AN ALGORITHM FOR DETERMINING SATELLITE ATTITUDE BY COMPARING PHYSICAL FEATURE MODELS TO EDGES DETECTED IN SATELLITE OR GROUND-BASED TELESCOPE IMAGERY

by

ERIC BRIAN REINHART
B.S. University of Florida, 2003

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Department of Mechanical, Materials, and Aerospace Engineering
in the College of Engineering & Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2007

ABSTRACT

This thesis discusses the development and performance of an algorithm created to calculate satellite attitude based on the comparison of satellite "physical feature" models to information derived from edge detection performed on imagery of the satellite. The quality of this imagery could range from the very clear, close-up imagery that may come from an unmanned satellite servicing mission to the faint, unclear imagery that may come from a ground-based telescope investigating a satellite anomaly. Satellite "physical feature" models describe where an edge is likely to appear in an image. These are usually defined by physical edges on the structure of the satellite or areas where there are distinct changes in material property. The theory behind this concept is discussed as well as two different approaches to implement it. Various simple examples are used to demonstrate the feasibility of the concept. These examples are well-controlled image simulations of simple physical models with known attitude. The algorithm attempts to perform the edge detection and edge registration of the simulated image and calculate the most likely attitude. Though complete autonomy was not achieved during this effort, the concept and approach show applicability.

ACKNOWLEDGMENTS

I would like to thank Dr. Roger W. Johnson for his guidance and direction (from a distance of more than 1500 miles). I would also like to thank my wife, Monica, and my children, Nathan and Sofia, for their patience and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS/ABBREVIATIONS

ARCSS	Autonomous Rendezvous and Capture Sensor System
ASSFC	Analysis-specific spacecraft-fixed coordinate
ASTRO	Autonomous Space Transport Robotic Operations
AVGS	Advanced Video Guidance Sensor
CAD	Computer aided design
DARPA	Defense Advanced Research Projects Agency (USA)
DART	Demonstration for Autonomous Rendezvous Technology
DCM	Direction cosine matrix
ECI	Earth Centered Inertial
ETS-7	Engineering Test Satellite 7
GUI	Graphical user interface
IFC	Image frame coordinate
IMU	Inertial measurement unit
ISAS	Institute of Space and Astronautical Science (Japan)
ISS	International Space Station
JAXA	Japan Aerospace Exploration Agency
LHS	Left hand side
MDSFC	Manufacturer-defined spacecraft-fixed coordinate
MUBLCOM	Multiple Paths, Beyond-Line-of-Sight Communications
NAL	National Aerospace Laboratory (Japan)
NASA	National Aeronautics and Space Administration (USA)

NASDA	National Space Development Agency (Japan)
NEXTSat	Next-Generation Serviceable Satellite
OEDS	Orbital Express Demonstration System
PXS	ProXimity Sensor
PXSM	ProXimity Sensor Marker
RHS	Right hand side
RTG	Radioisotope thermoelectric generator
SSA	Space situational awareness
TASAT	Time-domain analysis and simulation for advanced tracking
Vis-STAR	Vision-based Software for Track, Attitude, and Ranging
XSS-10	Experimental Satellite System 10
XSS-11	Experimental Satellite System 11

CHAPTER 1: INTRODUCTION

In most cases the attitude of a spacecraft is critical for mission success. In many cases the mission of the payload depends on extremely precise knowledge of the satellite's attitude, and with few exceptions even satellites with the simplest of payloads require attitude knowledge and control in order to point their solar arrays at the sun to generate power. Obvious exceptions include satellites powered by nuclear reactors or radioisotope thermoelectric generators (RTG) (Wertz & Larson, 1999), short-term missions operated only by battery, and satellites with the solar cells located all over the external of the spacecraft bus, but these are rare exceptions. Even in these cases the payload will most likely still require attitude control.

Knowing the importance of attitude knowledge and control, designers often use sensors in redundancy to determine attitude and attitude dynamics in order to ensure the satellite will continually operate and meet mission requirements. Attitude sensors that are currently being employed include inertial measurement units (IMU), star sensors, sun sensors, horizon sensors, and magnetometers. IMUs can track the satellite's attitude without information derived from external indicators (Sun, Earth, stars, etc.), but because of the degradation in the IMU's absolute attitude knowledge over time, it periodically requires an update from sensors that use those external cues, like star sensors (Wertz & Larson, 1999). The other sensors require inputs derived from those external indicators mentioned above (Sun, Earth, stars, etc.) to determine attitude.

