2012

# Comparison of Three Machine Vision Pose Estimation Systems Based on Corner, Line, and Ellipse Extraction for Satellite Grasping

Jeremy A. Luckett
*West Virginia University*

# Comparison of Three Machine Vision Pose Estimation Systems Based on Corner, Line, and Ellipse Extraction for Satellite Grasping

By

Jeremy A. Luckett

Thesis submitted to the

College of Engineering and Mineral Resources

At West Virginia University

in partial fulfillment to the requirements

for the degree of

Master of Science

in

Aerospace Engineering

Accepted By:

Dr. Marcello Napolitano, Committee Chairperson

Dr. Giacomo Marani

Dr. Powsiri Klinkhachorn

Dr. Gianfranco Doretto

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia

2012

Keywords: Machine Vision, Satellite Servicing, Corner, Line, Ellipse, Pose Estimation

## Abstract

The primary objective of this research was to use three different types of features (corners, lines, and ellipses) for the purpose of satellite grasping with a machine vision-based pose estimation system. The corner system is used to track sharp corners or small features (holes or bolt) in the satellite; the lines system tracks sharp edges while the ellipse system tracks circular features in the satellite. The corner and line system provided 6 degrees of freedom (DOF) pose (rotation matrix and translation vector) of the satellite with respect to the camera frame, while the ellipse system provided 5 DOF pose (normal vector and center position) of the circular feature with respect to the camera frame. Satellite grasping is required for on-orbit satellite servicing and refueling. Three machine vision estimation systems (base on line, corner, and ellipse extraction) were studied and compared using a simulation environment. The corner extraction system was based on the Shi-Tomasi method; the line extraction system was based on the Hough transform; while the ellipse system is based on the fast ellipse extractor. Each system tracks its corresponding most prominent feature of the satellite. In order to evaluate the performance of each position estimation system, six maneuvers, three in translation (xyz) and three in rotation (roll pitch yaw), three different initial positions, and three different levels of Gaussian noise were considered in the virtual environment. Also, a virtual and real approach using a robotic manipulator sequence was performed in order to predict how each system could perform in a real application. Each system was compared using the mean and variance of the translational and rotational position estimation error. The virtual environment features a CAD model of a satellite created using SolidWorks which contained three common satellite features; that is a square plate, a marman ring, and a thruster. The corner and line pose estimation systems increased accuracy and precision as the distance decreases allowing for up to 2 centimeters of accuracy in translation. However, under heavy noise situations the corner position estimation system lost tracking and could not recover, while the line position estimation system did not lose track. The ellipse position estimation system was more robust, allowing the system to automatically recover, if tracking was lost, with accuracy up to 4 centimeters. During both approach sequences the ellipse system was the most robust, being able to track the satellite consistently. The corner system could not track the system throughout the approach in real or virtual approaches and the line system could track the satellite during the virtual approach sequence.

## Acknowledgments

First, I would like to thank my family for all of the support that they have given me throughout this journey; without their help this would have been much more difficult. I would also like to thank my committee chair and research advisor, Dr. Marcello Napolitano, for providing me many opportunities to perform research which allowed me to learn many different areas.

I would like to thank my remaining committee members, Dr. Gianfranco Doretto, Dr. Powsiri Klinkhachorn, and Dr. Giacomo Marani, for providing advice and guidance throughout this journey and preparing me for life after graduate school. I would especially like to thank Andreas Felipe Velasquez, for all of the help, mentoring, and friendship he has given. I would like also like to thank the research group working at the West Virginia Robotics Technology Center, (Mr. Andrew Dow, Mr. Patrick Lewis, and Dr. Sanghoon Han) for their advice and assistance as to what to expect after graduate school.

Thank you to all the students and research professors in the Flight Controls Research Lab for the encouragement and support: Dr. Yu Gu, Dr. Brad Seanor, Dr. Srikanth Gururajan, Dr. Jason Gross, Dr. Kerri Phillips, Frank Barchesky, Matt Rhudy, Daniele Tancredi, Amanda McGrail, Giovanni DeNunzio, Matteo Dariol, and Matteo Guerra.

Finally, I would like to thank West Virginia University, The Benjamin M. Statler College of Engineering and Mineral Resources, and the Mechanical and Aerospace Engineering Department for giving me a great education, great friends, and a second home.  I will always be a Mountaineer.

# Contents

## List of Figures

# List of Tables

## Nomenclature

| | | |
|---|---|---|
| DOF | = | Degrees Of Freedom |
| FEDALPE | = | Feature Extraction, Detection and Labeling, Pose Estimation |
| MV | = | Machine Vision |
| Pose | = | Position and Orientation Estimation |

# 1. Introduction

The use of machine vision based pose estimation systems and robotic manipulators to capture satellites has been studied for many decades, with initial work being published as early as 1989 (1) (2). The pose (position and orientation estimation) of the target satellite with respect to the robotic manipulator is required by the robot controller in order to compute the trajectories to catch/grapple/grasp the satellite. With varying satellite designs, different features need to be tracked by the MV (Machine Vision) system so that the system can track any type of satellite. It should also be noted that the same part of a satellite can appear differently at different distances.

The goal of this research is to compare three different types of features extractors, within the general scheme known as FEDALPE (Features extraction, detection and labeling, post estimation technique, see Section 3.4). The original FEDALPE (3) only uses corners to determine pose, however in this effort two modified version are used, one that includes lines and another that uses ellipses. The original corner system is used to track sharp corners or small features such as holes or bolts. The line system tracks sharp edges while the ellipse system tracks circular features in the satellite (ellipse extraction is used since the projection of a circular feature of the satellite in the image plane of a camera is an ellipse). These types of features are common in satellites (see Figure 1, Figure 2, and Figure 3). Corners can be seen in holes, bolt heads (see Figure 1), lines can be found in access panels, antennas, solar panel arrays (see Figure 2), and circular features can be found on thrusters and marman rings (see Figure 3).


Figure 1: Example of Corner Features on ANDE (Atmospheric Neutral Density Experiment)

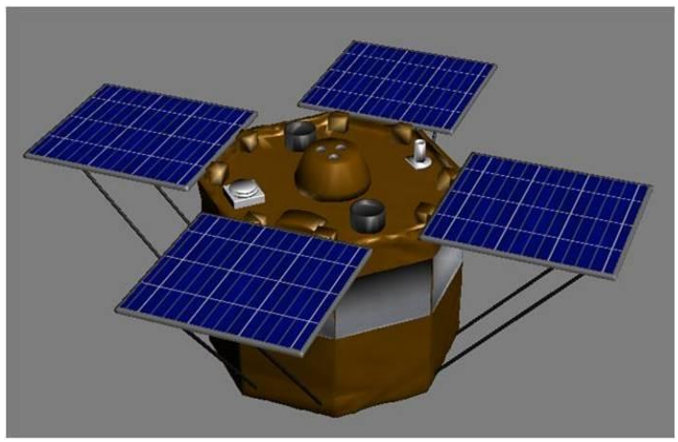

Figure 2: Example of Line Features on AcrimSat (Active Cavity Radiometer Irradiance Monitor Satellite)


Figure 3: Example of Circle Features on AOSO (Advanced Orbiting Solar Observatory)

This research effort characterizes and compares the pose estimation systems based on the estimation error of the pose. The potential applications of this research may include sensor fusion in order to use all three systems (corner, lines and ellipses) together.

This thesis will cover the FEDALPE methodology of different methods in order to compare different features. During the feature extraction phase, features such as corners, lines, or ellipses are extracted from the image. These features are then fed through the detection and labeling phase in order to solve the correspondence problem; i.e. find the desired feature to track from the extracted features. By using the desired features, the position is estimated. An overview of this approach can be seen in Section 3. 4.

# 2. Project and Theory Literature Review

The following sections will review prior results in the satellite servicing field along with capturing and tracking satellites. Other applications for the proposed machine vision techniques will then be discussed.

## 2. 1. Project Background

Currently, there are over 900 satellites in Earth's orbit (4); however most of these are non-functional. Over time, the gradual break-down of the non-functional satellites has resulted in the existence of approximately 19,000 individual objects in Earth's atmosphere (5). Figure 4 depicts the debris around the Earth's orbit.

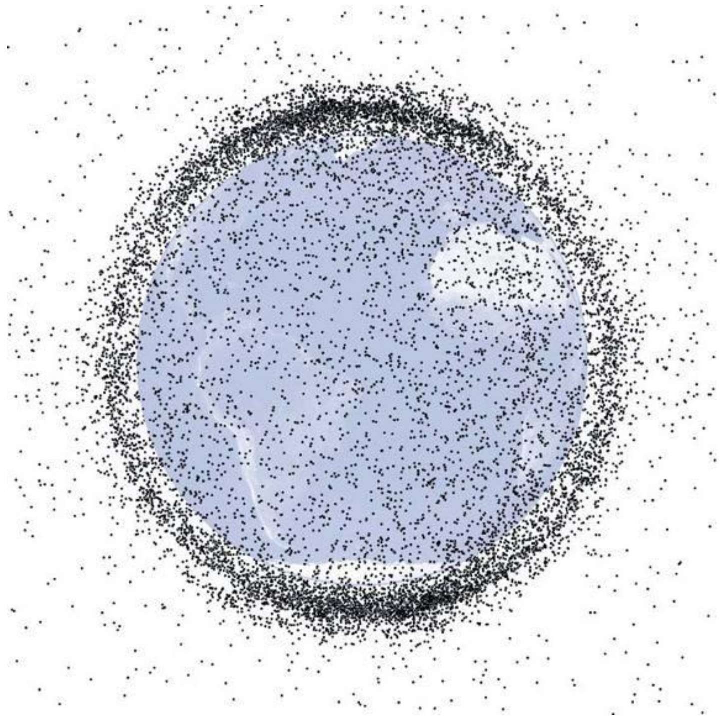

Figure 4: Image depicting debris in Earth's Orbit **(6)**

The presence of these objects in uncontrolled orbits has led to collisions with fragments of other satellites or other countries satellites. For example, as recently as 2009 a Russian satellite (Cosmos 2251) collided with the US satellite, Iridium (7). This new class of space accidents prompted the United States Congress to direct NASA to investigate space servicing in order to repair or transport non-operational satellites (8). It is envisioned that this program will provide substantial benefits, including repairing or refueling of satellites (at a minimum "orbit modification"), reuse of previous satellites by installing new modern equipment and increased lifespan of satellites (9). That effort has already begun with satellite capture and can be categorized in four different stages: Aided Rendezvous, Aided Rendezvous and Docking, Unaided Rendezvous, and Unaided Rendezvous and Capture (10).

### 2. 1. 1. Aided Rendezvous

The Demonstration of Autonomous Rendezvous Technology (DART) mission was the first mission (2005), in order to determine if a spacecraft could track and follow another spacecraft (the MUBLCOM) autonomously. To do this, the DART performed maneuvers 1000 kilometers behind the MUBLCOM satellite in order to evaluate the precision. The guidance system used an Advanced Video Guidance Sensor (AVGS), which gathered data from laser signals and reflected off targets that were mounted on the MUBLCOM satellite. The system also featured two GPS receivers, one GPS on DART and another on MUBLCOM, in order to collect relative position and velocity data. Ultimately, the DART satellite collided with the MUBLCOM due to lack of fuel (11).

### 2. 1. 2. Aided Rendezvous and Docking

The Engineering Test Satellite No. 7 (ETS-7) was developed and launched (1997) from NASDA in Japan. It consisted of a target satellite and a chaser satellite. In this setup, the chaser satellite would approach and attempt to have a rendezvous docking with the target satellite. To dock with the satellite a robotic manipulator was used. The target satellite was located by different sensors based on the distance between the target and the chaser satellites. The main navigation sensor for the approach (2 meters < 520 meters) used a set of modulated laser sources and CCD detectors, the reflectors in this system were mounted on the target. A set of LED array light sources and CCD detectors, combined with Micro CCR on the target, allowed for the calculation of relative position and attitude of the target during the docking phase (<2 meters) . The system also featured a GPS Receiver for the main navigation sensor during the relative approach (<500 meters). Using this approach, and after several attempts the ETS-7 chase mated with the target (12).

Another example of aided rendezvous and docking is given by the Orbital Express mission, which was developed by DARPA in 2009 with assistance from Boeing Advanced Systems for the chase satellite and Ball Aerospace for the target vehicle. This package consisted of a chaser (ASTRO) and target satellite (NEXTSat). ASTRO was equipped with NASA's AVGS package which contained a combination of video and lasers that used cube reflectors that were attached to NEXTSat. There were reflectors for the long range target (LRT) and the short range target (SRT) as seen in Figure 5 (13).

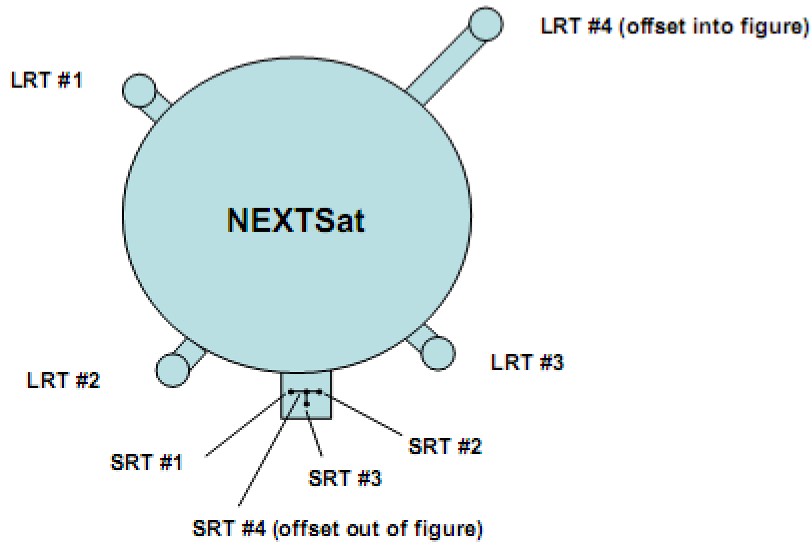


Figure 5: Sensor Layout on NEXTSat **(13)**

Once close enough, the NEXTSat captured the ASTROC by using a robotics manipulator and a grapple fixture called the Probe Fixture Assembly (PFA) that was installed on ASTRO (14).

Europe sent the next aided rendezvous and docking system (2008) called the Automated Transfer Vehicle: Jules Verne (ATV-1, ATV-JV). The ATV carried various equipment, supplies, water, fuel, and gases to the ISS Station (15). During both the rendezvous and long-range phase, GPS and Relative GPS were used along with a linearized Kalman filter in order to filter and process the data (16).

Japan sent up an unmanned carrier called the H-II Transfer Vehicle (HTV) for docking with the ISS (1998). The HTV used GPS signals for its long range approach, and then used a Rendezvous Sensor (RVS) which was guided by reflectors on the ISS. When the HTV was closer than 10 meters to the ISS, the ISS crew sent commands in assist docking (17) (18).

### 2. 1. 3. Unaided Rendezvous

The XSS-10 was developed by the Air Force Research Laboratory to promote micro-satellite technology. Its goals involved performing three points of autonomous inspection about a “resident satellite”. The goal was accomplished through finding a star pattern on the bottom of the target satellite as seen in Figure 6 (19).

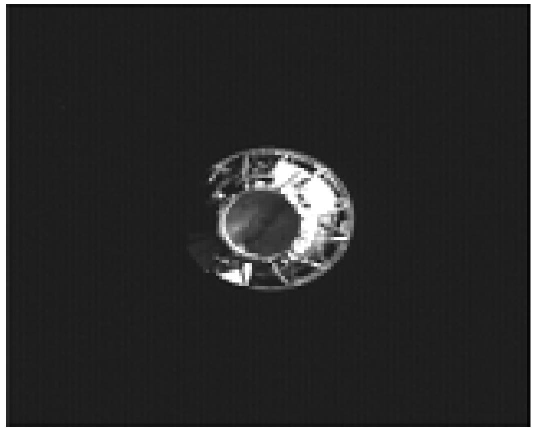

Figure 6: Star Pattern for XSS-10 Tracking

The PRISMA mission was the next program to demonstrate unaided rendezvous; however, the primary objects also included formation flying. This mission involved two satellites, Mango (the chaser) and Tango (the target). The experiments were performed in four different categories; autonomous formation flying, autonomous rendezvous, proximity operations, and the final approach and recede. During the autonomous formation (20 meters – 5000 meters) the form of control came from relative GPS. During the homing autonomous rendezvous section the main position data came from their vision based system (VBS). This system consisted of four cameras, one for long range, one for short range, one that can see the target for the phone mission, and one with a specialized lens that could operate at high light conditions at close range. The target was then equipped with 5 LED's to provide sufficient features to track and would operate in a synchronous pulse. However, the vision system could also work with satellites that did not have LED's, but the complete geometry of the target was needed (20).

#### 2. 1. 4. Unaided Rendezvous and Capture

The Front-End Robotics Enabling Near-term Demonstration (FREND), shown in **Figure 7** is a prototype from the Naval Research Laboratory (NRL) sponsored by the Defense Advanced Research Projects Agency (DARPA). This robotic arm uses a TriDAR system in order to locate the satellite providing 6 DOF relative position estimate. As its robotic arm gets closer, it uses machine vision in order to correct the relative position estimate by looking at the marman ring and bolt holes (10) (21).

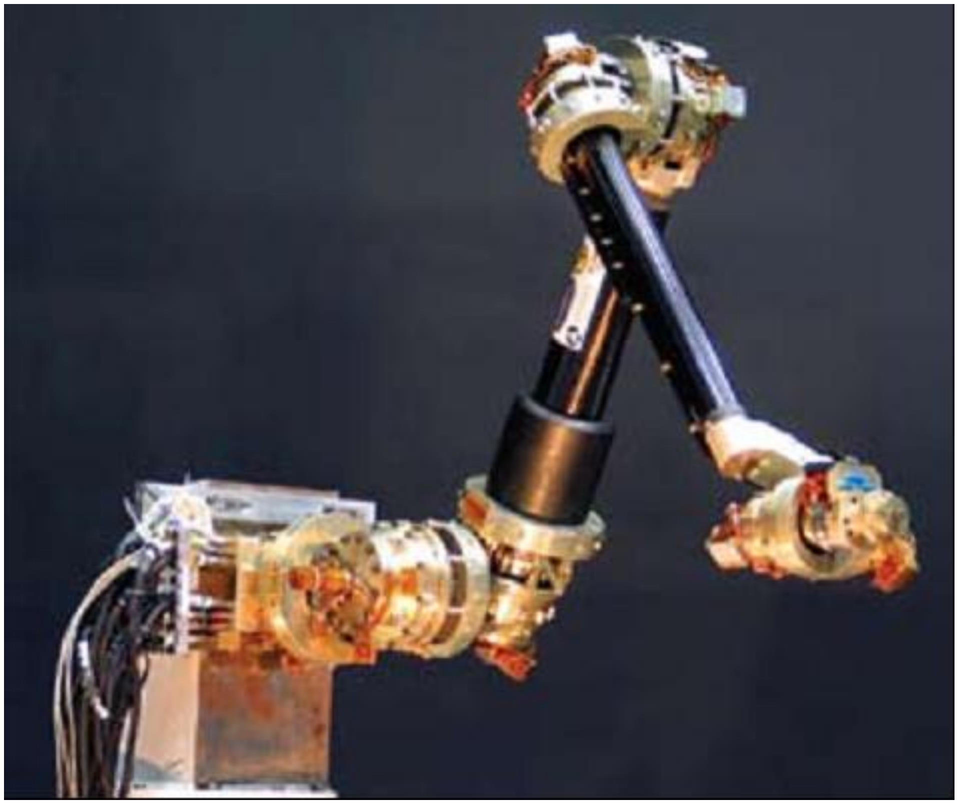

Figure 7: FREND Robotic Arm **(21)**

The SMART-OLEV (Orbital Live Extension Vehicle), developed by the Swedish Space Corporation (SSC) is very different from the FREND system. During the long phase it uses a far field camera (compared to a scanner); the system then switches to stereovision (compared to monovision) when it is closer than five meters. It also uses the thrusters to grapple instead of using the marman ring (22) (23). An overview of all the missions and years they were used are shown in

Table **1**.

Table 1: Summary of Previous Satellite Tracking Missions

| Satellite Name | Year | Type of Mission |
|---|---|---|
| DART | 2005 | Aided Rendezvous |
| ETS-7 | 1997 | Aided Rendezvous and Docking |
| Orbital Express | 2007 | Aided Rendezvous and Docking |
| ATV-JV | 2008 | Aided Rendezvous and Docking |
| HTV | 2009 | Aided Rendezvous and Docking |
| XSS-10 | 2003 | Unaided Rendezvous |
| PRISMA | 2010 | Unaided Rendezvous |
| FREND | 2009 | Unaided Rendezvous and Capture |
| SMART-OLEV | 2010 | Unaided Rendezvous and Capture |

## 2. 2. Servicing Satellites

To service satellites, the similar features of each satellite must be considered, as there are many different satellites used for communication, observing earth, military, and interplanetary studies (24). All of these satellites have unique features or targets that could be useful to track the satellite. Along with these different features, the main focus must be to locate these features at all times, from long range to grappling. To know which features to track, a grappling location on the satellite must be defined. In the next section, some important aspects of a servicing mission will be discussed.

#### *2. 2. 1. 1. Grappling Locations*

Attitude and Orbital Control (AOC) thrusters along with the main thrusters (Apogee Thrusters) are present on nearly every satellite, due to the need to be able to maneuver and control a satellite. This is required specifically for changing and adjusting orbits. Thrusters have also been proposed as a satellite grappling location (25). In order to capture a satellite via the thrusters a probe needs to be inserted into the thruster and then expanded, thus not allowing the satellite to move. Figure 8 and Figure 9 show examples of a thruster grappling system.

