Next, the ellipse with p_x will be the ellipse to run the test point against. If the test point is within the ellipse, as well as within the length portion of the ellipsoid, the test point is within

the ellipsoid. The entire equation for ellipsoidal point inclusion can be referenced in Appendix B.

$$
\begin{aligned}
&Cyl \in p = distance <= 1, where \\
&distance = (xPoint^2/xRadius^2) + (yPoint^2/yRadius^2),\ where \\
&xRadius = radius\ of\ X\ axis\ of\ elliptical, \\
&yRadius = radius\ of\ Y\ axis\ of\ elliptical, \\
&xPoint = \vec{p}.x^2, \\
&yPoint = \vec{p}.y^2
\end{aligned}
$$

Figure 34: A Brief Equation Exemplifying the Final Portion of the Ellipsoid Point Inclusion Equation

The ellipsoid filtering model possesses the ability to determine if a specific, feature matched point resides within the model. However, the filtering model needs to be correctly placed on the point cloud to correctly filter undesirable points. Figure 35 illustrates the new chronological process of relative pose estimation provided the filtering model. The red boxes in Figure 35 indicate the added methodology related to removing noisy data. Obviously, the placement of the filtering model needs to occur prior to filtering invalid points but after the placement of the point cloud. Naturally, the first placement choice positions the filtering model on the center of the point cloud. However, the point cloud may have outliers, or any other noisy data, that may invalidate the placement of the filtering model. Thus, resulting in a potentially even worse relative pose estimation. Thus, a specific value least affected by outliers must be calculated. Mathematically, the median for a set of points is the least affected by outliers. Therefore, the median of the points in the point cloud will be utilized to place the filtering model. Finally, with the position of the filtering model, the math for ellipsoidal point inclusion, and a visual representation to determine the filtered points, more accurate relative pose estimations can be computed.

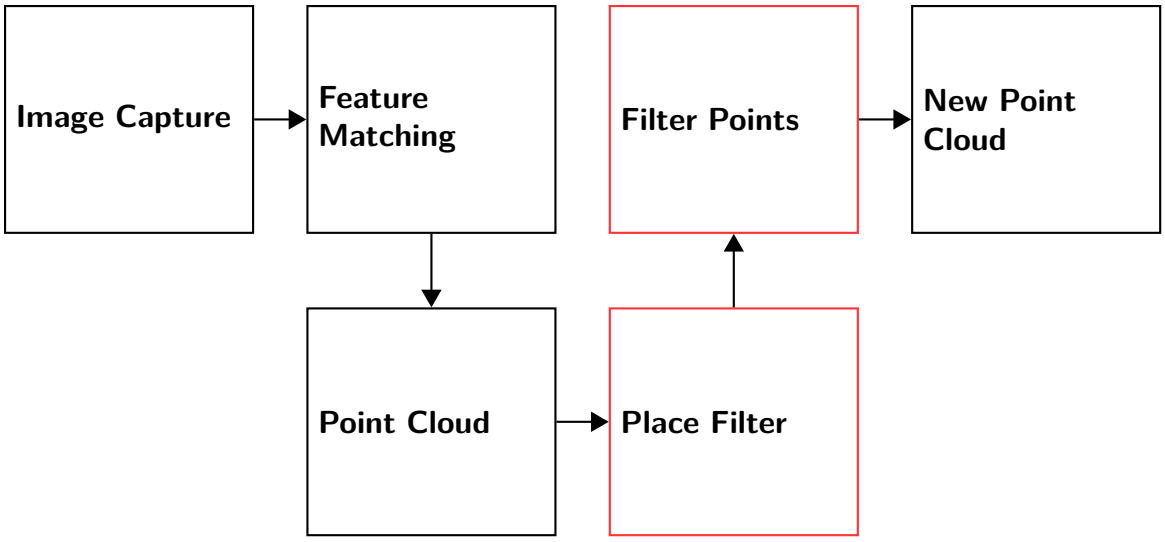


Figure 35: Chronological Order for Relative Pose Estimations Including Filtering

(a) The Filtering With the Model Visible

(b) The Filtering With the Model Invisible

Figure 36: The Ellipsoid Model Filtering Out Invalid Points and Calculating the Relative Pose Estimation

Table 2: Filtering Model Data for Flight 3.3

Filtering Model Flight 3.3									
Filter Y/N	Time/Filter	Time/ICP	Time	ICP	MAEX	MAEY	MAEZ	PD	PK
N	0ms	0.1153ms	0.1153ms	34	0.3036	0.0999	0.1544	0	0
Y	0.0026ms	0.127ms	0.129ms	40	0.2270	0.0908	0.2123	167	5837

Figure 36 and Table 2 indicate the utilization of the filtering model for flight 3.3. Figure 36 displays the filtering model, the inclusive red points, and the invalid blue points. Chronologically, the final relative pose estimation for the red points were calculated after the removal of the blue points. Table 2 indicates the speed of filtering, speed of ICP, speed of the overall relative pose estimation, and the number of points removed from the filtering. Similarly, Table 2 exhibits the times values and axis errors related to relative pose estimations. Further, PD and PK indicate how many points were invalid and valid, respectively. The addition of PD and PK, $PD + PK = PT$, equals the total number of points prior to filtering. For this flight, flight 3.3, the filtering model further reduces the errors related to the physical world imagery; however, the time for ICP increases slightly. Additional results on the filtering model for other flights, including the virtual world imagery, can be referenced in Sections 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8.

Finally, following the completion of the filtering model, the corrections to the hardware mounting errors, and camera calibration corrections, a more accurate relative pose estimation can be calculated. Figure 37 indicates a pose estimation with the above errors accounted for. Once the pose estimation is calculated, an equivalent virtual relative pose estimation can be calculated utilizing an equivalent, virtual set of imagery. Finally, with equivalent physical and virtual imagery and pose estimations, a statement can be made to the validity of how well the virtual world accurately represents the physical world, and if so, to within what margin of error?

Figure 37: Relative Pose Estimation for Flight 3.3 After Error Corrections

3.7 Virtual World Data

To appropriately determine how well the virtual world depicts the physical world, both sets of data need to have equivalent testing measures. First, the physical data needs to be collected and adjusted for errors. Sections 3.2, 3.4, 3.5, and 3.6 illustrate the corrections needed for the physical world data. Next, the virtual data needs to be collected. However, the virtual data does not need all the corrections the physical data utilizes. For example, the hardware mounting errors are only associated with the physical data collection. Thus, no measurements for translation and rotation for the virtual data need to be applied. However, the virtual cameras still need good camera calibrations. Finally, the filtering of the virtual points might help with potential

outliers. Since the virtual world does not have the physically associated errors, the amount of invalid points will be minimized.

(a)

(b) Left Virtual Image

(c) Right Virtual Image

Figure 38: Relative Pose Estimation for Virtual Imagery

First, the virtual data needs to be collected. To collect the virtual data, virtual

cameras were placed on the virtual tanker. Then, for each relative pose estimation for a pair of physical imagery, a virtual pair of images are captured from the same perspective. Figure 38 illustrates both a pair of captured virtual imagery and the associated relative pose estimation. Figures 38b and 38c are the virtual re-creations from the top left and top right physical world images in Figure 38a. These virtual images are inserted into the stereo vision pipeline and are the baseline for the virtual world approximations. Additionally, the yellow, red, and blue points are associated with the physical imagery. The orange, green and purple points are associated with the virtual world. The orange points display the virtual imagery point cloud. The green points indicate the sensed position after completing ICP for the virtual imagery. Finally, the purple points illustrate the outliers filtered from the filtering model referenced in Section 3.6.

3.8 Understanding GNUPlot Graphs

A complete understanding of the pose estimation comparisons between the physical and virtual world require a certain understanding of the GNUPlot graphs created during this research. Figure 39 exemplifies the nature of plotting utilized during this research to compare relative pose estimations. Initially, the title indicates whether the imagery used during the pose estimations and error calculations are from the physical or virtual imagery using "Real" or "Virtual" as indicators. In Figure 39, the plot data was calculated from a set of the physical imagery. After the type of data, the specific 6 DOF axis is denoted in the title by an X, Y, or Z for "Position" and "Orientation". The X,Y,Z axes for orientation correlate to roll, pitch, and yaw. Additionally, the flight number can be located in the title. A set of relevant statistics have been calculated to further discuss the center and spread of data.

To enhance the visual depiction of the GNUPlots, the (Mean Absolute Error)

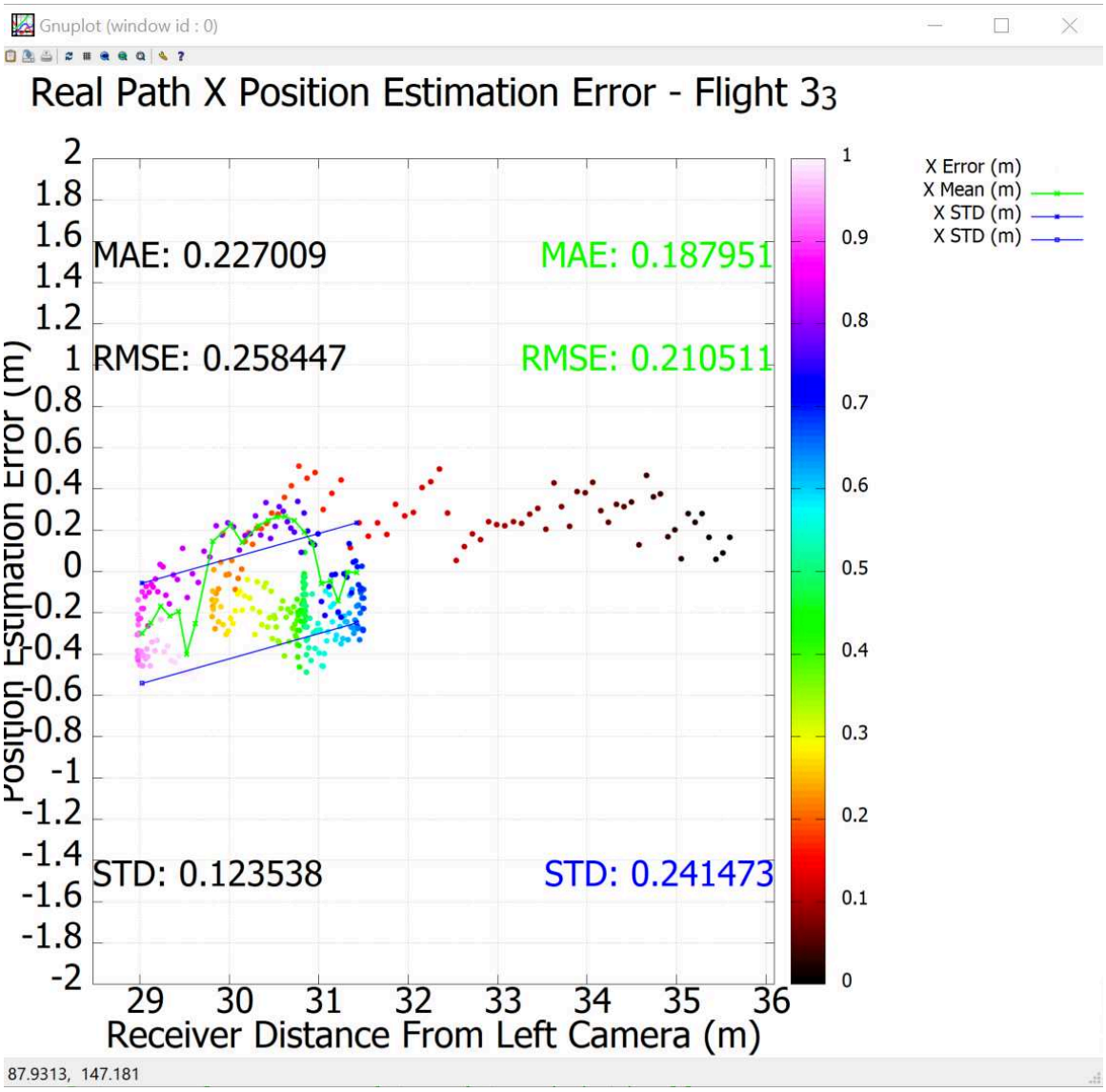


Figure 39: GNUPlot for Flight 3.3

(MAE), Root Mean Squared Error (RMSE), and standard deviation are calculated. The MAE statistic relays the mean of the error while maintaining absolute value. The absolute value portion helps avoids situations where the error would minimize between the negative and positive axes and bias towards 0 incorrectly. The MAE generally acts as the average-bound of the points. Similarly, RMSE communicates another mean of the error; however, instead of the absolute value, the RMSE squares the individual errors. The squaring during RMSE also helps avoid situations where

the error would incorrectly minimize between the positive and negative axes. But, due to the squaring, the RMSE exponentially grows with outliers. Thus, the RMSE proves to be an upper-bound for mean values whereas MAE functions as the average-bound [40]. Finally, the standard deviation helps represent the spread of the errors throughout the flight.

During aerial refueling, corrections based on the last few seconds of error may increase pose estimation accuracy. The errors are grouped into distance based classifications to further categorize the data. Next, the data is parameterized by time using the HSV color model. Time-wise, the black color indicates the oldest data points while the pink ones illustrate the newest data points. The HSV gradient on the right depicts where the data points fall between the time of the first data point and the last data point. Further, there are statistics placed on the graph to portray the data statistically. The black text (left) on the graph represent the statistics related to the entire set of data. The green text and blue text (right) on the graph represent the statistics related to the time based portion. Only data within the past 10 seconds are included in the time based calculations. Additionally, the time based data is seperated into categories based on distance, specifically every 0.1 meters. The time and distance based categorizations allow the visualization of the data, especially the slopes, as illustrated by the green line in Figure 39. However, due to potential lack of numbers within each bucket, the standard deviation is calculated for the entire time based data, instead of each individual bucket. Therefore, the standard deviation (blue) lines are a linear slope. Finally, the GNUPlots can be understood to best illustrate the physical and virtual data to allow comparisons between the two domains.

3.9 Compare Physical vs. Virtual

To appropriately determine how well the virtual world represents the physical world, each movement within the 6 DOF must be compared. To compare each individual axis, the 6 DOF pose estimations were individually plotted and analyzed on their own graphs using GNUPlot. Figure 40 illustrates each axis, X/Y/Z and Roll/Pitch/Yaw, using distance from the camera to the sensed model distance as the x axis and error between the sensed model and truth model as the y axis of the graph. To better understand the graph, recall from Section 2.3.2 that the refueling point is 30 meters from the cameras' position. Furthermore, the graph illustrates both the physical and virtual world errors. Assuming camera calibrations have been corrected, hardware mounting errors accounted for, and noisy points filtered, a final comparison between the physical and virtual errors can be asserted.

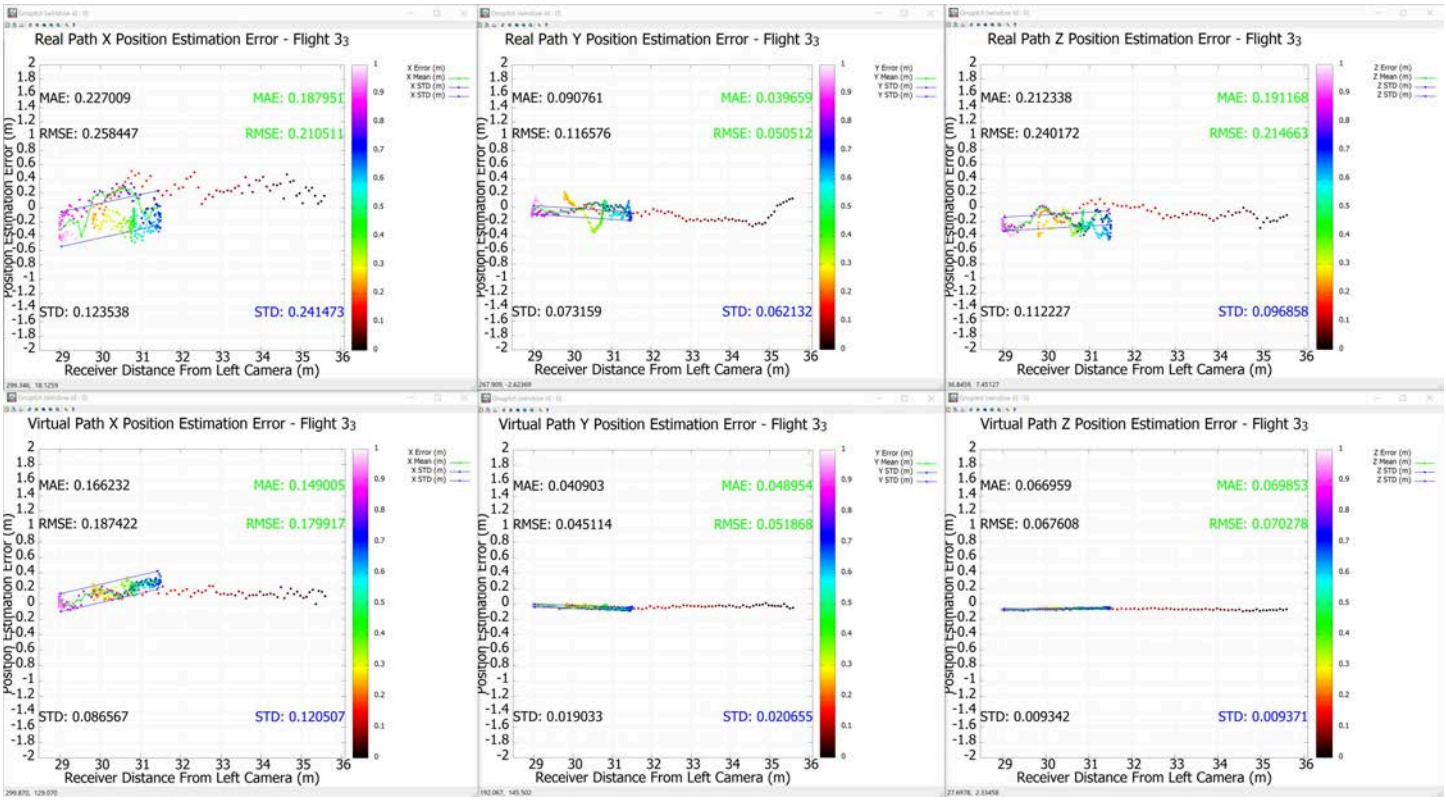


Figure 40: GNUPlot Showcase for Physical vs. Virtual

Figure 40 illustrates 3 DOF, specifically the relative position errors between the tanker and receiver. Moreover, Figure 40 contains data for the physical and virtual imagery. The title of each GNUPlot indicates the various type and axis of the recorded data. Referencing Figure 40, the physical and virtual x axis errors differ by roughly 10 centimeters, the y axis by 2 centimeters, and the z axis by 15 centimeters. Furthermore, notice that the x axis is the only virtual axis that does not look linear. However, all of the average errors for both the physical and virtual world are consistently under 30 centimeters. Additional results for comparisons between the physical and virtual world can be found in Chapter 4.

IV. Results and Analysis

The different camera calibrations for each flight resulted in differing comparisons for each flight. Thus, no single value sufficiently approximates the physical and virtual world results. Therefore, the results in this section are organized by flight. Within each flight section, a series of comparisons are made between the pre vs. post positioning, post vs. virtual positioning, pre vs. post orientation, post vs. virtual orientation, pre vs. post Infrared (IR) positioning, and pre vs. post IR orientation. The "pre" data is that from Section 3.3 that resulted in the initial pose estimations prior to any error correct from the physical data sets. The "post" data references the data from Section 3.6 and Figure 37 that resulted in the post error correction from the physical data sets. Lastly, the "virtual" data refers to the virtual data results from Section 3.7 collected by placing the truth data within the virtual world and executing equivalent experiments from the physical world.

Attached to each flight section are tables that indicate the error statistics from each axis related to their respective flight. The tables' first three rows indicate the "pre", "post", and "virtual" errors respectively. The fourth row indicates the improvement multiplier from the "pre" data to the 'post"data. The fifth row indicates the improvement multiplier from the "post" data to the "virtual" data. The improvement multiplier indicates the error reduction as a multiplier based off the original error. The equation can be calculated as $\frac{OriginalError}{NewError}$. Table 3 displays the formulas used to calculate the error reduction between each set of data as well as the format of the tables displayed in this section. Each flight also contains the corrected intrinsic matrices to obtain better camera calibrations. Both the initial and correct camera calibration matrices are available as indicated in Appendix A.

As described in Section 3.5, the cameras were secured into place from the first flight through the last flight. Therefore, the translation and rotation errors should

Table 3: Example Table Illustrating Error Reduction Formulas

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	Pre1	Pre2	Pre3	Pre4	Pre5	Pre6
PostCorrection	Post1	Post2	Post3	Post4	Post5	Post6
Virtual	V1	V2	V3	V4	V5	V6
PreToPostImprovement	$\frac{Pre1}{Post1}$	$\frac{Pre2}{Post2}$	$\frac{Pre3}{Post3}$	$\frac{Pre4}{Post4}$	$\frac{Pre5}{Post5}$	$\frac{Pre6}{Post6}$
PostToVirtualImprovement	$\frac{Post1}{V1}$	$\frac{Post2}{V2}$	$\frac{Post3}{V3}$	$\frac{Post4}{V4}$	$\frac{Post5}{V5}$	$\frac{Post6}{V6}$

have been maintained throughout all of the flights. Thus, each flight will have the same translation and rotation error correction values. Figure 41 displays values used to correct for the translation and rotation error.

$$T : \begin{Bmatrix} 2.015 & 0.406 & 0.2 \end{Bmatrix}$$
$$R : \begin{Bmatrix} -0.820 & 0.189 & -4.381 \end{Bmatrix}$$

Figure 41: Translation and Rotation Correction Values

4.1 Flight 1

4.1.1 EO

$$M1 : \begin{Bmatrix} 1403.56666731844689 & 0.00000000000000 & 626.85671545770185 \\ 0.00000000000000 & 1232.69651230282329 & 514.05511522043673 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2 : \begin{Bmatrix} 1392.54871835380345 & 0.00000000000000 & 674.85646521247349 \\ 0.00000000000000 & 1221.39419921652461 & 512.81030733293278 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 42: Flight 1 Electro-Optical (EO) Corrected M1/M2 Camera Calibrations Matrices

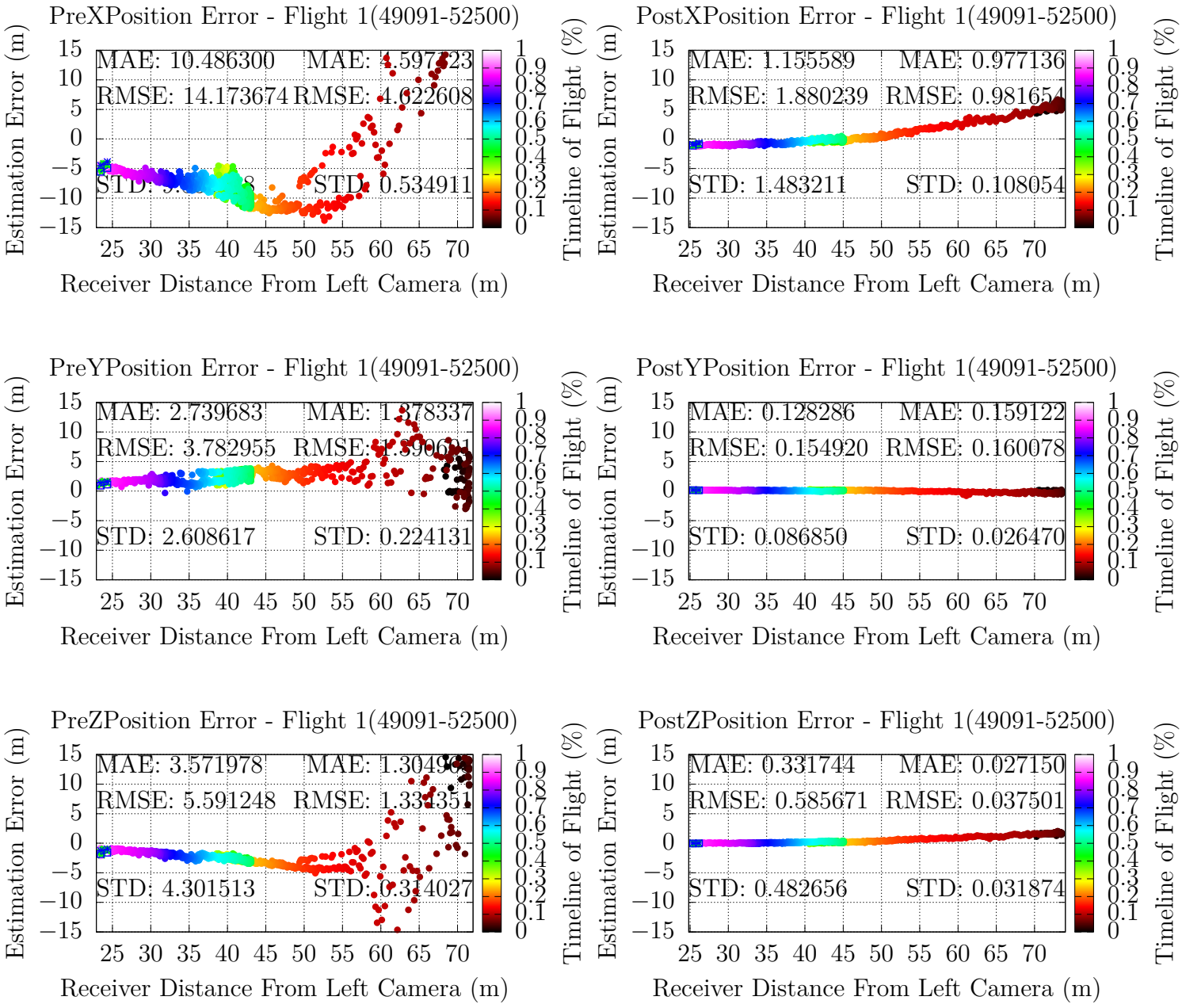


Figure 43: Flight 1 Comparing PreCorrections and PostCorrections for Positioning

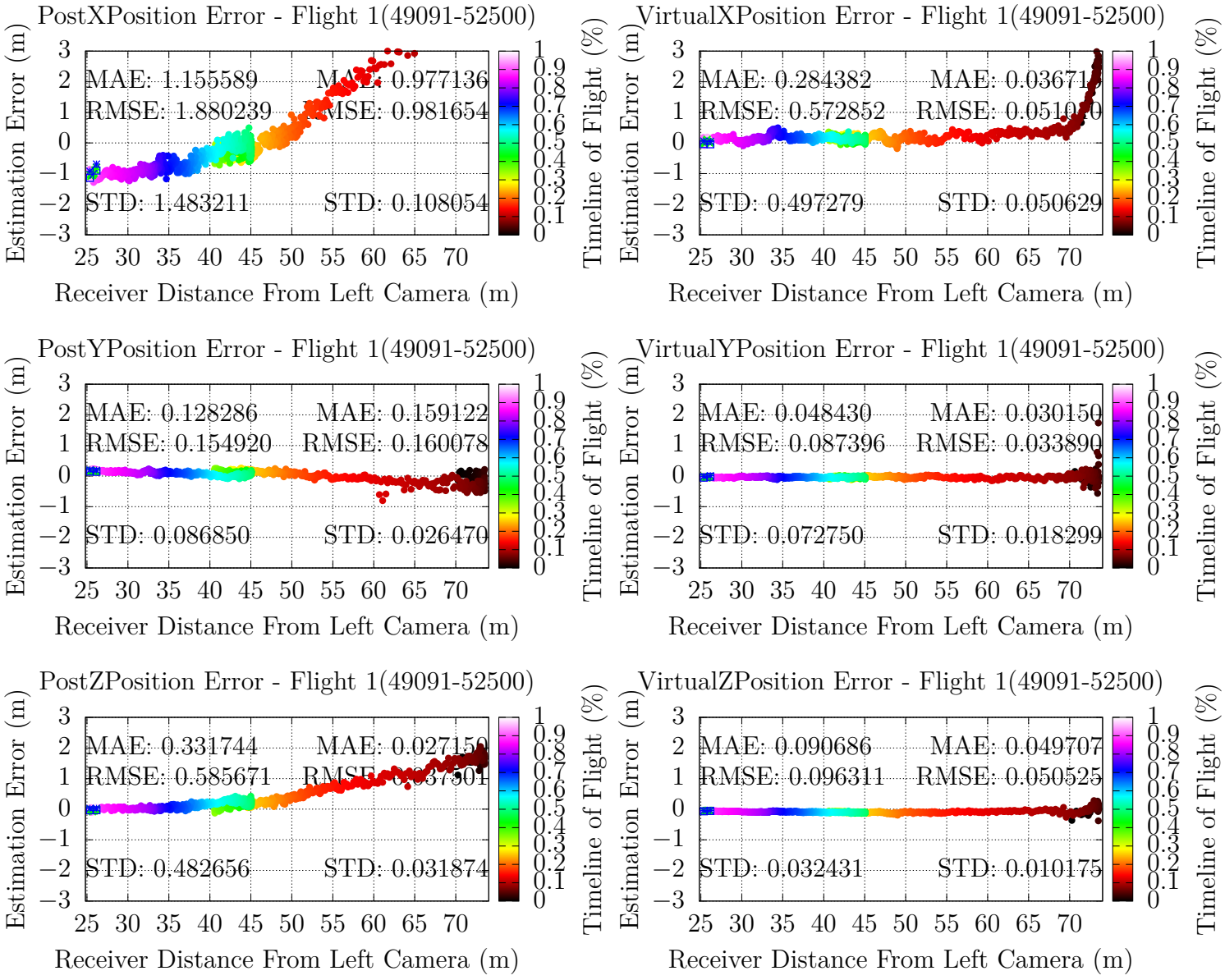


Figure 44: Flight 1 Comparing PostCorrections and Virtual for Positioning

Table 4: Position Error Estimation Statistics for Flight 1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	10.486	2.7396	3.5719	9.5357	2.6086	4.3015
PostCorrection	1.0845	0.1345	0.3936	1.5607	0.0839	0.4673
Virtual	0.3767	0.0469	0.0775	0.4786	0.0525	0.0287
PreToPostImprovement	8.6692	19.368	8.0749	5.1098	30.091	8.2050
PostToVirtualImprovement	1.8789	1.8678	4.0787	2.2609	0.598095	15.282