Development, construction, and launch have long been known to be the most expensive phases of a satellite's life cycle. If the attitude control system on the satellite loses the ability to correctly sense those external cues, even with redundant systems, it most likely means the end of the mission, even if communication can still be maintained. If this occurs during initial system

checkout, all of that investment in development, construction, and launch is lost. The operators may be able to re-establish proper attitude with procedures and checklists, but they may also still be acting without knowing exactly what the satellite is doing. The problem is that there are no external (to the satellite) sensors designed to watch and analyze the satellite.

Currently a great deal of research is being done on ground-based telescopes by a number of different communities all over the world. Much of the research is astronomically focused, but there is a large part that is focused on tracking satellites in Earth's orbit. This is an understandably important field of study and has several applications. One of the most important applications includes tracking and studying debris. Another important application is to track and image the space shuttle (afrlhorizons, 2004) or any other manned spacecraft (International Space Station – ISS) for damage. This can also be done for any other unmanned satellite as well, to ensure proper solar array or antenna deployment for example. Most of these telescope sites are for research and not used for any kind of satellite operations, but with the proper tools and contingency operations plans there are no obvious reasons why these sites cannot be used to aid in satellite anomaly recovery. For those companies who have spent the money on development, construction, and launch, it may be worth spending the money to employ these telescopes for analysis of their own on-orbit satellites.

Of course due to the high reliability and redundancy of current satellites, the likelihood that a satellite would only lose its attitude sensing system and still be recoverable and/or usable is very low. Observations from ground-based telescopes in support of satellite anomaly resolution may potentially provide timely attitude determination, but most likely only to investigate for structural damage and not for recovery. This does not mean that the conditions above will never exist; the likelihood is just extremely low.

There are more promising applications for the ability to use external (to the satellite) sensors for attitude determination that are currently of interest to the space community, and these applications range from servicing the ISS to refueling aging satellites to re-supplying a manned mission to Mars. Recent experimental emphasis has been placed on autonomous rendezvous and proximity operations as a precursor to future on-orbit satellite refueling and servicing missions as well as future Mars exploration missions. Some of the most noteworthy projects within the past ten years include Japan's National Space Development Agency's (NASDA) Engineering Test Satellite 7 (ETS-7), the United States Air Force's Experimental Satellite System 10 (XSS-10) and Experimental Satellite System 11 (XSS-11), the National Aeronautics and Space Administration's (NASA) Demonstration for Autonomous Rendezvous Technology (DART), and the Defense Advanced Research Projects Agency's (DARPA) Orbital Express Demonstration System (OEDS) program.

For a satellite whose mission is to service/maintain/refuel another satellite, knowledge of relative attitude is critical. If in the future these types of satellite missions use video or infrared imagery, this type of algorithm holds the potential to be the main attitude calculation tool. Of the projects listed above, only the OEDS program used imagery in any way to determine attitude, so study of autonomous attitude determination for the future of this mission area is vital.

The problem with these approaches may be that the analysis of the imagery could take days or weeks to analyze. In the case of the satellite anomaly the satellite may be unrecoverable by that time. The purpose of this thesis is to calculate satellite attitude from real-time or near-real-time appropriately resolved imagery taken from any ground telescope or servicing satellite. From these calculations a satellite operator could possibly determine satellite dynamics, plan a

corrective action, and send the command to the satellite within one orbital period, or a servicing satellite could autonomously dock with any generic satellite in need of repair.

CHAPTER 2: LITERATURE REVIEW

The following chapter discusses some of the basic concepts that were reviewed in preparation for the development of this algorithm. Very few have published papers on attitude estimation from external imagery. Because of this, most of the available information is on the basic concepts behind this effort, so this algorithm was developed from fundamental concepts which include coordinate systems and coordinate system transformations as well as image processing.

Several coordinate systems are used in this analysis. The two most important coordinate systems are the Earth-centered inertial (ECI) coordinate system and the spacecraft-fixed coordinate system because the relationship between these determines the attitude of the satellite. Additionally as an intermediate step the image frame coordinate system is used.