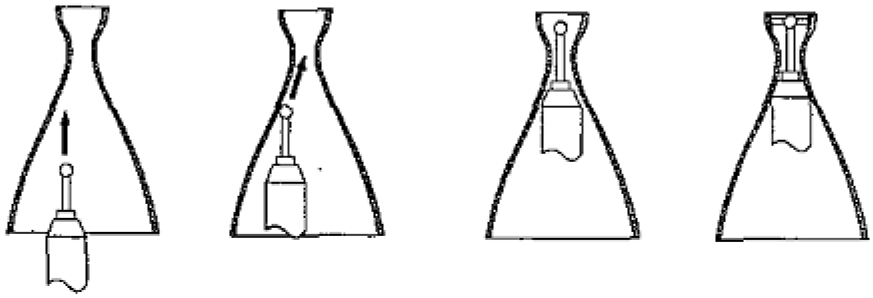

Figure 8: Example of a Thruster Grappling System **(25)**

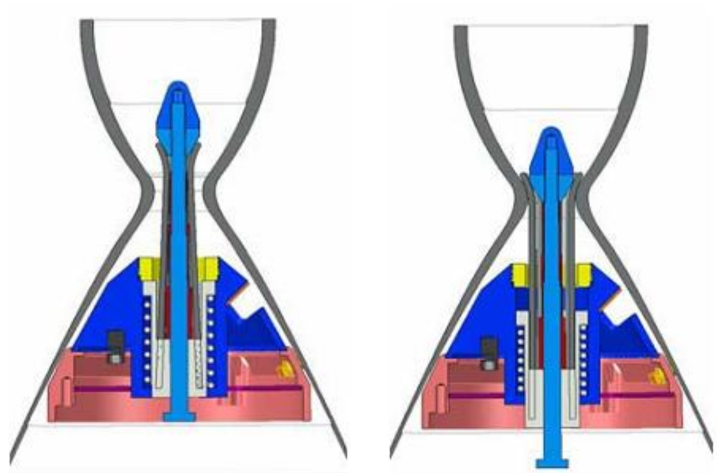

Figure 9: SMART-OLEV Thrust Grappling System (23)

The marman ring connects the satellite and booster rocket, and is the common feature in every satellite. The marman ring contains two parts, one attached to the booster rocket and one to the spacecraft. When the satellite and booster rocket reach the point in the mission where they need to separate, explosive bolts, which connect the two sections, explode, allowing separation. An example of the marman ring can be seen in Figure 10.

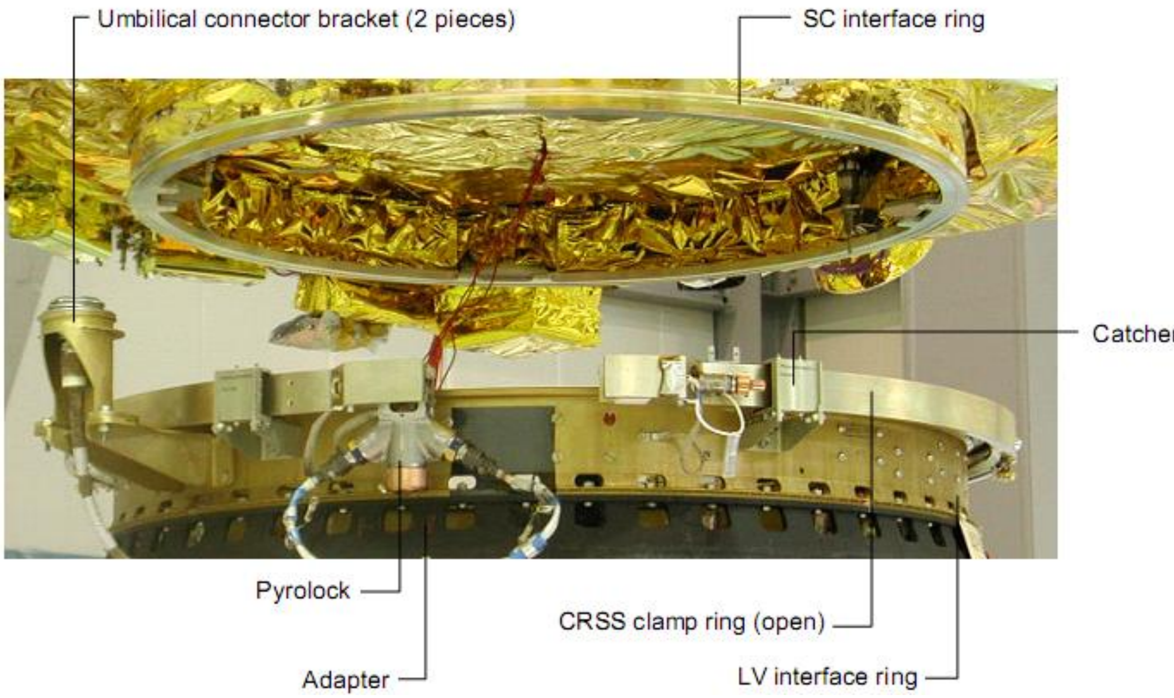


Figure 10: Example of a Marman Ring **(26)**

Due to recent missions being focused on the use of a robotic arm grappling the marman ring, and a camera being mounted on the grappling arm, the main features that are used on will be close to the marman ring in order to insure the features will be seen

#### *2. 2. 1. 2. Approach*

The approach to a satellite is very critical in order to grapple and capture a satellite in all of the previously mentioned methods and can be divided into three different sections; long-range rendezvous, short-range rendezvous, and finally capture. During the long-range rendezvous (5000 meters – 300 meters) the primary objective is to collect technical information about the orbit of the target satellite. With the orbit of the target satellite known, the chaser satellite can know the relative velocity and position throughout the rest of the mission. When the chaser satellite is within 300 meters it is in the short-range rendezvous until it is within several meters. During this phase it is important to gain attitude knowledge, along with position knowledge. The relative speed and attitude is minimized in order to dock with the satellite in the capture phase. After the chaser satellite closes to within a few meters of the target satellite, the capture phase of the mission starts. The capture phase is the highest risk of the grasping sequence due to the small tolerance in distances and error that can occur. Within a few meters, the chaser satellite must deploy the robotic arm in order to grasp the satellite. The robotic arm must then be guided in order to align and grasp the satellite, without colliding with the satellite. After capture, the satellite must then be stabilized in order to create a rigid body. An overview of this phase is shown in Figure 11. The current research will focus on distances closer than 3m away from the target satellite.

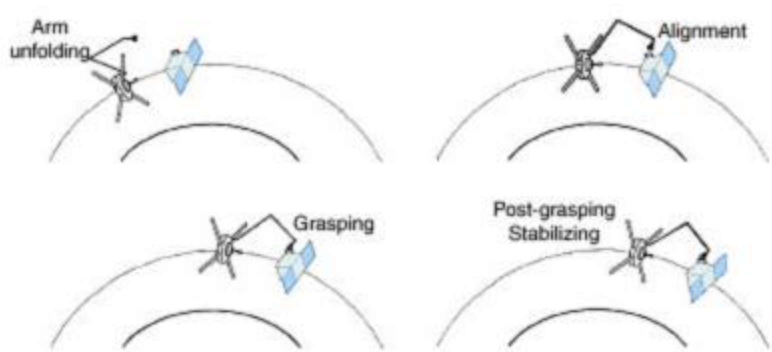


Figure 11: Capturing sequence **(27)**

#### *2. 2. 1. 3. Features on Satellites*

There are many different features that can be found within the view of a satellite. Some of these features are points, corners, lines, and circles. Points can be considered to be bolts or screws that at a certain distance they do not have a particular shape. Lines can be a part of a square or triangular sensor; these lines can be broken down further into the corners of the intersecting lines. There can also be circles like the marman ring or thrusters that are used to detach or control the satellite. Finally, satellites can be classified into cooperative satellites or non-cooperative satellites. A cooperative satellite is a satellite that was designed for capture and has features built on it to track like the NEXTSat, where a non-cooperative satellite was not designed for repair and/or servicing missions.

Figure 12: NEXTSat satellite

The NEXTsat satellite, shown in Figure 12, is a cooperative satellite that has an example of each of the features; the marman ring being a circle, the LEDs as points and the solar array as lines and corners.

## 2. 3. Machine Vision Literature Review

The machine vision theory that is covered in the literature review contains the basic purpose and reasons of use for various systems, including how various systems were used in industrial applications, then specifically in robotics and space robotics. Papers that have similar methods, or are used for satellite tracking, will be discussed more elaborately.

### 2. 3. 1. Corner Detector

Corners are one of the most commonly used geometric features used for pose estimation. There are different types of corner detectors, including Harris, Shi and Thomasi, SUSAN, and FAST (28). They have been used in geo-localization by being able to locate buildings (29). They also have been used in robotics for industry (30), or for research in order to allow robots to pick up objects (31). This allows more processes to be automated and run without human interaction. More specifically it has been used in space robotics in order to capture a satellite (32). For example, on the AVGS mission there were LED lights that were put onto the satellite to allow the satellite to be tracked and captured (13) (33).

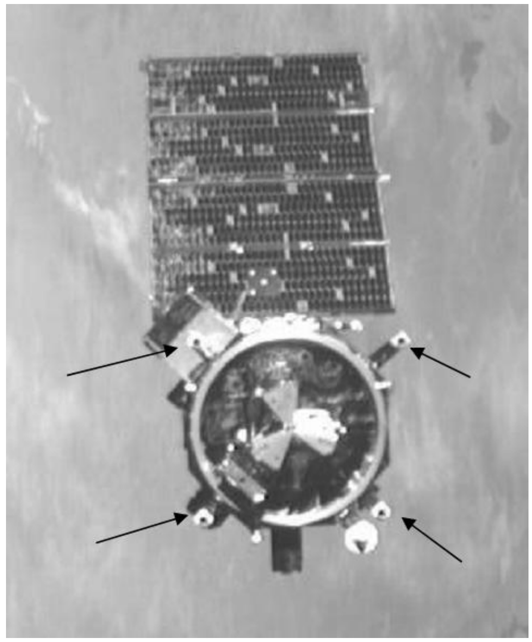

Figure 13: Targets Tracked on AVGS Mission **(13)**

Figure 13 shows the targets that were used as corners. These were markers that were fixed and designed into the satellite for purposes of tracking.

The corner detector used was originally designed to track a Boeing 747 to simulate refueling of a KC-135 tanker aircraft. Here they used the FEDALPE approach of Feature Extraction, Detection and Labeling, and Position Estimation. In order to extract various features, the Harris and Susan methods were user. They were subjected to tests to assess and quantify accuracy, speed, robustness, and included testing of poor contrast images. During the accuracy tests, the algorithms were compared to see if all the true corners would be extracted, while no false corners were detected. The robustness tests included image perturbations which included presence of noise in the image, variations of image contrast, motion blur, and noise added into the input image. Under these conditions the Harris extractor proved to be more accurate, while the Susan method is computationally more efficient.

In order to detect and label the correct points, the mutual nearest point (MNP) and the Maximum Clique Detection (MCD) methods were assessed. These methods were compared by their computational effort, virtual image analysis, and real image analysis. Under these tests, the MCD proved the best performance. In order to find the position the detected points were analyzed with the Gaussian Least Square Difference Correlation (GLSDC) algorithm and the Lu, Hager and Mjolsness (LHM) algorithm. During various tests comparing accuracy, robustness, points with noise, labeling errors, errors with initial conditions, and errors in tracking, it was determined that the LHM method was more accurate and stable (3). This will be discussed in further detail in Chapter 3. 5.

### 2. 3. 2. Line Detector

Line detectors have been widely used in MV systems and POSE systems, including systems with monocular vision (34) and with stereo vision (35), needing CAD models of the target (36), not needing a CAD model (37) or just needing a CAD model for initialization (38). There have also been papers published for robotic manipulators to grab objects for industry use (39), for more humanoid robots to climb robots (40), and for satellite missions (35) (41) (42). Figure 14 shows a rendering of the SNAP-1 (Surrey Nanosatellite Applications Platform) Nanosatellite used in (41).

Figure 14: Picture of the SNAP Satellite **(43)**

In the effort described in (35), the operation is divided into five different sections and used four different sensors. During the first phase, far-range rendezvous (>300 meters), a microwave radar is used. During the closing phase (300 meters – 15 meters) a monocular camera and laser range finder are used. The final approach phase is divided in two sections, during the first segment (15 meters – 5 meters) the same monocular camera and laser range finder are used; in the second segment (5 meters – 1 meter) two coordinated cameras are used. During the final segment (<1 meter) a hand-eye camera is used. They explain that a high field of view lens (>80°) cannot be used during the final approach phase due to a high level of distortion, but due to recent developments in technology this is not an issue in the research presented in this paper.

In the main part of (35), a stereo vision approach is described for the second segment of the final approach phase to determine the position of a satellite using a rectangular pattern. This pattern is only partially seen in each system, but together can see the entire pattern. Inside of their system, the image is filtered, and then Canny edge detection is applied. After the edge map has been acquired, a Hough transform is used to detect lines, which are then used to determine the sides of the rectangle. The detected lines are then improved to sub-pixel accuracy, and the vertices of the lines are determined. Finally the position is calculated using the corners that are found. This algorithm was tested using a 3D virtual reality scenario as seen in Figure 15.

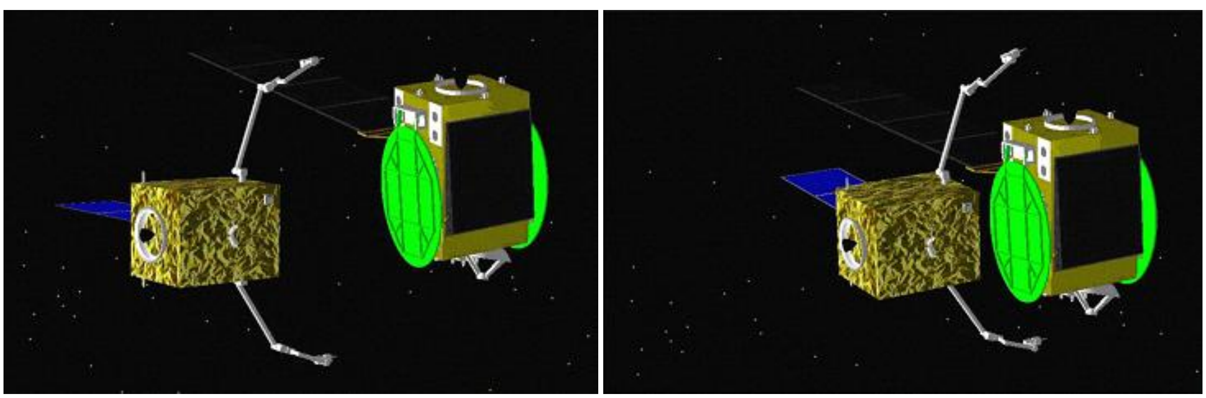

Figure 15: Initial and Final position for 3D Simulation **(35)**

### 2. 3. 3. Circle/Ellipse Detector

Circle detectors have been used extensively for industrial applications to grab objects (44), to capture satellites using a robotic manipulator (45), identify passing vehicles (46), and to have intelligent transportation systems by detecting the position of traffic signs (47). There has been additional research in finding the position of circles or spheres when knowing the radius and without knowing the radius (48), solving for the duality problem (49) which is caused when trying to find the position of a single circle (see Section 3. 7. 3), or using stereo vision (50) or two circles (51) to correct for the duality problem.

In (51), an ellipse extractor system is used to track the wheels of vehicles in order to allow determining where in space a vehicle is located, and what direction it is traveling. This could be used in drive assistance systems; however it requires a 3D model of the target. In order to track the vehicle a new approach was developed that could track objects even when motionless. To do this, the image was smoothed using a square kernel and then normalized. After being normalized, blobs were detected and labeled; if there were missing spaces in the blobs they would be filled. Ellipses are then extracted from the blobs, which allowed for the ellipse parameters and normal to be found.

To capture a satellite, a system has to be robust and allow for many different scenarios (45). In (45), a robotic manipulator was used to capture a satellite using the marman ring. This can be accomplished by a hand camera that is attached to the manipulator and a monitoring camera which is placed away from the target. This would allow for full view of the target at all times along with detection if the grappling sequence was slipping. This system was tested using a solar simulator, along with various time delays.

# 3. Methodology

In this chapter, all methodology is discussed including basic definitions on transformation matrices, coordinate frames, and the corner, line, and ellipse pose systems. Inside of each pose system, the feature extraction process is discussed, along with the detection and how to calculate the position estimation.

## 3. 1. Coordinate Frames and Transformation Matrix

The two object involved in the pose estimation are described by reference frames. Each frame has been defined as shown in Figure 16:

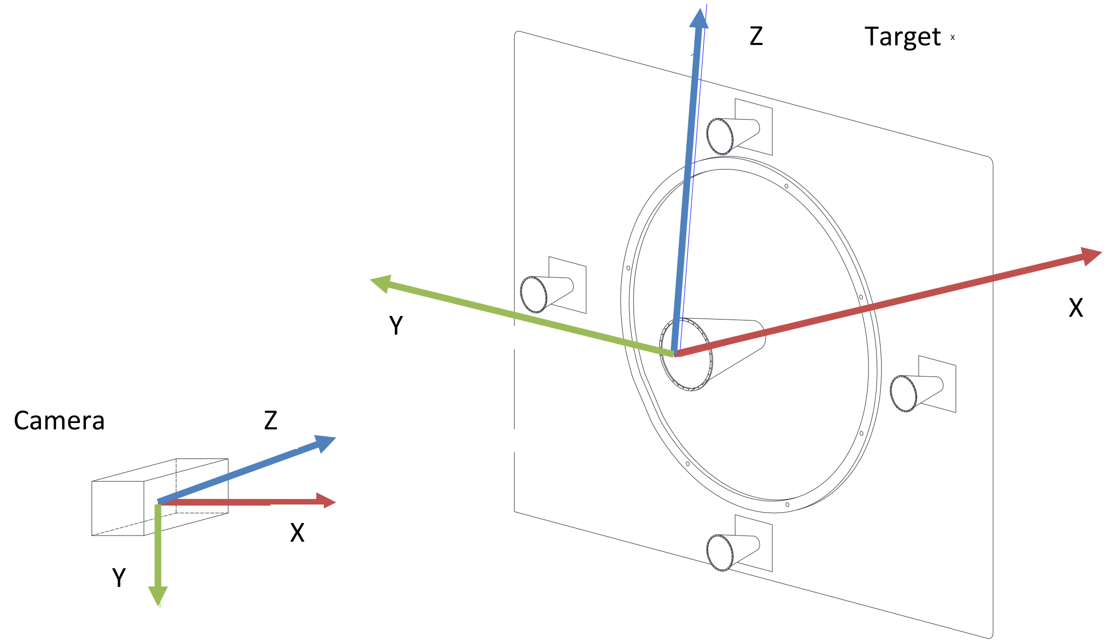


Figure 16: Camera and Target Coordinate Frames

The relative position and orientation of the target (satellite) with respect to the camera can be defined using a rotation matrix and a translation vector. The rotation matrix of the target with respect to the camera frame is defined in equation (1):

$$ {}_{C}^{T}Rot = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \tag{1} $$

Where ‘n’ is the coordinates of the target x axis with respect  to the camera frame. This is true for ‘o’ to the y axis, and ‘a’ to the z axis

The rotation matrix contains information about the relative orientation of the three axes of the two reference frames. If the axes of the two reference frames are parallel, the rotation matrix is given by the identity matrix. The translation vector is defined in equation (2):

$$ {}_{C}^{T}Trans = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{2} $$

This gives the translation between the origin of the target axis with respect to the frame axis.

These two can be written in form of a 4x4 matrix called the transformation matrix as defined in equation (3).

$$ {}_{C}^{T}T = \begin{bmatrix} {}_{C}^{T}Rot & {}_{C}^{T}Trans \\ 0 & 1 \end{bmatrix} \tag{3} $$

or

$$ {}_{C}^{T}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

Then a point X in the target frame using homogeneous coordinates, (which is required to be consistent with the transformation matrix notation) is expressed as,

$$ X_T = \begin{bmatrix} x_T \\ y_T \\ z_T \\ 1 \end{bmatrix} \tag{5} $$

With respect to the camera frame we have

$$ X_C = X_T \, {}_{C}^{T}T \tag{6} $$

## 3. 2. Camera Model

The pin-hole model of the camera is commonly (3) (39) (47) (52) used in order to find the projection of a 3D point $[X, Y, Z]^T$ to the 2D image plane of the camera as a pixel $(u,v)^T$. A sample of the pinhole camera can be seen in Figure 17.