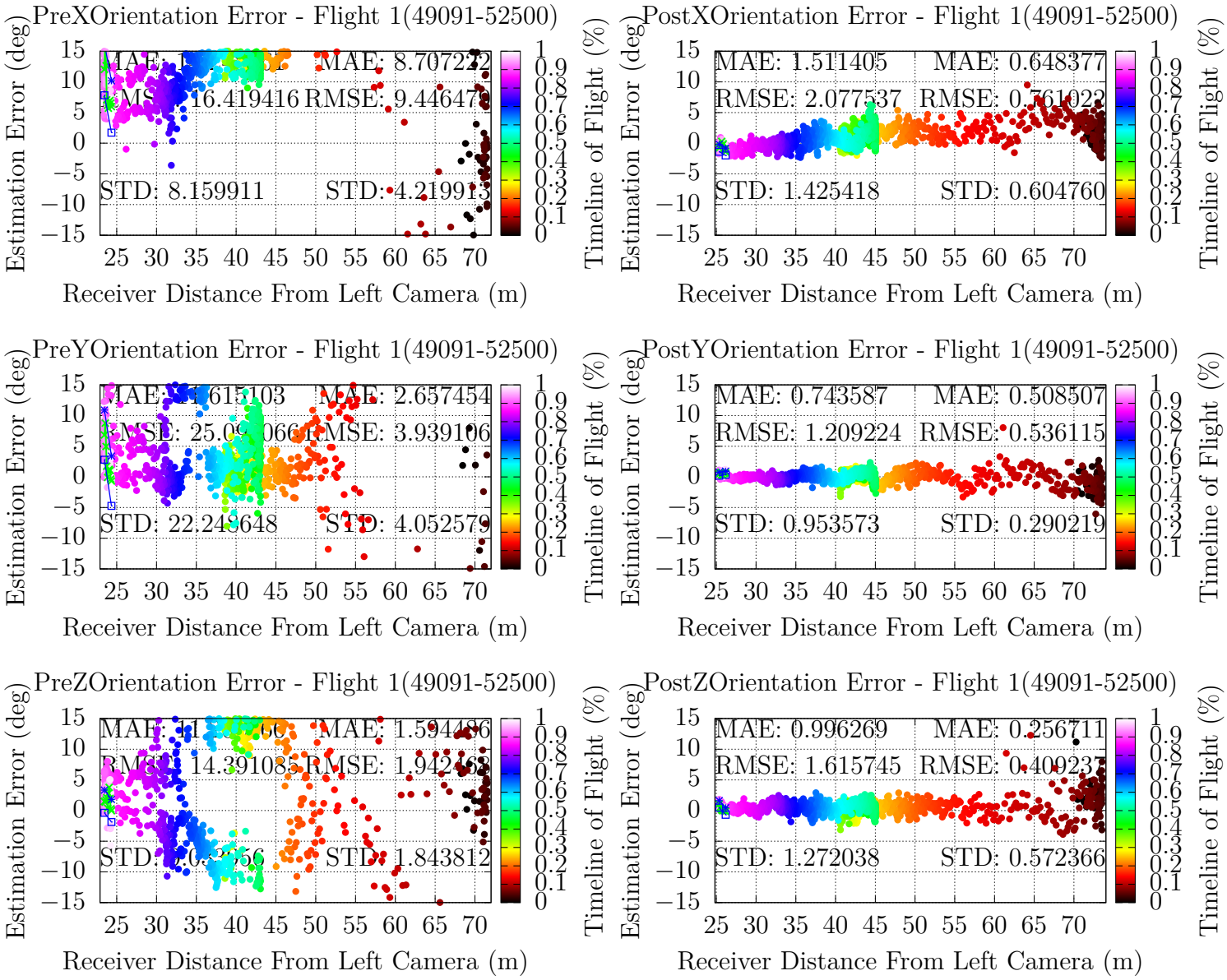


Figure 45: Flight 1 Comparing PreCorrections and PostCorrections for Orientation

Table 5: Orientation Error Estimation Statistics for Flight 1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	14.248	11.615	11.202	8.1599	22.2486	9.0340
PostCorrection	0.6825	1.7152	1.2551	0.9353	1.2351	1.1584
Virtual	0.2903	0.8742	0.7502	1.0340	0.8923	1.2029
PreToPostImprovement	19.876	5.7718	7.9252	7.7244	17.013	6.7987
PostToVirtualImprovement	1.3510	0.9620	0.6730	-0.095	0.3842	-0.036

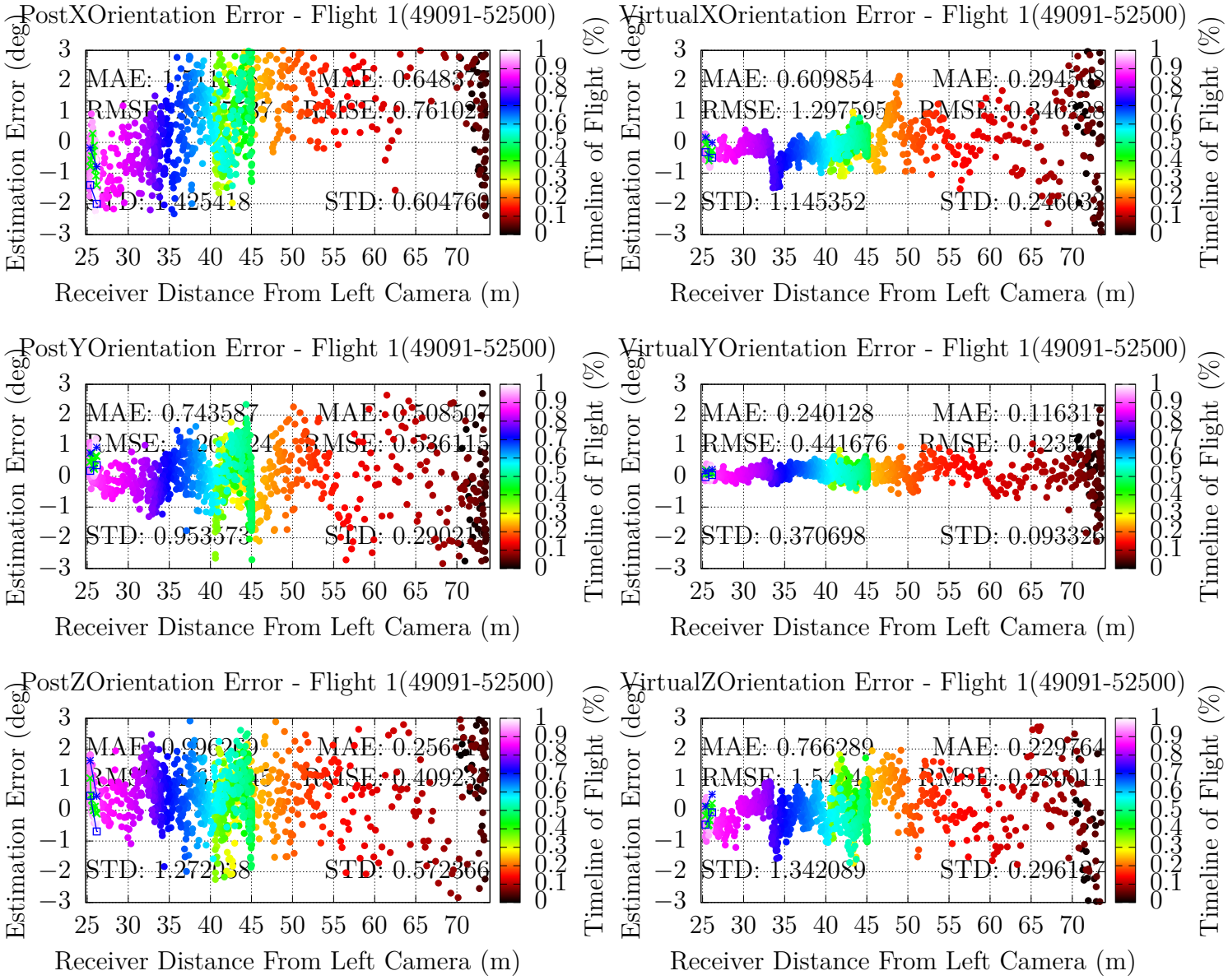


Figure 46: Flight 1 Comparing PostCorrections and Virtual for Orientation

4.1.2 IR

$$M1: \begin{Bmatrix} 923.83404710484842 & 0.00000000000000 & 518.62469390373735 \\ 0.00000000000000 & 899.21554704076482 & 394.45951124543234 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2: \begin{Bmatrix} 924.81885422055518 & 0.00000000000000 & 522.13431499205490 \\ 0.00000000000000 & 901.69398381234043 & 393.36129425441288 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 47: Flight 1 IR Corrected M1/M2 Camera Calibrations Matrices

Table 6: Position Error Estimation Statistics for Flight 1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	4.4577	4.4788	4.4456	3.3251	3.2963	2.5008
PostCorrection	0.9590	0.2913	0.4633	0.8920	0.3619	0.1715
PreToPostImprovement	3.6483	14.375	8.5955	2.7277	8.1083	13.582

Table 7: Orientation Error Estimation Statistics for Flight 1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	20.193	54.746	20.144	13.015	17.875	14.531
PostCorrection	1.5698	2.5276	2.2954	1.5609	3.7549	4.6269
PreToPostImprovement	11.863	20.659	7.7758	7.3381	3.7604	2.1405

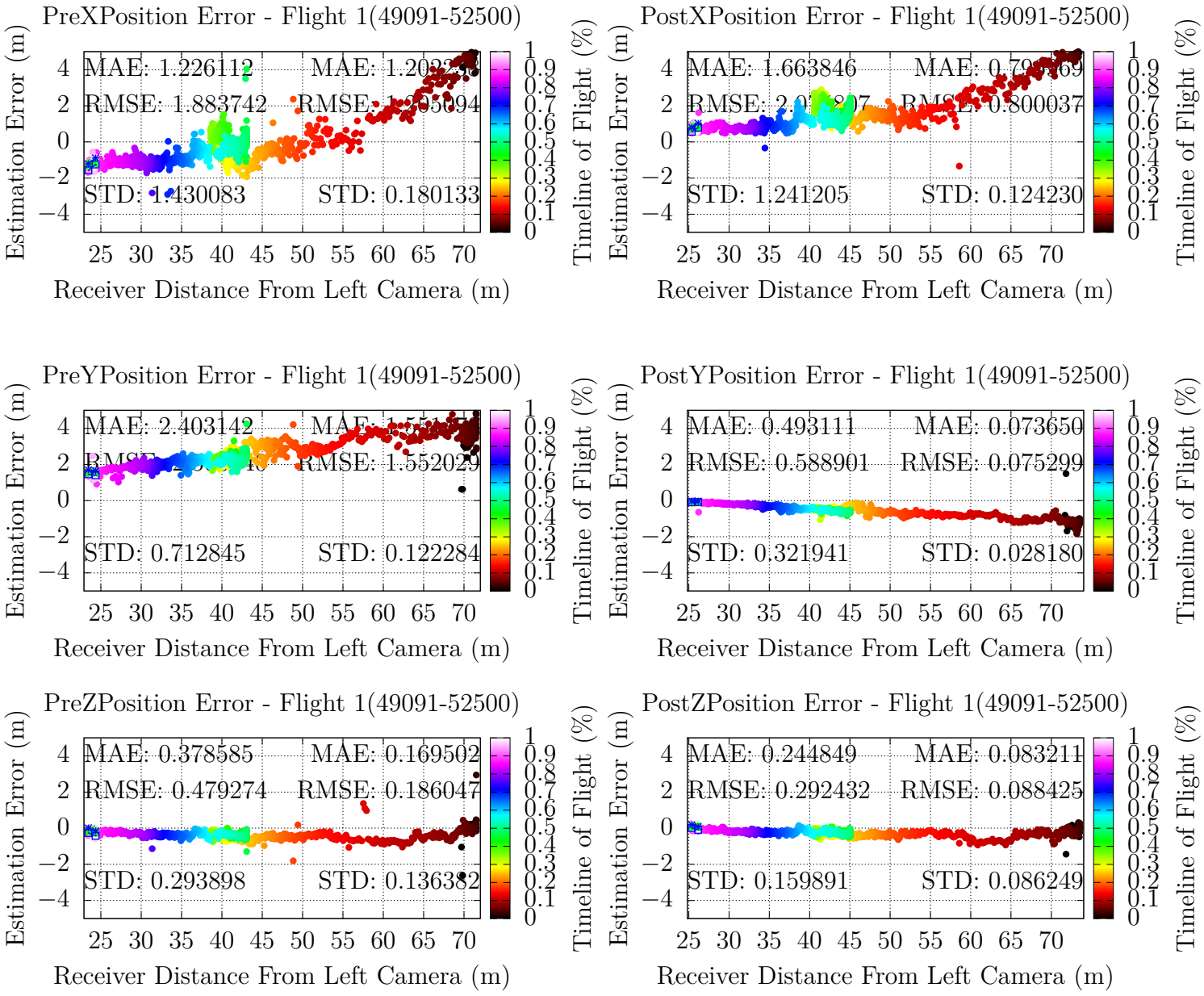


Figure 48: Flight 1 Comparing PreCorrections and PostCorrections for Positioning

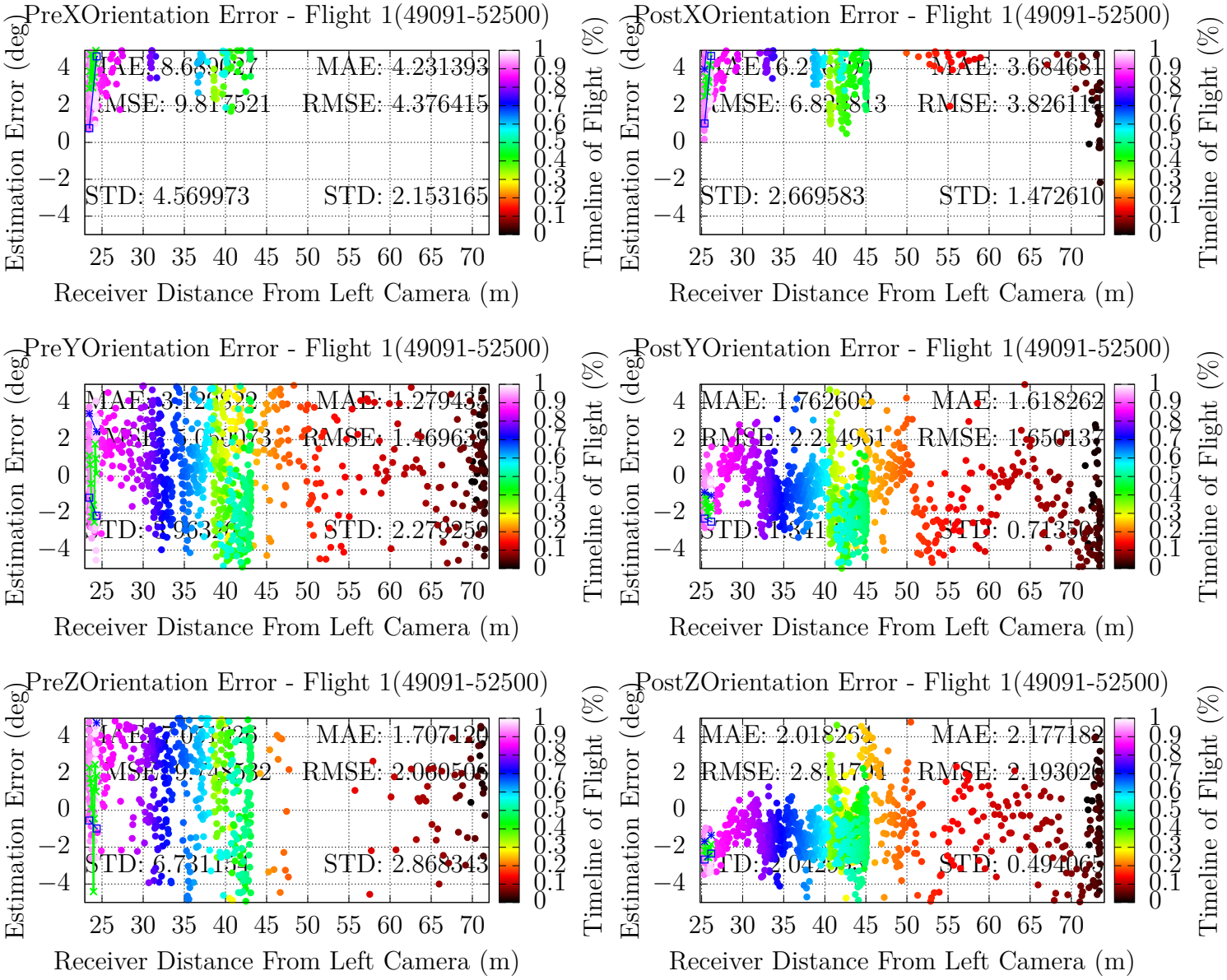


Figure 49: Flight 1 Comparing PreCorrections and PostCorrections for Orientation

4.2 Flight 2

4.2.1 EO

$$M1 : \begin{Bmatrix} 1337.06756915102187 & 0.00000000000000 & 661.65827931425383 \\ 0.00000000000000 & 1248.82317941853785 & 503.05934774158976 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2 : \begin{Bmatrix} 1324.72864741448211 & 0.00000000000000 & 709.07592807865922 \\ 0.00000000000000 & 1236.79254486871309 & 500.61317136502981 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 50: Flight 2 EO Corrected M1/M2 Camera Calibrations Matrices

Table 8: Position Error Estimation Statistics for Flight 2

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	1.0965	1.7237	0.3129	0.4940	0.4346	0.2452
PostCorrection	0.5665	0.6951	0.5238	0.4152	0.3542	0.3358
Virtual	0.2006	0.0389	0.0651	0.0707	0.0169	0.0122
PreToPostImprovement	0.9356	1.4798	-0.402	0.1898	0.2270	-0.269
PostToVirtualImprovement	1.8240	16.869	7.0461	4.8727	19.959	26.524

Table 9: Orientation Error Estimation Statistics for Flight 2

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	4.3917	2.0860	1.2961	2.3314	1.5390	1.3680
PostCorrection	1.8264	1.4179	1.9114	1.3132	1.0507	0.8712
Virtual	0.1577	0.6171	0.3210	0.1324	0.2415	0.2407
PreToPostImprovement	2.4046	0.4712	-0.322	0.7754	0.4647	0.5702
PostToVirtualImprovement	10.582	1.2977	4.9545	8.9184	3.3507	2.6194