ECI coordinate system

The ECI coordinate system is the most commonly used inertial reference frame for Earth-orbiting satellites. It is defined with the origin at the center of the Earth. The x-axis points toward the vernal equinox, the z-axis is along the Earth's rotation axis, and the y-axis completes the coordinate system (Kelso, 2006). This coordinate system is important because it provides an inertial reference for attitude and attitude dynamics. Figure 1 is a graphical representation of the ECI coordinate system.

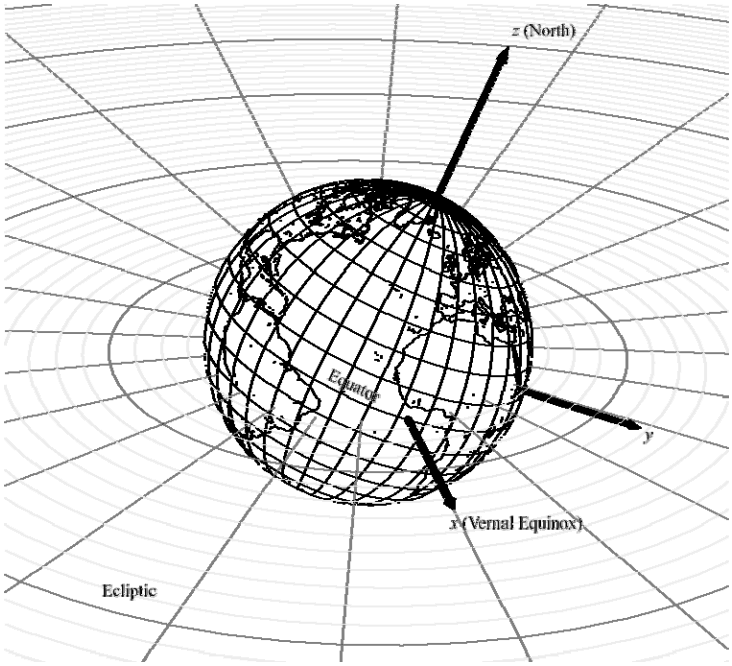

Figure 1: Basic Definition of the ECI Coordinate System (Kelso, 2006)

Spacecraft-fixed coordinate system

This coordinate system is fixed to the body of the satellite and can be arbitrarily defined. In many cases it is defined so that the x-axis is in the general direction of the velocity while on orbit, the z-axis points toward the center of the earth (nadir) while on orbit, and the y-axis completes the coordinate system (Wertz & Larson, 1999).

Image frame coordinate system

This coordinate system is defined so that the origin is located at the upper left corner of the resulting image (see Figure 2). The x-axis is along the top of the image, the y-axis is down the left side of the image, and the z-axis is into the image plane (Mathworks, 2007).

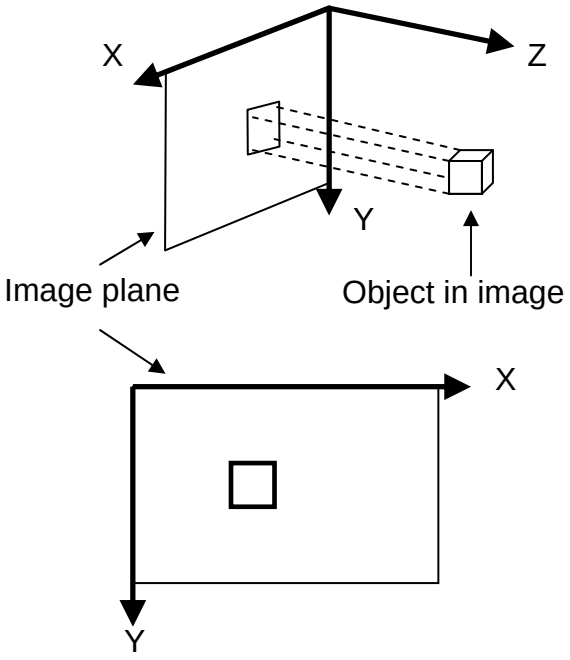

Figure 2: Basic Definition of the Image Frame Coordinate System (Mathworks, 2007)

Vectors defined in a specific coordinate frame in three dimensional space can be represented in a separate coordinate frame through a sequence of three rotations about three axes. This is known as Euler's rotation theorem. The matrix representation of the combined rotation is known as the direction cosine matrix (DCM). Because of the orthogonality condition, the transpose of the DCM is equal to its inverse. This allows for an easy conversion from one coordinate system to another and back again. To determine the DCM for a coordinate system transformation in some cases it may be easier to approach it from the opposite direction because of this property.