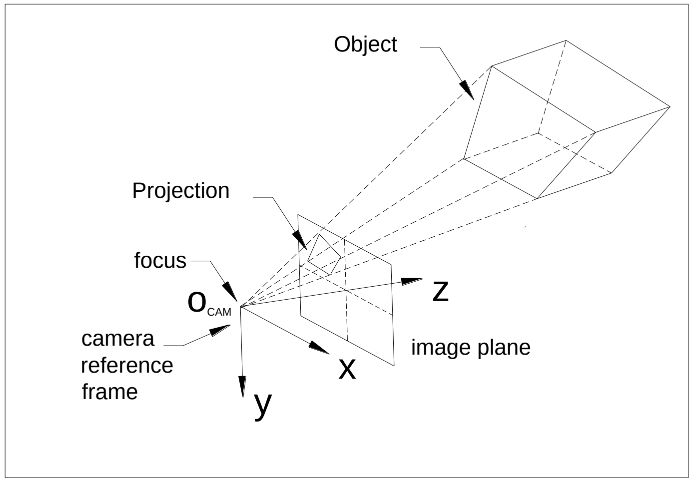


Figure 17: Pin-hole Model of a Camera

In this model $(u,v)^T$ can be defined as seen in equation (7):

$$u = \frac{Xf}{Z} \text{ and } v = \frac{Yf}{Z} \tag{7}$$

## 3. 3. Quaternion

The Euler angle representation of a rotation matrix, as defined in (53), has been used during this document as a representation of rotation. However, this convention is not unique and can cause ambiguities. Representation with a quaternion can be used to solve this issue, the quaternion can be seen in equation (8):

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{8}$$

In this equation $[O_x, O_y, O_z]$ define the origin of the target frame with respect to the camera frame. $[q_1, q_2, q_3, q_0]$ are used in order to find the unit vector, V, with respect to the camera frame that the target frame is rotated about, along with the amount of rotation θ in radians on the range of [-π,π]. Equation (9) and equation (10) show how to obtain the vector and rotation amount (53) (54).

$$[q_1, q_2, q_3] = \vec{V} * \sin\left(\frac{\theta}{2}\right) \tag{9}$$

$$q_0 = cos\left(\frac{\theta}{2}\right) \tag{10}$$

The transformation matrix can also be found from a set of quaternions using equation (11).

$$T = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) & x \\ 2(q_0q_3 + q_1q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) & y \\ 2(q_1q_3 - q_0q_2) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

## 3. 4. Overview of FEDALPE

The three methods used involve corners or points, lines, and circles or ellipses. In all of the proposed methods the FEDALPE approach discussed in (3) is used. In the Feature Extraction section, every feature of a certain type may be extracted from the image. This will contain an abundance of features, in which the desired features will be contained. An example of this may be seen in Figure 18:

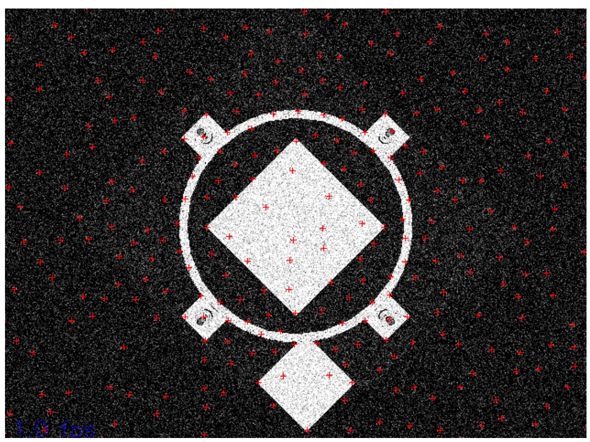

Figure 18: Extracted Corners

During the detection phase the extracted features are passed through various tests and thresholds in order to determine the correct features to track. Additional tests are done in order to tell if these features correspond to the previous time step.  An example of this can be seen in Figure 19.

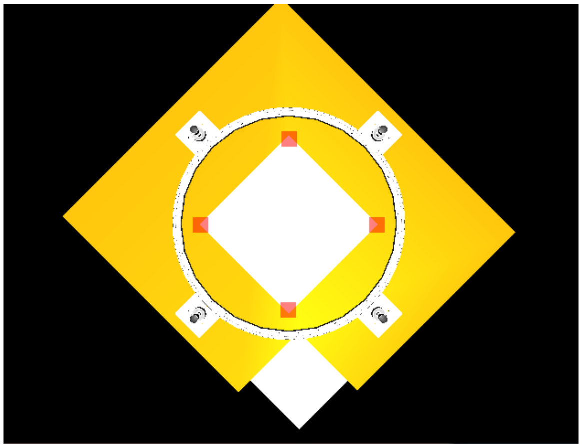

Figure 19: Detected Corners

The labeled features then go through a position estimator, which is specific for the type of feature, in order to determine the position of the target with respect to the camera as seen in Figure 20: FEDALPE Overview

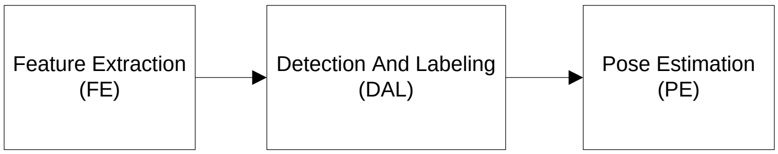


**Figure 20: FEDALPE Overview**

## 3. 5. Corner Feature Detection System

### 3. 5. 1. Feature Extraction

There are many different types of corner extractors such as the Harris, Susan, Fast, and the Shi and Tomasi (28). The corner detection used in this effort was the Shi and Tomasi method. The input of this method is a grayscale 2-D image. Figure 21 shows each of the respective colors (red, green, and blue);

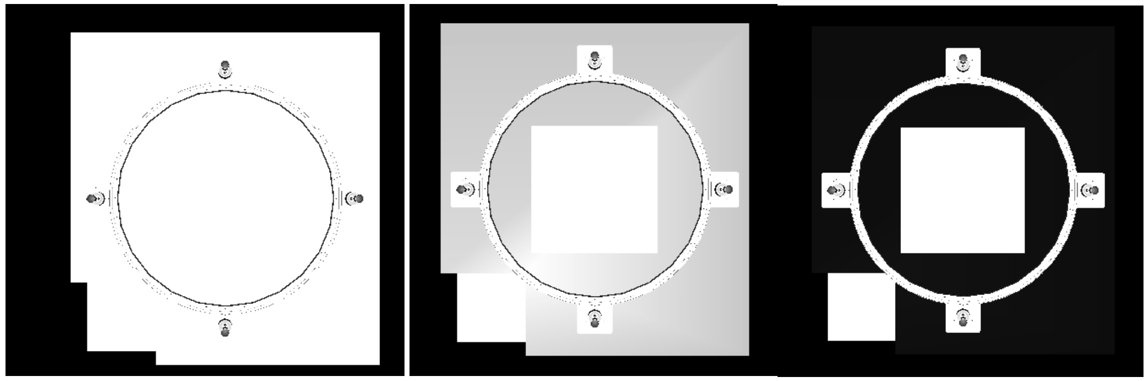

Figure 21: Individual Components of Image (R-G-B)

From this image, patches of the image called kernels are formed from small groups of pixels (3-9). If the direction derivatives are calculated inside of the kernels, a matrix of intensity derivatives can be found as seen in equation (12):

$$M = \begin{bmatrix} I_x^2 & I_{xy} \\ I_{yx} & I_y^2 \end{bmatrix} \tag{12}$$

In this equation M is the matrix of intensity derivatives, I is the intensity of the image kernel, and $I_{xx}$, $I_y$, $I_{xy}$, and $I_{yx}$ are the direction derivatives. The eigenvalues of M are then found. Using this information, the following conclusions are made:

1) If $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$ then no point of interest has been found
2) If $\lambda_1 \approx 0$ and $\lambda_2$ is a large positve number or $\lambda_1$ is a large positve number and $\lambda_2 \approx 0$ then an edge has been found
3) If $\lambda_1$ is a large positive number and $\lambda_2$ is a large positive number then a corner has been found

Using these conclusions, if the minimum of $\lambda_1$ and $\lambda_2$ is calculated to be above a threshold, λ, a corner is declared. The lower the threshold λ, is set, the fewer corners will be extracted. This can be seen in equation (13).

$$\min(\lambda_1, \lambda_2) > \lambda \tag{13}$$

Figure 22 shows all the corners that are found in the blue channel with no noise added (55) (56).

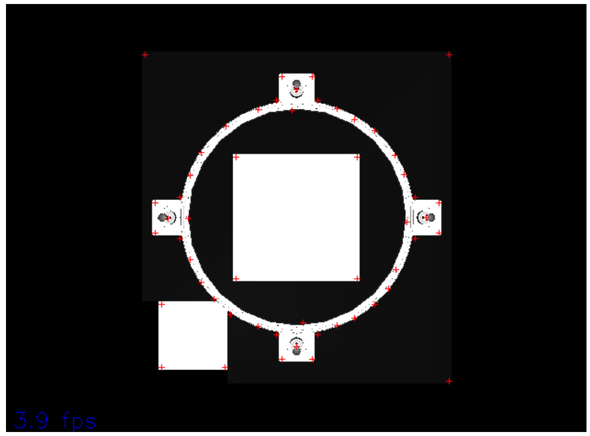


Figure 22: Extracted Corners

The output of this system is the location in image plane (in pixels) of the extracted corners.

### 3. 5. 2. Detection and Labeling

The detector and labeling system uses the previous time steps transformation matrix and the known position of the corners with respect to the origin of the target for correctly detecting and labeling the corners. Using these two values the location of the points with respect to the camera frame can be found using equation (14):

$$Tmatrix * [x_i \; y_i \; z_i \; w_i]_{target} = [x_i \; y_i \; z_i \; w_i]_{camera} \tag{14}$$

In this equation [x,y,z] is the location of the point, and w is equal to 1. The coordinate system that is being used for the associated group of points is denoted by 'target' or 'camera'. Using the standard pin-hole based model the location of the points on the image frame can be found. This can be seen in (7). The point closest to the previous time steps point is denoted as the correct point. This can be seen in equation (15):

$$min\left([u_i \; v_i\,] - \left[u_{ref} \; v_{ref}\,\right]\right) = Detected\ Point \tag{15}$$

In this equation [u,v] refers the point of the corner in the image plane, while 'i' refers to the corner being compared to the reference ('ref').

### 3. 5. 3. Position Estimation

The position estimation system uses the GLSDC (Gaussian Least Square) algorithm. The GLSDC algorithm is an iterative algorithm that minimizes a cost function based on the projected and detected locations of the corners, which can be seen in equation (16).

$$\bar{X}_{i+1} = \bar{X}_i + R_i^{-1} * A_i^T * W * \Delta G_i \tag{16}$$

where

$$R_i = A_i^T * W * A_i \text{, where } A_i = \left.\frac{\partial G_i}{\partial X}\right|_{X=\bar{X}_i} \tag{17}$$

and

$$\Delta G_i = G_{Detected} - G_{Projected} \tag{18}$$

W is usually set to the covariance matrix of the estimation error and $X_i$ is the previous time step's final estimation. For the first time step, an initial condition is used. For more information on this system, see (3).

## 3. 6. Line Feature Detection System

### 3. 6. 1. Feature Extraction

The Feature Extraction of the line detector is done by using an intensity (grayscale) image and an edge detector. In order to get all edges in the image the Canny edge detector was used. These edges are then put through the Hough transform in order to get 'r' and 'θ'. 'r' represents the distance of the line away from the center of the line, while 'θ' represents the slope of the line (see Figure 23).

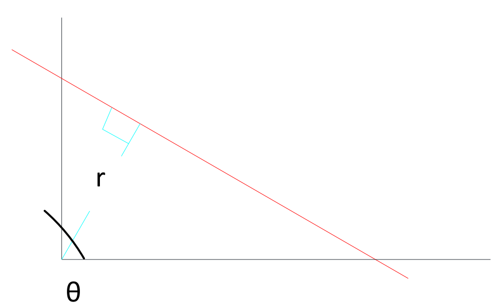


Figure 23: Explanation of r and θ

Each of the highest values of the Hough transform can be assumed to be a locally strong line. The locally strong parts of the Hough transform can then be used to find all possible lines. An example of all possible line configurations can be seen in Figure 24.

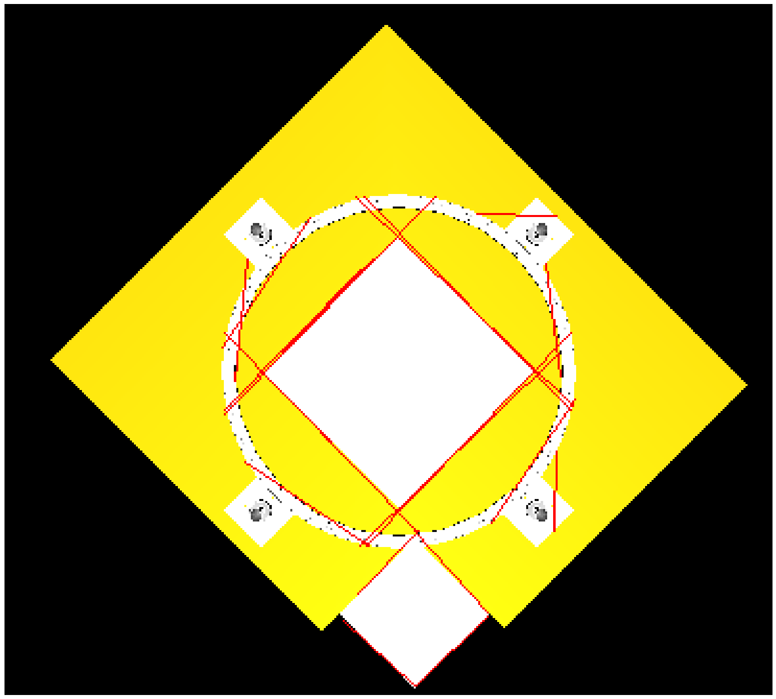

Figure 24: Extracted Lines

The OpenCV function 'HoughLinesP' was used to extract all possible lines. This function has an input of a binary image, and outputs all extracted lines. It also allows for additional parameters such as the minimum length a line must be and the maximum line gap. Line gap considers the maximum length that two points of the same line may be in order to link them to the same line.

### 3. 6. 2. Detection and Labeling

The goal of the detection and labeling system is to give the intersections of the four lines, describing the square plate, in the world frame to the position estimator. In order to detect and label the lines correctly the estimated transformation matrix, T, of the previous time step is used. From the previous time step, the slope and intercept with the v-axis (of the image plane) of the previous time step lines are calculated. In order to do this a point on the line (x0, y0, z0) must be known along with the parallel vector to the line (In target frame). These values are then projected on the camera frame by using equation (19).

$$V_C = V_T * T \tag{19}$$

$$P_T = T$$

where

$$P_T = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \tag{20}$$

is a point in the line, and

$$V_T = \begin{bmatrix} V_x \\ V_y \\ V_z \\ 0 \end{bmatrix} \tag{21}$$

is a parallel vector to the line. The equation of the line, which represents a generic point in the line, is then,

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = t \begin{bmatrix} V_x \\ V_y \\ V_z \\ 0 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \begin{bmatrix} tV_x + x_0 \\ tV_y + y_0 \\ tV_z + z_0 \\ 0 \end{bmatrix} \tag{22}$$

Using the camera model, the equation of the line becomes,

$$u = f * \frac{x}{z} + \ c_x = f * \frac{V_x * t + x_0}{V_z * y + z_0} + c_x \tag{23}$$

$$v = f * \frac{y}{z} + \ c_y = f * \frac{V_y * t + y_0}{V_z * y + z_0} + c_y \tag{24}$$

where ($x_0$, $y_0$, $z_0$) is a particular point on the line and ($c_x$, $c_y$) are the principle point which describes the center of the image plane. If u = 0 then v = b, then b can be solved for by using equation (25),

$$b = f * \frac{V_y * t + y_0}{V_z * t + z_0} + c_y \tag{25}$$

where

$$t = -\frac{f * x_0 + z_0 * c_x}{V_x * f + V_z * c_x} \tag{26}$$

If v = 0 then m = -b/u, this can be seen in equation (27)

$$m = -\frac{b}{f} * \frac{V_x * t + x_0}{V_z * t + z_0} + c_x \tag{27}$$

Where

$$t = -\frac{f * y_0 + z_0 * c_y}{V_y * f + V_z * c_y} \tag{28}$$

Given the slope, m, and the v-intercept, b, of the four lines of the rectangular feature, their intersection on the image plane can be found for calculating position as in equation (29) and (30).

$$x = \frac{b_{i+1} - b_i}{m_i - m_{i+1}} \tag{29}$$

$$y = m_i * x + b_i \tag{30}$$

Where i represents the four different lines and $m_i$ and $y_i$ are the slope and intersection of line i. The x and y locations on the camera frame are given to the position estimator in order to calculate position. The lines and intersections are labeled as seen in Figure 25 and Figure 26.

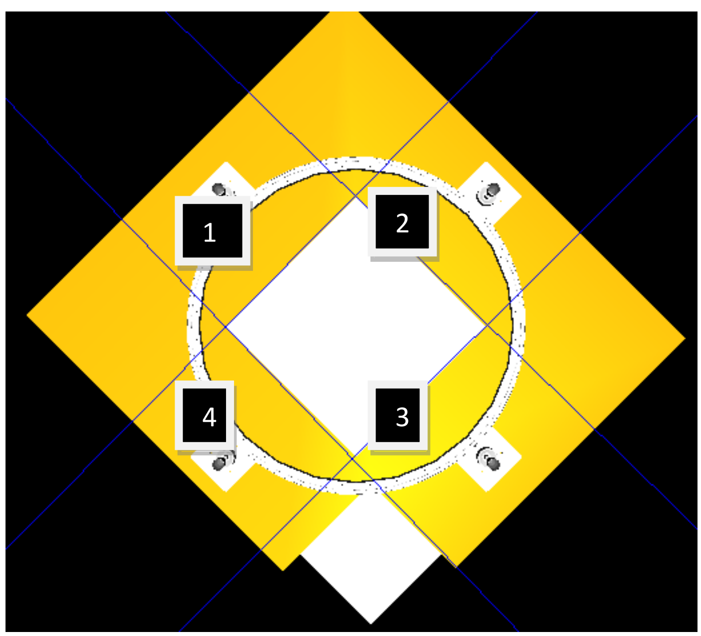


Figure 25: Labeling Numbers of Lines

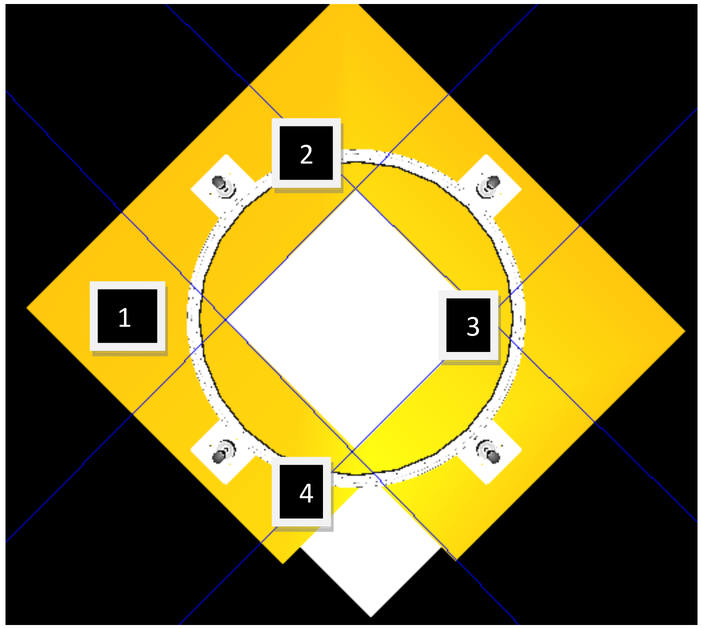


Figure 26: Labeling Numbers of Intersecting Lines

### 3. 6. 3. Position Estimation

In order to find pose of the line system, a two step approach is used. First the coordinates of the line intersections in the world frame are found; next, the transformation matrix of the target with respect to camera using those coordinates is calculated. In order to calculate the coordinates of the intersection points are redefined as seen in equation (31).

$$z_i = t_i \qquad (31)$$

$$x_i = \frac{t_i}{f} u_i$$

$$y_i = \frac{t_i}{f} v_i$$

In this equation, 'i' represents one of the four intersection points, u and v are the coordinates in the image plane of the corresponding intersection 3d points. Using the camera model and the definition of a rectangle, t can be solved for by using equation (32) (52).

$$\begin{bmatrix} -\frac{u1}{f} & \frac{u2}{f} & -\frac{u3}{f} & \frac{u4}{f} \\ -\frac{v1}{f} & \frac{v2}{f} & -\frac{v3}{f} & \frac{v4}{f} \\ -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t1 \\ t2 \\ t3 \\ t4 \end{bmatrix} = \lambda \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{32}$$

where

$$\lambda^2 = \frac{f^2 d^2}{(\theta 2u2 - \theta 1u1)^2 + (\theta 2v2 - \theta 1v1)^2 + f^2(\theta 2 - \theta 1)^2} \tag{33}$$

The least squares method can be used to find the relationship of the projected points on the target frame. This can be modeled with respect to the 3d points on the camera frame and can be seen in (34).