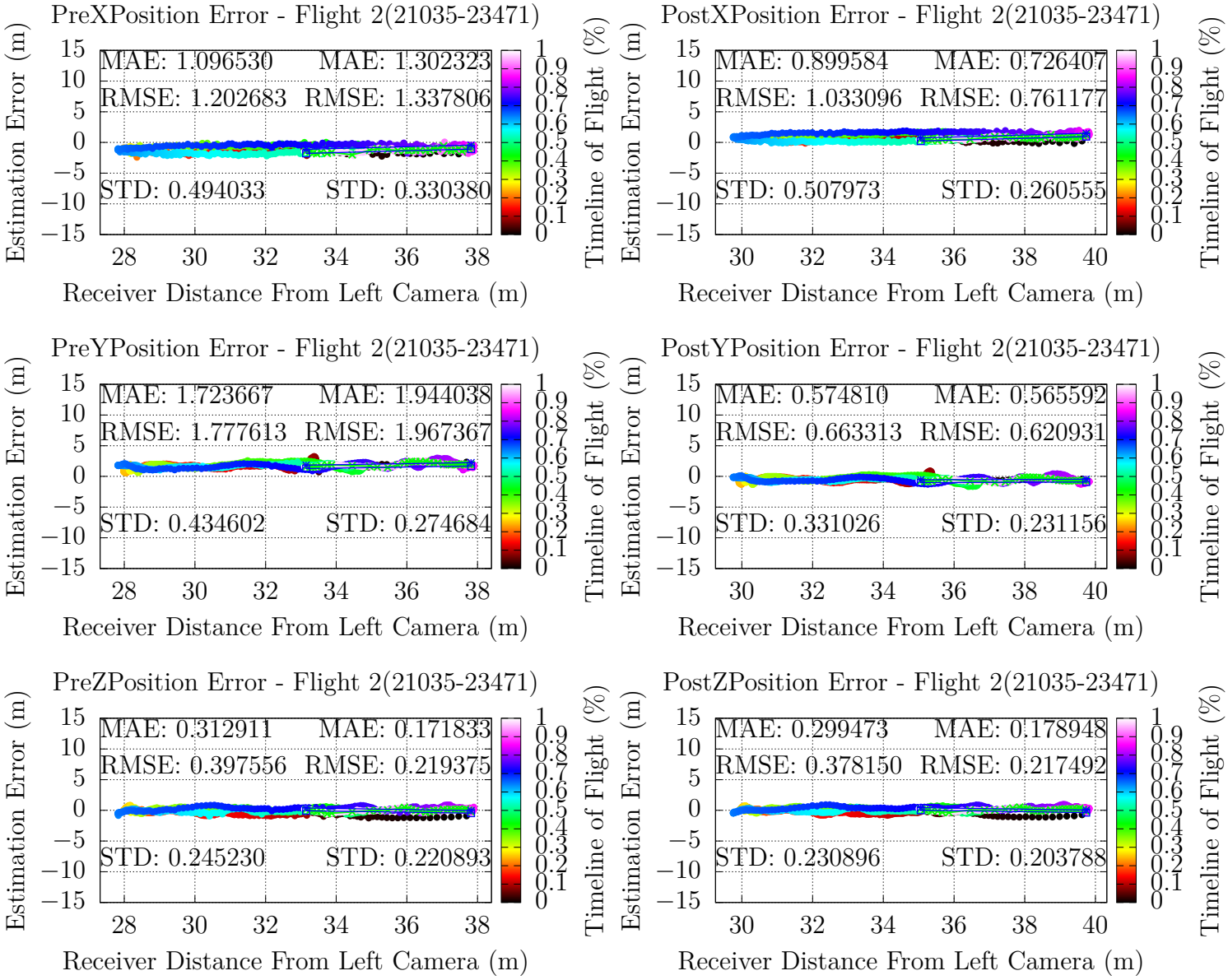


Figure 51: Flight 2 Comparing PreCorrections and PostCorrections for Positioning

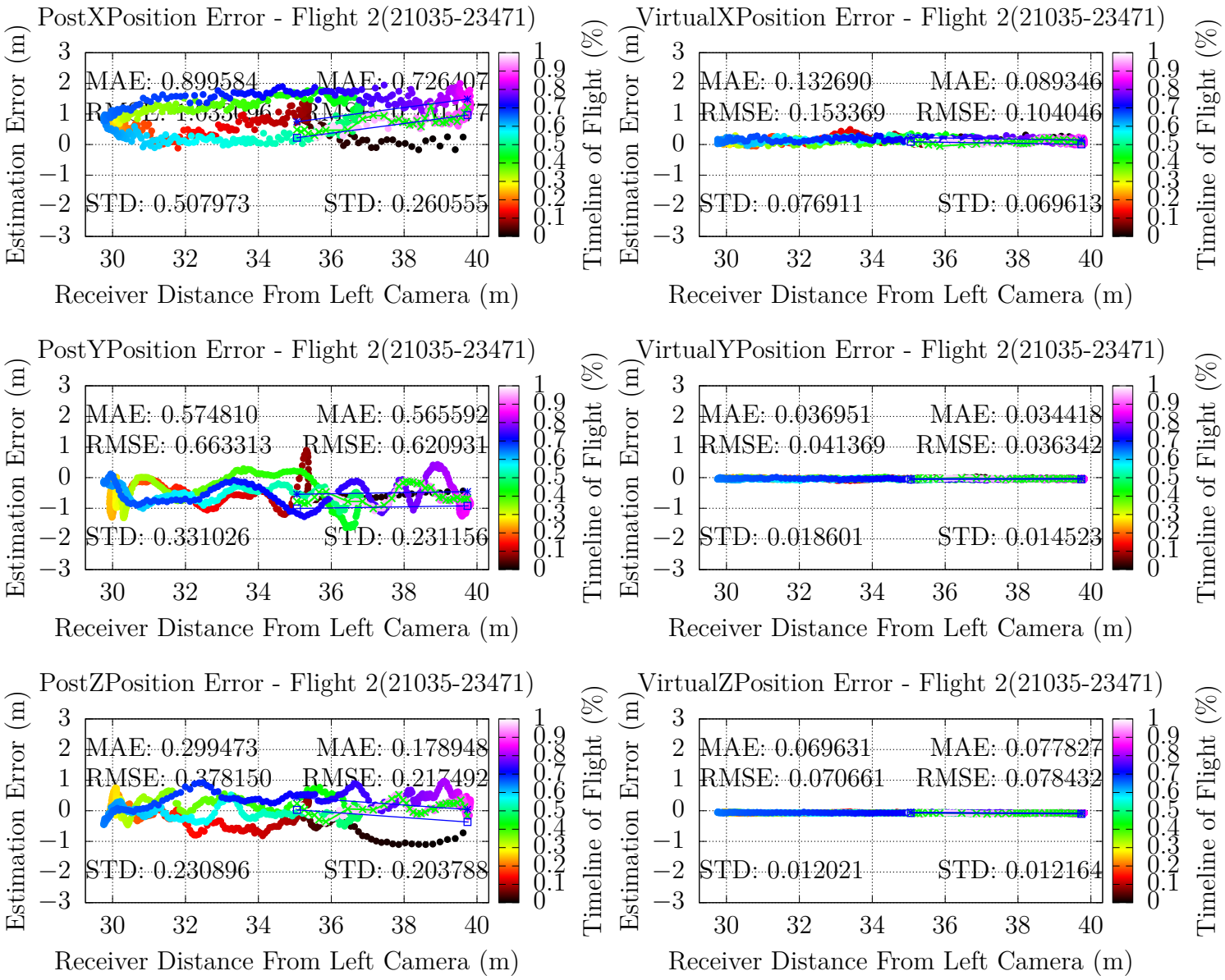


Figure 52: Flight 2 Comparing PostCorrections and Virtual for Positioning

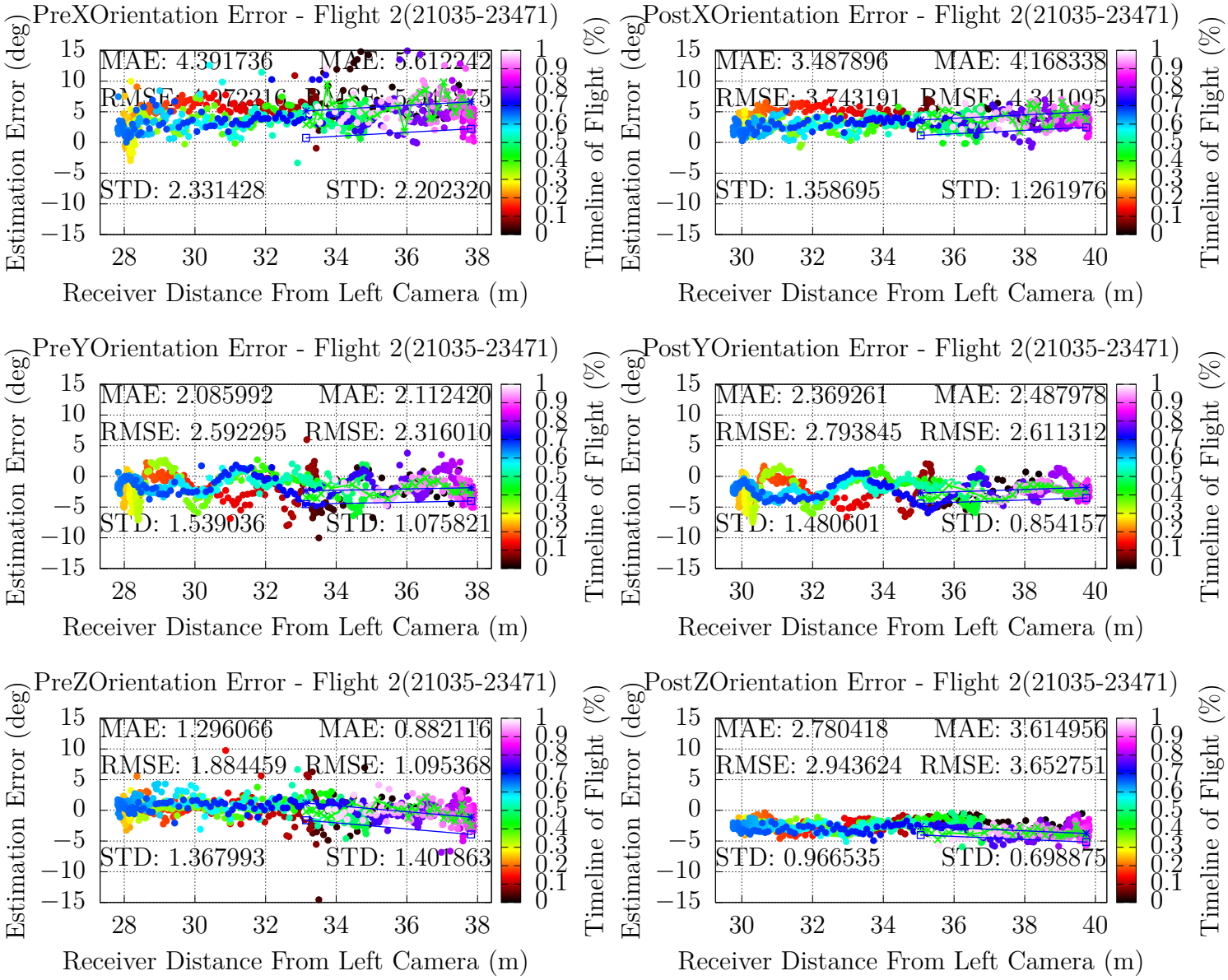


Figure 53: Flight 2 Comparing PreCorrections and PostCorrections for Orientation

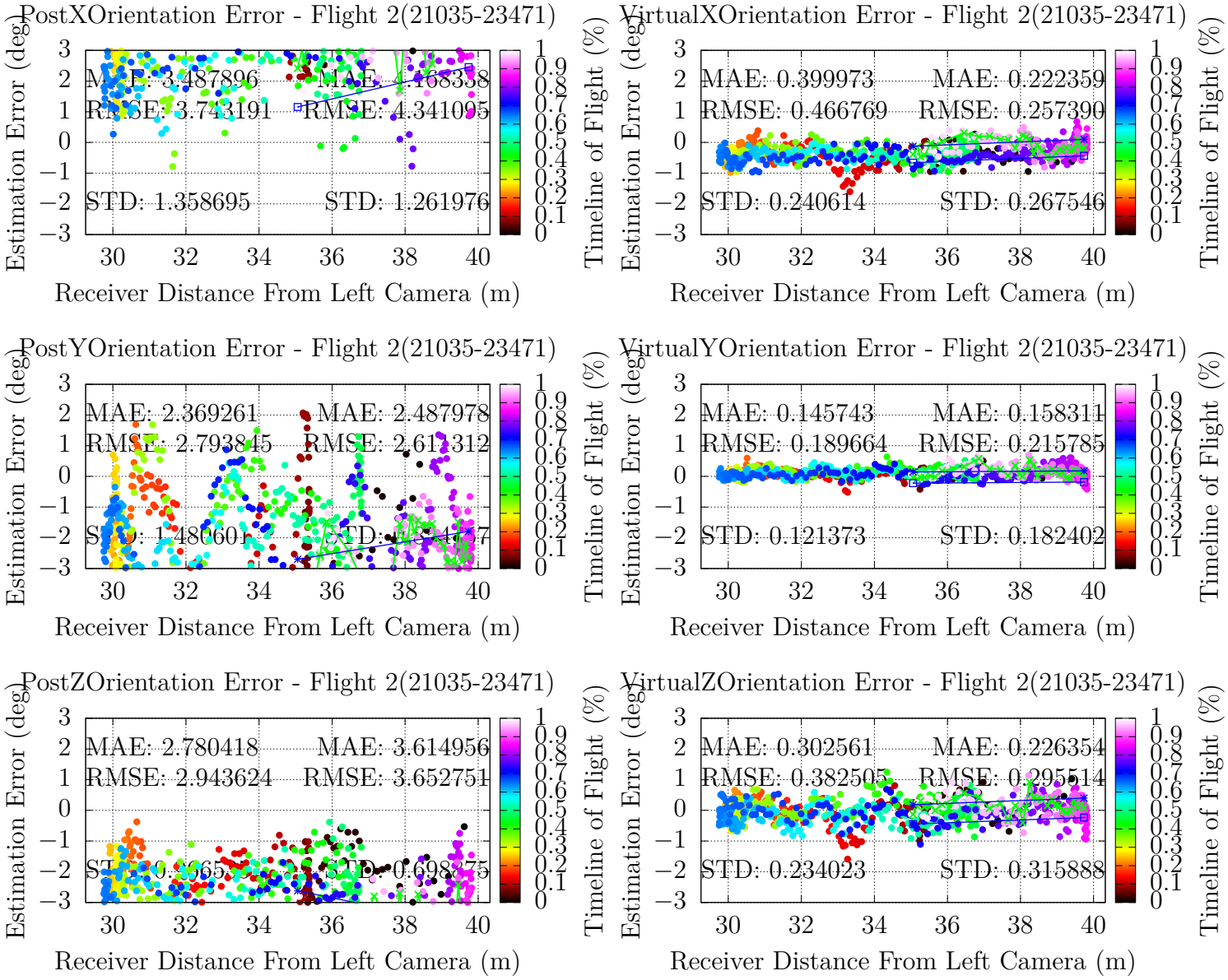


Figure 54: Flight 2 Comparing PostCorrections and Virtual for Orientation

4.2.2 IR

Due to hardware and wiring issues during flight tests, IR imagery does not exist for Flight 2.

4.3 Flight 3.1

4.3.1 EO

$$M1 : \begin{Bmatrix} 1451.46451339497708 & 0.00000000000000 & 633.98551411169683 \\ 0.00000000000000 & 1228.24807722601008 & 482.65405740523227 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2 : \begin{Bmatrix} 1442.36726601806345 & 0.00000000000000 & 683.78104221064871 \\ 0.00000000000000 & 1220.00586008914524 & 478.71722455329063 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 55: Flight 3.1 EO Corrected M1/M2 Camera Calibrations Matrices

Table 10: Position Error Estimation Statistics for Flight 3.1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	3.0952	1.7890	2.8176	1.4882	0.7658	0.5264
PostCorrection	1.1539	0.6666	0.3590	1.1190	0.6579	0.2209
Virtual	0.1649	0.0293	0.0515	0.0885	0.0212	0.0234
PreToPostImprovement	1.6824	1.6838	6.8485	0.3299	0.1640	1.3830
PostToVirtualImprovement	5.9976	21.751	5.9709	11.644	30.033	8.4402

Table 11: Orientation Error Estimation Statistics for Flight 3.1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	2.4847	4.1748	2.1810	3.0123	1.3391	1.5808
PostCorrection	3.4657	1.1707	2.8171	0.8936	0.6849	0.8558
Virtual	0.4820	0.4733	0.4685	0.5201	0.2861	0.3808
PreToPostImprovement	-0.283	2.5661	-0.226	2.3710	0.9552	0.8472
PostToVirtualImprovement	6.1903	1.4735	5.0130	0.7181	1.3939	1.2474

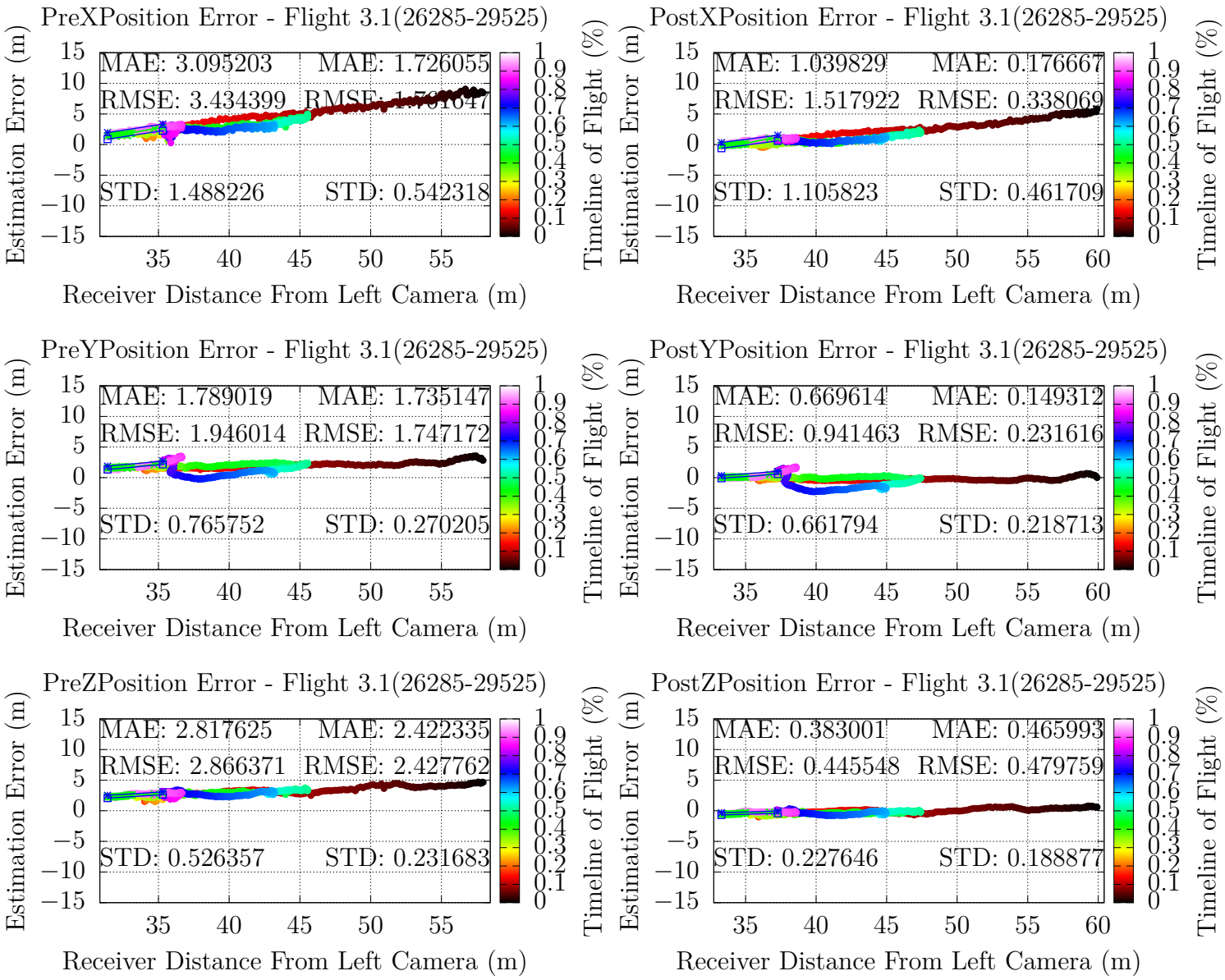


Figure 56: Flight 3.1 Comparing PreCorrections and PostCorrections for Positioning

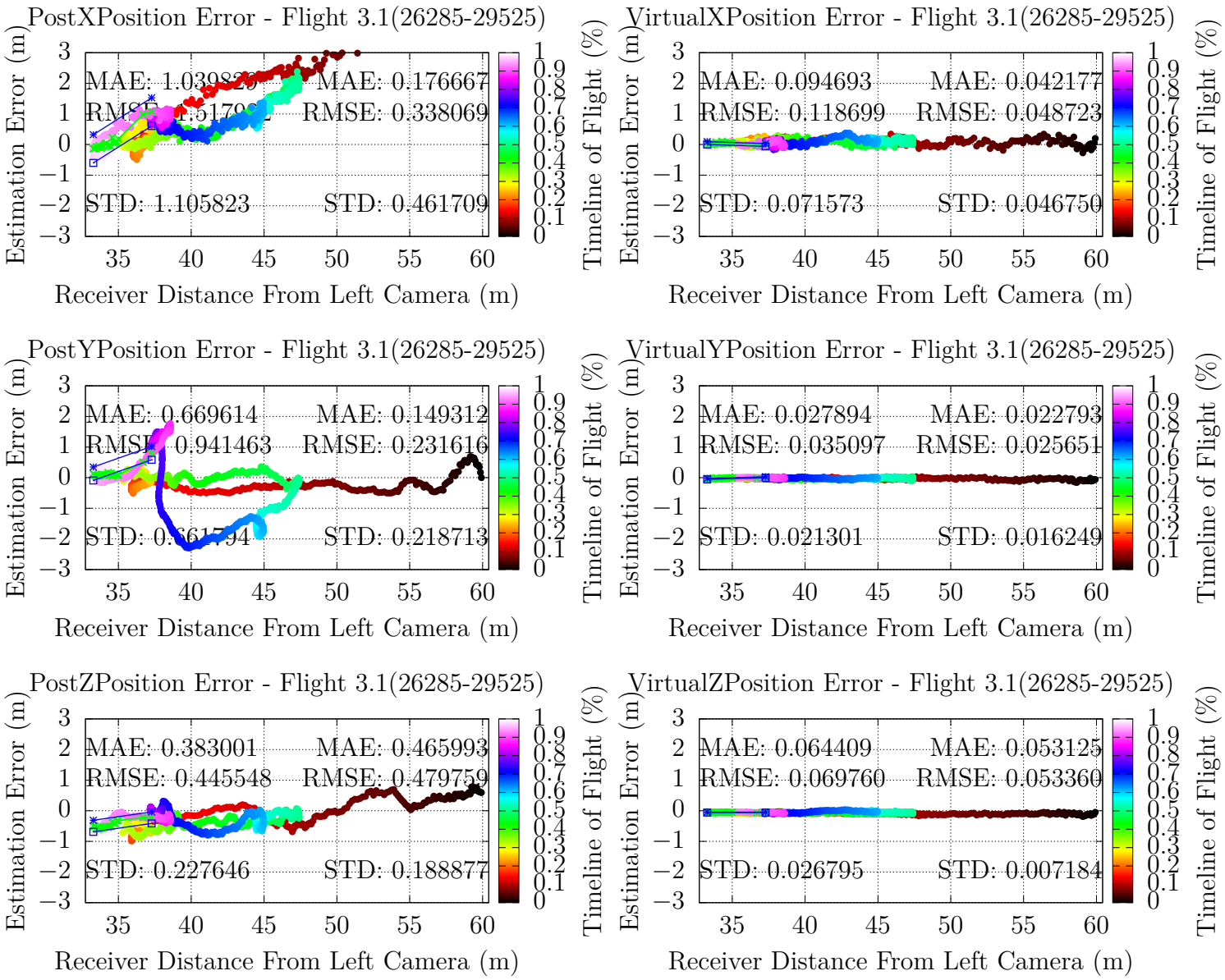


Figure 57: Flight 3.1 Comparing PostCorrections and Virtual for Positioning

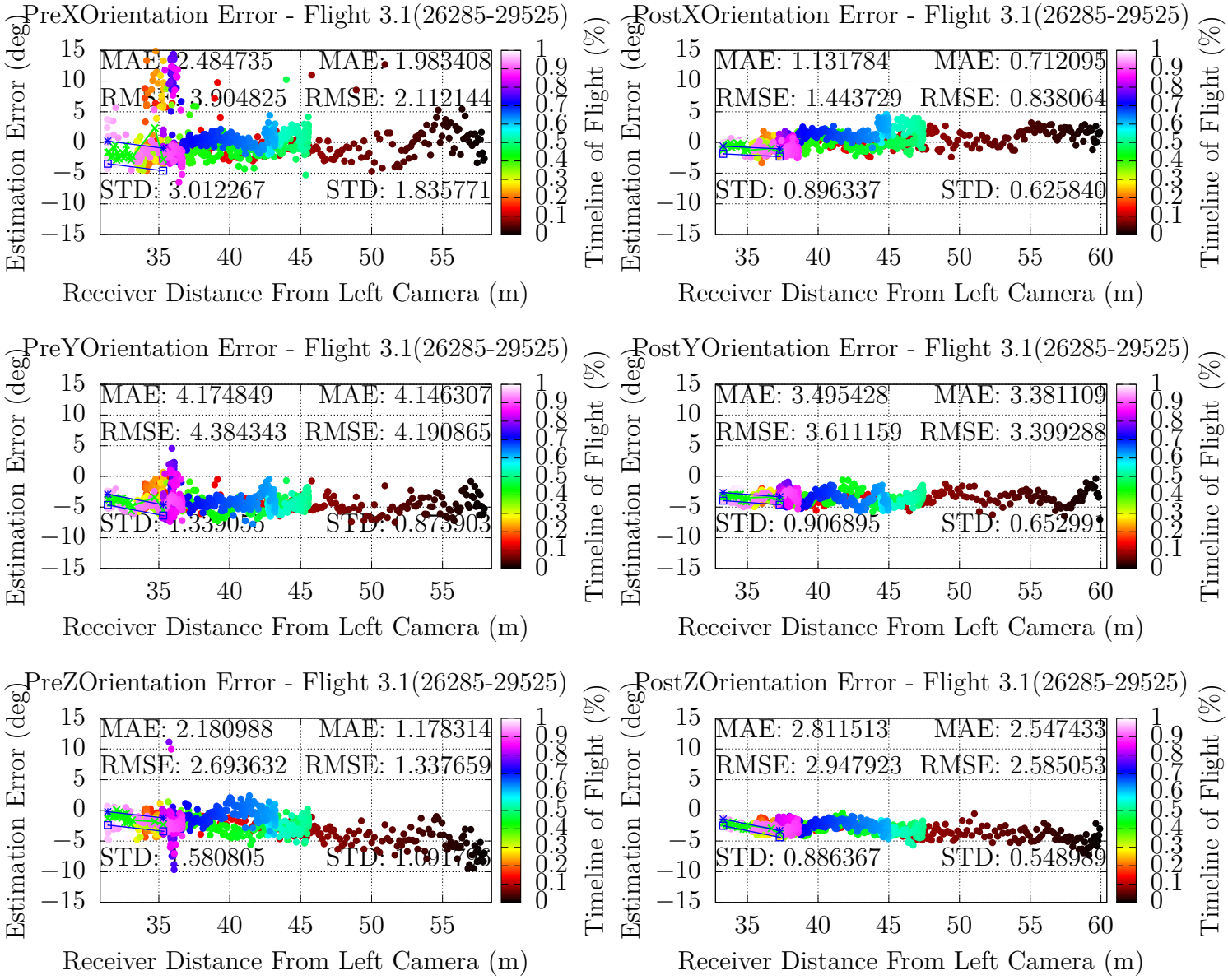


Figure 58: Flight 3.1 Comparing PreCorrections and PostCorrections for Orientation

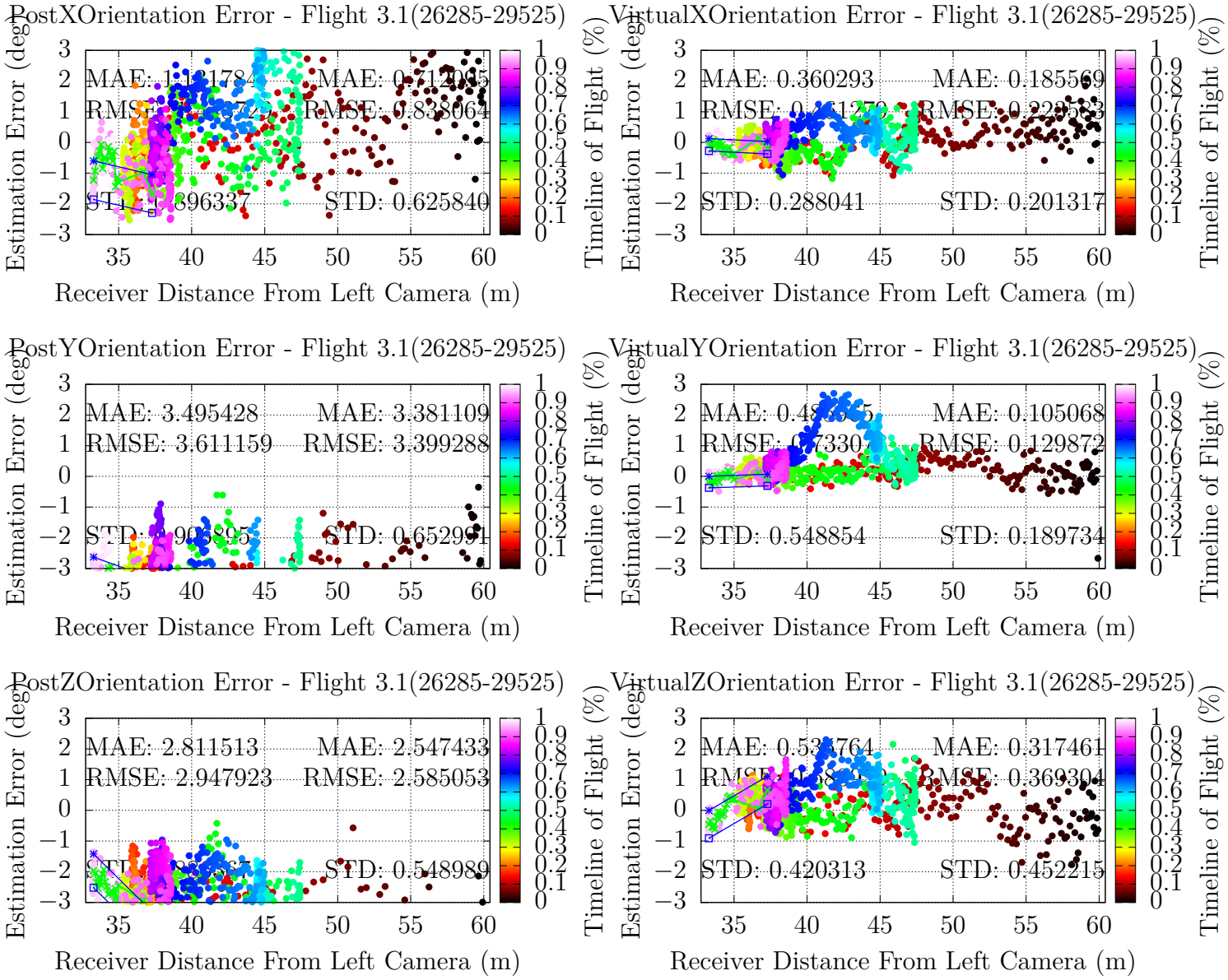


Figure 59: Flight 3.1 Comparing PostCorrections and Virtual for Orientation

4.3.2 IR

Due to hardware and wiring issues during flight tests, IR imagery does not exist for Flight 3.1.