In 1986 John Canny published a paper that became a cornerstone of image processing describing what became known as the "Canny Edge Detector," which uses a multi-step approach to improve upon the edge detectors of the time (Canny, 1986). The Canny edge detector is one

of the keys to the success of this project. Recall that the approach that this project takes is to compare edges detected in imagery to edges defined in a physical model. Edge detection acts somewhat as an image filter allowing only sharp intensity gradients to pass, so good edge detection will derive key data for use in the comparison.

Satellite imaging using ground-based electro-optical telescopes is improving. By using adaptive optics and other image processing tools such as multi-frame blind deconvolution (Schultz, et al., 1997) image resolution is continuously improving, but very little effort has gone toward attitude estimation using ground-based telescopes. A similar effort was studied, but that procedure used a matched filter-type approach (Wood, 1996), which requires previous imagery of the object to perform the comparison. There has been research done to develop anomalous satellite behavior detection algorithms (Maron, 1998). This research attempted to autonomously classify a satellite's behavior as normal or anomalous using ground-based telescope imagery, which is useful as a space situational awareness (SSA) tool but not as a tool for the satellite operator.

Much more work has been done on developing satellite docking technology. In 1997 Japan's National Space Development Agency (NASDA), which was merged with the Institute of Space and Astronautical Science (ISAS) and the National Aerospace Laboratory of Japan (NAL) to form the Japan Aerospace Exploration Agency (JAXA) in 2003, launched Engineering Test Satellite 7 (ETS-7). The main mission of this experimental satellite was to "develop the rendezvous docking and space robot technologies which are essential for the future space activities such as supply logistics to the international space station and to realize the in-orbit satellite servicing (JAXA, 2003)." To do this the ETS-7 consisted of a "chaser" satellite and a "target" satellite that were used to conduct various robotic and rendezvous docking experiments.

Though this experiment did use a sensor called the ProXimity Sensor (PXS) to determine relative attitude, the sensor did not use video. Additionally, the "target" satellite was equipped with the PXS Marker (PXSM) which was required by the PXS to determine relative attitude.

In 2003 the United States Air Force launched the XSS-10, which was a 24 hour experiment designed to demonstrate autonomous navigation and proximity operations around its Delta II rocket body (Davis, 2003). Though the satellite took photographic images of the rocket body during its mission, they were not used for attitude determination. Then, in 2005 the Air Force launched the XSS-11, which further demonstrated rendezvous and proximity operations around its Minotaur I rocket body (Kirtland, 2005). Again, the XSS-11 was equipped with the ability to take photographic images but did not use them to determine the attitude or attitude dynamics of the rocket body.

In 2005 NASA launched the Demonstration for Autonomous Rendezvous Technology (DART) satellite, which was supposed to "rendezvous with and perform a variety of maneuvers in close proximity to the Multiple Paths, Beyond-Line-of-Sight Communications (MUBLCOM) satellite, without assistance (autonomously) from ground personnel (Armstrong, 2006)." Unfortunately not all of the mission objectives were accomplished because of a collision between the two satellites. For attitude determination DART was to use the Advanced Video Guidance Sensor (AVGS), which is a laser-based, not video-based, tracking system.

The AVGS was later used in the Autonomous Rendezvous and Capture Sensor System (ARCSS) on DARPA's Orbital Express Demonstration System (OEDS) program, which was launched in 2007. In addition to the AVGS, Orbital Express used the Vision-based Software for Track, Attitude, and Ranging (Vis-STAR), which worked in conjunction with ARCSS to provide a method for passively determining the "serviceable" satellite's attitude (Weismuller & Leinz,

2006). Vis-STAR was software on board the servicing satellite (Autonomous Space Transport Robotic Operations – ASTRO) that was used to independently calculate the attitude of the client ("serviceable") satellite (Next-Generation Serviceable Satellite – NEXTSat) using a video-based approach (Weismuller & Leinz, 2006) and is the only program that is similar to the algorithm developed from work done on this thesis. All attitude determination done by Orbital Express was done by the AVGS (laser-based) and the Vis-STAR software (video-based).