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}_i = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ z_t \\ 1 \end{bmatrix}_i \tag{34}$$

Where $x_t$ is 0 due to the target being in the y-z plane and 'i' being the corner number. This can be expanded for the 4 corners simultaneously and expressing the components of the transformation matrix as the unknown vector.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ x_3 \\ y_3 \\ z_3 \\ x_4 \\ y_4 \\ z_4 \end{bmatrix} = [A_{LS}] \begin{bmatrix} o_x \\ a_x \\ p_x \\ o_x \\ o_y \\ p_y \\ o_z \\ a_z \\ p_z \end{bmatrix} \tag{35}$$

where

$$A_{LS} = \begin{bmatrix} y_1' & z_1' & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & y_1' & z_1' & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & y_1' & z_1' & 1 \\ y_2' & z_2' & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & y_2' & z_2' & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & y_2' & z_2' & 1 \\ y_3' & z_3' & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & y_3' & z_3' & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & y_3' & z_3' & 1 \\ y_4' & z_4' & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & y_4' & z_4' & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & y_4' & z_4' & 1 \end{bmatrix} \quad \textbf{(36)}$$

[$x_c$, $y_c$, $z_c$] are the coordinates of the corners of the rectangle with respect to the camera, while $x_t$, $y_t$, $z_t$ in the target frame. The above equations lead to the calculations of the transformation and the rotation components by taking the pseudo inverse, as shown in equation (37).

$$\begin{bmatrix} o_x \\ a_x \\ p_x \\ o_x \\ o_y \\ p_y \\ o_z \\ a_z \\ p_z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{camera\ frame} * pinv(A_{LS}) \quad \textbf{(37)}$$

In order to get the first column of the rotation matrix the (38) is used:

$$\begin{bmatrix} n_x \\ n_y \\ n_y \end{bmatrix} = \frac{cross\left(\begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix}, \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}\right)}{norm(cross\left(\begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix}, \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}\right))} \quad \textbf{(38)}$$

A single value decomposition step is added to make sure the columns of the rotation matrix are orthonormal.

## 3. 7. Ellipse Feature Detection System

### 3. 7. 1. Feature Extraction

The input of the ellipse extraction process is the edge binary map of the image, obtained typically by a Canny edge detector. The fast ellipse extraction system (57) is used, which uses the edge map to do a four stage process which encompasses line extraction, arc extraction, extended arc extraction, and finally ellipse extraction. During the first stage, segments of at least two adjacent pixels from the edge map are combined in order to create lines. These lines can be described by their starting and ending positions, midpoint, and slope.

Lines inside of a defined kernel size are then merged in order to create arcs. In order to create an arc, two lines arcs passed through two tests. The first test compares the intersection angle of the two lines; this value must be between zero and forty five. The tangential error is then calculated and must be less than a certain threshold. Figure 27 and Figure 28 show examples of the intersection and tangential errors.

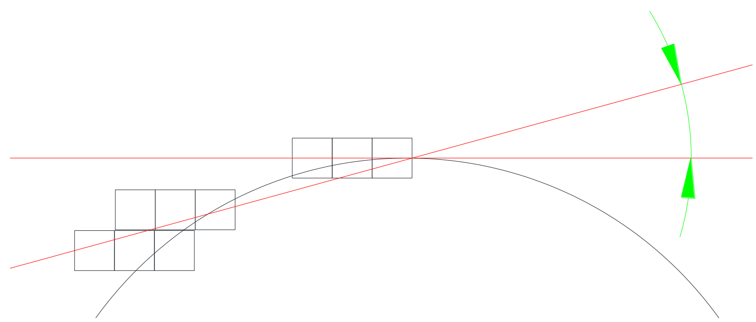

Figure 27: Intersection Error for Arcs

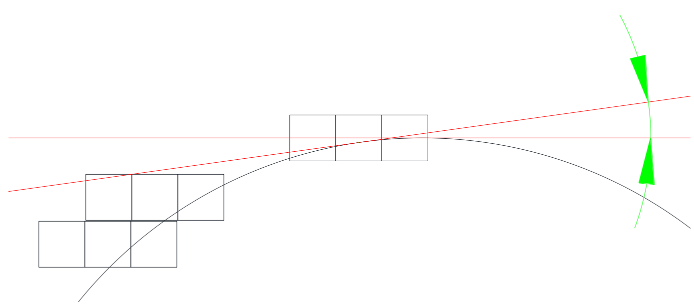

Figure 28: Tangential Error for Arcs

These arcs are very small, making them inadequate for accurate ellipse estimation. Due to this reason they are combined in order to create extended arcs. These extended arcs are created by comparing the absolute distance, relative distance, gap angles, inner angles, tangential error and line beams of the two arcs.

Merging similar and overlapping extended arcs are then used to create ellipses. The tangential error, line beams, and ellipse contour are compared in order to combine more ellipses which can then be used to calculate their center points ($x_E$,$y_E$), major semi-axis (a), minor semi-axis ($b_e$), and an orientation angle (α) (57). A summary of this process can be seen in Figure 29. The coefficients of multiple extracted ellipses are the output of the extraction system.

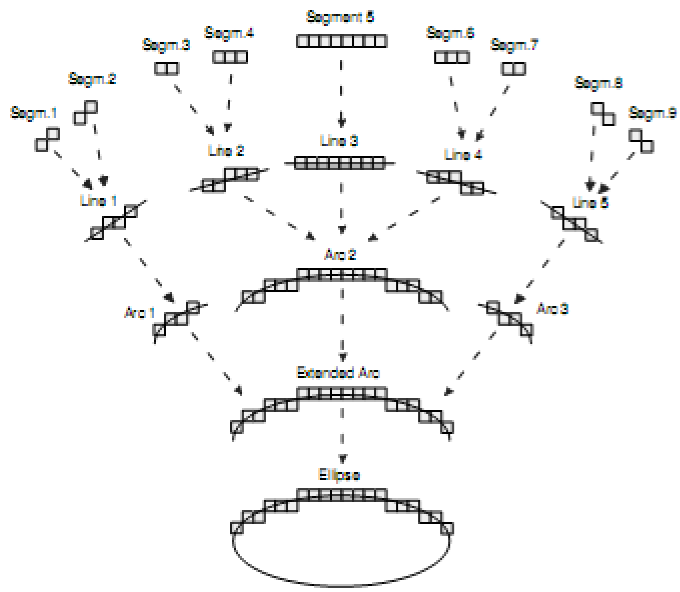


Figure 29: Overview of Ellipse Extraction **(57)**

### 3. 7. 2. Detection and Labeling

The inputs of the detection system are the geometric parameters of the extracted ellipses along with the geometric parameters of the ellipse corresponding to the projection of the circular feature in the image plane using the transformation matrix of the previous time. This ellipse is referred as the reference ellipse. This allows the system to compare the geometric parameters by using the equation (39).

$$E_i = (a_{ref} - a_i)^2 + (b_{e,ref} - b_{e,i})^2 + (x_{ref} - x_i)^2 + (y_{ref} - y_i)^2 \qquad \textbf{(39)}$$

In this equation, ‘ref’ refers to the geometric parameters of the reference ellipse, while ‘i’ is the extracted ellipse. The ellipse with the smallest value of the cost function is then declared as the detected ellipse and its geometry parameters are used to analytically find its quadratic equation on the image plane. The coefficients of the equation of the detected ellipse are the output of the detection system

### 3. 7. 3. Position Estimation

The quadratic equation of the detected ellipse and the method described in (48) are used to find the normal vector and the coordinates of the center of the ring with respect to the camera frame. In order to do this the equation in the camera frame is found using the detected ellipse equation. After the equation in the camera frame is found, a new reference frame is found in which the equation of the elliptical cone has the canonical form. Next, the two planes that intersect the cone in circles of the same diameter of the ring can be found. These steps can be seen below in Figure 30.

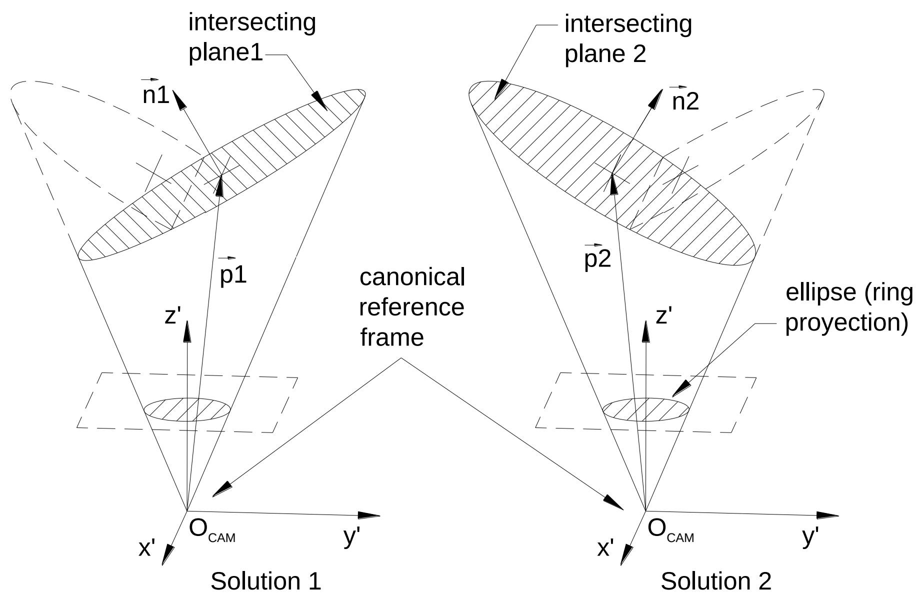


Figure 30: Canonical reference frame and intersecting planes

There are two possible circles that can generate the same solution to the position estimation problem (two normal and two center position vectors). Finally, the found normal and position vectors are transformed back to the original camera frame. Although the method of (58) can correct for the duality problem, the duality is solved by comparing the two possible normal values with a reference normal obtained from the transformation matrix.

# 4. Experiment Setup and Procedure

With different features and satellites, one satellite could not be used in order to compare all of the FEDALPE schemes. To fairly compare the different techniques, a virtual satellite mockup (see Figure 33) was created with features from various satellites in order to have the best features from different satellites. Therefore, there will be at least one of these features on every satellite that could be grappled for servicing.

## 4. 1. Virtual Setup and Procedure

For testing, the main experiments will be performed in MATLAB by using a Virtual Reality Toolbox (VRT) (59) which allows for exact maneuvers of the satellite or camera to simulate a grappling sequence. The Virtual World consists of a two VRML files which include the model of the satellite mockup and camera respectively as seen in Figure 31.

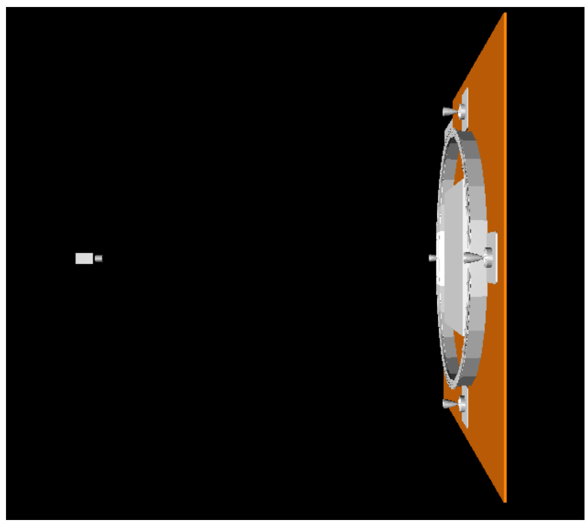

Figure 31: Virtual World

These VRML files are created using SolidWorks. The use of a virtual environment allows for knowledge of every position and orientation of each object and to compare them, and also allows changing the field of view of the camera. The simulation also features a number of graphic user interface (GUI) menus to allow the user to set the position of the camera or satellite, and motion to be performed. Some of the codes were written as Simulink s-functions (corners), while some were C++ code wrapped as an s-function. The model of the Line can be seen in Figure 32.

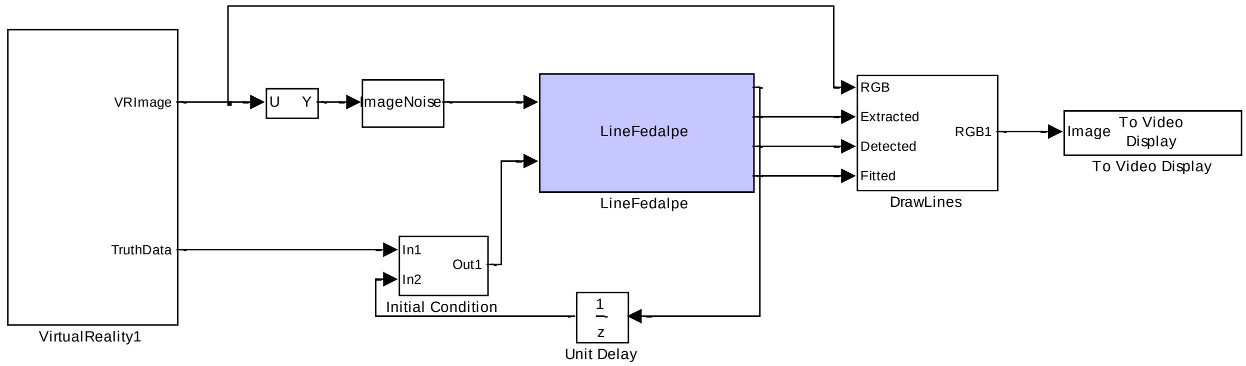


Figure 32: Simulink Line Model

## 4. 2. Satellite Mockup

The target used for the virtual reality contains a marman ring for grasping, along with a center panel which represents a control panel or heat padding. In this target, the corner and line FEDALPE's are targeting the center panel while the circle detector will target the marman ring. In order to keep the targets the same size, the marman ring was scaled 120% during the corner and line tests and the center panel was reduced 80% in the marman ring test. The corner/line and circle targets used for virtual reality are shown below in Figure 33.

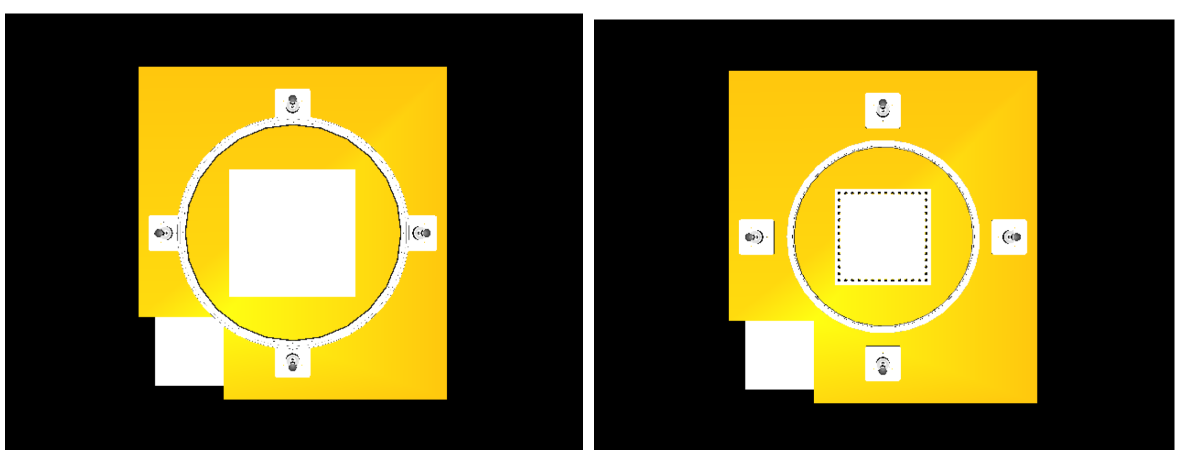

Figure 33: Examples of Satellite Targets

## 4. 3. Tests

Using the targets shown above, a static and dynamic test was performed. The static tests included movements in each degree of freedom, while the dynamic test contained an approach to have a more realistic grappling sequence. The movement in each direction has allowed for comparison between the movements in order to determine if there are movements that are related to certain errors, along with determining if there is a change in performance as the target become larger in the image plane. This has also allowed for comparisons between the algorithms in terms of performance. This is done by comparing the mean and standard deviation rotation and translational errors. In addition to these tests, the co-variance and the variance of the signal will be analyzed; this shows if any signals are related along with the dispersion of the signal. The dynamic tests allow for a qualitative analysis on how the system could work in real life.

The camera was placed orthogonally with respect to the satellite for the static tests at a far distance (3 meters), close distance (2 meters), and near distance (1.2 meters). In these tests, the deflections involved 10 centimeters for translational movements and 10 degrees for rotational movements. The resolution was fixed at a standard VGA resolution (640x460 pixels). The field of view of the camera was set at 70 degrees. Figure 34 below shows a side view of the close static test.

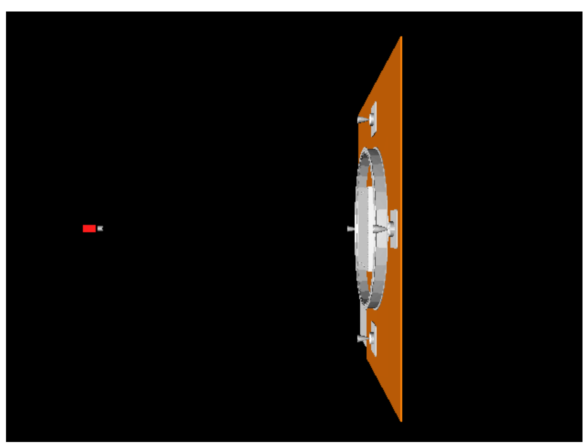

Figure 34: Side view of static tests

For the approach the camera started at 2.5 meters away and off center from the target with minor angular deflections, and approached to 0.5 meters away with rotational deflections in order to center the target in the screen. The resolution and field of view for the camera are allowed to vary in order to let the system work as best as possible.

All of these tests have included 3 different levels of noise in order to try and simulate more realistic scenarios. The Gaussian noise that was added could be caused from electronic noise from the robotic system. This type of noise is modeled with a Gaussian curve by using a mean and variance. For all the tests the mean was set to zero while the variance is varied from 0 (no noise) to 0.05 (low level of noise) to 0.1 (high level of noise). These values were found by evaluating the systems to find what levels the systems would work best. The results of the noise can be seen below in Figure 35, Figure 36, and Figure 37, both the RGB image and in the result of the canny detector. All of the tests can be seen in the test matrix shown in Table 2. All tests will be named in the convention of Case Algorithm_NoiseLevel_Position_Movement, with a '?' denoting the variable that will be changed.

Table 2: Test Matrix

| Algorithm | Noise Level | Position | Movement |
|---|---|---|---|
| Corner | No Noise | 3 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 1.2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | Low Noise | 3 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 1.2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | High Noise | 3 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 1.2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| Line | No Noise | 3 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 1.2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | Low Noise | 3 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 1.2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | High Noise | 3 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 1.2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| Ellipse | No Noise | 3 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 1.2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | Low Noise | 3 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 1.2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | High Noise | 3 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |
| | | 1.2 m | X-10 cm/Y-10 cm/Z-10cm/rX-10 deg/rY-10 deg/rZ-10deg |

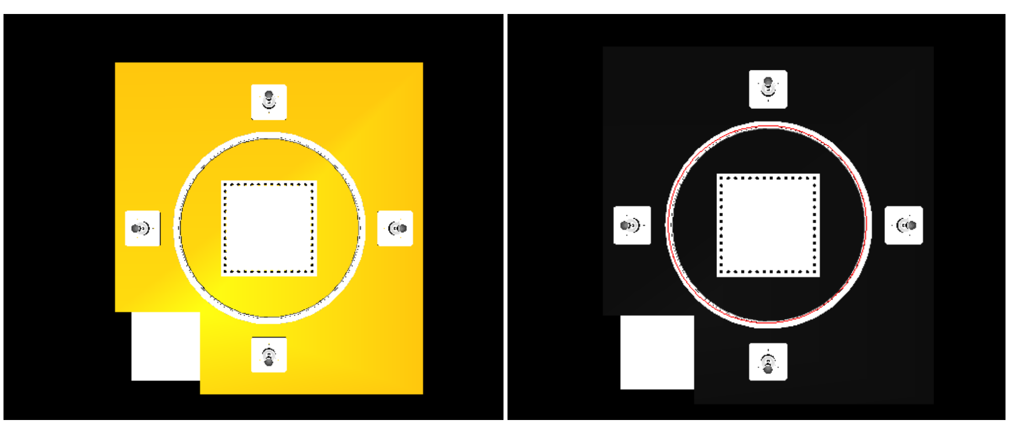

Figure 35: RGB Image and Black/White (BW) Image (No Noise)

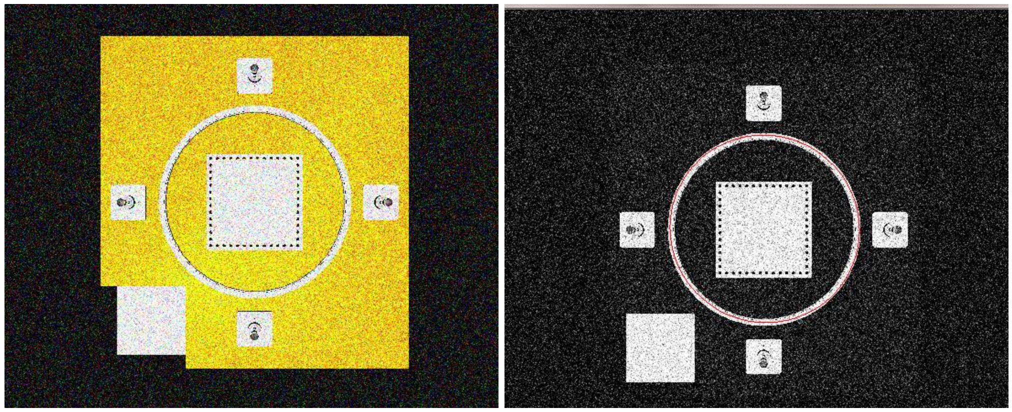

Figure 36: RGB Image and BW Image (Low Noise)