4.4 Flight 3.3

4.4.1 EO

$$M1 : \begin{Bmatrix} 1441.46451339497708 & 0.00000000000000 & 653.98551411169683 \\ 0.00000000000000 & 1228.24807722601008 & 502.65405740523227 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2 : \begin{Bmatrix} 1432.36726601806345 & 0.00000000000000 & 703.78104221064871 \\ 0.00000000000000 & 1220.00586008914524 & 498.71722455329063 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 60: Flight 3.3 EO Corrected M1/M2 Camera Calibrations Matrices

Table 12: Position Error Estimation Statistics for Flight 3.3

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	1.2261	1.6877	1.4702	0.7575	0.4384	0.2765
PostCorrection	0.5473	0.3463	0.2344	0.3094	0.3947	0.1934
Virtual	0.2233	0.0363	0.0546	0.0826	0.0234	0.0198
PreToPostImprovement	1.2403	3.8735	5.2722	1.4483	0.1107	0.4297
PostToVirtualImprovement	1.4510	8.5399	3.2930	2.7458	15.868	8.7677

Table 13: Orientation Error Estimation Statistics for Flight 3.3

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	1.8926	3.1002	0.9439	0.9236	1.0291	0.8356
PostCorrection	3.1233	1.3504	2.6940	0.9358	0.7080	0.6407
Virtual	0.3247	0.7190	0.4389	0.3901	0.2543	0.3187
PreToPostImprovement	-0.394	1.2958	-0.649	-0.013	0.4535	0.3042
PostToVirtualImprovement	8.6190	0.8782	5.1381	1.3989	1.7841	1.0104

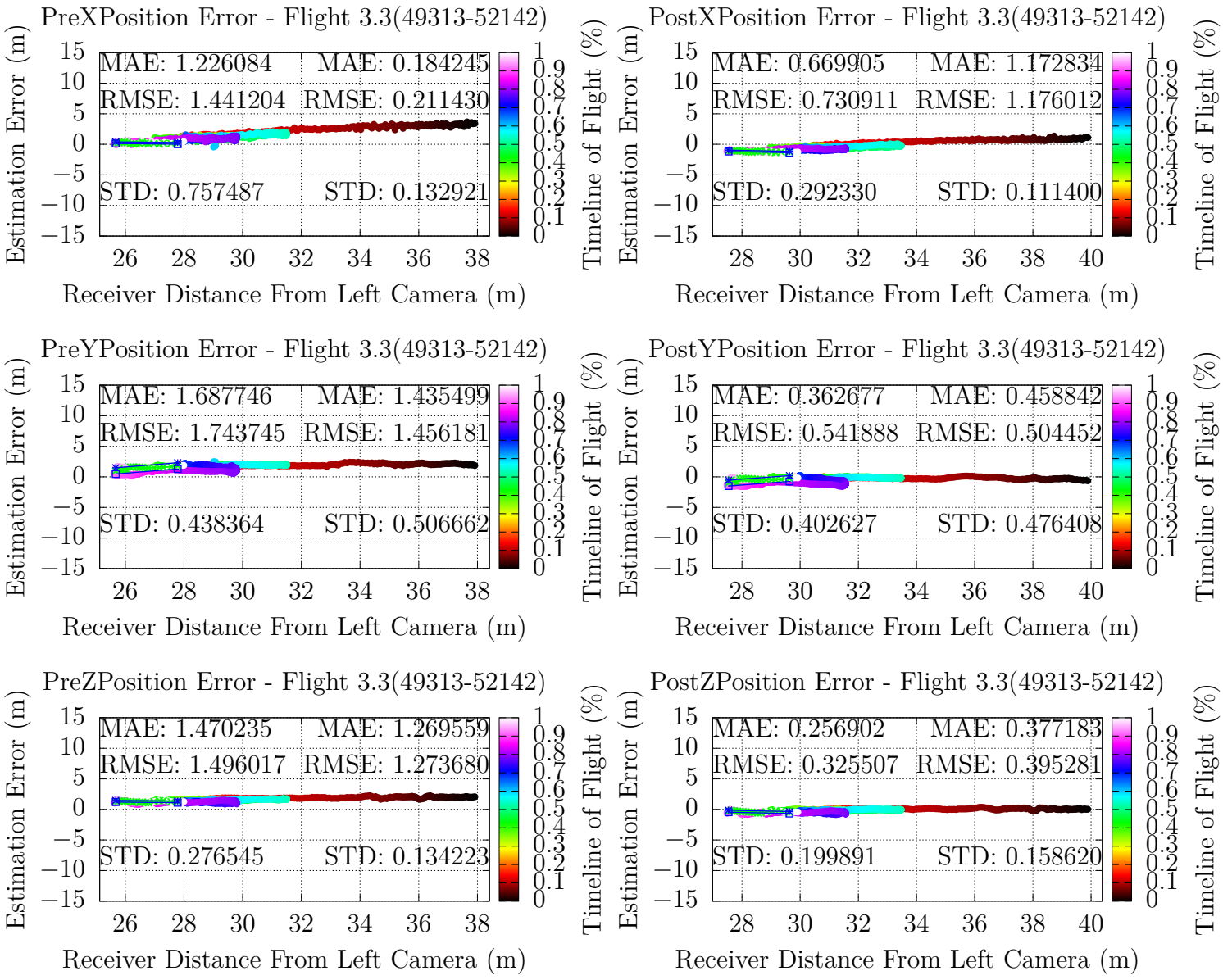


Figure 61: Flight 3.3 Comparing PreCorrections and PostCorrections for Positioning

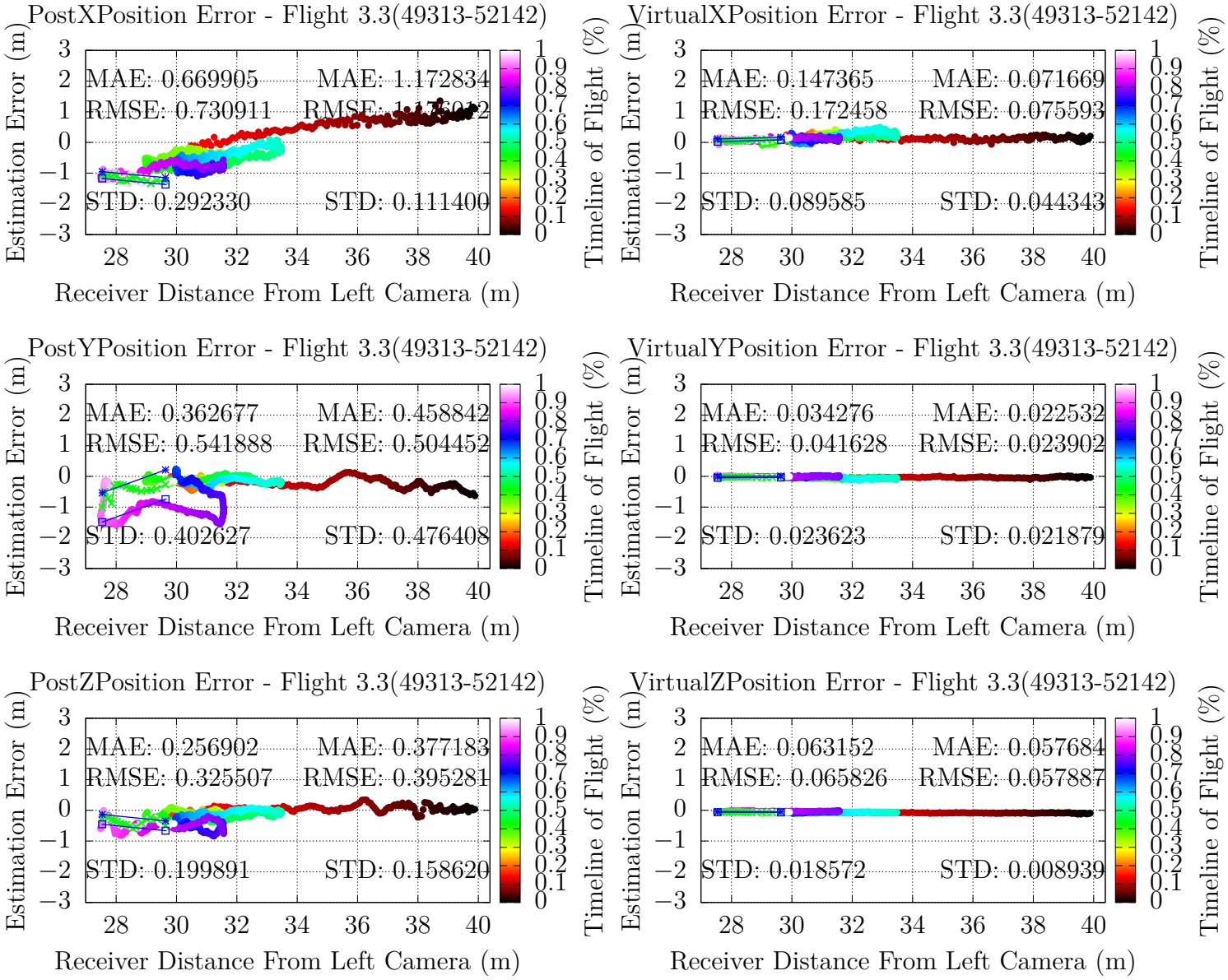


Figure 62: Flight 3.3 Comparing PostCorrections and Virtual for Positioning

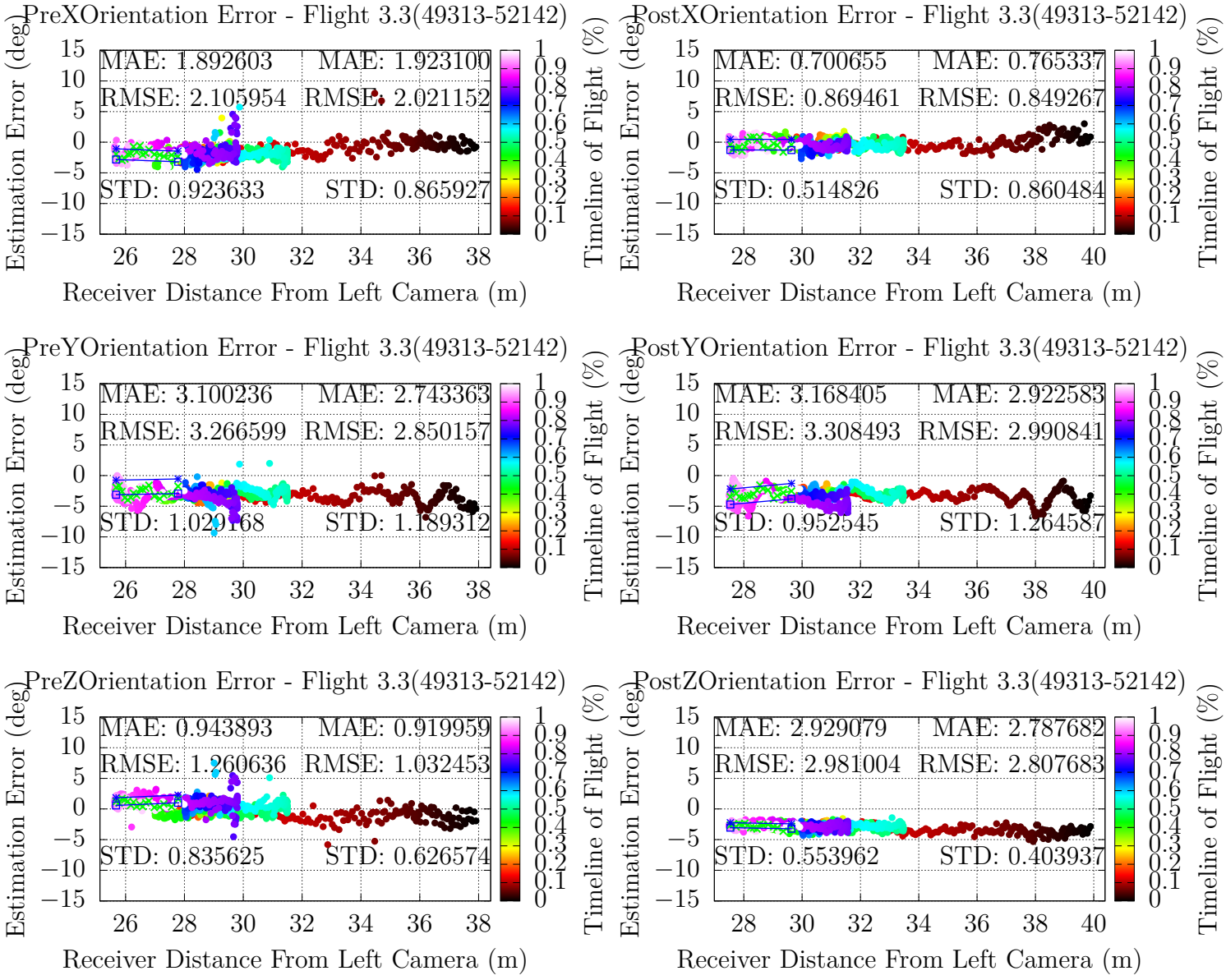


Figure 63: Flight 3.3 Comparing PreCorrections and PostCorrections for Orientation

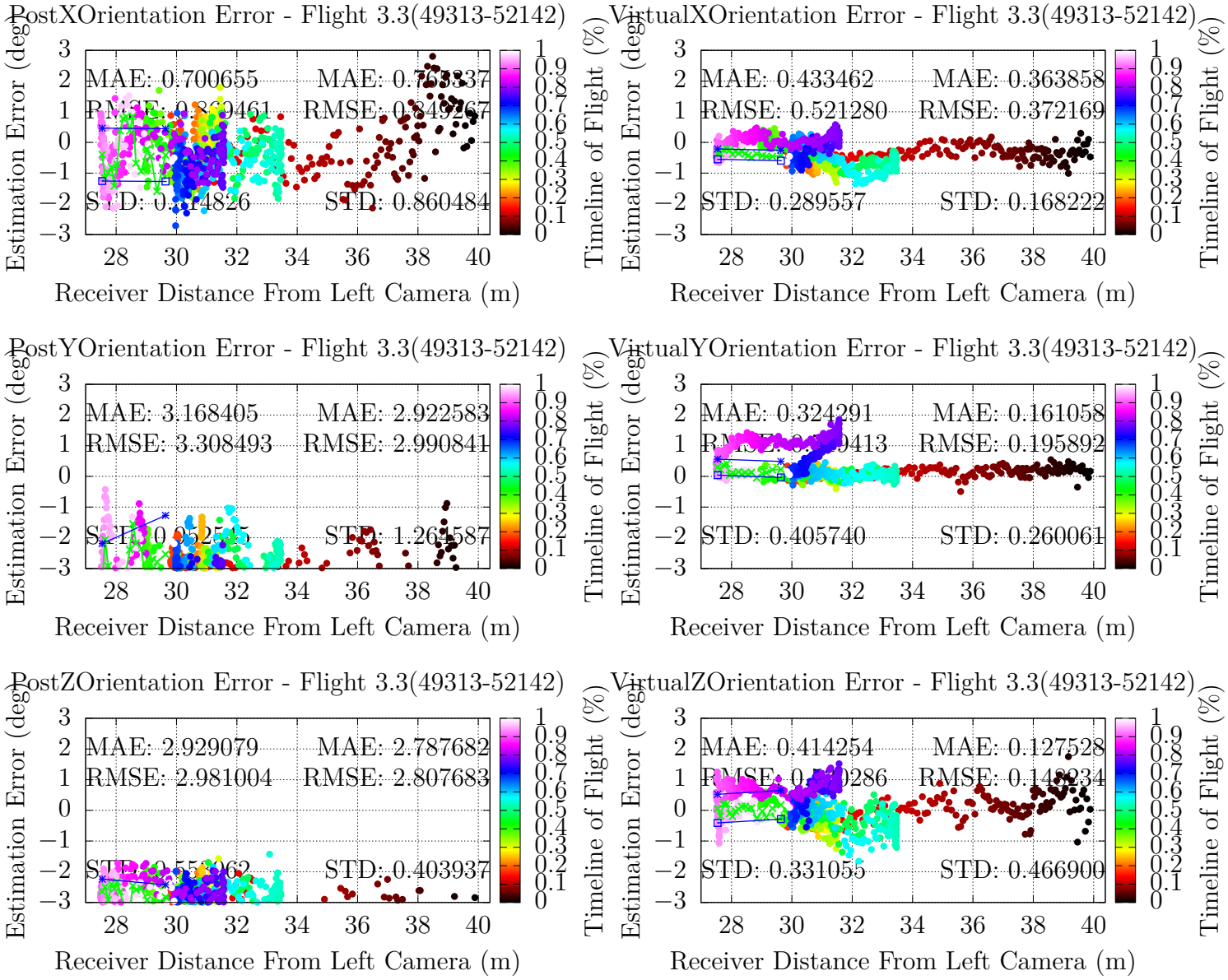


Figure 64: Flight 3.3 Comparing PostCorrections and Virtual for Orientation

4.4.2 IR

Due to hardware and wiring issues during flight tests, IR imagery does not exist for Flight 3.3.

4.5 Flight 5.1

4.5.1 EO

$$M1 : \begin{Bmatrix} 1350.69122843467449 & 0.000000000000008 & 665.31230214507218 \\ 0.00000000000000 & 1110.61664458947257 & 502.18293270488499 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2 : \begin{Bmatrix} 1352.90259830721971 & 0.00000000000000 & 712.03827622918266 \\ 0.00000000000000 & 1105.07619854918539 & 496.74734662364131 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 65: Flight 5.1 EO Corrected M1/M2 Camera Calibrations Matrices

Table 14: Position Error Estimation Statistics for Flight 5.1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	1.3062	2.8228	0.8572	0.8589	1.7281	0.9359
PostCorrection	1.7331	1.2481	0.5104	1.0276	1.3191	0.4244
Virtual	0.1138	0.0170	0.0532	0.0727	0.0120	0.0183
PreToPostImprovement	-0.246	1.2617	0.6795	-0.164	0.3101	1.2052
PostToVirtualImprovement	14.229	72.418	8.5940	13.135	108.93	22.191

Table 15: Orientation Error Estimation Statistics for Flight 5.1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	5.9920	6.0438	5.5364	9.3527	17.1753	4.3163
PostCorrection	3.8742	2.2867	5.0565	2.0589	0.7660	0.7322
Virtual	0.3029	0.4290	0.3959	0.3678	0.2289	0.3379
PreToPostImprovement	0.5466	1.6430	0.0949	3.5426	21.422	4.8950
PostToVirtualImprovement	11.790	4.3303	11.772	4.5979	2.3464	1.1669

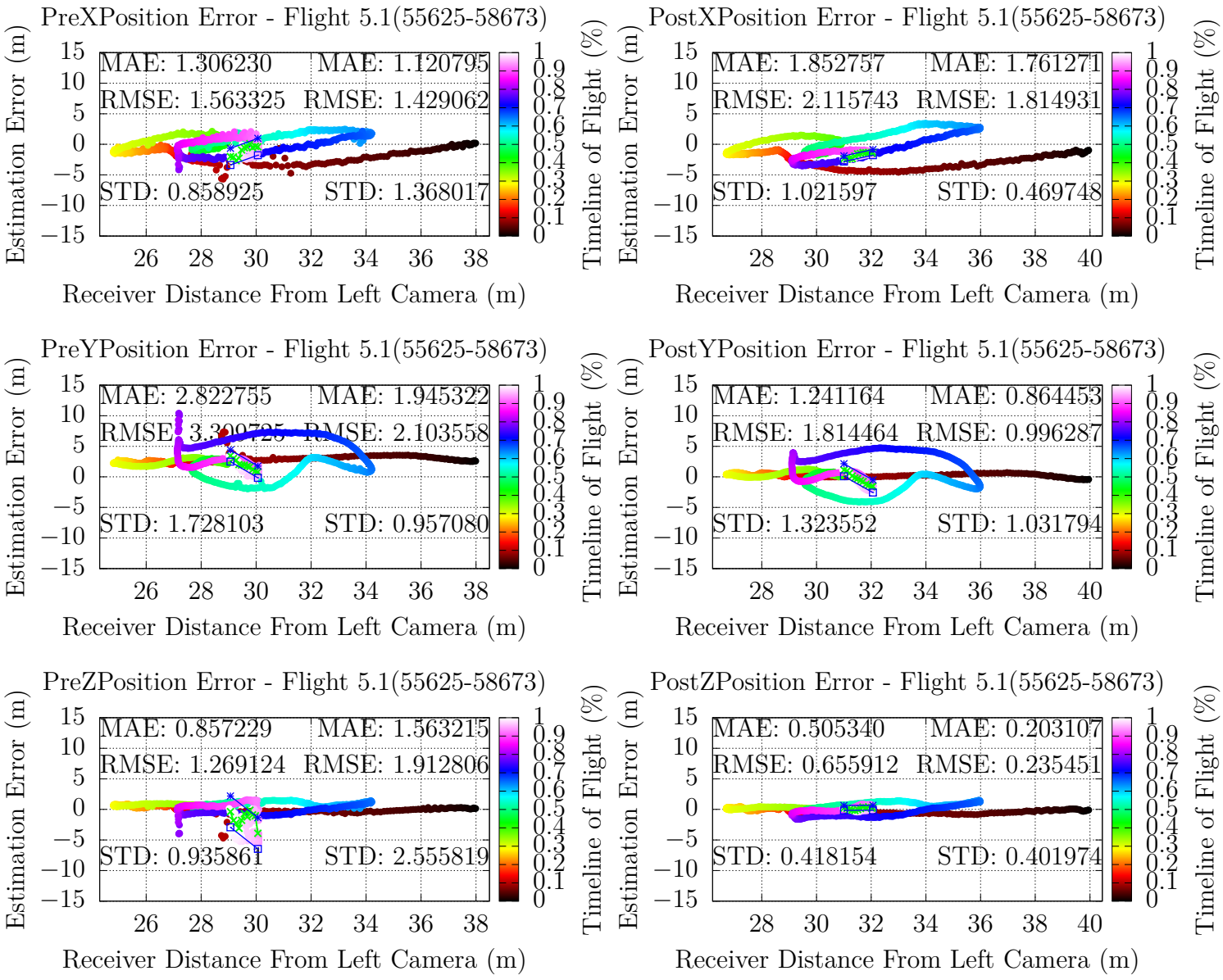


Figure 66: Flight 5.1 Comparing PreCorrections and PostCorrections for Positioning

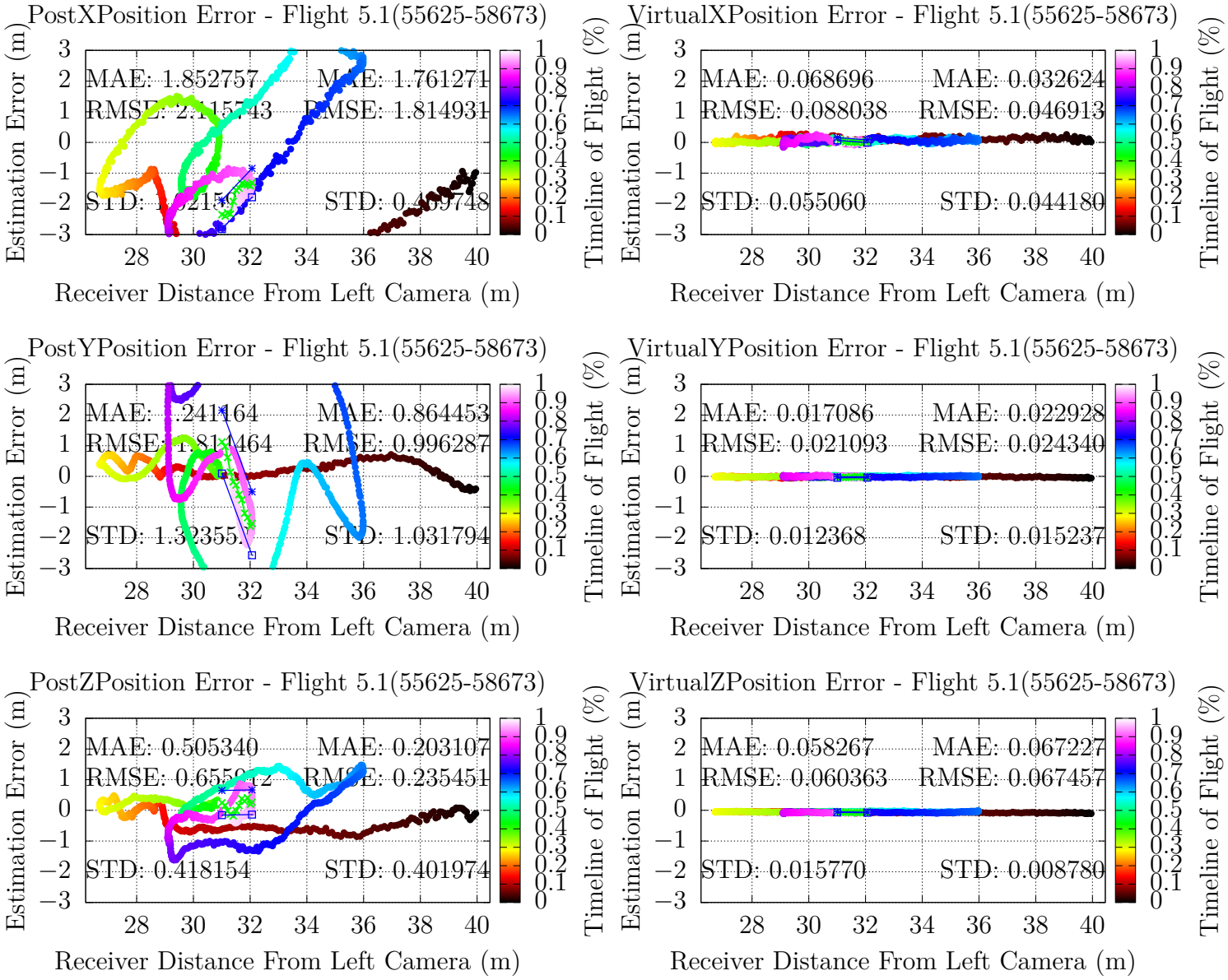


Figure 67: Flight 5.1 Comparing PostCorrections and Virtual for Positioning

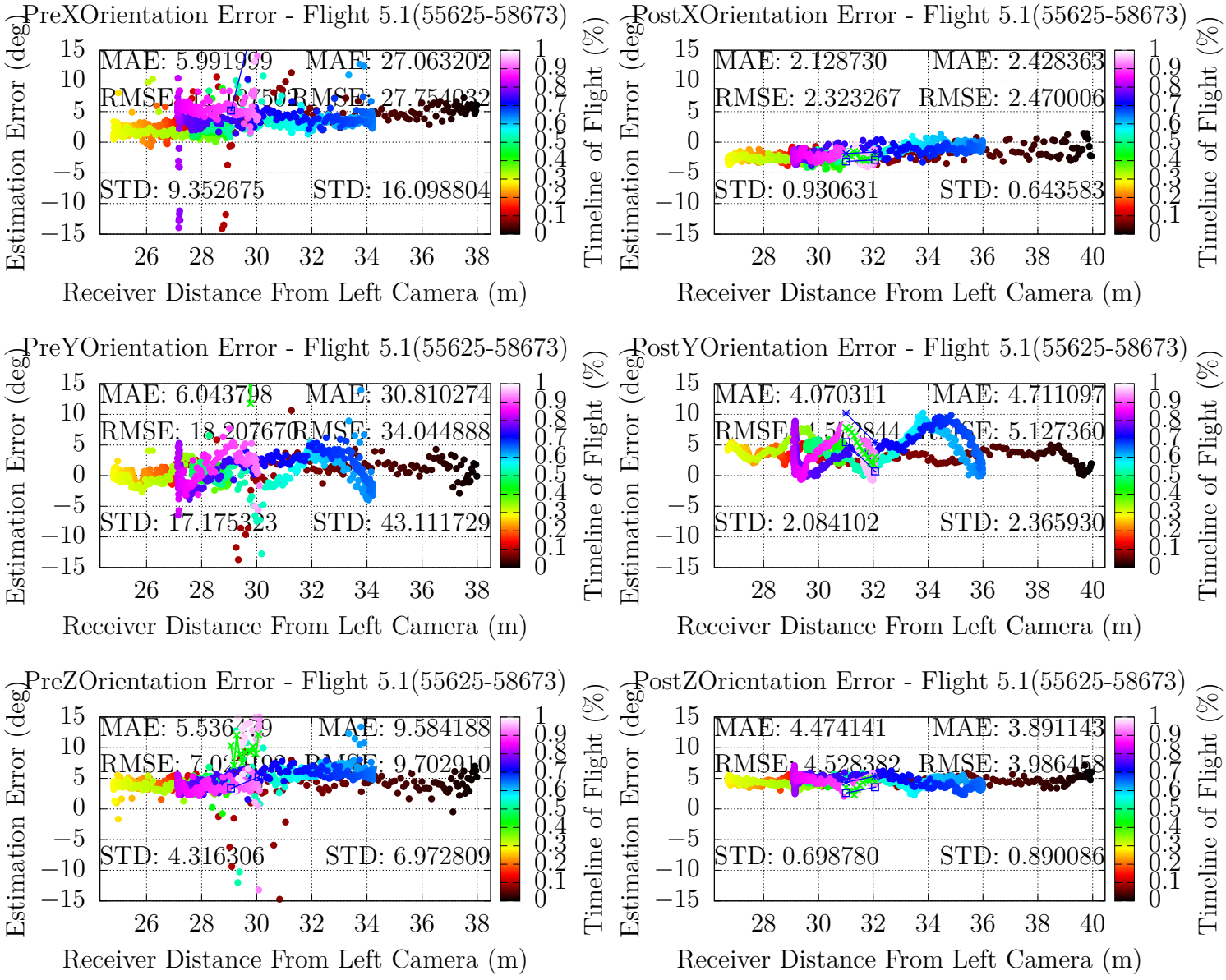


Figure 68: Flight 5.1 Comparing PreCorrections and PostCorrections for Orientation