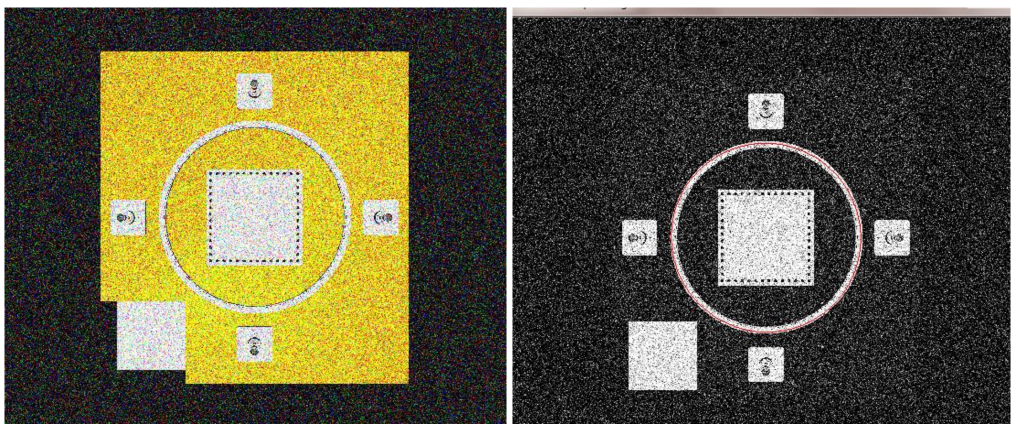

Figure 37: RGB Image and BW Image (High Noise)

# 5. Results

The sum of squares of individual error in the translational component is used to calculate the translation error. This can be seen in equation (40).

$$Translation\ Error = \sqrt{(p_{x,truth} - p_{x,estimated})^2 + (p_{y,truth} - p_{y,estimated})^2 + (p_{z,truth} - p_{z,estimated})^2} \quad \textbf{(40)}$$

In order to calculate the rotation error for the lines and corners the following equations will be used:

$$Rotation\ Error\ Matrix = R_{truth} * (R_{estimated})^{-1} \quad \textbf{(41)}$$

This gives the difference between rotation matrices, which can then be converted into roll, pitch, and yaw. Using the errors of roll, pitch, and yaw the rotation error is calculated using equation (42).

$$Rotation\ Error = \sqrt{(r_{e,i})^2 + (p_{e,i})^2 + (y_{e,i})^2} \quad \textbf{(42)}$$

The rotation error for ellipses cannot be calculated with the same approach since not all the DOF are observable. In this scenario equation (43) will be used in order to calculate rotation error.

$$Rotation\ Error = \operatorname{asin}\left\{cross\left(\begin{bmatrix} n_{x,truth} \\ n_{y,truth} \\ n_{z,truth} \end{bmatrix}, \begin{bmatrix} n_{x,estimated} \\ n_{y,estimated} \\ n_{z,estimated} \end{bmatrix}\right)\right\} \quad \textbf{(43)}$$

## 5. 1. Corner Results

The corner system results in the case of 'no noise' and various positions are shown in Table 3. In this table it, all tests that as the positions of the camera become close to the target the rotation and translational error is reduced from approximately 9 cm to about 2 cm. Also in Table 5, if noise is introduced the rotational error is increased from approximately 3 degrees to about 5 degrees. This system remained to be between around 9 centimeters to around 2 cm of accuracy for all the levels of noise, along with less than 5 degrees of rotational error at all levels. There appears to be no motion that greatly affects the system.

| Case<br>Corners<br>No Noise<br>Movements | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
|---|---|---|---|---|---|---|---|---|
| **3 [m]** | $\mu_{trans\ error}$ [m] | 0.09592 | 0.0978 | 0.09157 | 0.07170 | 0.11128 | 0.09875 | 0.07470 |
| | $\sigma_{trans\ error}$ [m] | 0 | 0 | 0 | 0.01643 | 0 | 0 | 0 |
| | $\mu_{rot\ error}$ [deg] | 3.34657 | 3.4719 | 1.22226 | 2.48333 | 1.98249 | 3.51753 | 0.13967 |
| | $\sigma_{rot\ error}$ [deg] | 0 | 0 | 0 | 1.22009 | 0 | 0 | 0 |
| **2 [m]** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [m] | 0.06032 | 0.0502 | 0.04410 | 0.04496 | 0.05040 | 0.05049 | 0.05072 |
| | $\sigma_{trans\ error}$ [m] | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $\mu_{rot\ error}$ [deg] | 1.15254 | 1.1013 | 1.09331 | 1.16844 | 0.31311 | 1.17765 | 1.66725 |
| | $\sigma_{rot\ error}$ [deg] | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1.2 [m] (grasping)** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [m] | 0.02258 | 0.0228 | 0.02300 | 0.02358 | 0.02746 | 0.02083 | 0.02260 |
| | $\sigma_{trans\ error}$ [m] | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | $\mu_{rot\ error}$ [deg] | 0.00268 | 0.5461 | 0.36954 | 0.29918 | 0.09699 | 0.11344 | 0.05218 |
| | $\sigma_{rot\ error}$ [deg] | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 3: Corners with No Noise vs. Position

| Corners Low Noise Movements | Parameters | Base Position | 10deg Rx | 10deg Ry | 10deg Rz | 0.1m dx | 0.1m dy | 0.1m dz |
|---|---|---|---|---|---|---|---|---|
| **3 [m]** | $\mu_{trans\ error}$ [mm] | 0.06909 | 0.08364 | 0.071704 | 0.072299 | 0.092445 | 0.070068 | 0.064898 |
| | $\sigma_{trans\ error}$ [mm] | 0.01586 | 0.01405 | 0.016435 | 0.015059 | 0.015725 | 0.011903 | 0.014417 |
| | $\mu_{rot\ error}$ [deg] | 3.63334 | 3.16950 | 2.483338 | 2.387058 | 3.282826 | 3.382961 | 2.880761 |
| | $\sigma_{rot\ error}$ [deg] | 2.02047 | 1.42979 | 1.220091 | 1.125884 | 1.915566 | 1.758372 | 1.629985 |
| **2 [m]** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.04774 | 0.04300 | 0.038346 | 0.039331 | 0.037681 | 0.044287 | 0.043695 |
| | $\sigma_{trans\ error}$ [mm] | 0.00606 | 0.00627 | 0.006336 | 0.007029 | 0.007179 | 0.00638 | 0.007419 |
| | $\mu_{rot\ error}$ [deg] | 1.35653 | 1.67205 | 1.260986 | 1.271136 | 1.596232 | 1.428215 | 1.309243 |
| | $\sigma_{rot\ error}$ [deg] | 0.82139 | 0.86285 | 0.596061 | 0.56427 | 0.896295 | 0.735669 | 0.639325 |
| **1.2 [m] (grasping)** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.02064 | 0.02186 | 0.021025 | 0.020958 | 0.024355 | 0.018438 | 0.022082 |
| | $\sigma_{trans\ error}$ [mm] | 0.00198 | 0.00205 | 0.002092 | 0.002181 | 0.002519 | 0.002057 | 0.002351 |
| | $\mu_{rot\ error}$ [deg] | 0.46762 | 0.50740 | 0.493931 | 0.46041 | 0.596052 | 0.474587 | 0.485952 |
| | $\sigma_{rot\ error}$ [deg] | 0.26233 | 0.26171 | 0.231435 | 0.212846 | 0.328394 | 0.246261 | 0.2448 |

Table 4: Corners with Low Noise vs. Position

| Corners High Noise Movements | Parameters | Base Position | 10deg Rx | 10deg Ry | 10deg Rz | 0.1m dx | 0.1m dy | 0.1m dz |
|---|---|---|---|---|---|---|---|---|
| **3 [m]** | $\mu_{trans\ error}$ [mm] | 0.06493 | 0.08047 | 0.216361 | 0.072694 | 0.090572 | 0.068843 | 0.060349 |
| | $\sigma_{trans\ error}$ [mm] | 0.02103 | 0.01796 | 0.136774 | 0.022736 | 0.021664 | 0.01952 | 0.019111 |
| | $\mu_{rot\ error}$ [deg] | 5.48579 | 5.06262 | 22.90096 | 4.303111 | 4.589107 | 5.034612 | 4.446131 |
| | $\sigma_{rot\ error}$ [deg] | 3.89928 | 3.33057 | 17.8458 | 3.71748 | 2.93317 | 4.008632 | 2.786767 |
| **2 [m]** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.05080 | 0.04491 | 0.044101 | 0.04021 | 0.039457 | 0.041209 | 0.044243 |
| | $\sigma_{trans\ error}$ [mm] | 0.01003 | 0.01021 | 0 | 0.00777 | 0.009135 | 0.008248 | 0.00827 |
| | $\mu_{rot\ error}$ [deg] | 2.26282 | 2.39645 | 1.093315 | 1.697099 | 2.201565 | 2.0722 | 1.838207 |
| | $\sigma_{rot\ error}$ [deg] | 1.3646 | 1.59549 | 0 | 0.897402 | 2.230142 | 1.357616 | 0.913948 |
| **1.2 [m] (grasping)** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.02087 | 0.02163 | 0.020966 | 0.020292 | 0.0256 | 0.020077 | 0.247458 |
| | $\sigma_{trans\ error}$ [mm] | 0.00295 | 0.00260 | 0.003249 | 0.003048 | 0.00589 | 0.005828 | 0.022587 |
| | $\mu_{rot\ error}$ [deg] | 0.74471 | 0.62674 | 0.803212 | 0.00281 | 1.104166 | 0.934358 | 42.63634 |
| | $\sigma_{rot\ error}$ [deg] | 0.45412 | 0.34384 | 0.697228 | 0.001346 | 1.275487 | 1.551977 | 6.655634 |

Table 5: Corners with High Noise vs. Position

A sample of the z motion can be seen in Figure 38 (corresponding to Case Corner_Low Noise_ 1.2m_XMotion). This test is at 1.2 m with low noise and the satellite performing a translation in the X-direction. This shows how the system tracks throughout the motion that is performed. The error in translation can be seen in Figure 39. It can be seen that the error is slightly increased as the target moves away. However, this was not seen in every test case. This can be assumed to be due to the corner detector being in pixel accuracy. It can be seen in Figure 40 that this error can be considered to be caused from the Z direction (camera frame).

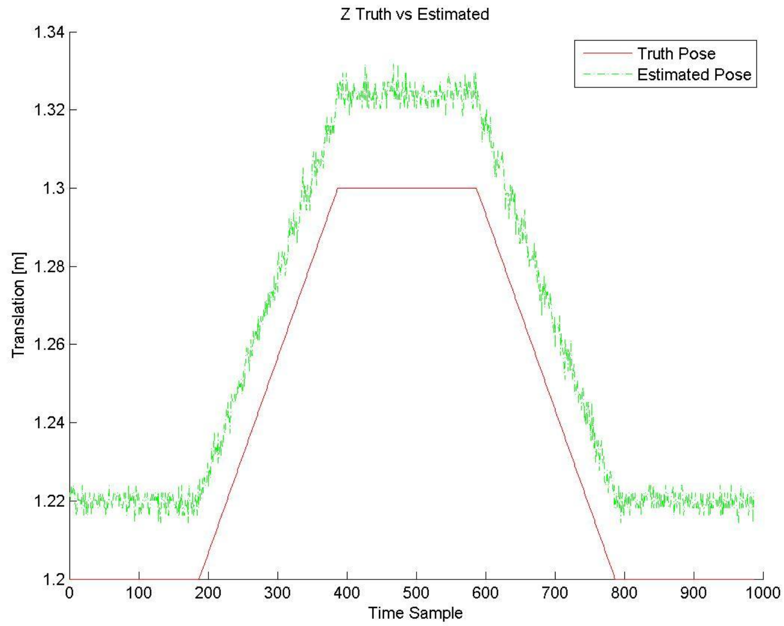


Figure 38: Z (Camera Frame) Truth vs. Estimated

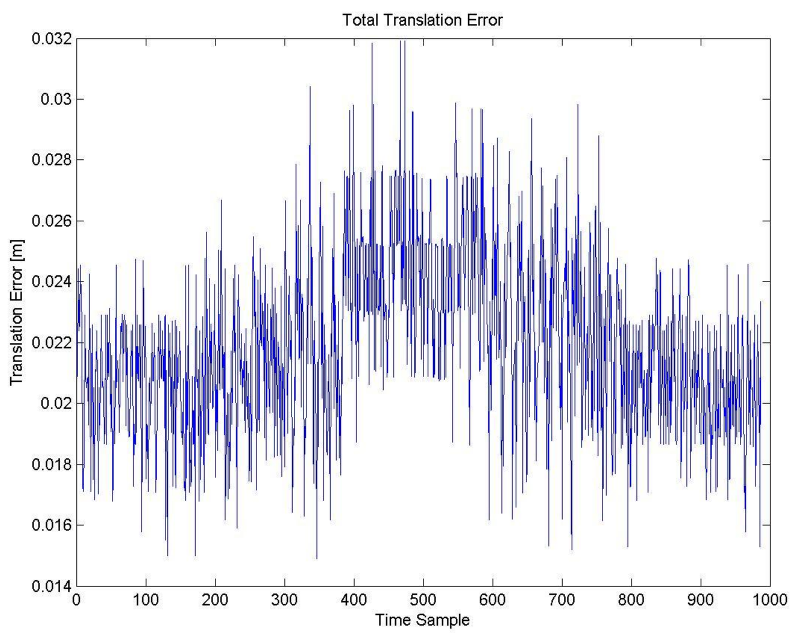


Figure 39: Translation Error

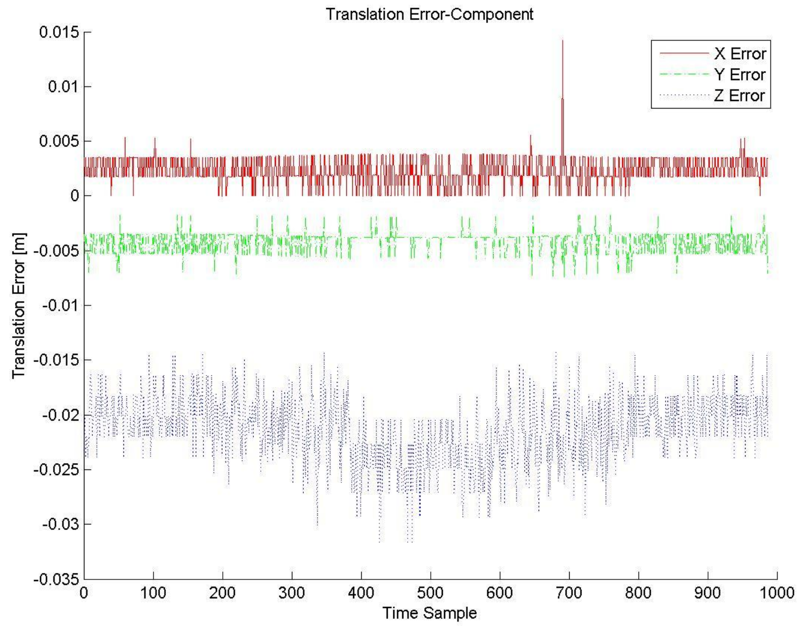


Figure 40: Translation Components

## 5. 2. Line Results

The line system data for all positions with low noise are shown below in Table 7. It can be seen that the maximum translational error is ~8 centimeters, while at the closest position it can be accurate within 1 centimeter. However, the rotational error can exceed 7 degrees.

| Lines No Noise Movements | Parameters | Base Position | 10deg Rx | 10deg Ry | 10deg Rz | 0.1m dx | 0.1m dy | 0.1m dz |
|---|---|---|---|---|---|---|---|---|
| **3 [m]** | $\mu_{trans\ error}$ [mm] | 0.04203 | 0.04380 | 0.031868 | 0.062998 | 0.044778 | 0.082891 | 0.044588 |
| | $\sigma_{trans\ error}$ [mm] | 1.12E-16 | 3.49E-17 | 6.28E-17 | 0 | 3.49E-17 | 1.12E-16 | 5.58E-17 |
| | $\mu_{rot\ error}$ [deg] | 4.47944 | 5.77361 | 4.455635 | 7.919148 | 0.766063 | 7.701367 | 10.66514 |
| | $\sigma_{rot\ error}$ [deg] | 5.36E-15 | 1.43E-14 | 1.79E-15 | 1.16E-14 | 1.12E-16 | 1.16E-14 | 1.79E-14 |
| **2 [m]** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.01021 | 0.01511 | 0.019212 | 0.020009 | 0.015073 | 0.007211 | 0.009965 |
| | $\sigma_{trans\ error}$ [mm] | 1.22E-17 | 1.92E-17 | 5.58E-17 | 1.74E-17 | 3.49E-17 | 1.57E-17 | 2.09E-17 |
| | $\mu_{rot\ error}$ [deg] | 5.34609 | 3.76689 | 4.872765 | 1.464069 | 2.727492 | 0.573797 | 1.588821 |
| | $\sigma_{rot\ error}$ [deg] | 8.03E-15 | 3.57E-15 | 6.25E-15 | 1.34E-15 | 0 | 1.12E-15 | 4.46E-16 |
| **1.2 [m] (grasping)** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.00945 | 0.00628 | 0.007414 | 0.007558 | 0.009693 | 0.008092 | 0.008964 |
| | $\sigma_{trans\ error}$ [mm] | 0.00212 | 6.1E-18 | 7.85E-18 | 1.83E-17 | 0.002143 | 8.72E-18 | 1.22E-17 |
| | $\mu_{rot\ error}$ [deg] | 1.34174 | 0.65889 | 0.786813 | 0.795181 | 1.941729 | 0.276222 | 1.772649 |
| | $\sigma_{rot\ error}$ [deg] | 0.58136 | 7.81E-16 | 5.58E-16 | 7.81E-16 | 0.456553 | 4.46E-16 | 2.01E-15 |

Table 6: Lines with No Noise vs. Position

| Lines Low Noise Movements | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
|---|---|---|---|---|---|---|---|---|
| **3 [m]** | $\mu_{trans\ error}$ [mm] | 0.06346 | 0.07900 | 0.081784 | 0.063703 | 0.075561 | 0.070229 | 0.078802 |
| | $\sigma_{trans\ error}$ [mm] | 0.04284 | 0.05053 | 0.053728 | 0.044469 | 0.045539 | 0.042946 | 0.039374 |
| | $\mu_{rot\ error}$ [deg] | 5.79897 | 7.12050 | 4.87669 | 5.900335 | 5.400128 | 5.830717 | 6.103323 |
| | $\sigma_{rot\ error}$ [deg] | 3.07069 | 3.37684 | 2.860308 | 2.573246 | 2.671141 | 2.692735 | 2.808617 |
| **2 [m]** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.01208 | 0.01952 | 0.016369 | 0.023356 | 0.01734 | 0.013385 | 0.012549 |
| | $\sigma_{trans\ error}$ [mm] | 0.00376 | 0.00582 | 0.006182 | 0.00975 | 0.007408 | 0.004896 | 0.004336 |
| | $\mu_{rot\ error}$ [deg] | 3.68576 | 3.38516 | 3.354849 | 4.744756 | 3.672168 | 3.690321 | 3.772223 |
| | $\sigma_{rot\ error}$ [deg] | 1.51862 | 1.54081 | 1.412255 | 1.577316 | 1.567295 | 1.407731 | 1.46127 |
| **1.2 [m] (grasping)** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.00945 | 0.00879 | 0.008993 | 0.011056 | 0.009693 | 0.009766 | 0.009992 |
| | $\sigma_{trans\ error}$ [mm] | 0.00212 | 0.00172 | 0.002248 | 0.001988 | 0.002143 | 0.002515 | 0.002136 |
| | $\mu_{rot\ error}$ [deg] | 1.34174 | 0.95290 | 0.761531 | 0.816236 | 1.941729 | 0.610188 | 1.788189 |
| | $\sigma_{rot\ error}$ [deg] | 0.58136 | 0.48409 | 0.18542 | 0.206977 | 0.456553 | 0.291285 | 0.394881 |

Table 7: Lines with Low Noise vs. Position

| Lines High Noise Movements | Parameters | Base Position | 10deg Rx | 10deg Ry | 10deg Rz | 0.1m dx | 0.1m dy | 0.1m dz |
|---|---|---|---|---|---|---|---|---|
| **3 [m]** | $\mu_{trans\ error}$ [mm] | 0.05974 | 0.0765 | 0.061115 | 0.058145 | 0.062854 | 0.067172 | 0.069821 |
| | $\sigma_{trans\ error}$ [mm] | 0.03766 | 0.03366 | 0.029586 | 0.032048 | 0.033938 | 0.037835 | 0.027596 |
| | $\mu_{rot\ error}$ [deg] | 5.91663 | 7.20117 | 5.504577 | 6.401511 | 6.096729 | 6.000919 | 5.638553 |
| | $\sigma_{rot\ error}$ [deg] | 3.25404 | 3.99585 | 2.620006 | 2.884483 | 2.592894 | 3.368313 | 2.56001 |
| **2 [m]** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.01245 | 0.02139 | 0.018158 | 0.023318 | 0.01977 | 0.014736 | 0.012386 |
| | $\sigma_{trans\ error}$ [mm] | 0.00406 | 0.00596 | 0.006284 | 0.010229 | 0.007667 | 0.006045 | 0.004378 |
| | $\mu_{rot\ error}$ [deg] | 3.68897 | 3.56041 | 3.345156 | 4.635284 | 4.009568 | 3.510829 | 3.743093 |
| | $\sigma_{rot\ error}$ [deg] | 1.60286 | 1.46077 | 1.43876 | 1.798292 | 1.667298 | 1.249412 | 1.392058 |
| **1.2 [m] (grasping)** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.00947 | 0.00819 | 0.00942 | 0.01117 | 0.00927 | 0.01010 | 0.00899 |
| | $\sigma_{trans\ error}$ [mm] | 0.00217 | 0.00173 | 0.00227 | 0.00223 | 0.00219 | 0.00247 | 0.00211 |
| | $\mu_{rot\ error}$ [deg] | 1.32524 | 1.04846 | 0.79773 | 0.87036 | 2.01420 | 0.67942 | 1.7729 |
| | $\sigma_{rot\ error}$ [deg] | 0.59028 | 0.46093 | 0.20102 | 0.21322 | 0.41171 | 0.27725 | 0.39238 |

Table 8: Lines with High Noise vs. Position

The z (camera frame) estimation vs. truth data is shown for an x translation in the satellite in Figure 41 along with the error in x direction (satellite frame) in Figure 42. In most cases the error in translation, specifically in z, is increased as distance is increased.