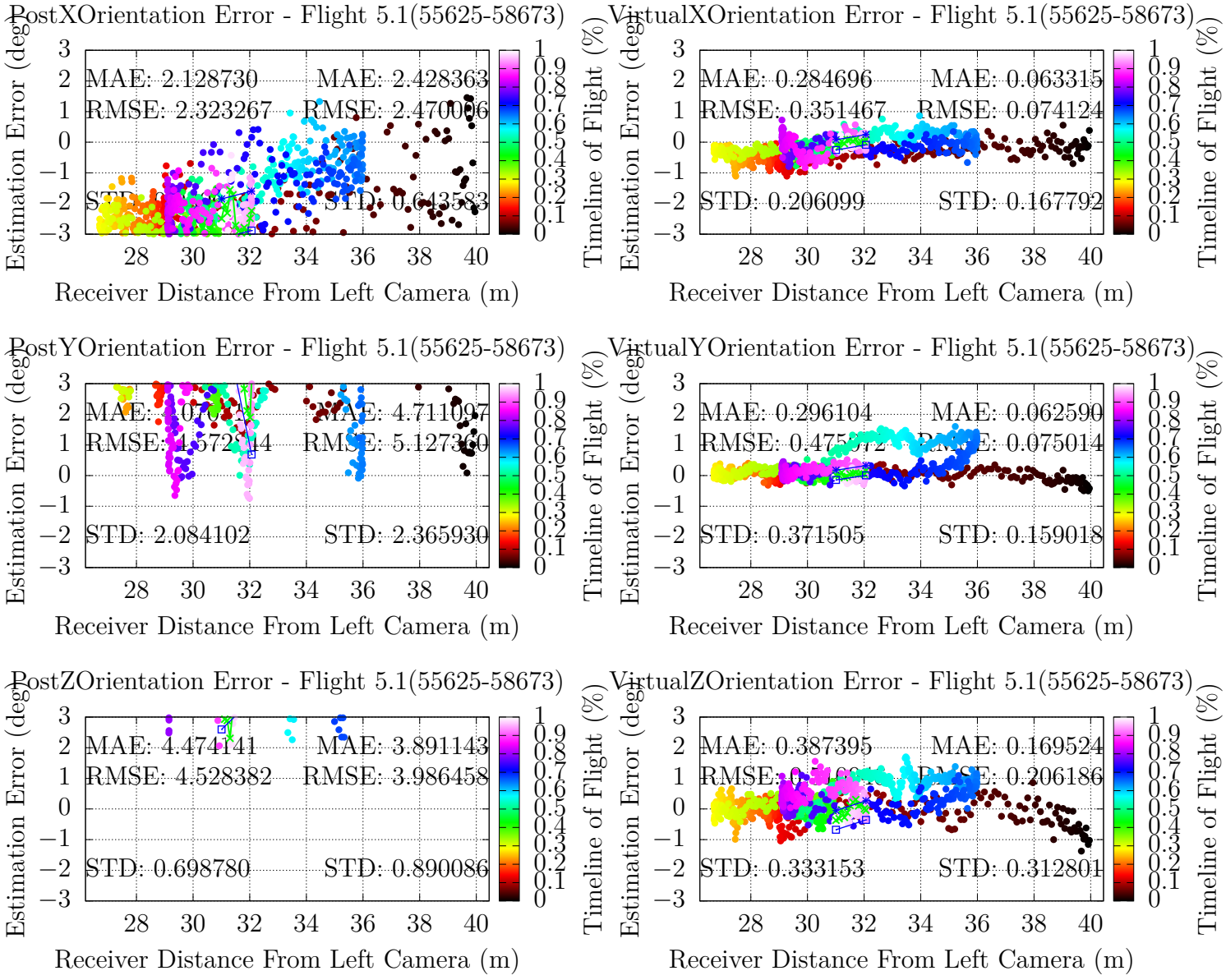


Figure 69: Flight 5.1 Comparing PostCorrections and Virtual for Orientation

4.5.2 IR

$$M1 : \begin{Bmatrix} 1101.70335040437567 & 0.00000000000000 & 540.08459668110436 \\ 0.00000000000000 & 1030.28385125849775 & 416.48093926172663 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2 : \begin{Bmatrix} 1097.55641124206613 & 0.00000000000000 & 538.24060774757015 \\ 0.00000000000000 & 1030.33557522777187 & 415.94667644611860 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 70: Flight 5.1 IR Corrected M1/M2 Camera Calibrations Matrices

Table 16: Position Error Estimation Statistics for Flight 5.1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	6.3307	4.9690	4.1310	3.3251	3.4875	4.0331
PostCorrection	3.2586	1.5322	1.4483	2.4611	1.2186	0.9973
PreToPostImprovement	0.9428	2.2430	1.8523	0.3511	1.8619	3.0440

Table 17: Orientation Error Estimation Statistics for Flight 5.1

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	27.294	48.849	16.177	12.720	28.198	8.2246
PostCorrection	31.833	10.217	7.0745	26.4762	8.1672	6.7165
PreToPostImprovement	-0.143	3.7811	1.2867	-0.520	2.4526	0.2245

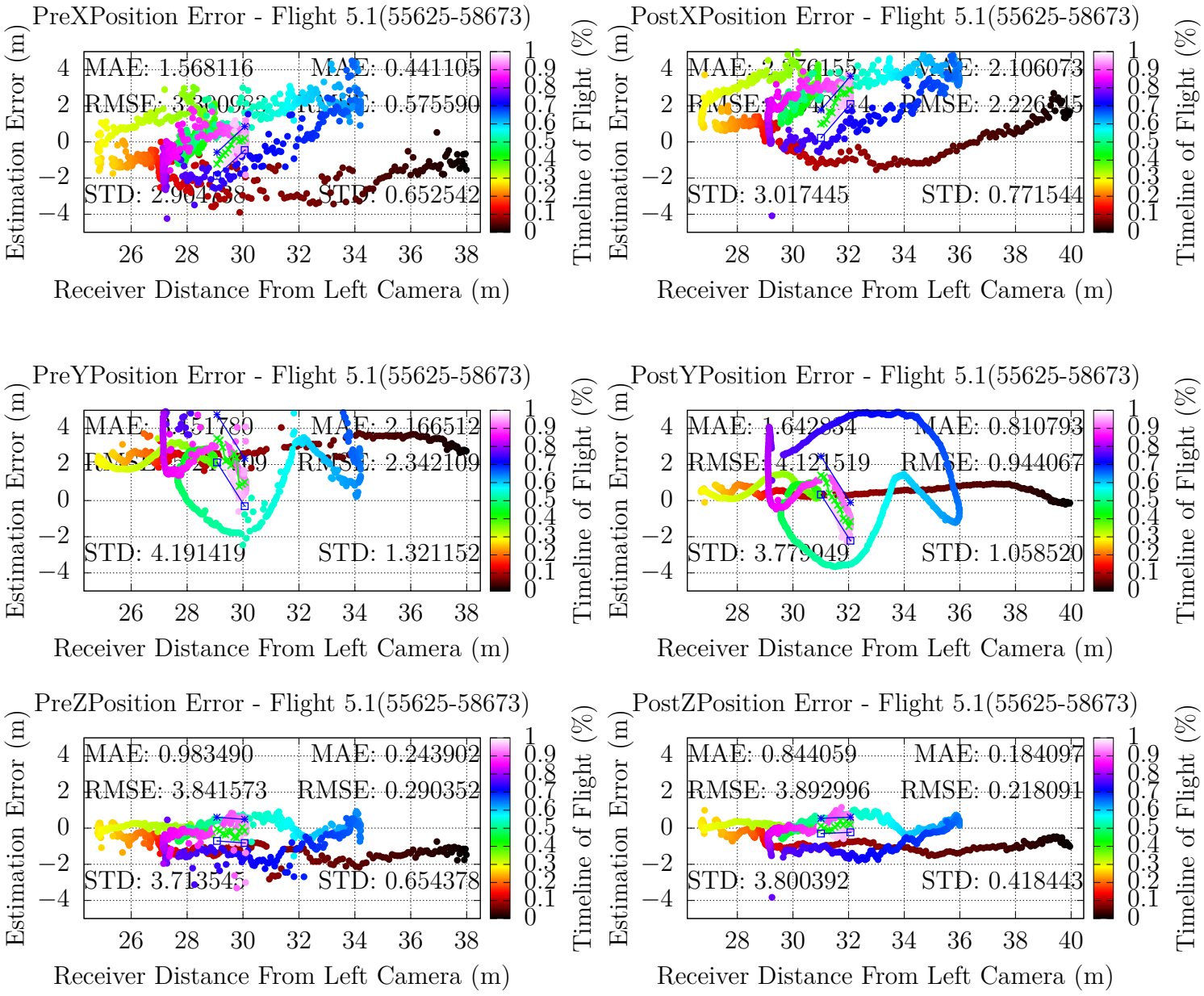


Figure 71: Flight 5.1 Comparing PreCorrections and PostCorrections for Positioning

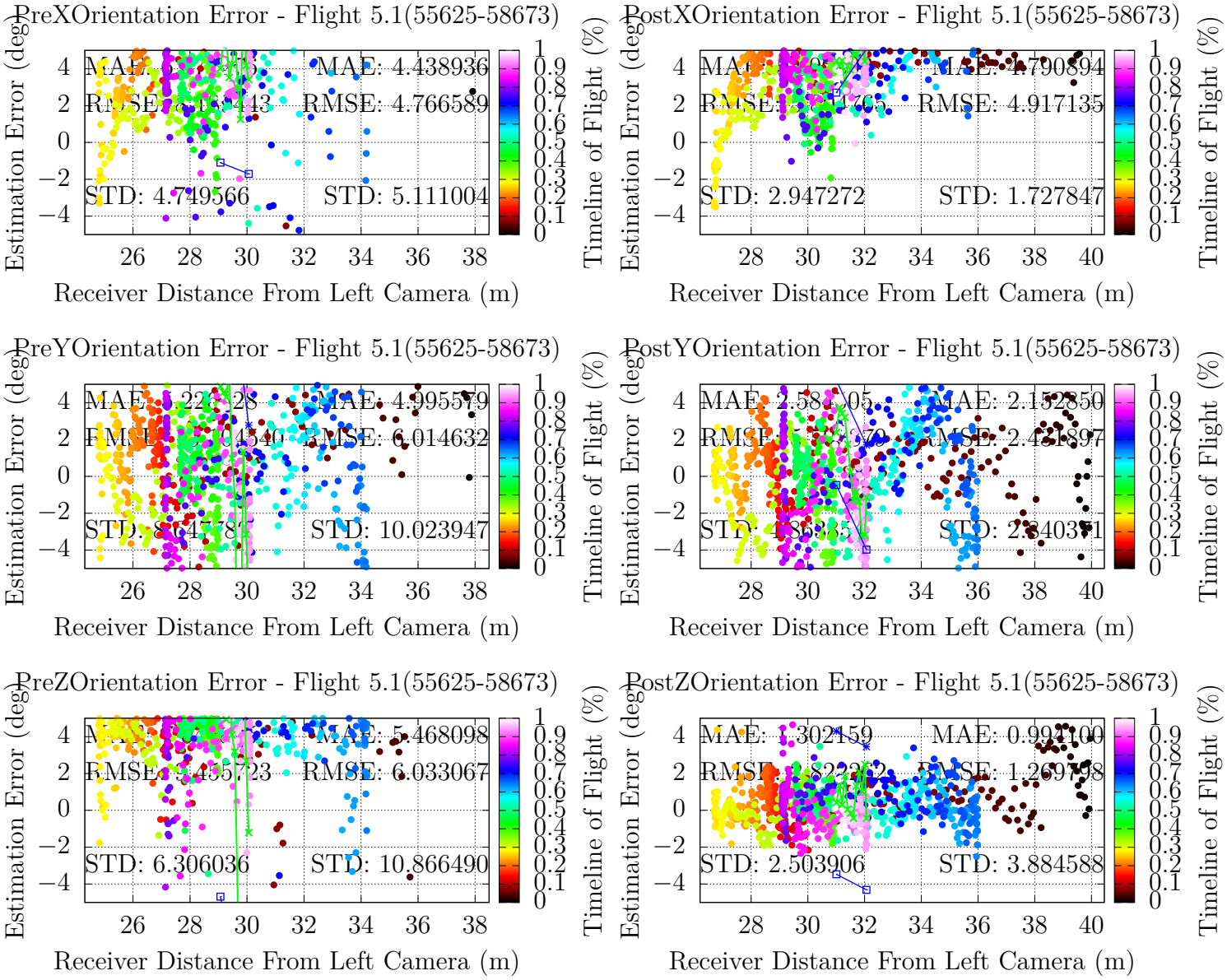


Figure 72: Flight 5.1 Comparing PreCorrections and PostCorrections for Orientation

4.6 Flight 5.2

4.6.1 EO

$$M1 : \begin{Bmatrix} 1329.69122843467449 & 0.00000000000000 & 665.31230214507218 \\ 0.00000000000000 & 1220.61664458947257 & 499.18293270488499 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2 : \begin{Bmatrix} 1327.90259830721971 & 0.00000000000000 & 712.03827622918266 \\ 0.00000000000000 & 1220.07619854918539 & 496.74734662364131 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 73: Flight 5.2 EO Corrected M1/M2 Camera Calibrations Matrices

Table 18: Position Error Estimation Statistics for Flight 5.2

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	2.9199	2.9749	1.5862	2.6895	0.8034	0.8033
PostCorrection	1.8701	0.8292	0.5353	1.4316	0.3952	0.4557
Virtual	0.2063	0.0471	0.0701	0.1083	0.0324	0.0382
PreToPostImprovement	0.5614	2.5877	1.9632	0.8787	1.0329	0.7628
PostToVirtualImprovement	8.0650	16.605	6.6362	12.219	11.198	10.929

Table 19: Orientation Error Estimation Statistics for Flight 5.2

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	9.6474	4.1387	6.2061	7.1009	4.1798	4.9190
PostCorrection	1.5188	3.1744	1.7644	1.2799	1.9867	1.4632
Virtual	0.4383	0.8432	0.6753	0.4169	0.5459	0.5445
PreToPostImprovement	5.3520	0.3038	2.5174	4.5480	1.1039	2.3618
PostToVirtualImprovement	2.4652	2.7647	1.6128	2.0700	2.6393	1.6872

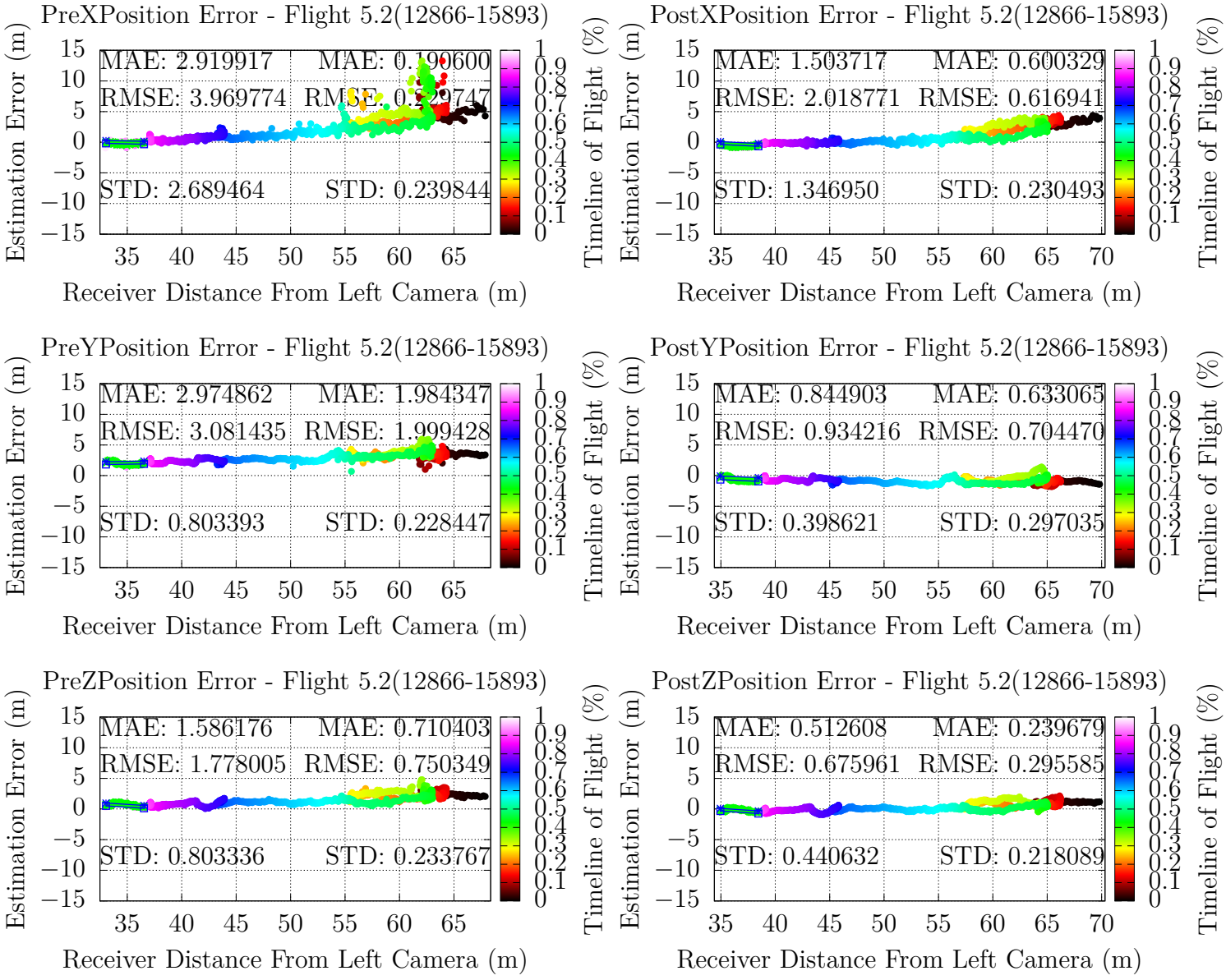


Figure 74: Flight 5.2 Comparing PreCorrections and PostCorrections for Positioning

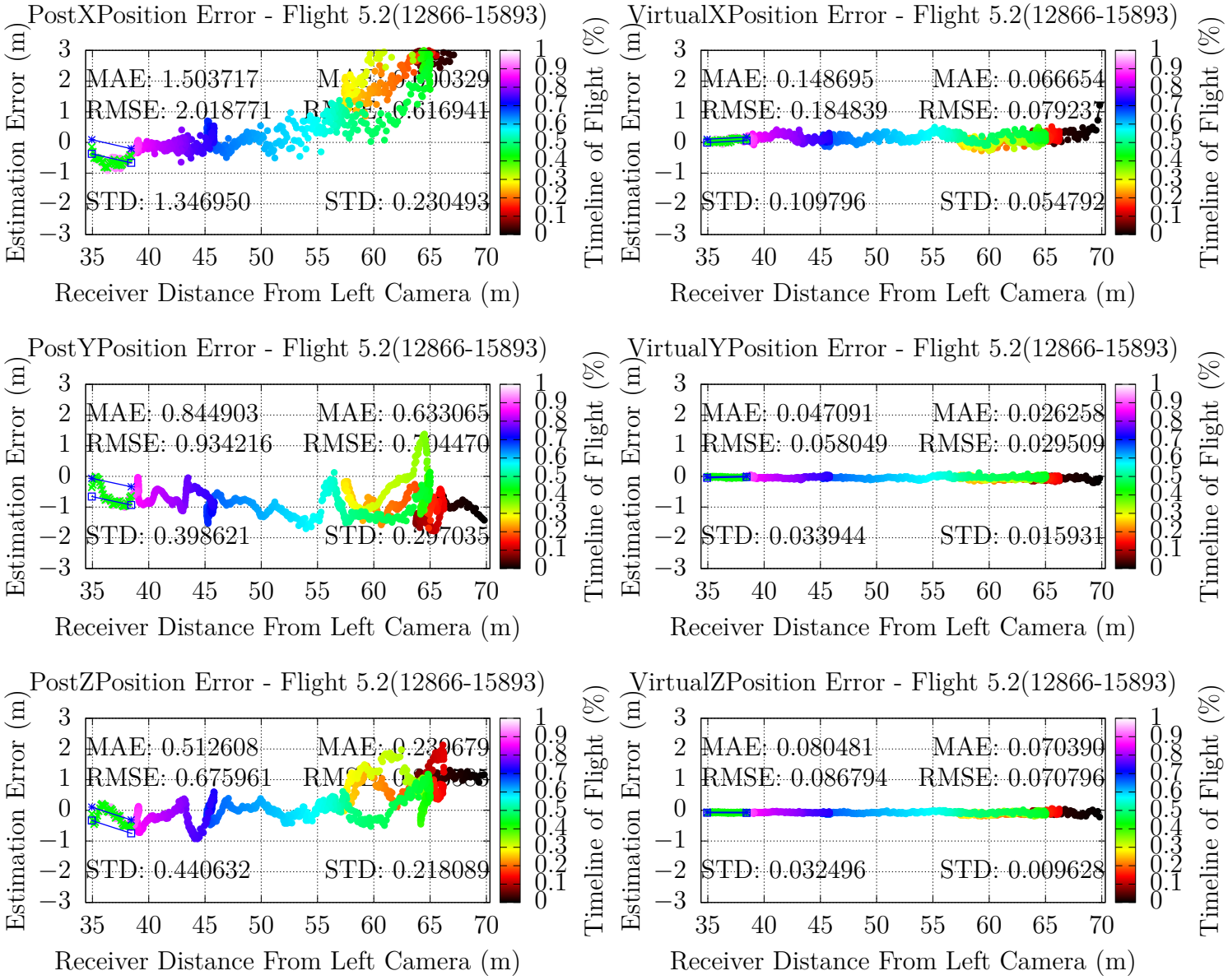


Figure 75: Flight 5.2 Comparing PostCorrections and Virtual for Positioning

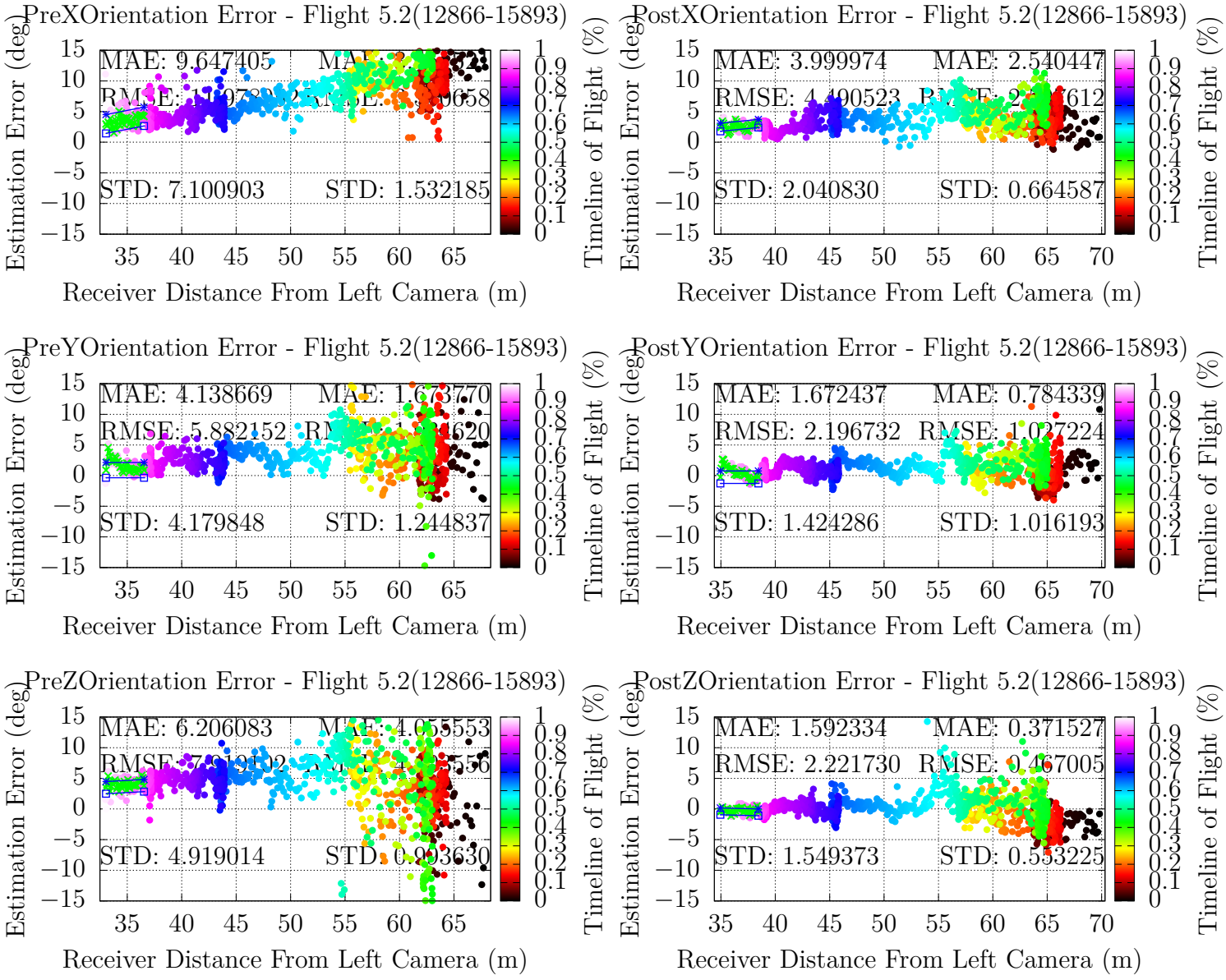


Figure 76: Flight 5.2 Comparing PreCorrections and PostCorrections for Orientation