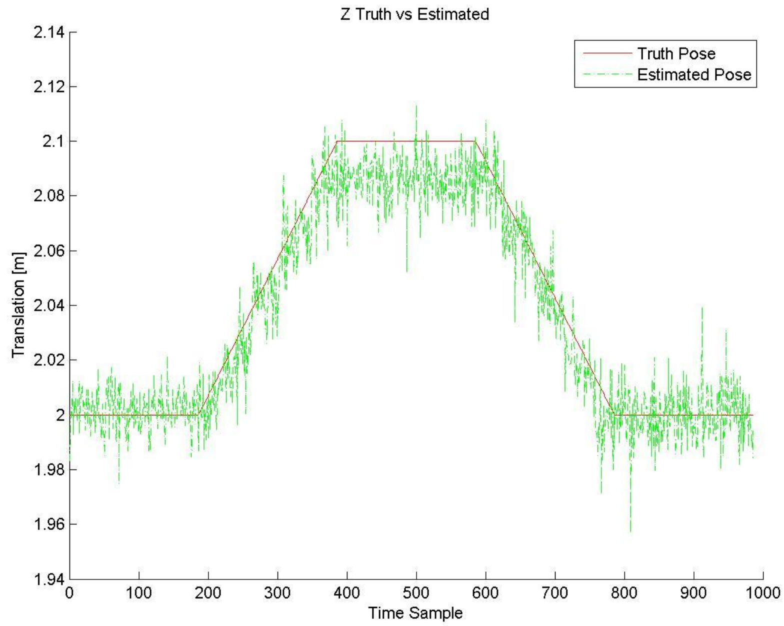


Figure 41: Z (Camera Frame) Estimation vs. Truth

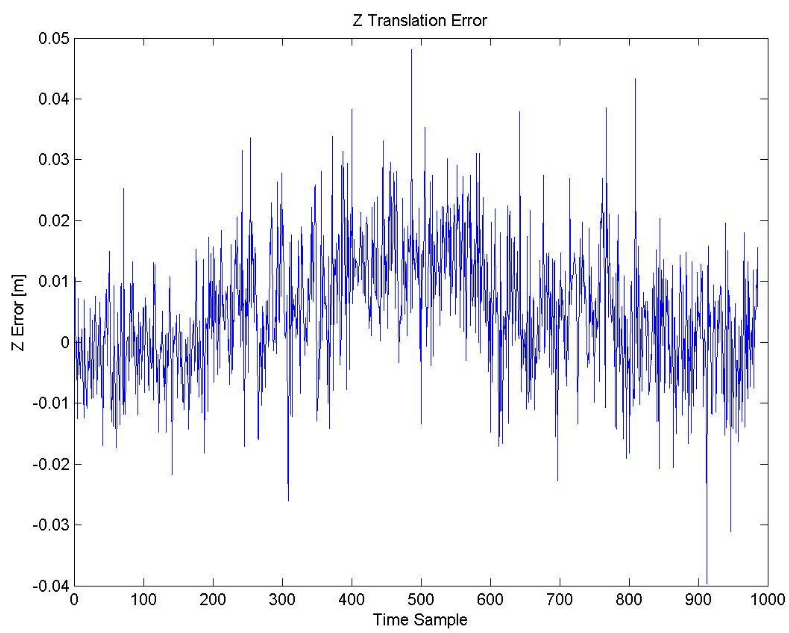


Figure 42: Translation Error in Z (Camera Frame)

## 5. 3. Ellipse Results

The ellipses system data for all positions with a low level of noise are shown below in Table 10. It can be seen that the maximum translational error is ~6 centimeters at the closest position. However, in all tests at the non-grappling places the mean error is ~4 centimeters. The increased error at the closest position is due to the ellipse not being detected for several times steps in the test. This is due to parameters in the ellipse extraction that prevents ellipses beyond an amount of pixels to be considered as an ellipse.

| Ellipses No Noise Movements | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
|---|---|---|---|---|---|---|---|---|
| **3 [m]** | $\mu_{trans\ error}$ [mm] | 0.03034 | 0.02402 | 0.01618 | 0.00433 | 0.02608 | 0.01720 | 0.02099 |
| | $\sigma_{trans\ error}$ [mm] | 6.28E-17 | 6.28E-17 | 2.09E-17 | 6.97E-18 | 2.44E-17 | 3.14E-17 | 3.14E-17 |
| | $\mu_{rot\ error}$ [deg] | 11.2377 | 10.6456 | 5.20318 | 2.44156 | 7.59366 | 9.65287 | 6.68608 |
| | $\sigma_{rot\ error}$ [deg] | 0 | 0 | 6.25E-15 | 4.46E-15 | 1.52E-14 | 1.43E-14 | 3.57E-15 |
| **2 [m]** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.03799 | 0.03825 | 0.01520 | 0.02751 | 0.03003 | 0.04171 | 0.02319 |
| | $\sigma_{trans\ error}$ [mm] | 3.49E-17 | 8.37E-17 | 0 | 2.79E-17 | 6.97E-17 | 3.49E-17 | 1.39E-17 |
| | $\mu_{rot\ error}$ [deg] | 9.70756 | 11.4963 | 2.62273 | 9.29735 | 4.81271 | 12.3559 | 8.89037 |
| | $\sigma_{rot\ error}$ [deg] | 1.25E-14 | 1.79E-15 | 3.57E-15 | 1.25E-14 | 9.82E-15 | 3.57E-15 | 1.79E-15 |
| **1.2 [m] (grasping)** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.03226 | 0.05755 | 0.100717 | 0.048055 | 0.037795 | 0.030305 | 0.052854 |
| | $\sigma_{trans\ error}$ [mm] | 1.39E-17 | 2.09E-17 | 4.18E-17 | 1.05E-16 | 1.12E-16 | 4.18E-17 | 6.97E-18 |
| | $\mu_{rot\ error}$ [deg] | 3.10548 | 12.4418 | 18.4841 | 10.6413 | 8.21647 | 5.24702 | 10.6003 |
| | $\sigma_{rot\ error}$ [deg] | 8.93E-16 | 1.96E-14 | 2.5E-14 | 1.07E-14 | 1.43E-14 | 1.07E-14 | 2.68E-14 |

Table 9: Ellipses with No Noise vs. Position

| Ellipses Low Noise Movements | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
|---|---|---|---|---|---|---|---|---|
| **3 [m]** | $\mu_{trans\ error}$ [mm] | 0.03547 | 0.03019 | 0.03747 | 0.03536 | 0.03308 | 0.03261 | 0.037523 |
| | $\sigma_{trans\ error}$ [mm] | 0.01581 | 0.01152 | 0.01771 | 0.01786 | 0.01779 | 0.01472 | 0.016034 |
| | $\mu_{rot\ error}$ [deg] | 9.69074 | 9.10668 | 14.0081 | -0.00217 | 9.31464 | 9.58376 | 9.045289 |
| | $\sigma_{rot\ error}$ [deg] | 2.93239 | 2.45774 | 8.41673 | 0.02658 | 2.54705 | 3.03689 | 3.139329 |
| **2 [m]** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.0378 | 0.03583 | 0.04103 | 0.03744 | 0.03237 | 0.03527 | 0.034209 |
| | $\sigma_{trans\ error}$ [mm] | 0.00995 | 0.01021 | 0.01460 | 0.01459 | 0.00999 | 0.01458 | 0.009736 |
| | $\mu_{rot\ error}$ [deg] | 9.92003 | 10.1619 | 13.8783 | 10.6198 | 9.62673 | 8.66742 | 10.68111 |
| | $\sigma_{rot\ error}$ [deg] | 2.77232 | 2.84864 | 6.50835 | 6.62988 | 2.80496 | 3.68271 | 3.054022 |
| **1.2 [m] (grasping)** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.05245 | 0.05928 | 0.06219 | 0.06371 | 0.04731 | 0.05711 | 0.060255 |
| | $\sigma_{trans\ error}$ [mm] | 0.01516 | 0.01387 | 0.01987 | 0.02461 | 0.01536 | 0.01898 | 0.031594 |
| | $\mu_{rot\ error}$ [deg] | 10.3840 | 11.6969 | 11.4290 | 13.4243 | 10.0219 | 10.2759 | 11.33696 |
| | $\sigma_{rot\ error}$ [deg] | 3.39923 | 2.88933 | 4.08649 | 6.26197 | 3.54158 | 3.93652 | 4.595171 |

Table 10: Ellipses with Low Noise vs. Position

| Ellipses High Noise Movements | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
|---|---|---|---|---|---|---|---|---|
| **3 [m]** | $\mu_{trans\ error}$ [mm] | 0.04292 | 0.03764 | 0.041458 | 0.03661 | 0.043465 | 0.037663 | 0.049123 |
| | $\sigma_{trans\ error}$ [mm] | 0.01927 | 0.01643 | 0.016626 | 0.019587 | 0.026103 | 0.019601 | 0.028187 |
| | $\mu_{rot\ error}$ [deg] | 10.5934 | 9.90599 | 14.98471 | 8.241418 | 10.32158 | 10.19419 | 10.23229 |
| | $\sigma_{rot\ error}$ [deg] | 3.49455 | 3.29964 | 8.296751 | 5.855199 | 2.562067 | 3.101524 | 4.389991 |
| **2 [m]** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.04546 | 0.03849 | 0.038344 | 0.044672 | 0.038163 | 0.046831 | 0.041631 |
| | $\sigma_{trans\ error}$ [mm] | 0.01996 | 0.01375 | 0.017358 | 0.016136 | 0.011903 | 0.025445 | 0.018953 |
| | $\mu_{rot\ error}$ [deg] | 10.6721 | 10.0328 | 10.69906 | 12.51614 | 10.53628 | 11.61426 | 11.2701 |
| | $\sigma_{rot\ error}$ [deg] | 3.66251 | 3.19761 | 6.993912 | 7.526111 | 2.875376 | 4.334617 | 4.004272 |
| **1.2 [m] (grasping)** | **Parameters** | **Base Position** | **10deg Rx** | **10deg Ry** | **10deg Rz** | **0.1m dx** | **0.1m dy** | **0.1m dz** |
| | $\mu_{trans\ error}$ [mm] | 0.06035 | 0.08023 | 0.06211 | 0.06632 | 0.06867 | 0.07049 | 0.06866 |
| | $\sigma_{trans\ error}$ [mm] | 0.02553 | 0.05819 | 0.03441 | 0.03362 | 0.06148 | 0.04152 | 0.04540 |
| | $\mu_{rot\ error}$ [deg] | 11.4420 | 13.9722 | 11.8924 | 13.2677 | 12.3615 | 12.7564 | 12.1372 |
| | $\sigma_{rot\ error}$ [deg] | 4.12571 | 5.94624 | 6.82778 | 6.90654 | 5.37743 | 0.03014 | 5.01971 |

Table 11: Ellipses with High Noise vs. Position

The z motion is shown for an x translation in the satellite frame in Figure 43 along with the translation error in Figure 44. It can be seen that as the distance is increased, the error remains constant. This is due to the determination of the center of the ellipse is considered to be sub-pixel accuracy.

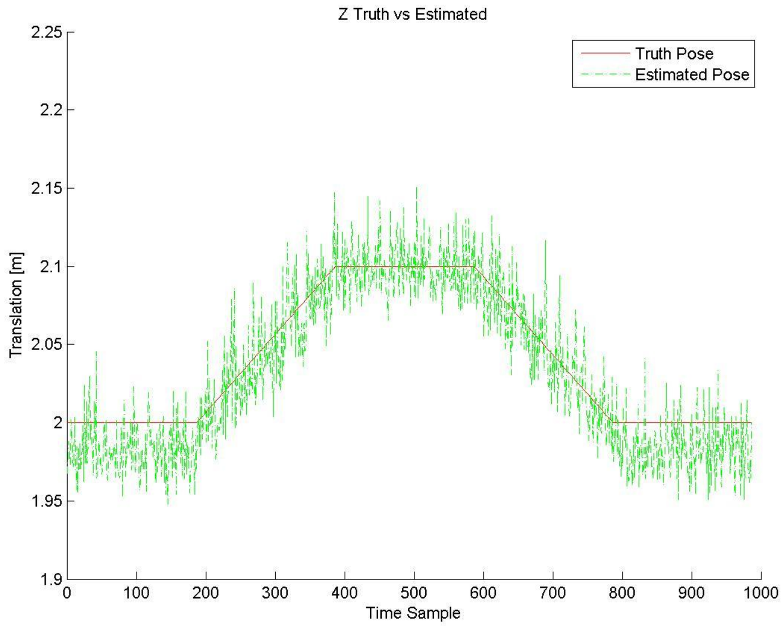


Figure 43: Z (Camera Frame) Truth vs. Estimated

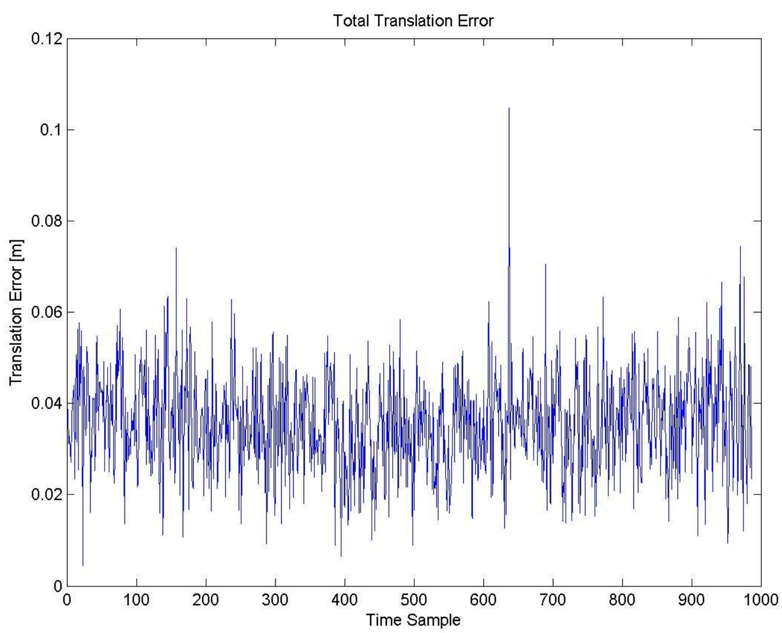


Figure 44: Total Translation Error

# 6. Analysis of Results

## 6. 1. Comparing the Three Pose Estimation System

### 6. 1. 1. Effect of the Position on the accuracy and precision

The analysis of Figure 45, Figure 46, and Figure 47 provides a comparison of the accuracy of the approach as position changes. While the corner system becomes more accurate as it gets closer, the line system becomes highly accurate as position changes. It can also be seen that the ellipse system does not vary until the closest position. This can be due to the loss of tracking in the system, which is caused by a maximum size an ellipse may be in pixels during the extraction stage.

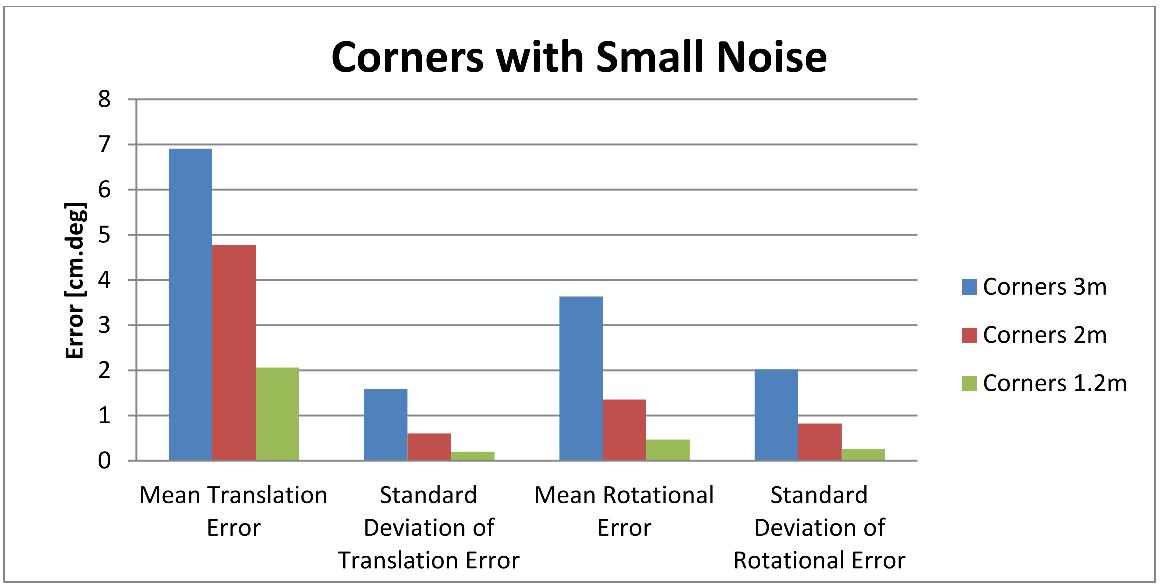


Figure 45: Corners with Low Noise vs. Position

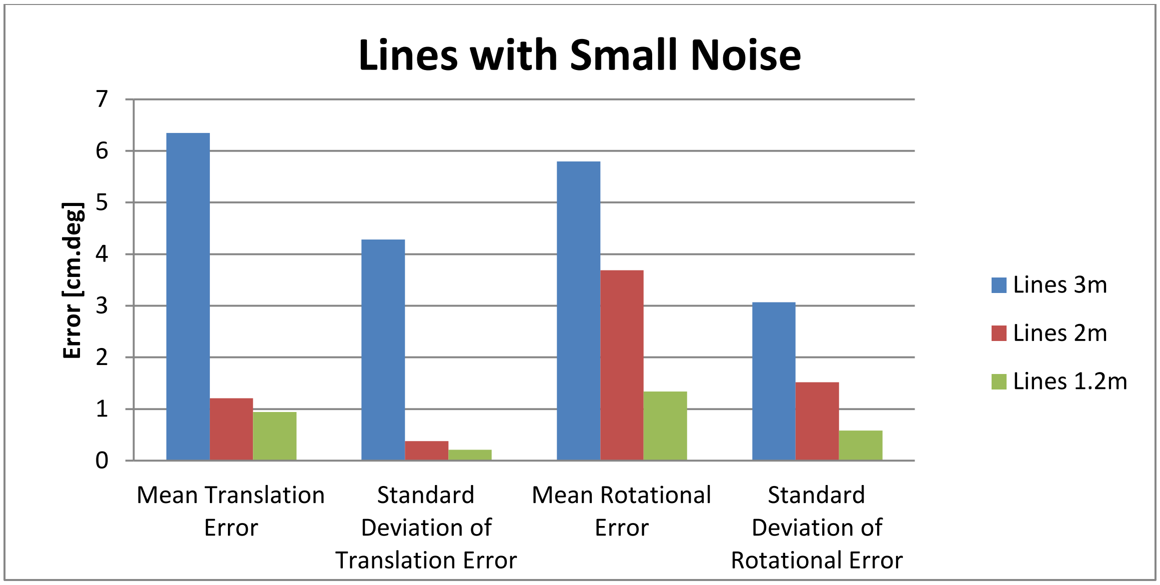


Figure 46: Lines with Low Noise vs. Position

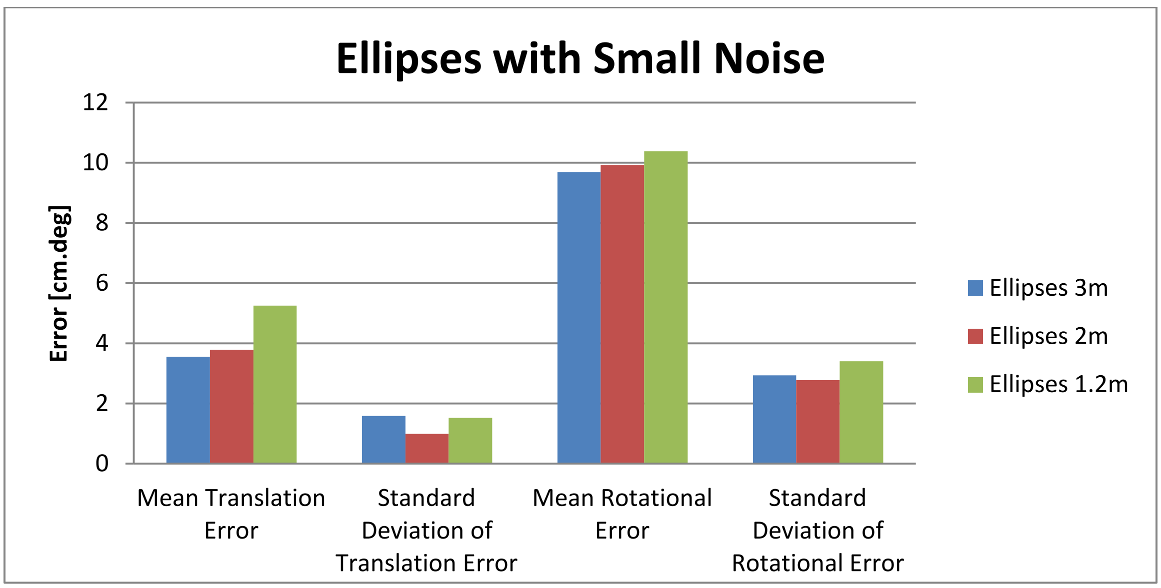


Figure 47: Ellipses with Low Noise vs. Position

### 6. 1. 2. Effect of Algorithm in the accuracy and precision

In Figure 48, Figure 49, and Figure 50 a comparison of each system and how they react to noise can be seen. It can be seen that the mean rotational error in the corner system increases while the other systems remain constant. All systems standard deviation of translation and rotation errors increased as noise increases.