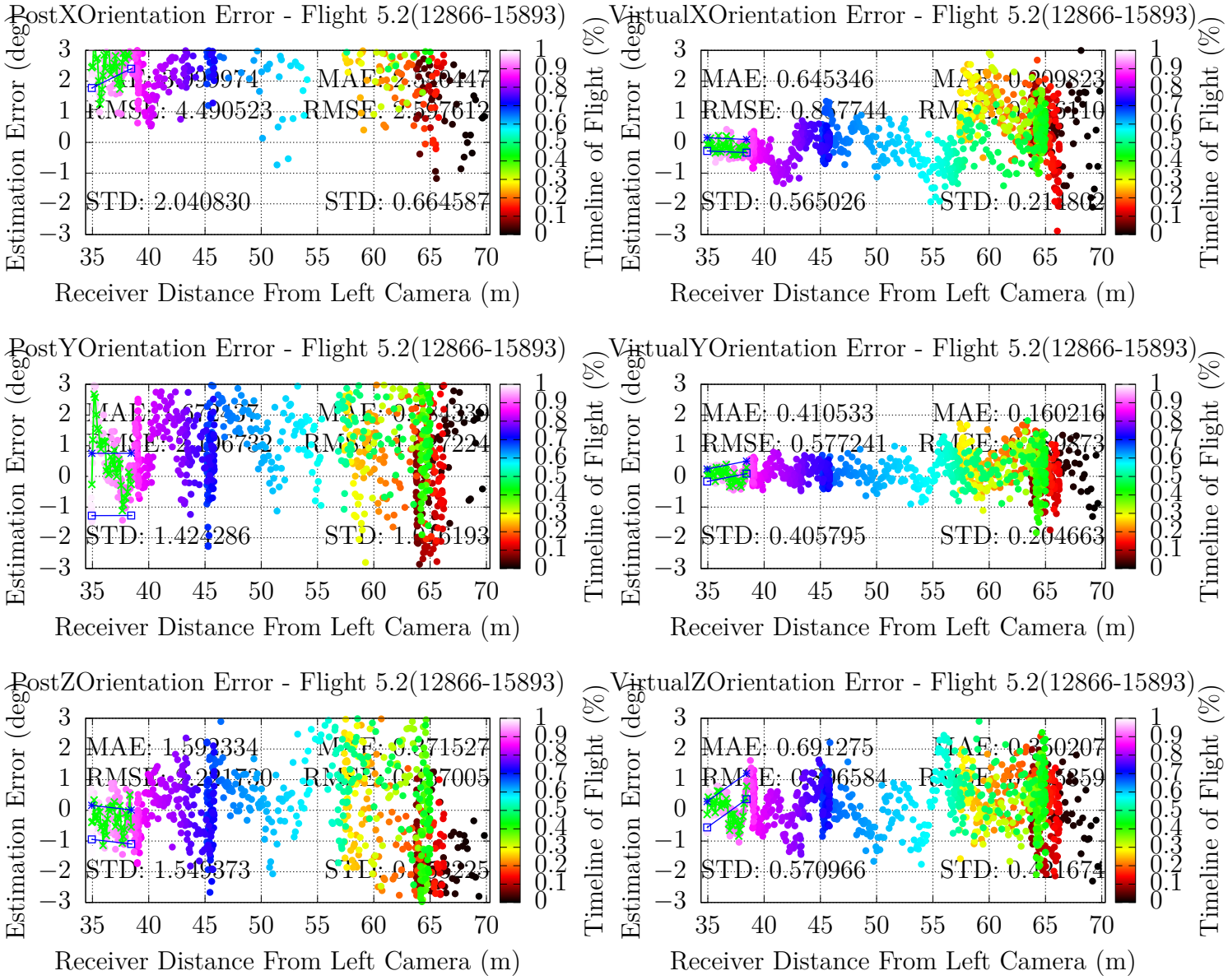


Figure 77: Flight 5.3 Comparing PostCorrections and Virtual for Orientation

4.6.2 IR

$$M1: \begin{Bmatrix} 952.70335040437567 & 0.00000000000000 & 525.08459668110436 \\ 0.00000000000000 & 939.28385125849775 & 381.48093926172663 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2: \begin{Bmatrix} 954.55641124206613 & 0.00000000000000 & 522.24060774757015 \\ 0.00000000000000 & 931.33557522777187 & 381.54667644611860 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 78: Flight 5.2 IR Corrected M1/M2 Camera Calibrations Matrices

Table 20: Position Error Estimation Statistics for Flight 5.2

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	2.9889	2.9621	1.2053	1.7832	1.1395	0.8423
PostCorrection	2.9320	0.5654	1.0148	1.9603	0.6596	0.6617
PreToPostImprovement	0.0194	4.2389	0.1877	-0.090	0.7276	0.2729

Table 21: Orientation Error Estimation Statistics for Flight 5.2

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	19.623	63.237	16.177	9.9308	32.446	8.2246
PostCorrection	34.042	9.349	5.6262	25.932	6.3359	4.8910
PreToPostImprovement	-0.424	5.7640	1.8753	-0.617	4.1209	0.6816

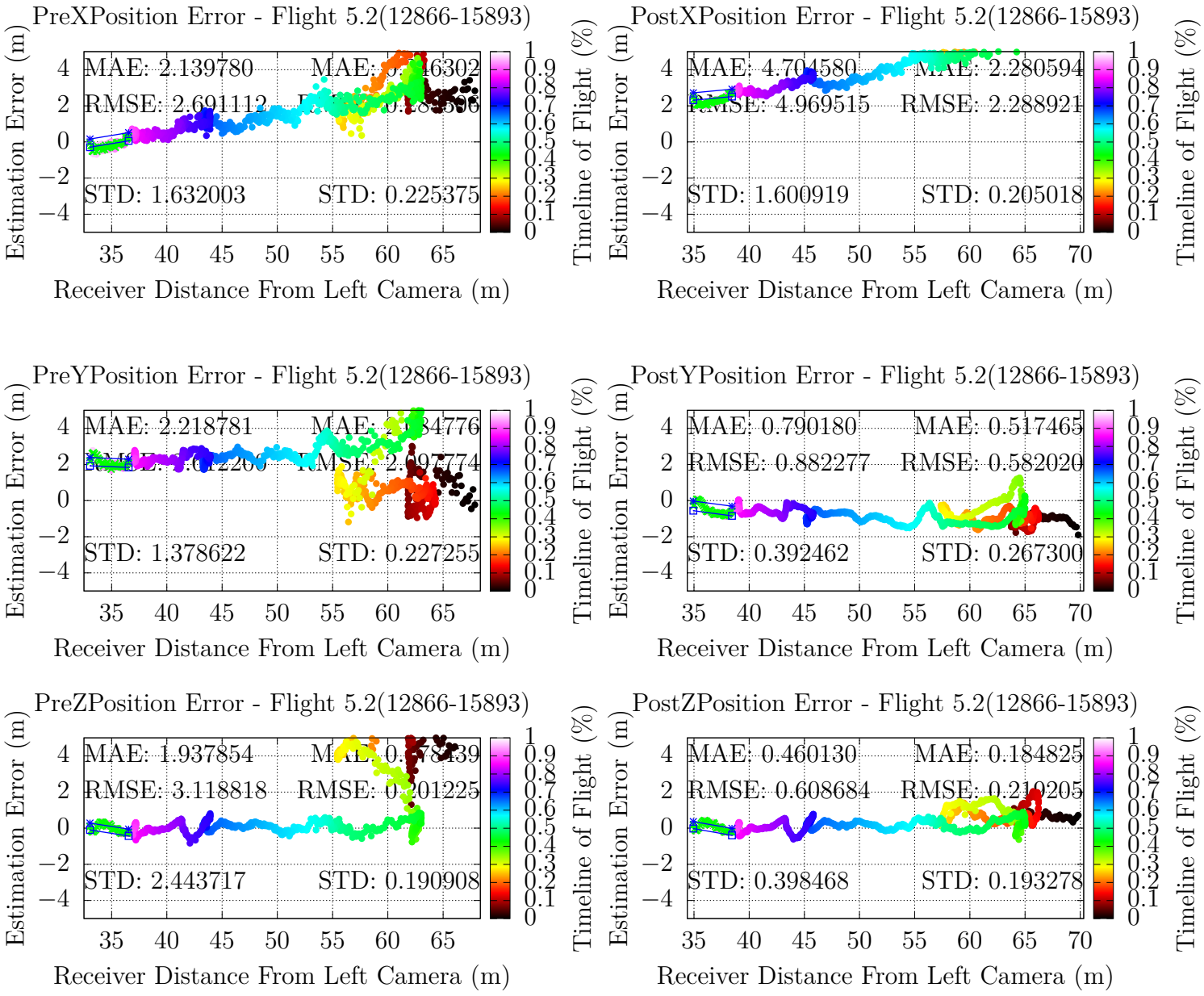


Figure 79: Flight 5.2 Comparing PreCorrections and PostCorrections for Positioning

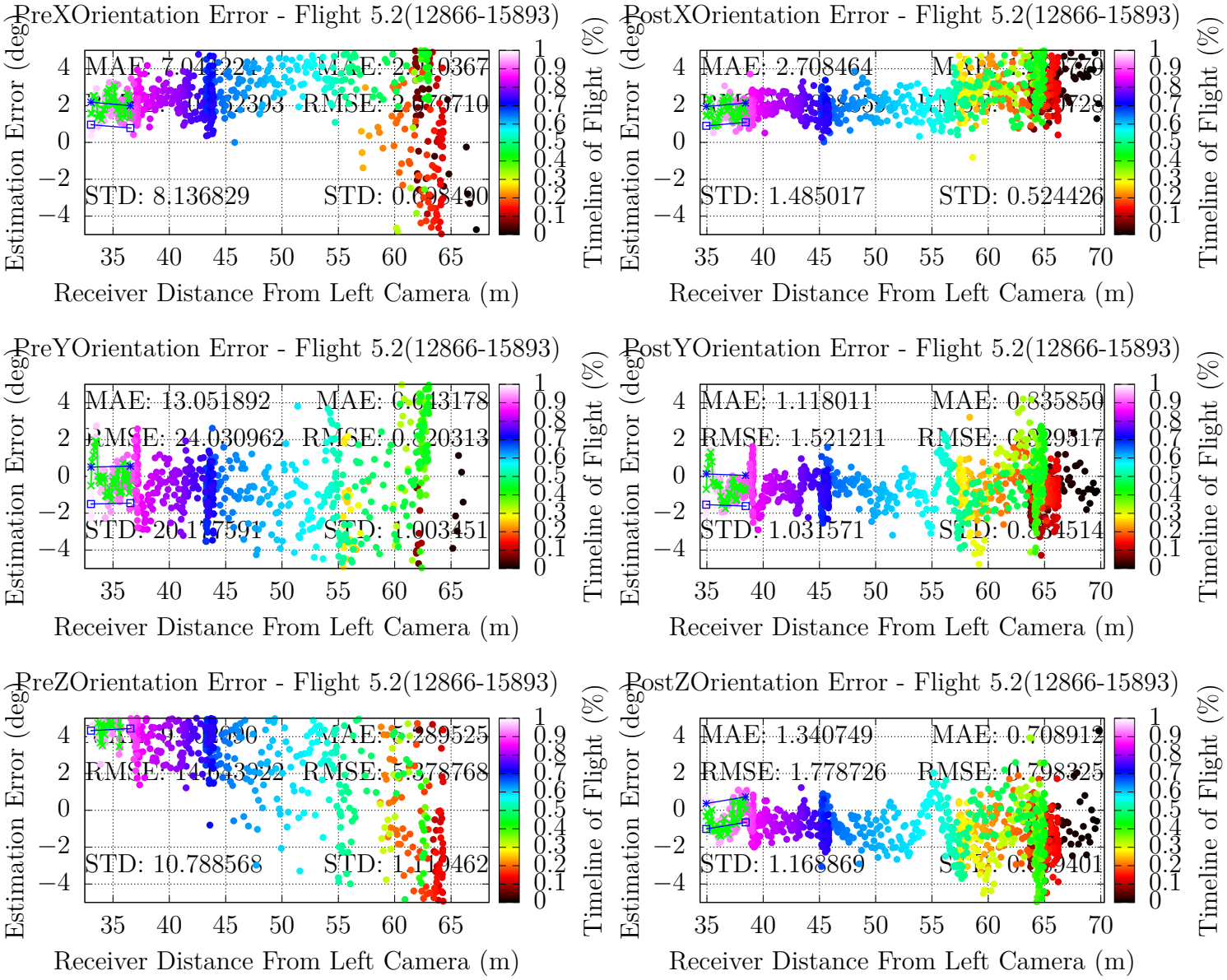


Figure 80: Flight 5.2 Comparing PreCorrections and PostCorrections for Orientation

4.7 Flight 5.3

4.7.1 EO

$$M1 : \begin{Bmatrix} 1329.69122843467449 & 0.00000000000000 & 645.31230214507218 \\ 0.00000000000000 & 1210.61664458947257 & 499.18293270488499 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2 : \begin{Bmatrix} 1327.90259830721971 & 0.00000000000000 & 692.03827622918266 \\ 0.00000000000000 & 1210.07619854918539 & 496.74734662364131 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 81: Flight 5.3 EO Corrected M1/M2 Camera Calibrations Matrices

Table 22: Position Error Estimation Statistics for Flight 5.3

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	1.2850	1.8235	1.1953	1.4244	0.7483	0.6909
PostCorrection	0.8706	0.6991	0.4211	1.2174	0.4677	0.4329
Virtual	0.3411	0.0533	0.0762	1.0093	0.1512	0.1409
PreToPostImprovement	0.4760	1.6084	1.8385	0.1700	0.5999	0.5960
PostToVirtualImprovement	1.5523	12.116	4.5263	0.2062	2.0933	2.0724

Table 23: Orientation Error Estimation Statistics for Flight 5.3

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	5.0271	2.8415	5.5625	3.7812	2.2558	4.0482
PostCorrection	1.5815	2.3272	1.1939	1.2469	2.6133	2.1941
Virtual	0.6167	1.0311	0.8389	2.1621	3.9579	2.9776
PreToPostImprovement	2.1790	0.2210	3.6591	2.0325	-0.137	0.8450
PostToVirtualImprovement	1.5645	1.2570	0.4232	-0.423	-0.340	-0.263

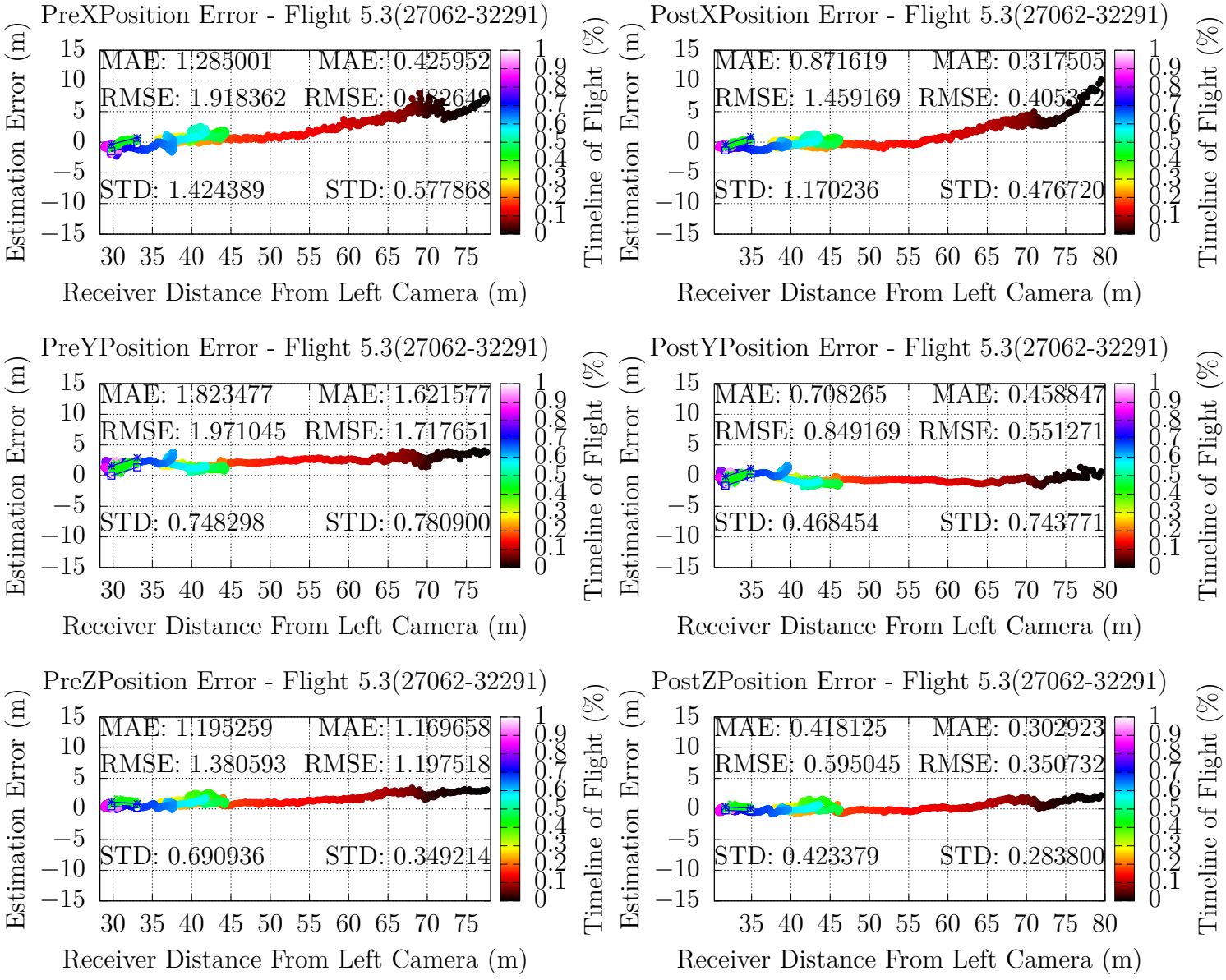


Figure 82: Flight 5.3 Comparing PreCorrections and PostCorrections for Positioning

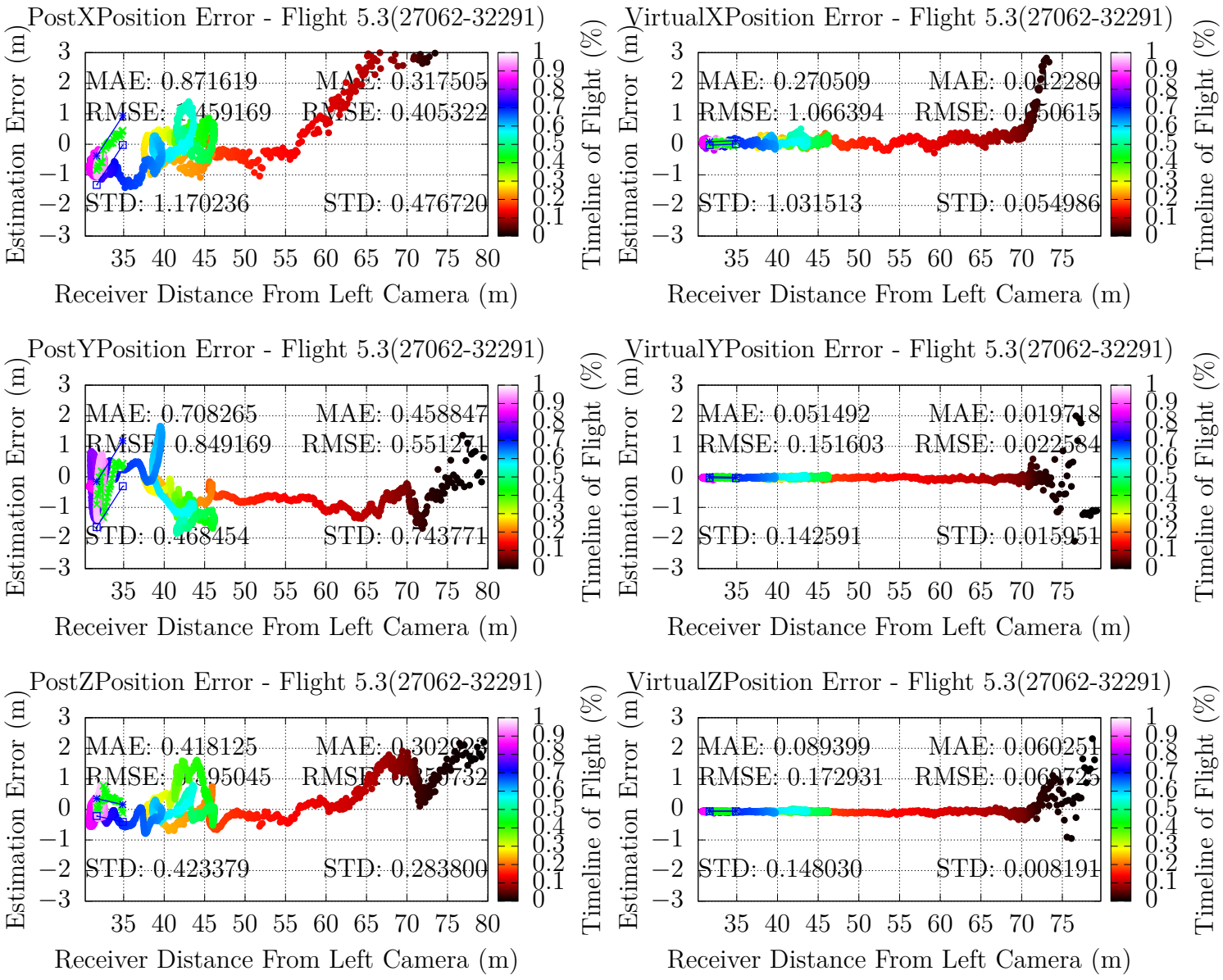


Figure 83: Flight 5.3 Comparing PostCorrections and Virtual for Positioning

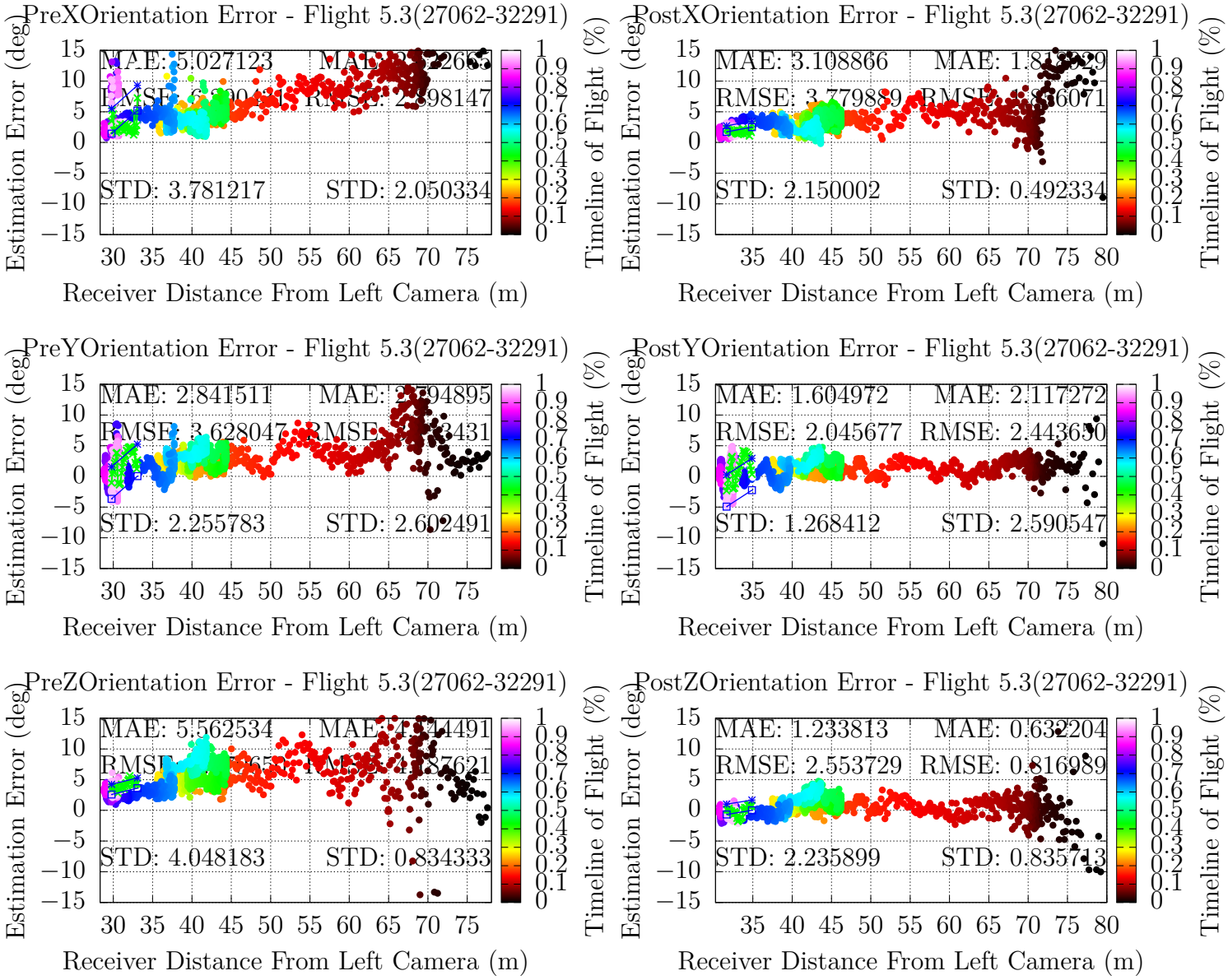


Figure 84: Flight 5.3 Comparing PreCorrections and PostCorrections for Orientation

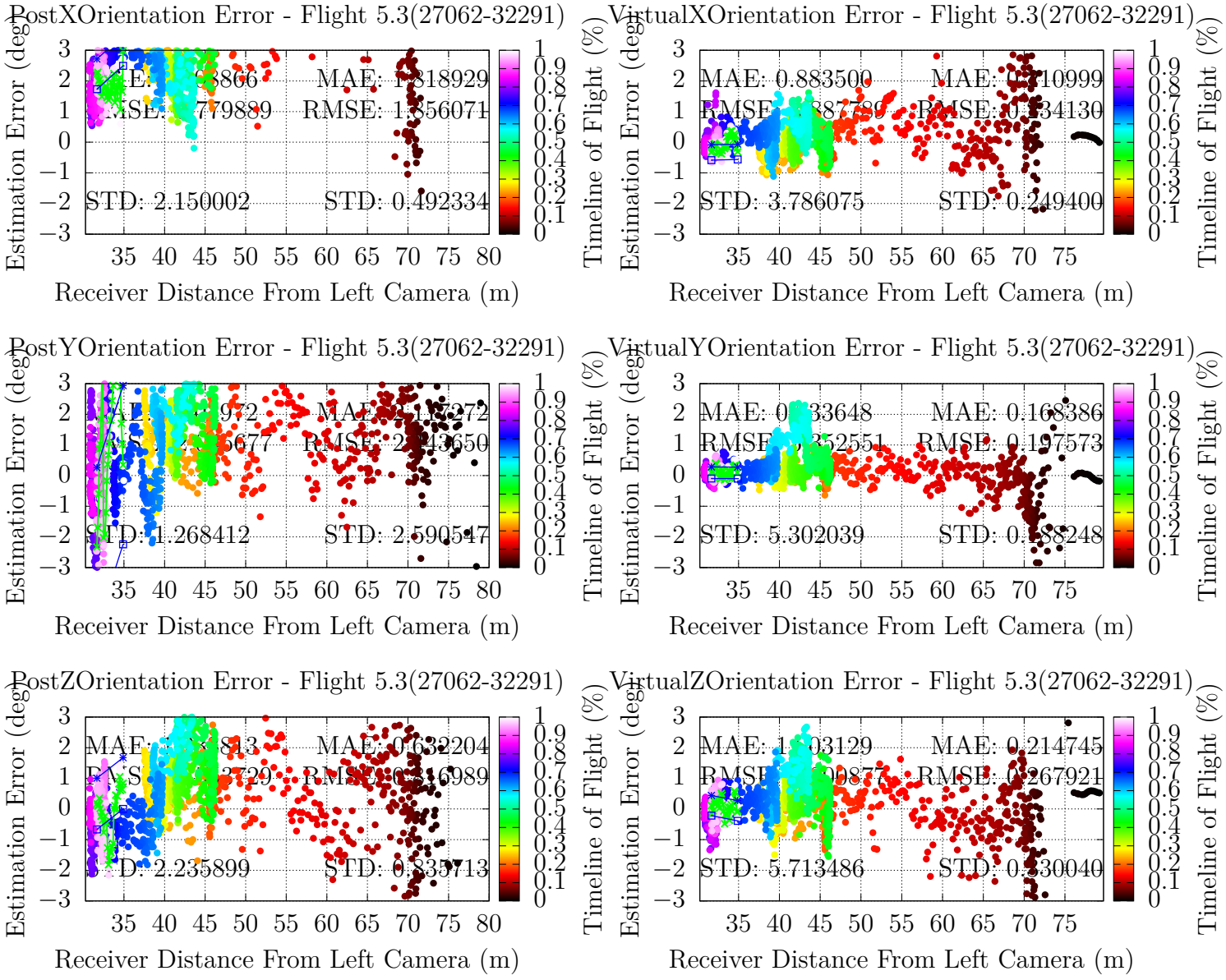


Figure 85: Flight 5.3 Comparing PostCorrections and Virtual for Orientation

4.7.2 IR

$$M1: \begin{Bmatrix} 962.70335040437567 & 0.00000000000000 & 521.08459668110436 \\ 0.00000000000000 & 950.28385125849775 & 375.98093926172663 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2: \begin{Bmatrix} 964.55641124206613 & 0.00000000000000 & 518.54060774757015 \\ 0.00000000000000 & 955.33557522777187 & 375.94667644611860 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 86: Flight 5.3 IR Corrected M1/M2 Camera Calibrations Matrices

Table 24: Position Error Estimation Statistics for Flight 5.3

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	4.2587	2.1319	1.6986	2.501	1.3016	1.3411
PostCorrection	4.1141	1.2240	1.4942	3.6542	0.9692	1.4987
PreToPostImprovement	0.0351	0.7417	0.1368	-0.316	0.3429	-0.105

Table 25: Orientation Error Estimation Statistics for Flight 5.3

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	17.864	67.197	15.557	9.0916	29.719	7.0882
PostCorrection	15.868	11.337	3.1616	16.188	6.6189	3.3956
PreToPostImprovement	0.1258	4.9272	3.9206	-0.438	3.4900	1.0875

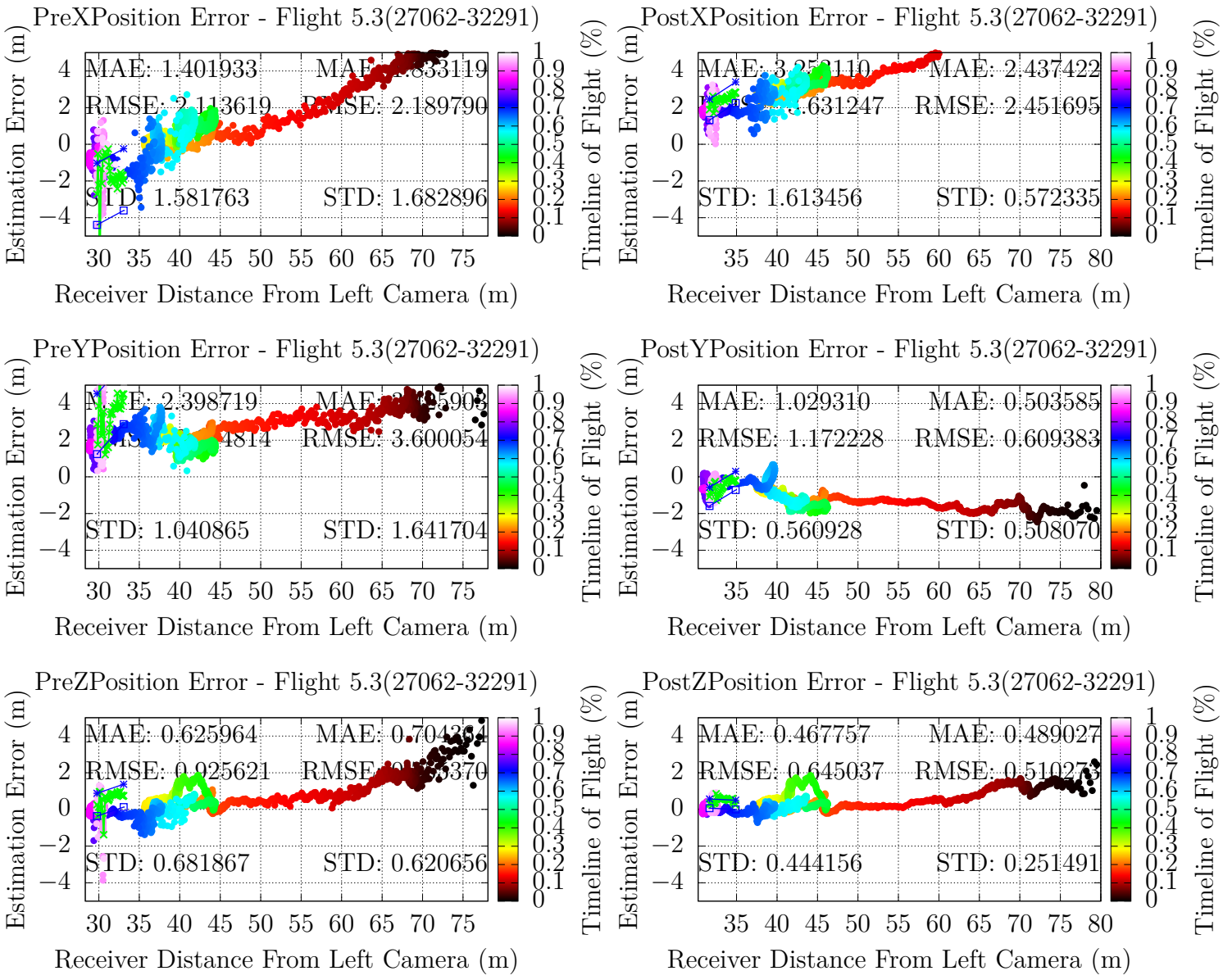


Figure 87: Flight 5.3 Comparing PreCorrections and PostCorrections for Positioning

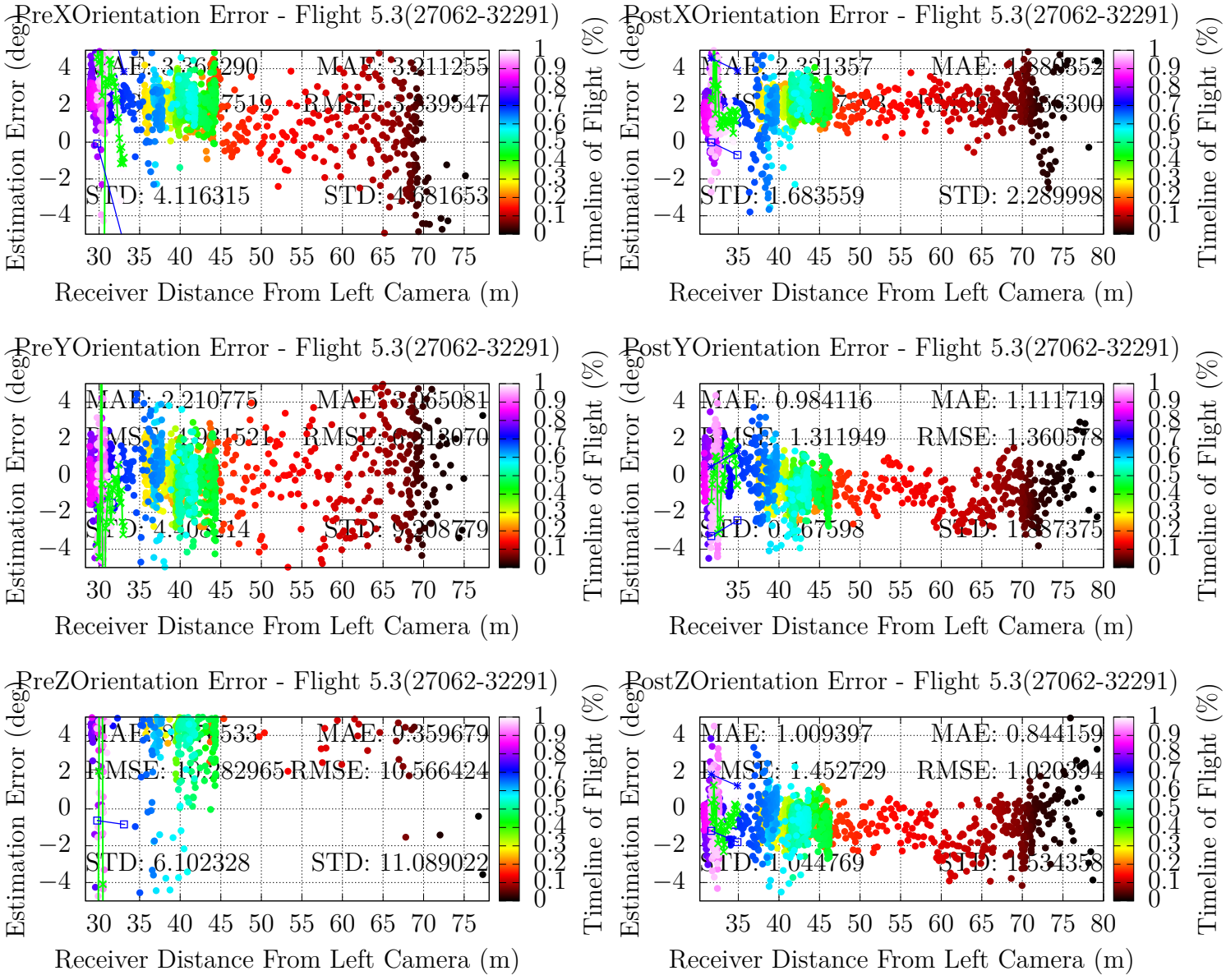


Figure 88: Flight 5.3 Comparing PreCorrections and PostCorrections for Orientation

4.8 Flight 6

4.8.1 EO

$$M1 : \begin{Bmatrix} 1269.69122843467449 & 0.00000000000000 & 635.31230214507218 \\ 0.00000000000000 & 1230.61664458947257 & 501.18293270488499 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2 : \begin{Bmatrix} 1267.90259830721971 & 0.00000000000000 & 682.03827622918266 \\ 0.00000000000000 & 1230.07619854918539 & 496.74734662364131 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 89: Flight 6 EO Corrected M1/M2 Camera Calibrations Matrices

Table 26: Position Error Estimation Statistics for Flight 6

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	1.7535	2.5807	0.3343	1.9811	0.7865	0.2381
PostCorrection	1.3181	0.3455	0.4144	2.3520	0.4118	0.3129
Virtual	0.6096	0.0940	0.0879	1.3924	0.2231	0.1039
PreToPostImprovement	0.3303	6.4695	-0.193	-0.158	0.9099	-0.239
PostToVirtualImprovement	1.1622	2.6755	3.7145	0.6892	0.8458	2.0116

Table 27: Orientation Error Estimation Statistics for Flight 6

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	10.651	1.6594	2.5947	5.3852	1.6631	3.4781
PostCorrection	2.0495	5.3921	2.2608	8.6047	4.4724	8.8495
Virtual	1.4586	1.2827	2.0156	9.0140	2.8604	9.6786
PreToPostImprovement	4.1969	-0.692	0.1477	-0.374	-0.628	-0.607
PostToVirtualImprovement	0.4051	3.2037	0.1217	-0.045	0.5636	-0.086

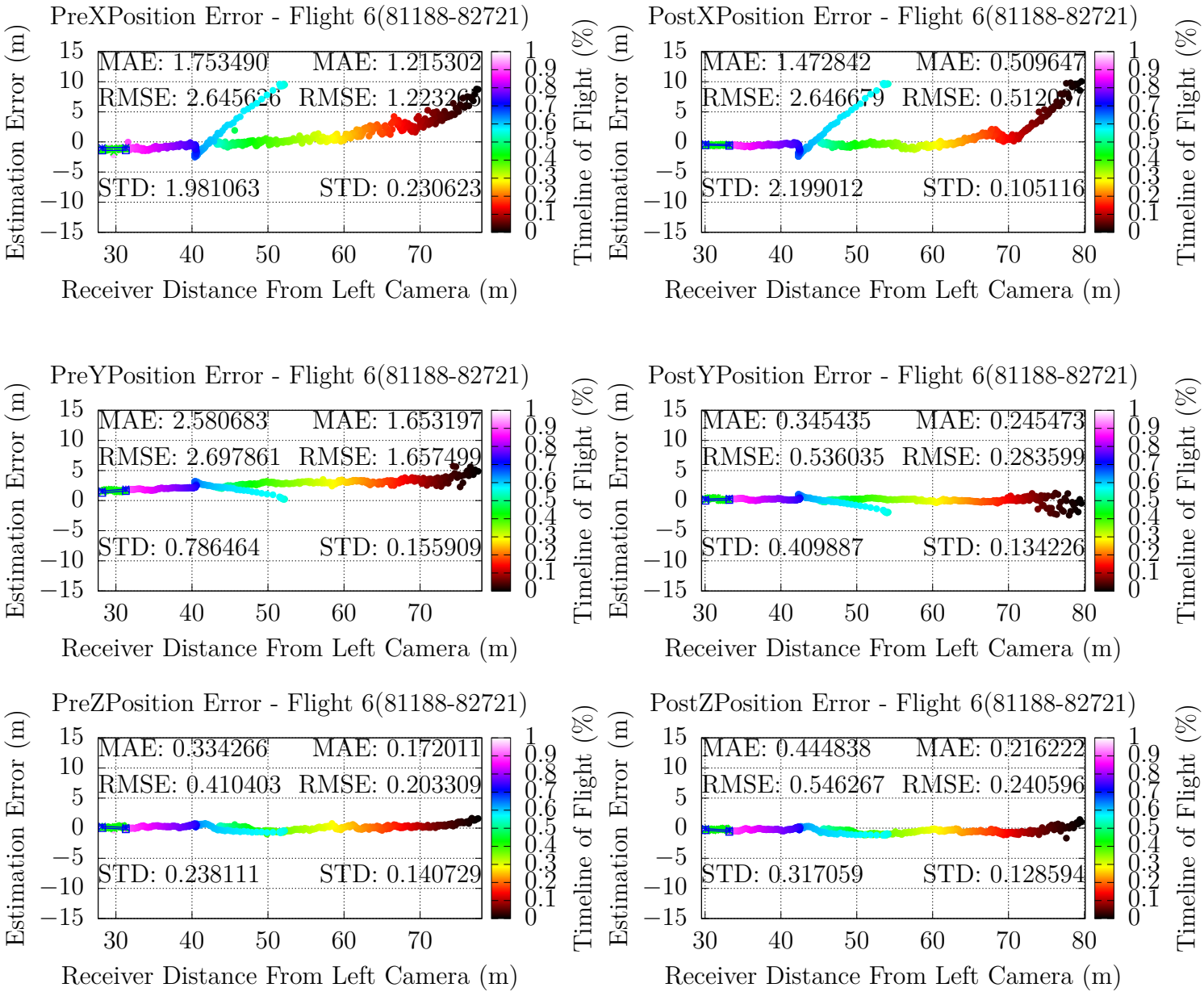


Figure 90: Flight 6 Comparing PreCorrections and PostCorrections for Positioning

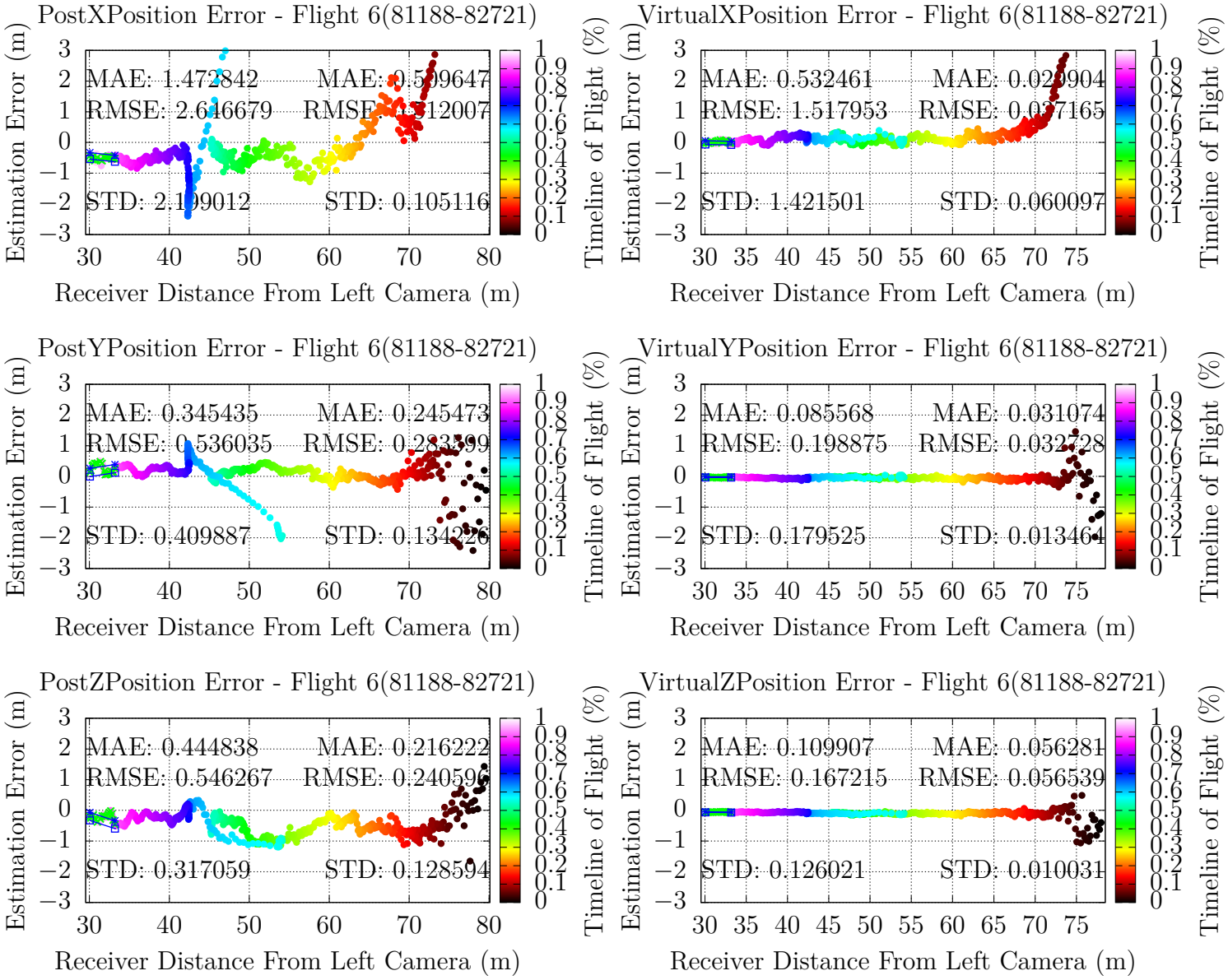


Figure 91: Flight 6 Comparing PostCorrections and Virtual for Positioning

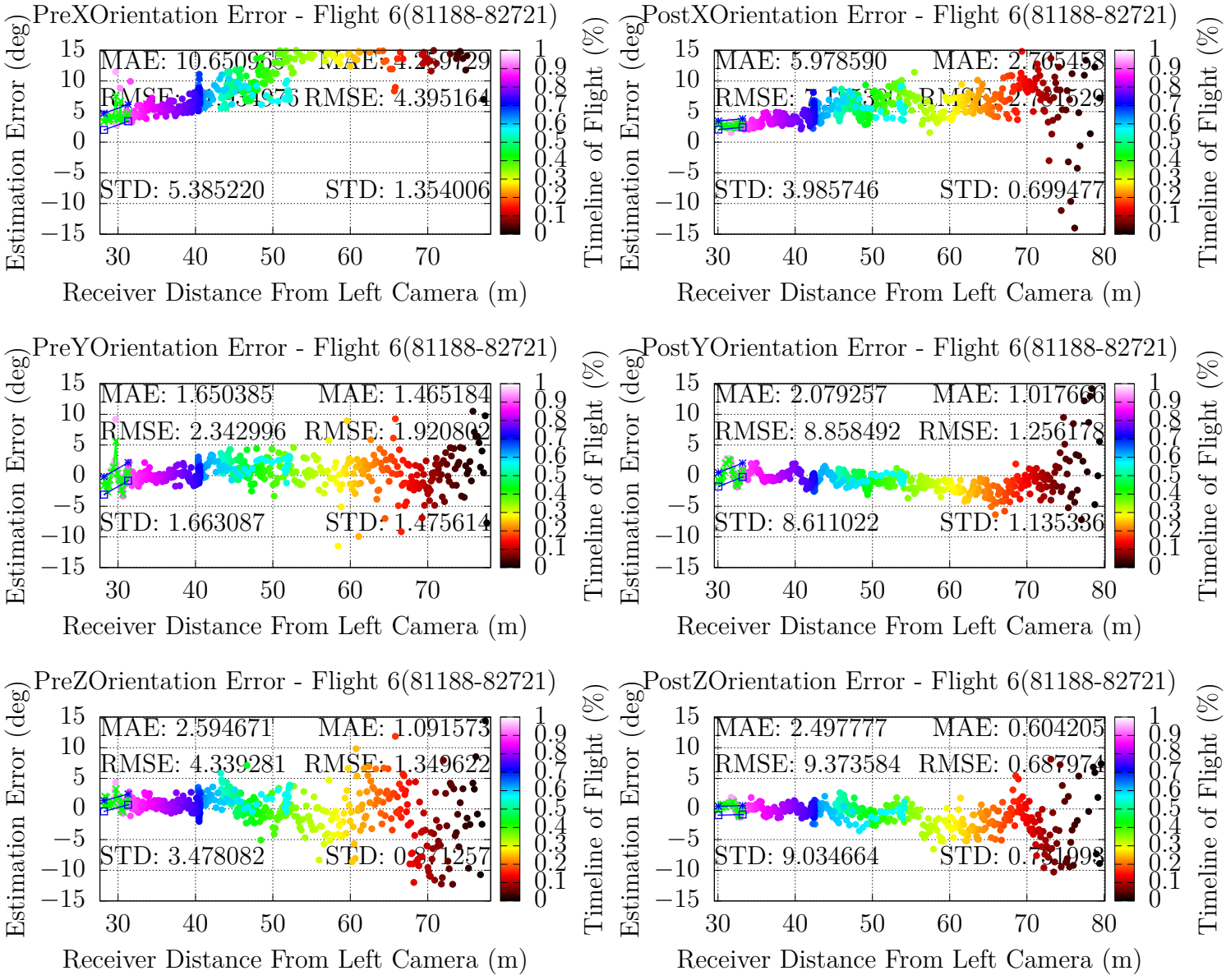


Figure 92: Flight 6 Comparing PreCorrections and PostCorrections for Orientation

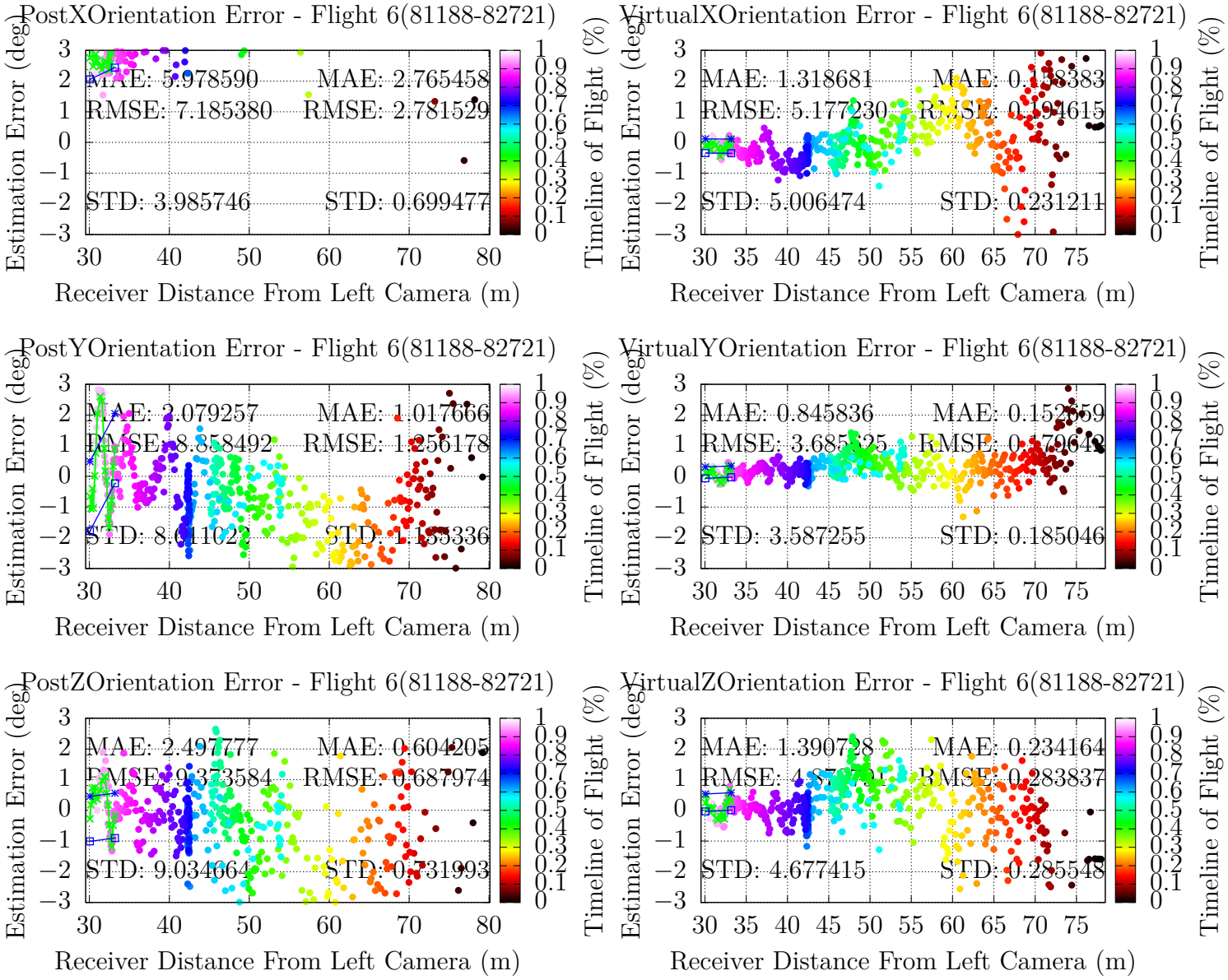


Figure 93: Flight 6 Comparing PostCorrections and Virtual for Orientation

4.8.2 IR

$$M1: \begin{Bmatrix} 963.90335040437567 & 0.00000000000000 & 517.98459668110436 \\ 0.00000000000000 & 960.28385125849775 & 395.08093926172663 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

$$M2: \begin{Bmatrix} 964.55641124206613 & 0.00000000000000 & 515.24060774757015 \\ 0.00000000000000 & 960.33557522777187 & 387.94667644611860 \\ 0.00000000000000 & 0.00000000000000 & 1.00000000000000 \end{Bmatrix}$$

Figure 94: Flight 6 IR Corrected M1/M2 Camera Calibrations Matrices

Table 28: Position Error Estimation Statistics for Flight 6

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	13.233	4.5211	4.1767	10.2328	3.9233	3.8468
PostCorrection	13.110	4.7743	3.9442	12.834	5.3804	4.2546
PreToPostImprovement	0.0094	-0.053	0.0589	-0.203	-0.271	-0.096

Table 29: Orientation Error Estimation Statistics for Flight 6

	XMae	YMae	ZMae	XStdDev	YStdDev	ZStdDev
PreCorrection	24.686	43.928	18.994	16.003	29.305	16.809
PostCorrection	43.005	20.095	8.846	29.662	13.875	9.8841
PreToPostImprovement	-0.426	1.1860	1.1472	-0.460	1.1121	0.7006