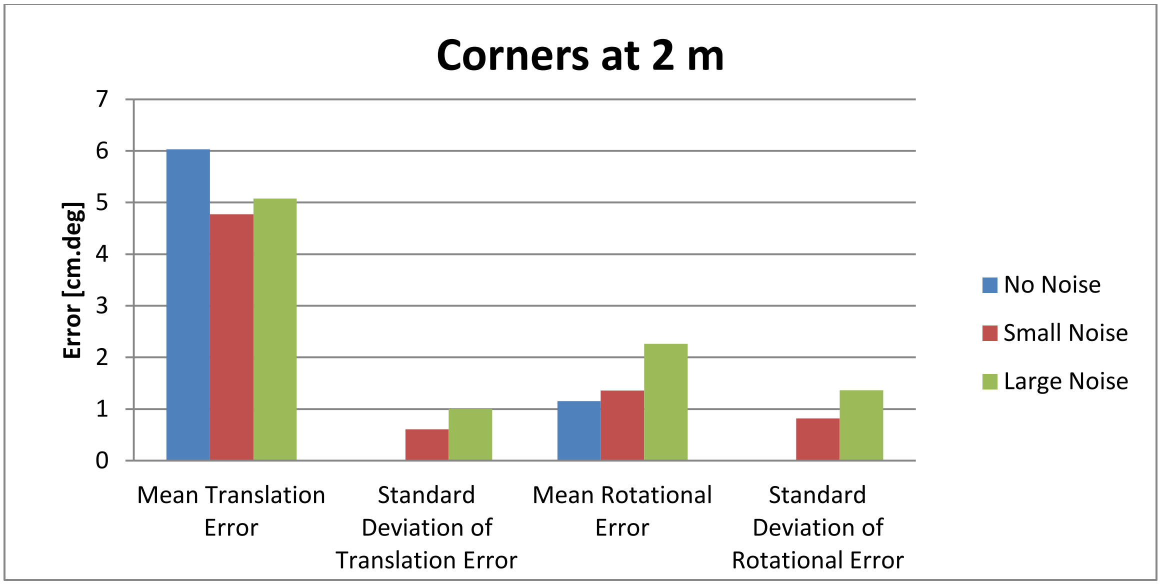


Figure 48: Corners at a Fixed Position vs. Noise

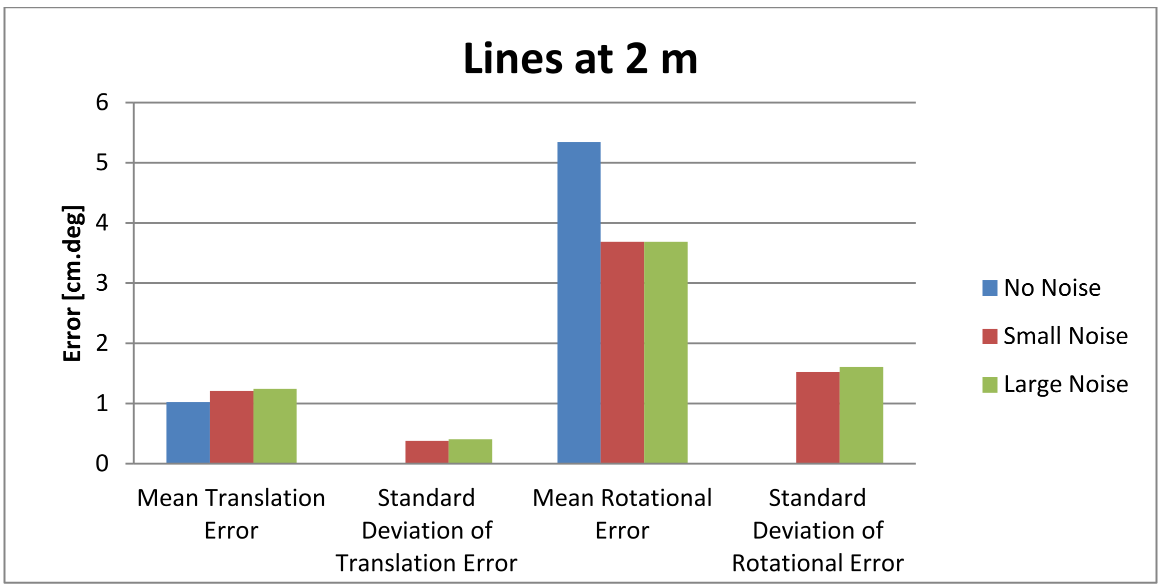


Figure 49: Lines at a Fixed Position vs. Noise

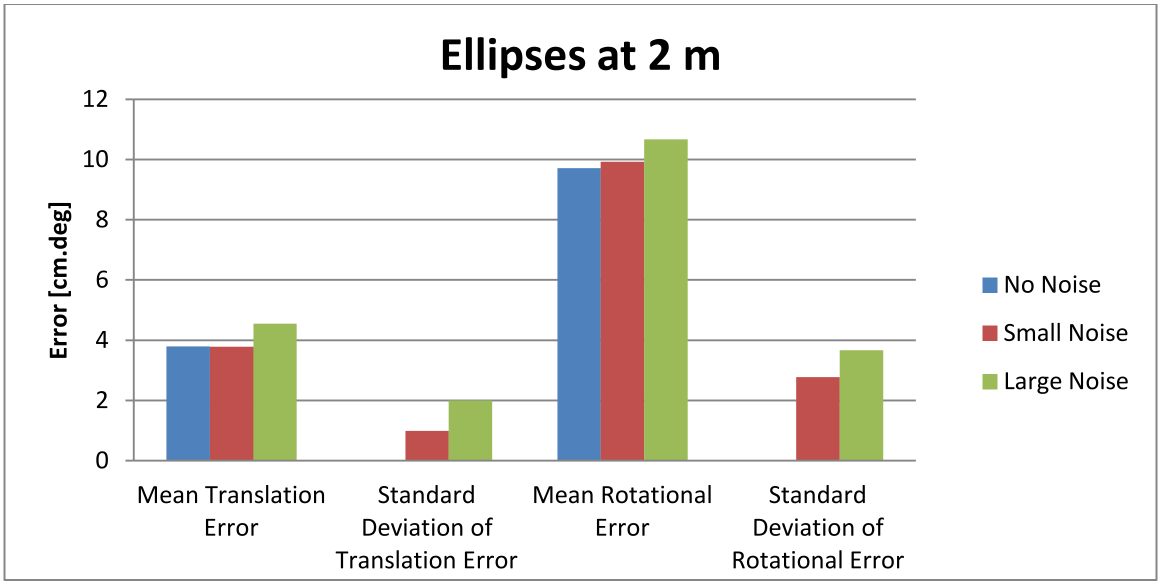


Figure 50: Ellipses at a Fixed Position vs. Noise

### 6. 1. 3. Effect of Noise on the accuracy and precision

Figure 51, Figure 52, and Figure 53 provide a direct comparison between the algorithms at a fixed level of noise. In all cases the ellipse system has the high error in rotation mean and standard deviation. However, at far distances the error of the ellipses is lower than the line and corner systems, at closer distances the line detector becomes the most accurate.

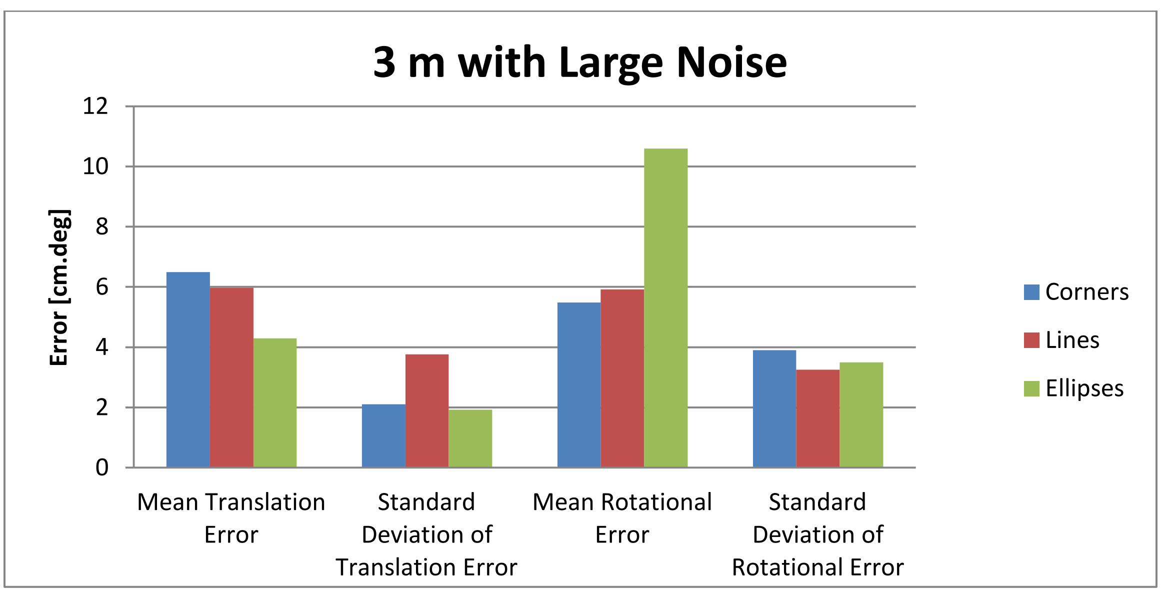


Figure 51: 3 meters and Fixed Noise vs. Algorithm

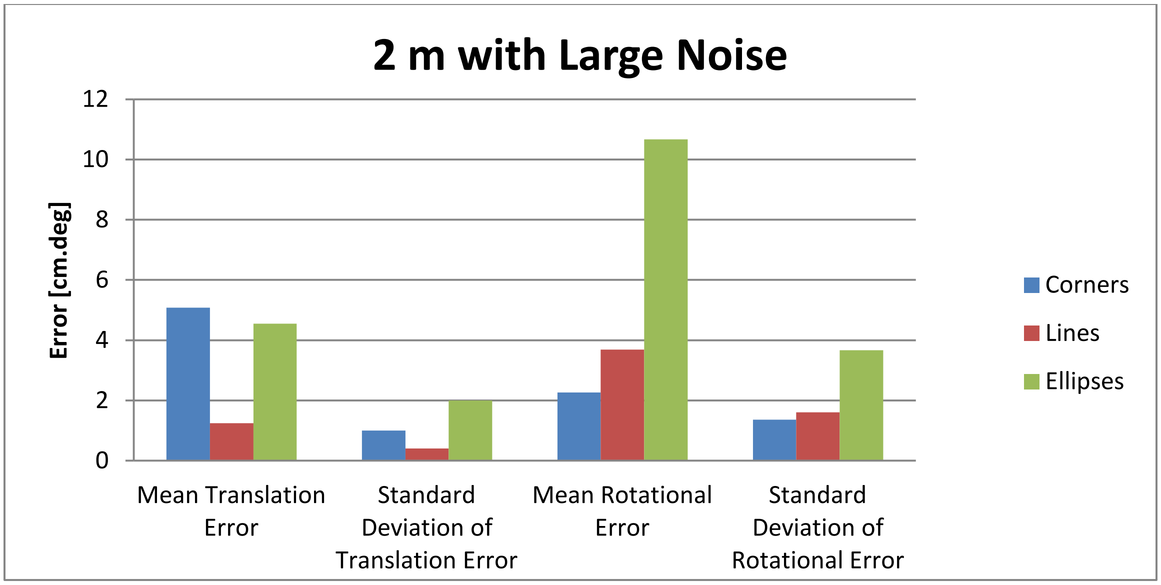


Figure 52: 2 meters with Fixed Noise vs. Algorithm

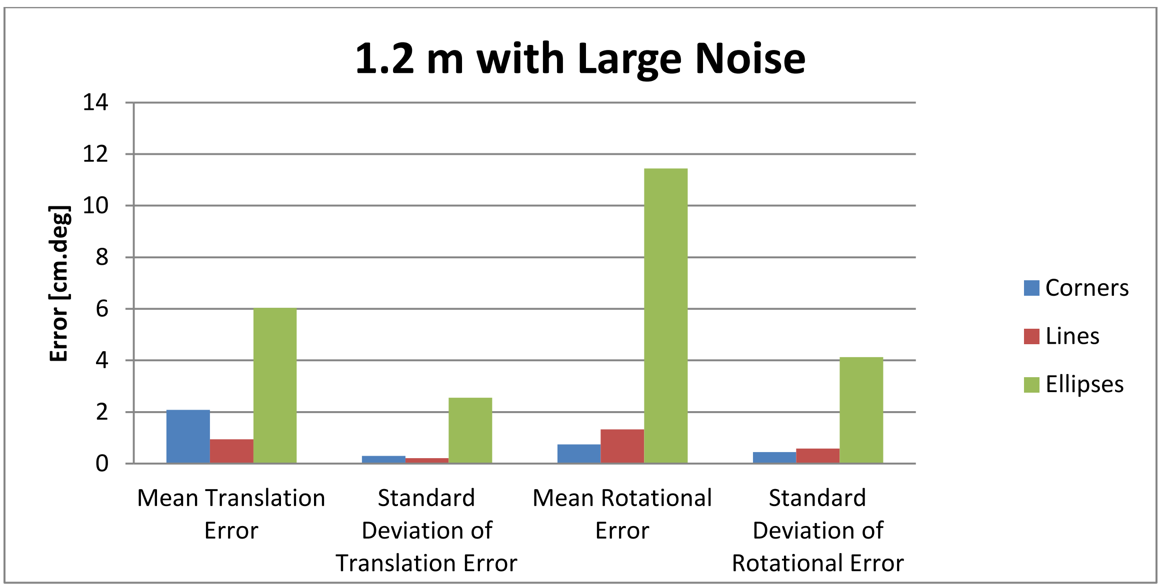


Figure 53: 1.2 meters with Fixed Noise vs. Algorithm

Figure 54, Figure 55, and Figure 56 shows each systems response to various maneuvers. It can be seen that the maneuvers have no major affect on the translational or rotational error. This could be due to there being not enough of a movement in order to cause a major change in error.

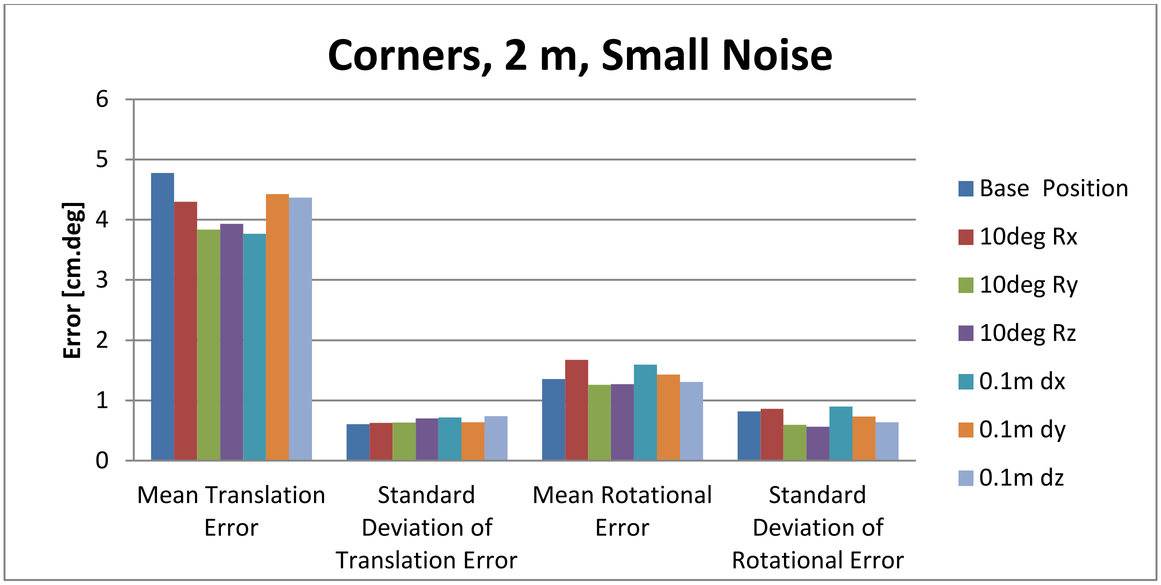


Figure 54: Corners at 2 meters with Low Noise vs. Movements

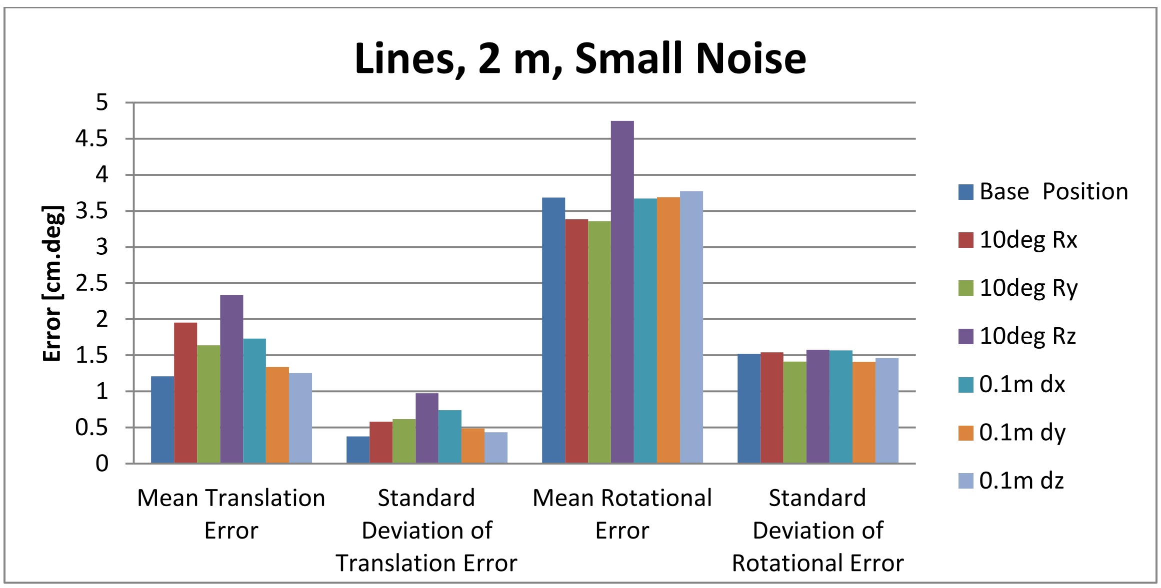


Figure 55: Lines at 2 meters with Low Noise vs. Movements

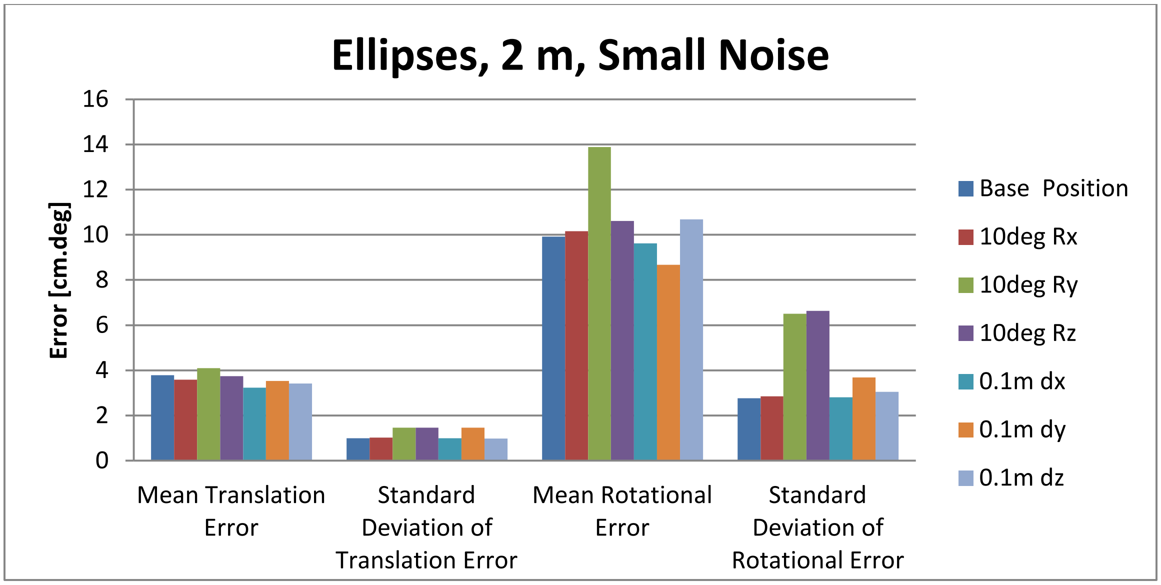


Figure 56: Ellipses at 2 meters with Low Noise vs. Movements

In several instances the corner and ellipse systems would be unable to maintain tracking of the target. In Figure 57 and Figure 58, the translational and rotational error can be seen as one of the instances where the ellipse system lost track. Translational error can near 40 centimeters along with rotational error becoming close to 40 degrees. The ellipse tracker could regain track due to the limited number of ellipses being extracted.

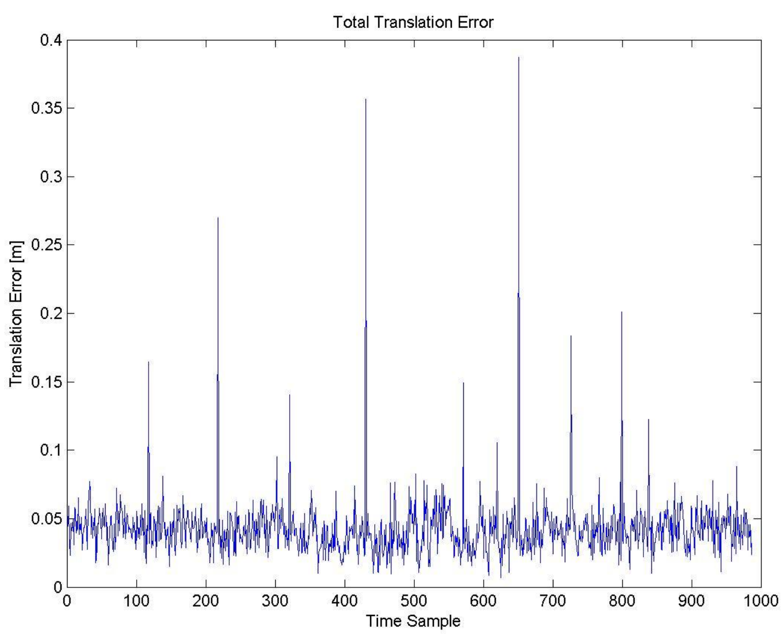


Figure 57: Lost Ellipse Tracker - Translational Error

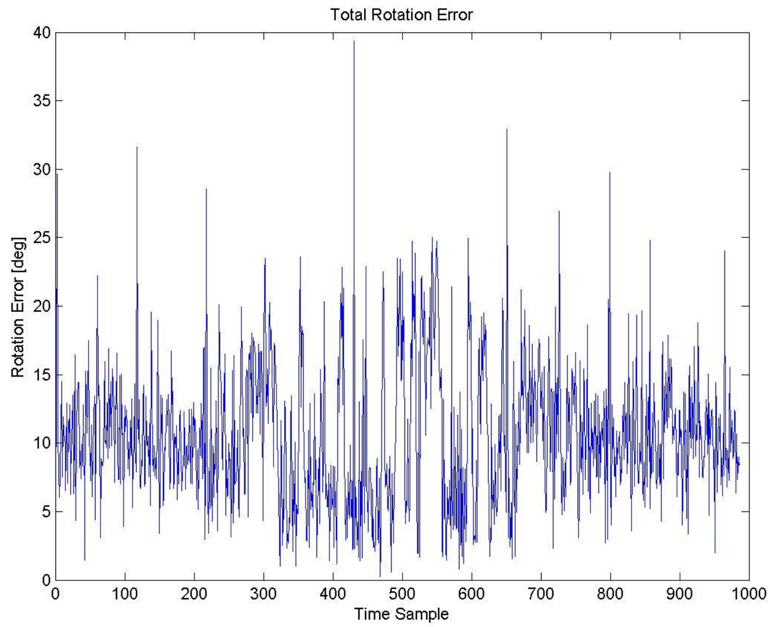


Figure 58: Lost Ellipse Tracker - Rotational Error

An example of the corner system losing track can be seen in Figure 59 and Figure 60. It can be seen as soon as the corner system loses track; it cannot regain lock on the target. This is due to the quantity of corners that can be extracted from any image. This did not happen in any of the virtual images with no noise, due to the perfect image.

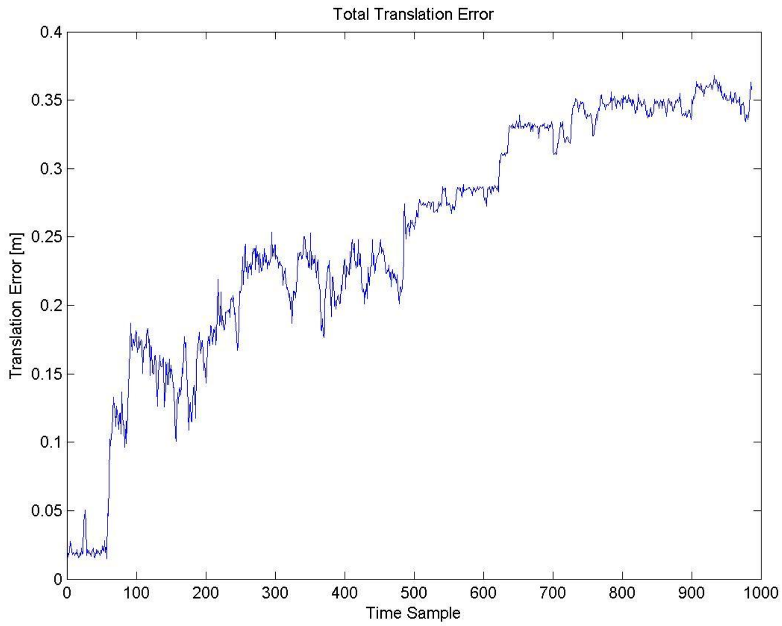


Figure 59: Lost Corner Tracker - Translational Error

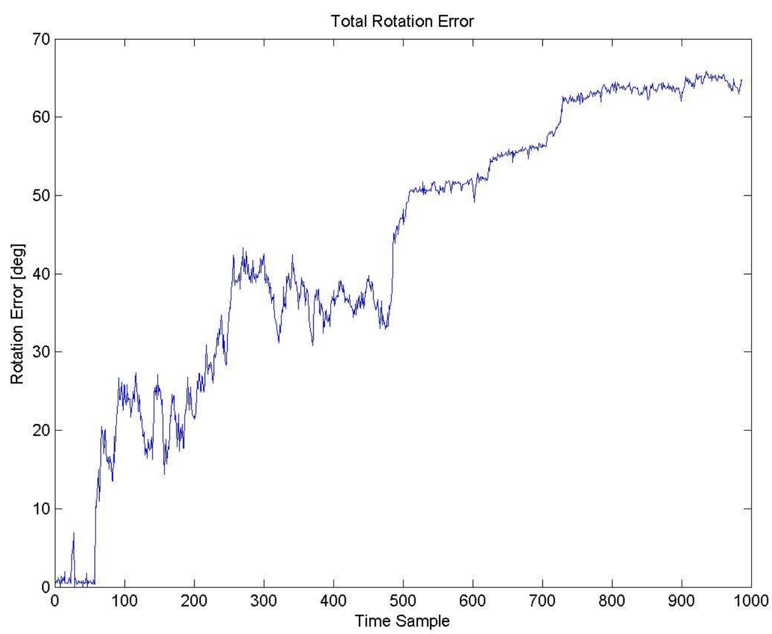


Figure 60: Lost Corner Tracker - Rotational Error

## 6. 2. Approach Results

### 6. 2. 1. Simulation

During the approach, the field of view of the camera along with resolution was changed in each system to allow for each system to be optimized. The ellipse system was set to use a higher field of view camera in order to reduce the size of the ellipse in pixels; this allows the system to remain tracked for the whole approach. With tuning the corner detector was never able to work due to the increased noise levels, while the line detector would occasionally lose track. If each system was able to be tuned iteratively, based on the distance of the target from the camera, it would allow for a very durable ellipse detector.

### 6. 2. 2. Mock-up

All three systems have been integrated into a full system that includes a Kalman Filter. The corner and line systems remain locked at a fixed position; however, once motion is presented the system loses track. The ellipse detector, due to the magnitude of lines presented, remains locked on the target throughout the approach sequence regardless of noise.

# 7. Conclusion

The following overall conclusions can be made:

- A corner, line, and ellipse MV based PE system were successfully implemented in a virtual reality environment and the translation and rotation errors were computed for different motions and test conditions in order to compare and evaluate all of the systems.
- As the distance gets smaller, the corner and line feature extraction schemes provide more accurate results.
- The presence of noise can potentially lead to loss of tracking capabilities for the 'corner detection' scheme while it introduces a substantial level of error for the 'line detection' scheme. With the presence of noise the ellipse system only increased the error in rotation.
- The "ellipse detection" scheme provided the most robust performance due to its capability of recovering from a "loss of track" condition and its lower sensitivity to noise.

The following conclusions can be made about accuracy and precision:

- The corner and line system's accuracy and precision was strongly related to the distance away from the target for both rotational and translational position estimation. However, the ellipse system was very robust and did not vary with distance for both rotational and translational position estimation.
- The line system was the most accurate at the closest distance for the translational position estimation, while the ellipse system was the most accurate at the farthest distance for the translational position estimation. The corner system was most accurate at all times for the rotational position estimation.

The following conclusions can be made about robustness:

- The corner system lost track several times and was not able to regain tracking, while the ellipse system lost track but was able to regain track. All of the results show that the ellipse system is the most robust system and could be used as a base tracking system. The number of features extracted (Corners ~1000s of features, Lines ~100s of features, Ellipses ~10 features) directly impacted the performance of the system in the mockup experiment. The ellipse system extracted far less features than the line system, which extracted far less features than the corner system

# 8. Recommendations

This work allows for the continuation of research by merging the line, corner, and ellipse systems into one system, for example, using Kalman Filter to allow the information to be combined into a single result. This would allow for the robustness of the ellipse system to maintain tracking, while using lines and corners to improve the pose estimation of the complete system, as well as to be able to find 6DOF pose. This would allow for the system to adaptively change as the target becomes closer, or farther away, using the most prominent features.

# 9. Bibliography


1. **Brian Wilcox, Kam Tso, Todd Litwin, Samad Hayati, Bruce Bon.** *Autonomous sensor-based dual-arm satellite grappling.* 1989.

2. *Control of Space Manipulator for Capturing a Tumbling Object.* **Sakawa, Zheng Hua Luo and Yoshiyuki.** Honolulu, Hawaii : IEEE Proceedings of the 29th Conference on Decision and Control, 1990.

3. *Machine Vision Algorithms for Autonomous Aerial Refueling for UAV's using the USAF Refueling Boom Method.* **Mario L. Fravolini, Marco Mammarella, Giampiero Camp, Marcello R. Napolitano, Mario Perhinschi.** s.l. : Studies in Computational Intelligence, 2010, Vol. 304/2010.

4. Nuclear Weapons and Global Security. *Unions of Concerned Scientists.* [Online] 10 07, 2011. [Cited: 02 07, 2012.] http://www.ucsusa.org/nuclear_weapons_and_global_security/space_weapons/technical_issues/ucs-satellite-database.html.

5. Falling Satellites & Space Junk: Q&A with Orbital Debris Expert Ray Williamson. *Space.com.* [Online] 09 22, 2011. [Cited: 02 7, 2012.] http://www.space.com/13053-falling-nasa-satellite-space-junk-orbital-debris-interview.html.

6. Achieving and Maintaining Orbit. *NASA Earth Observatory.* [Online] [Cited: 2 07, 2012.] http://earthobservatory.nasa.gov/Features/OrbitsCatalog/page3.php.

7. U.S. Satellite Destroyed in Space Collision. *Space.com.* [Online] 02 11, 2009. [Cited: 02 07, 2012.] http://www.space.com/5542-satellite-destroyed-space-collision.html.

8. Public Law 111-267. 10 11, 2010.

9. **NASA.** *On-Orbit Satellite Servicing Study.* 2010.

10. **NRL.** NRL Robotics Overview. [Online] July 29, 2009. [Cited: 02 11, 2012.] http://spirit.as.utexas.edu/~fiso/telecon/Henshaw_7-29-09/Henshaw%20FREND%20Overview%207_29_09.pdf.

11. **NASA.** Overview of the DART Mishap Investigation Results. [Online] [Cited: 12 12, 2012.] http://www.nasa.gov/pdf/148072main_DART_mishap_overview.pdf.

12. **Mitsushige Oda, Isoa Kawano, Kouichi Kibe, Funio Yamagata.** *ETS-7, A Rendezvous Docking and Space Robot Technology Experiment Satellite - Result of the Engineering Model development work -.* Tokyo,Japan : SICE, 1995. 318 C-4.

13. *Orbital Express AVGS Validation and Calibration for Automated Rendezvous.* **Andrew F. Heaton, Richard T. Howard, Robin M. Pinson.** Huntsville, AL : s.n.

14. **Andrew Ogilvie, Justin Allport, Michael Hannah, John Lymer.** Autonomous Satellite Servicing Using the Orbital Express Demonstration Manipulator System. Ontario, Canada : s.n.

15. **NASA.** ATV Jules Verne Docks with Station. [Online] NASA, 10 23, 2010. [Cited: 02 15, 2012.] http://www.nasa.gov/mission_pages/station/expeditions/expedition16/atv_dock.html.

16. *The Navigation and Communication Systems for the Automated Transfer Vehicle.* **I. Aguilar sanchez, D. Paris, F. Allard, N. Frischauf.** s.l. : IEEE, 1999.

17. *OrbitalHub.* [Online] 03 05, 2009. [Cited: 15 02, 2012.] http://orbitalhub.com/?p=403.

18. **Japan Aerospace Exploration Agency.** [Online] Japan Aerospace Exploration Agency, 2007. [Cited: 02 15, 2012.] http://iss.jaxa.jp/en/htv/operation/.

19. *XSS-10 Micro-Satellite Flight Deomonstration.* **Thomas M Davis, David Melanson.**

20. **Prosma Satellites.** About Prisma. *Prisma Satellites.* [Online] [Cited: February 11, 2012.] http://www.prismasatellites.se/about-prisma.aspx#The Objectives.

21. **NRL.** FREND: Pushing the Envelope of Space Robotics. *2008 NRL Reveiw.* [Online] 2008. [Cited: Februrary 11, 2012.] http://www.nrl.navy.mil/content_images/08Space_Kelm.pdf.

22. **NordicSpace.** The "Space Tug". [Online] [Cited: February 11, 2012.] http://www.nordicspace.net/PDF/NSA201.pdf.

23. *Simulation of the Docking Phase for the SMART-OLEV Satellite Servicing Mission.* **Clemen Kaiser, Peter Rank, Rainer Krenn, Klaus Landzettel.** s.l. : ESA.

24. Satellites of the USA. [Online] [Cited: 02 07, 2012.] http://space.skyrocket.de/sat_c_usa.htm.

25. *Impedancy Matching in Capturing a Satellite by a Space Robot.* **Nakanishi, Kazuya Yoshida and Hiroki.** Las Vegas, Nevada : s.n., October, 2003. IEEE/RSJ.

26. **Eurocket.** Eurock User's Guide:Chapter 5. *Eurocket.* [Online] August 2011. [Cited: February 11, 2012.] http://www.eurockot.com/joomla/index.php?option=com_content&view=article&id=20040718&Itemid=102.

27. *Autonomous Capture of a Tumbling Satellite.* **Ioannis Rekleitis, Eric Martin, Guy Rouleau, Regent L'archeveque, Kourosh Parsa, and Eric Dupuis.** 4, s.l. : Journal of Field Robotics, 2007, Vol. 24.

28. Wikipedia. *Corner Detection.* [Online] June 12, 2012. [Cited: June 28, 2012.] http://en.wikipedia.org/wiki/Corner_detection.

29. *Pose Estimation using Both Points and Lines for Geo-Localization.* **Srikumar Ramalingam, Sofien Bouaziz, Peter Strum.** Shanghai, China : IEEE International Conference on Robotics and Automation, 2011.

30. *Supervised Pose and Motion Estimation over Weakly Textured Industrial Objects.* **Payeur, Valentin Borsu and Pierre.** Montreal, QC : IEEE International Symposium on Robotic and Sensors Environments (ROSE), 2011.

31. *A Probabilistic Measurement Model for Local Interest Point Based 6 DOF Pose Estimation.* **Thilo Grundmann, Robery Eidenberger, Georg v. Wichert.** Taipei, Taiwan : IEEE/RSJ Internation Conference on Intelligent Robots and Systems, 2010.

32. *Feature-tracking Based Pose Estimatino for Autonomous Rendezvous and Docking.* **Daihou Wang, Hanghang Ren, Yiran Wu, Changhong Wang.** Harbin : IEEE Internation Symposium on Systems and Control in Aeronautics and Astronautics (ISSCAA), 2010.

33. *Vision-Based System for Satellite On-Orbit Self-Servicing.* **Guoliang Zhang, Hong Liu, Jie Wang, Zainan Jiang.** Xi'an, China : IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2008.

34. *Robotic Vision: 3D Object Recognition and Pose Determination.* **A. K. C. Wong, L. Rong, and X. Liang.** Victoria Canada : IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998.

35. *Pose Determination of Large Non-Cooperative Satellite in Close Range Using Coordinated Cameras.* **Xiadong Du, Bin Liang, Yanbo Tao.** Changchun, China : IEEE International Conference on Mechatronics and Automation, 2009.

36. *Real-Time Visual Tracking of Complex Structures.* **Drummond, Tom.** 7, s.l. : IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, Vol. 24.

37. *3D Post estimation from an n-Degree Planar Curved Feature in Two Perspective Views.* **MA, Li LI and Song De.** Vienna : International Conference on Pattern Recognition, 1996, Vol. 1.

38. *Online Camera Pose Estimation in Partially Known and Dynamic Scenes.* **Gabriele Bleser, Harald Wuest, Didier Stricker.** Santa Barbard, CA : IEEE/ACM International Symposium on Mixed and Augmented Reality, 2006.

39. *Robotic Vision: 3D Object Recognition and Pose Determination.* **A. K. C. Wong, L. Rong, and X. Liang.** Victoria, B.C., Canada : IEEE/RSJ International Converence on Intelligent Robots and Systems, 1998.

40. *Vision Guided Autonomous Stair Climbing.* **Yalin Xiong, Larry Matthies.** San Francisco, CA : IEEE International Conference on Robotics and Automation, 2000.

41. *Estimation Pose of Known Target Satellite.* **A. Cropp, P. Palmer, P. McLauchlan, and C. Underwood.** 15, s.l. : IEEE Electronics Letters, 2000, Vol. 36.

42. *Autonomous Distant Visual Silhouetting of Satellites.* **Mcinroy, J.** 2, s.l. : IEEE Transactions on Aerospace and Electronic Systems, 2008, Vol. 44.

43. Gunter's Space Page. *SNAP 1.* [Online] October 03, 2012. [Cited: March 13, 2012.] http://space.skyrocket.de/doc_sdat/snap-1.htm.

44. *Real-Time Tracking and Pose Estimation for Industrial Objects using Geometric Features.* **Youngrock Yoon, Guilherme N. DeSouze, Avinash C. Kak.** Taipei, Taiwan : IEEE International Conference on Robotics and Automation, 2003.

45. *Monitoring Techniques in a Layered Control Architecture for Reliable Robotic Satellite Capture.* **Noriyasu Inaba, Mitsugu Asano, Mitsushige Ode.** Sendal, Japan : IEEE/RSJ Internation Converence on Intelligent Robots and Systems, 2004.

46. *Matching 2-D Ellipses to 3-D Circles with Application to Vehicle Pose Identification.* **Marcus Hutter, Nathan Brewer.** Wellington : International Conference on Image and Vision Computing New Zealand (IVCNZ), 2010.

47. *3D Pose Determination and Non-Linear Image Registration of Circular Traffic Sign.* **Zhonggen, Ren Lei Cao Fang Yang.** Huanggang : International Symposium on Intelligence Information Processing and Trusted Computing (IPTC), 2010.

48. *Three-Dimensional Location Estimation of Circular Features for Machine Vision.* **Reza Safaee-Rad, Ivo Tchoukanov, Kenneth Carless Smith, and Bensiyon Benhabib.** 5, s.l. : IEEE Transations on Robotics and Automation, 2002, Vol. 8.

49. *Solving the Orientation-Duality Problem for a Circular Feature in Motion.* **Benhabib, D. He and B.** 4, s.l. : IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 1998, Vol. 28.

50. *Closed-form Solution for Circle Pose Estimation Using Binocular Stereo Vision.* **Liu, Y. Zheng and Y.** 21, s.l. : IEEE Electronics Letters, 2008, Vol. 44.

51. *Matching 2-D Ellipses to 3-D Circles with Application to Vehicle Pose Identification.* **Brewer, Marcus Hutter and Nathan.** Wellington : International Conference of Image and Vision Computing New Zealand (IVCNZ), 2009.

52. *Pose and Position Estimations of Regular Shapes in Monocular Images.* **Ren, Yuan.** Baoding : IEEE Conference on Machine Learning and Cybernetics, 2009.

53. **Wolfram Research, Inc.** Wolfram MathWorld. *Euler Angles.* [Online] 1999-2012. [Cited: June 10, 2012.] http://mathworld.wolfram.com/EulerAngles.html.

54. **Jazar, Reza N.** *Advanced Dynamics: Rigid Body, Multibody, and Aerospace Applications.* 2011.

55. *Good Features to Track.* **Shi, Jianbo.** Seattle, WA : IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994.

56. Wikipedia. *Corner Detection.* [Online] [Cited: May 22, 2012.] http://en.wikipedia.org/wiki/Corner_detection.

57. *Ellipse Detection in Digital Image Data using Geometric Features.* **Lars Libuda, Ingo Grothues, Karl-Friedrich Kraiss.**

58. *3D Location of Circular and Spherical Features by Monocular Model-Based Vision.* **Shiu, Y.C.** Cambridge, MA : IEEE International Conference on Systems, Man and Cybernetics, 1989.

59. **Humusoft and The MathWorks Inc.** Virtual Reality Toolbox Users Guide. 2001-2006.