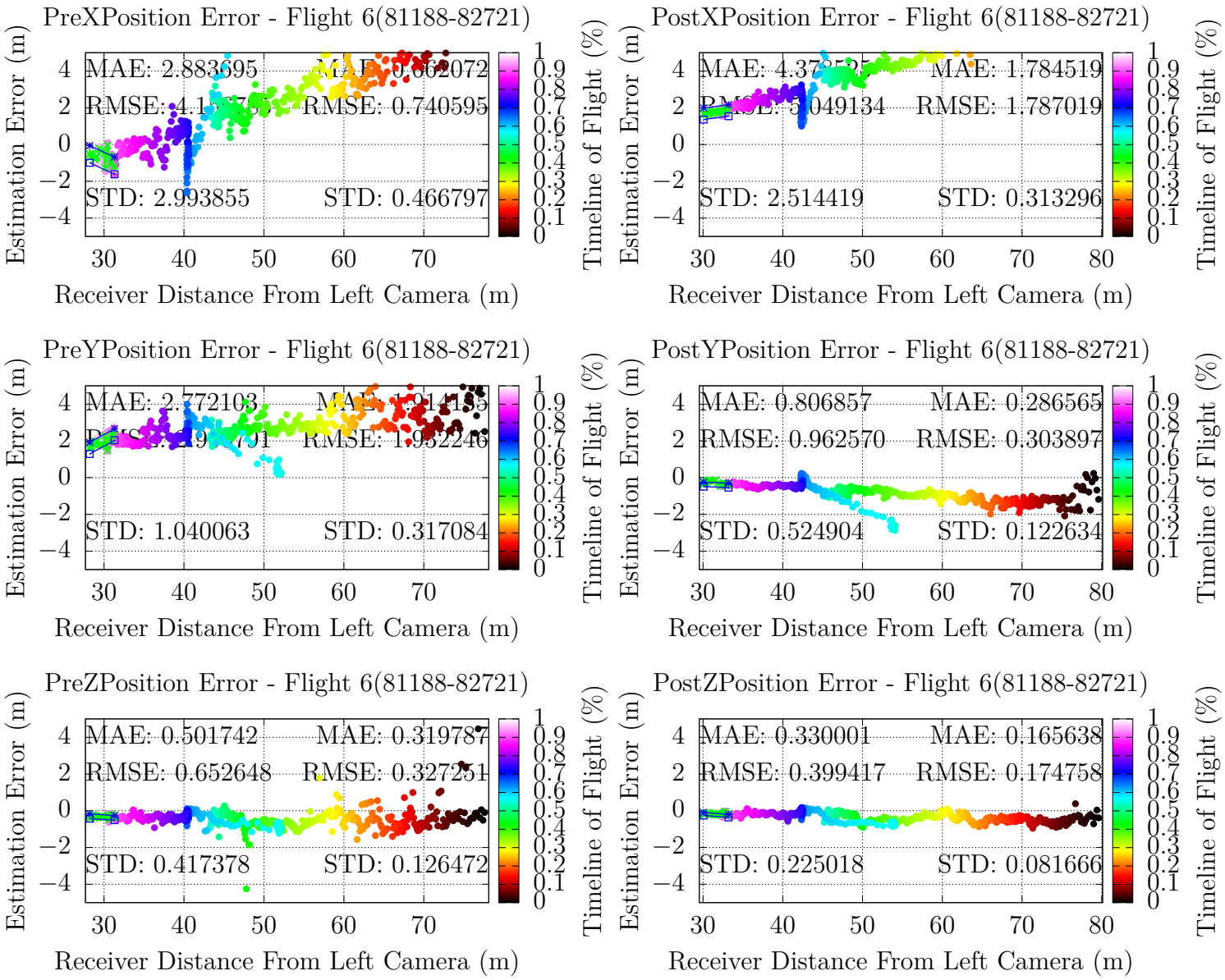


Figure 95: Flight 6 Comparing PreCorrections and PostCorrections for Positioning

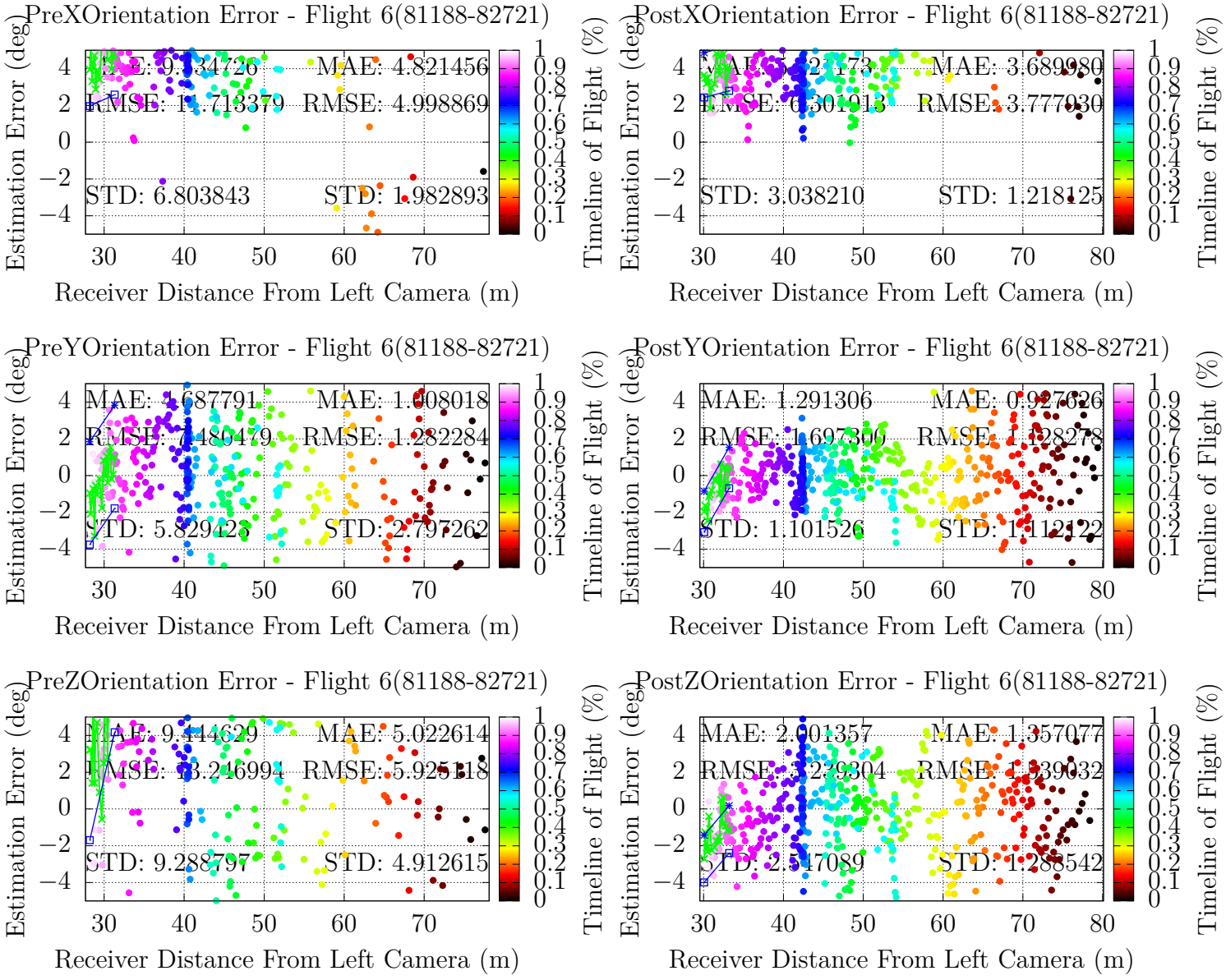


Figure 96: Flight 6 Comparing PreCorrections and PostCorrections for Orientation

4.9 PreCorrection vs. PostCorrection Analysis

When correcting for camera calibrations, the focus was correcting the points that lied around the refueling point of 30 meters. Including outliers, the average improvement after physical camera corrections is 3.24x better. Without outliers, the average improvement falls around 2x better. Although improvements were generally made to the data sets, some of the data sets did not improve. Table 26 illustrates how some of the errors increased instead of decreased for Flight 6. However, although the error increased trivially in the Z axis, the amount of error decreased substantially in the X and Y axes. Furthermore, some figures, such as Figure 51, illustrate the what looks to be the same exact graph. However, the PreCorrection data is actually offset with more error while the PostCorrection data more aligns along the 0 axis of the graph. Overall, the majority of the flights' errors decreased and now provide a better comparison to the virtual data sets.

4.10 IR Data Results

Although the IR data sets cannot be compared to virtual simulations at this time, the IR results can be analyzed to further discuss the viability of utilizing IR cameras as a source for stereo vision Automated Aerial Refueling (AAR) operations. Similar to the EO pre vs post results, the IR cameras perform better when correcting for the hardware mounting errors, correcting camera calibrations, and removing the outliers. On average, the post-correction data performs about 2.5x better than the pre-correction data; however, as expected, the EO cameras outperform the IR cameras due to the resolution of the cameras. A definitive comparison cannot be made between the EO and IR results due to the resolution differences as well as the type of cameras.

4.11 Physical vs. Virtual Analysis

With an exception to the orientation comparisons, the virtual data always outperforms the post correction physical data sets. On average, the increase in performance from the post data to the virtual data is 9.16x, inclusive of outliers. Without outliers, the virtual data is expected to perform 2-6x better. The exact improvement relies on how well the cameras are calibrated, environmental variables such as visibility of the receiver, and how well the features match during stereo vision.

4.12 Summary of Results

The results between the pre vs. post data sets are as expected. Due to the interconnection between camera calibrations, not all axes are expected to improve while modifying camera calibrations; however, the majority of them improve and reduce the amount of error in the pose estimations. The results between the physical (post) and virtual data sets also perform as expected. The virtual data performs near perfect compared to that of the physical data, which is expected. But, the differences between the physical and virtual data performs differently for each flights. Thus, it is difficult to provide an exact amount of difference, and instead only provide a range of errors when performing experiments in the physical and virtual world.

V. Conclusions

Current Automated Aerial Refueling (AAR) research efforts at The Air Force Institute of Technology (AFIT) utilize a 3D virtual engine to perform low-cost, rapid experiments; however, the results from the experiments have not yet been proven to equivalently compare to the physical world. The results from the experiments in March in combination with the equivalent virtual experiments provide a foundation of how well the virtual world approximates the physical world. Generally speaking, the virtual world provides 2-6x better results than that of the physical world; however, the same experiments could be further improved with a more refined set of experiments and camera calibrations from the physical world. Additionally, due to the current unavailability of an Infrared (IR) camera in the virtual world, virtual and physical IR data could not be compared. Potential error reductions could be provided with both Electro-Optical (EO) and IR parallel computations.

Although the approximations vary across the flights, the accuracy of the post estimations between the physical and virtual world can be narrowed down to the virtual world performing 2-6x better. The results from pre-correction to post-correction as well as post-correction to virtual all perform as expected. Additionally, this research further develops the notion that precise, correct camera calibrations significantly alter the error in the relative pose estimations. In addition to producing a comparison between physical vs. virtual data sets, the experiments conducted here have established a framework to compare and validate various AAR technologies from the physical to virtual world.

5.1 Future Work

The difficulty of determining the exact value for approximating the physical world from the virtual world lies with the camera calibrations. All of the camera calibrations had to be tested for these experiments and comparisons to work. Due to the interconnection between the camera calibration variables, it is incredibly difficult to perfectly correct the camera calibrations. Thus, with better initial camera calibrations, a more defined approximation value between the physical and virtual world could be calculated.

The exact value for the rotation and translation error from the lever arm offsets also play a factor into the errors. In the Results section, for example Figure 66, it can be seen that multiple error values are recorded at the same distance. If the correct lever arm offsets were recorded and calculated, the error values would ideally be the same at a given distance, instead of the sporadic behavior as shown. A more consistent set of behavior would allow for more consistent pose estimations and result in a better idea of the exact approximation from the physical to the virtual world.

To further enhance the discussion of IR imagery data sets, a virtual replica of the IR cameras could be created. This would be able to further analyze the IR imagery in the Results chapter. Additionally, this would provide a great comparison to determine how well the virtual IR compare to that of the physical IR cameras as well as compare the virtual IR improvements from the physical world to the EO improvements above.

Similarly, the background of the virtual imagery should be changed to simulate that of the real world. Instead of the constant and current virtual landscape in the background, a series of images should be created to replicate the physical world imagery. Then, compare those results to the ones above and determine if the approximation values stay within the same range. Additional research could modify the exposure settings, the extrinsic and distortion properties, and the type of camera

utilized.

Appendix A. Additional Results

The entire code base and results related to this project can be found in the X:// drive at AFIT. The flight data associated with this research are organized into folders within the X:// drive. The corrected camera calibrations can be found in their respective flight folders.

This research utilizes Git to maintain the related code base. The code related to this project is located in the AAR-tps-edwards2019 repository under the BradThesis branch. The calibration methods in this research were experimented under the CalibrationBranch branch.

Appendix B. Equations

Cylinder Point Inclusion Algorithm: (1)

$Cyl = Cylinder$

$\vec{CP} = Center\ Position\ of\ Cylinder$

$CylDCM = DCM\ (Direction\ Cosine\ Matrix)\ of\ Cyl$

$L = Length\ of\ Cyl$

$R = Radius\ of\ Cyl$

$\vec{TOC} = (CylDCM\ *\ 0, 0, L/2)\ +\ \vec{CP}$

$\vec{BOC} = (CylDCM\ *\ 0, 0, L/2)\ +\ \vec{CP}$

$p = Point\ in\ Question$

$\vec{pos} = Position\ of\ p$

$\vec{CV} = \vec{TOC} - \vec{BOC}$

$\vec{CPV} = \vec{pos} - \vec{BOC}$

$\vec{NCV} = \vec{CV}.normalize$

$\vec{PNCV} = \vec{NCV}\ *\ (\vec{NCV} \cdot \vec{CPV})$

$\vec{F} = (\vec{BOC}\ +\ \vec{CPV}) - (\vec{BOC}\ +\ \vec{PNCV})$

$\vec{POC} = \vec{BOC} + \vec{PNCV}$

$DTEP = \sqrt{(\vec{TOC}.x - \vec{CP}.x)^2 + (\vec{TOC}.y - \vec{CP}.y)^2 + (\vec{TOC}.z - \vec{CP}.z)^2}$

$DTNP = \sqrt{(\vec{POC}.x - \vec{CP}.x)^2 + (\vec{POC}.y - \vec{CP}.y)^2 + (\vec{POC}.z - \vec{CP}.z)^2}$

$Cyl \in p = (DTNP > DTNE)\ \&\&\ (Magnitude(\vec{F}) <= R),$

$where$

$CP : Center\ Point$

$S : Scale$

$TOC : Middle\ Top\ of\ Cylinder$

$BOC : Middle\ Bottom\ of\ Cylinder$

$CV : Cylinder\ Vector$

$CPV : Cylinder\ to\ Point\ Vector$

$PNCV : Projected\ Normal\ Cylinder\ Vector$

$F : Final\ Vector$

$POC : Point\ on\ Cylinder$

$DTEP : Distance\ To\ End\ Point$

$DTNP : Distance\ To\ New\ Point$

Ellipsoid Point Inclusion Algorithm: (2)

$Cyl = Cylinder$

$\vec{CP} = CenterPosition\ of\ Cylinder$

$CylDCM = DCM\ (Direction\ Cosine\ Matrix)\ of\ Cyl$

$L = Length\ of\ Cyl$

$R = Radius\ of\ Cyl$

$\vec{S} = ScaleofCyl - provides\ the\ ellipsoidal\ effect$

$\vec{TOC} = (CylDCM\ *\ 0, 0, L/2)\ +\ \vec{CP}$

$\vec{BOC} = (CylDCM\ *\ 0, 0, L/2)\ +\ \vec{CP}$

$p = Point\ in\ Question$

$\vec{pos} = Position\ of\ p$

$$\vec{CV} = \vec{TOC} - \vec{BOC}$$

$$\vec{CPV} = \vec{pos} - \vec{BOC}$$

$$\vec{NCV} = \vec{CV}.normalize$$

$$\vec{PNCV} = \vec{NCV} \ * \ (\vec{NCV} \cdot \vec{CPV})$$

$$\vec{F} = (\vec{BOC} \ + \ \vec{CPV}) - (\vec{BOC} \ + \ \vec{PNCV})$$

$$\vec{POC} = \vec{BOC} + \vec{PNCV}$$

$$DTEP = \sqrt{(\vec{TOC}.x - \vec{CP}.x)^2 + (\vec{TOC}.y - \vec{CP}.y)^2 + (\vec{TOC}.z - \vec{CP}.z)^2}$$

$$DTNP = \sqrt{(\vec{POC}.x - \vec{CP}.x)^2 + (\vec{POC}.y - \vec{CP}.y)^2 + (\vec{POC}.z - \vec{CP}.z)^2}$$

$$\vec{PICF} = CylDCM.transpose \ * \ (\vec{pos} - \vec{CP})$$

$$xRadius = \vec{S}.x \ * \ R$$

$$yRadius = \vec{S}.y \ * \ R$$

$$xDistance = (\vec{PICF}.x^2) \ / \ (xRadius^2)$$

$$yDistance = (\vec{PICF}.y^2) \ / \ (yRadius^2)$$

$$Distance = xDistance \ + \ yDistance$$

$$Cyl \in p = (DTNP > DTNE) \ \&\& \ Distance <= 1,$$

$where$

$CP : Center \ Point$

$S : Scale$

$TOC : Middle \ Top \ of \ Cylinder$

$BOC : Middle \ Bottom \ of \ Cylinder$

$CV : Cylinder \ Vector$

$CPV : Cylinder \ to \ Point \ Vector$

$PNCV : Projected \ Normal \ Cylinder \ Vector$

$F : Final\ Vector$

$POC : Point\ on\ Cylinder$

$DTEP : Distance\ To\ End\ Point$

$DTNP : Distance\ To\ New\ Point$

$PICF : Point\ in\ Cylinder\ Fram$

Bibliography

1. Air Force Reserve Command. Aerial refueling. Accessed: https://www.afrc.af.mil/News/Photos/igphoto/2000575522/ on 7 February 2019.

2. OpenCV. Camera Calibration and 3D Reconstruction. Accessed https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.h on Dec 8 2019.

3. Epipolar geometry. Accessed: https://en.wikipedia.org/wiki/Epipolar_geometry on 14 October 2019.

4. William Dallman and Scott Nykl. Infrared and Electro-Optical Stereo Vision for Automated Aerial Refueling. Master's thesis, Air Force Institute of Technology, 2019.

5. Christopher A Parsons. Improving Automated Aerial Refueling Stereo Vision Pose Estimation Using a Shelled Reference Model. Master's thesis, Air Force Institute of Technology, 2017.

6. James D. Anderson, Scott Nykl, and Thomas Wischgoll. Augmenting flight imagery from aerial refueling. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Daniela Ushizima, Sek Chai, Shinjiro Sueda, Xin Lin, Aidong Lu, Daniel Thalmann, Chaoli Wang, and Panpan Xu, editors, *Advances in Visual Computing*, pages 154–165, Cham, 2019. Springer International Publishing.

7. American Institute of Aeronautics and Aerospace. Aerial Refueling. Accessed: https://www.aiaa.org/microlesson37/ on Dec 8 2019.

8. Christopher Bolkcom and Jon D Klaus. Air Force Aerial Refueling Methods: Flying Boom versus Hose-and-Drogue. Accessed https://apps.dtic.mil/docs/citations/ADA472542 on Dec 8 2019.

9. SC De Vries. UAVs and Control Delays. Technical report, TNO Defence Security and Safety Soesterberg (Netherlands), 2005.

10. Rebecca Grant. Refueling the RPAs. Accessed: http://www.airforcemag.com/MagazineArchive/Pages/2012/March%202012/0312RPA.aspx on 14 October 2019.

11. Richard Szeliski. Computer Vision: Algorithms and Applications. Accessed: http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf on 14 October 2019.

12. Yoram Yekutieli, Rea Mitelman, Binyamin Hochner, and Tamar Flash. Analyzing Octopus Movements Using Three-Dimensional Reconstruction. *Journal of Neurophysiology*, 98(3):1775–1790, 2007.

13. Diane Berkenfeld, Dave Black, Mike Corrado, and Lindsay Silverman. Understanding Focal Length. Accessed https://www.nikonusa.com/en/learn-and-explore/a/tipsandtechniques/understandingfocallength.html on 6 Nov 2019.

14. MathWorks. Camera Calibration. Accessed: https://www.mathworks.com/help/vision/ug/camera-calibration.html on 6 Nov 2019.

15. Prateek Joshi. Camera Calibrations. Accessed: https://prateekvjoshi.com/2014/05/31/understanding-camera-calibration/ on 14 October 2019.

16. Zhengyou Zhang. A Flexible New Technique for Camera Calibration. Accessed: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.220.534&rep=rep1&type=pdf on 14 October 2019.

17. Guido Gerig. Image Rectification (Stereo) . Accessed: http://www.sci.utah.edu/ gerig/CS6320-S2012/Materials/CS6320-CV-F2012-Rectification.pdf on 14 October 2019.

18. Jace Robinson, Matt Piekenbrock, Lee Burchett, Scott Nykl, Brian Woolley, and Andrew Terzuoli. Parallelized Iterative Closest Point for Autonomous Aerial Refueling. In *International Symposium on Visual Computing*, pages 593–602. Springer, 2016.

19. Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, 2013.

20. Dmitry Chetverikov, Dmitry Svirko, Dmitry Stepanov, and Pavel Krsek. The Trimmed Iterative Closest Point Algorithm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 545–548. IEEE, 2002.

21. Paul Besl, IEEE, Neil McKay, and Member. A Method for Registration of 3-D Shapes. 1982.

22. Peter R Thomas, Ujjar Bhandari, Steve Bullock, Thomas S Richardson, and Jonathan L Du Bois. Advances in Air to Air Refuelling. *Progress in Aerospace Sciences*, 71:14–35, 2014.

23. Thomas R Stuart. Integrity Monitoring For Automated Aerial Refueling: A Stereo Vision Approach. Master's thesis, Air Force Institute of Technology, 2018.

24. Zachary C Paulson. Mitigating the Effects of Boom Occlusion on Automated Aerial Refueling Through Shadow Volumes. Master's thesis, Air Force Institute of Technology, 2018.

25. Phillip Swarts. Air Force Refueling Sorties on Top of the World: They're 'like and Organized Chaos'. 2016.

26. Wikipedia. Observational Error. Accessed https://en.wikipedia.org/wiki/Observational_error on 6 Nov 2019.

27. Jo-Anne Ting, Evangelos Theodorou, and Stefan Schaal. A Kalman Filter for Robust Outlier Detection. 2007.

28. Ba Nguyen and Tong Lin. The Use of Flight Simulation and Flight Testing in the Automated Aerial Refueling Program. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, page 6007, 2005.

29. Richard Burns, Curt Clark, and Ron Ewart. The Automated Aerial Refueling Simulation at the AVTAS Laboratory. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, page 6008, 2005.

30. Shuai An and Suozhong Yuan. Relative Position Control Design of Receiver UAV in Flying-Boom Aerial Refueling Phase. *ISA transactions*, 73:40–53, 2018.

31. Nicolas Fezans and Thomas Jann.

32. Marco Mammarella, Giampiero Campa, N.R. Napolitano, and Brad Seanor. GPS/MV Based Aerial Refueling for UAVs. 2008.

33. Nicholas J. Seydel . Stereo Vision: A Comparison of Synthetic Imagery VS. Real World Imagery for the Automated Aerial Refueling Problem. Accessed https://scholar.afit.edu/etd/1823/ on Dec 8 2019.

34. Scott Nykl, Chad Mourning, Mitchell Leitch, David Chelberg, Teresa Franklin, and Chang Liu. An Overview of the STEAMiE Educational Game Engine. In *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual*, pages F3B–21. IEEE, 2008.

35. Daniel T Johnson. Combined Stereo Vision and Inertial Navigation for Automated Aerial Refueling. Master's thesis, Air Force Institute of Technology, 2017.

36. TechTerms. BMP. Accessed https://techterms.com/definition/bmp on Dec 8 2019.

37. Victor Powell and Lewis Lehe. Principal Component Analysis Explained Visually. Accessed http://setosa.io/ev/principal-component-analysis/ on Dec 10 2019.

38. Josh Starmer with Statquest. StatQuest: Principal Component Analysis (PCA), Step-by-Step.

39. Blender Online Community. *Blender - A 3D Modelling and Rendering Package*. Blender Foundation, Blender Institute, Amsterdam,

40. T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? - Arguments against avoiding RMSE in the literature. Accessed https://www.geosci-model-dev.net/7/1247/2014/gmd712472014.pdf on Nov 6 2019.

Acronyms

AAR Automated Aerial Refueling. iv, vii, 1, 2, 4, 5, 6, 7, 8, 13, 14, 16, 17, 18, 19, 20, 26, 46, 116, 118, 1

AF Air Force. iv, 1, 6, 16, 1

AFB Air Force Base. vii, 2, 14, 17, 25, 32

AFIT The Air Force Institute of Technology. iv, 4, 16, 17, 19, 22, 26, 118, 121, 1

bmp bitmap. 26

DGPS Differential GPS. viii, 14, 18, 21, 22, 23, 40, 41

DOF Degrees of Freedom. 18, 55, 58, 59

EO Electro-Optical. ix, x, xi, xii, 13, 14, 16, 18, 24, 25, 26, 61, 69, 74, 79, 84, 92, 100, 108, 116, 118, 119

GPS Global Positioning System. 2, 14, 15, 16, 22, 23, 40

ICP Iterative Closest Point. 7, 12, 13, 18, 27, 52, 55

IMU Inertial measurement units. viii, 2, 14, 15, 18, 21, 22, 40, 41

IR Infrared. vi, ix, xi, xii, 14, 16, 18, 24, 26, 60, 66, 74, 79, 84, 89, 97, 105, 113, 116, 118, 119

long-wave infrared long-wave infrared. 26

MAE Mean Absolute Error. 36, 39, 56, 57

MB Megabyte. 33, 35

PCA Principal Component Analysis. 44, 46

pgm Portable Gray Map Image. 26

PNG Portable Network Graphic. 26

RMSE Root Mean Squared Error. 13, 56, 57

TB Terabyte. 33, 35

TPS Test Pilot School. 2, 14

UAV unmanned aerial vehicle. 1, 2, 6, 16

USAF United States Air Force. 6

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
19–03–2020	Master's Thesis	Sept 2018 — Mar 2020

4. TITLE AND SUBTITLE

DETERMINING VIRTUAL PRACTICALITY FROM PHYSICAL STEREO VISION IMAGES AND GPS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Bradley S. French

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-20-M-020

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

Current research efforts for AAR at AFIT utilize Stereo Computer Vision to compute a relative pose between a tanker and receiver aircraft. Due to costs, time, and availability, it can be onerous to test these algorithms using actual AF aircraft. Our solution to this problem consists of using a 3D Graphics Engine to simulate AAR endeavors. However, the question then arises, "Does the virtual world accurately represent the physical world?" This can be explored by comparing a set of truth data to a similar set of virtual data. First, a set of truth data is collected using physical aircraft. Next, using the same flight path as that of the truth data, a set of virtual data is collected. Finally, a comparison of the physical and virtual data can provide information regarding how well the virtual world accurately represents the physical world, and if so, to within what margin of error? The results show that the virtual world roughly approximates the physical world but performs 2-6x better on average.

15. SUBJECT TERMS

Automated Aerial Refueling, Simulations

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	149	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x4395 scott.nykl@afit.edu