The Vis-STAR software operates by comparing silhouettes detected in the active camera's field of view to scaled (based on range) library reference images showing views of the serviceable satellite in all rotations. Boeing claims that this attitude determination method can achieve sub-degree accuracy in each of the relative pitch, roll, and yaw axes (Weismuller & Leinz, 2006). This method only requires a computer and visible/IR images. In addition Vis-STAR is able to accommodate image processing against cluttered backgrounds (looking down toward Earth) (Weismuller & Leinz, 2006). It is noted that Vis-STAR does use edge tracking for secondary position determination. Additionally the literature states:

> "When correlating against library images, client satellites with near-symmetrical geometry can have competing attitudes with scores close to those of the actual orientation. In this case, surface details are used as a discriminator to reject the incorrect orientation. Fortunately, real-world satellites have enough dissimilar features for each orientation to make this an effective approach when needed. For in-corridor (within field of view) cases, edge tracking can again be used to bring out embedded features which do not affect the silhouette view, but which have sufficient contrast with the background of the vehicle to produce an edge. Often, *a priori* knowledge of the client vehicle is enough to eliminate competing attitudes from consideration (Weismuller & Leinz, 2006)."

The Vis-STAR software has problems with multiple solutions on symmetrical bodies. It actually uses edge detection only as a discriminating factor between several possible orientations.

In all but one of the applicable cases stated above, the target/client/serviceable satellites were required to have equipment or specific modifications to allow for attitude determination by the servicing satellite. The other applicable case, the use of the Vis-STAR software, required a library of reference images. This means that even though no additional equipment was required to be put onto the client satellite, a library of ground images was still needed from defined angles and views.

CHAPTER 3: METHODOLOGY

The basic purpose of this algorithm is to determine the attitude of a satellite using the comparison between a satellite "physical feature" model (related to a physical CAD model) and information derived from edge detection done on imagery of the satellite. The output is an array of possible direction cosine matrices (DCMs) that describe the transformation from the inertial (ECI) coordinate system to the manufacturer-defined spacecraft-fixed coordinate (MDSFC) system of the satellite. This chapter describes the theory behind the algorithm and two ways in which the theory was applied.

3.1: Theory

3.1.1: Assumptions

In order for this algorithm to possibly be operationally implemented there are several realistic assumptions that need to be made about what information is known about the satellite, the imaging system, and the imagery itself. These were taken into account during the development of this algorithm to help prove or disprove feasibility.

The first satellite assumption is that the on-orbit satellite configuration is known and that its "feature definition" model has been modeled to match that configuration using the MDSFC system. This approach may not be useful for a satellite that has been severely damaged while on

orbit or a satellite that has articulating parts in an unknown configuration, possibly like a solar array. The reason for this is that the model is matched to the imagery for attitude determination.

The second satellite assumption is that there are enough features in the "feature definition" model to calculate the attitude correctly. Symmetric satellites or satellites composed mostly of "secondary feature definitions" as described below will naturally be more difficult, if not impossible, to solve.

The first imaging system assumption is that the focal plane array can be characterized in such a way that a pixel to length conversion can be determined. In the application presented later in this chapter the instantaneous horizontal and vertical fields of view of the focal plane array, the pixel array dimensions, and the range to the satellite for each image are assumed to be known before the processing is done. There are many different image processing techniques that are used to improve resolution or improve the data content in imagery, but to use this algorithm a pixel to length conversion (or equivalent) must be known.

The second imaging system assumption is that the direction cosine matrix that defines the transformation from the ECI coordinate system to the IFC system as described in Chapter 2 must be known for each image. Though the ECI to IFC DCM is used by this algorithm, it does not calculate it.

The final imaging system assumption is that the imagery is panchromatic. The imagery comes from light in the visible spectrum with grayscale intensity. It does appear that with further development and research multi-spectral or other multi-channel imagery may be able to be used. Additionally detectors of other parts of the electromagnetic spectrum may also be used such as infrared or ultraviolet light detectors, but the "feature definition" model may need to be

changed to reflect what parts of the satellite will present light intensity gradients in the environment and under the conditions that the satellite operates.

The first imagery assumption is that the time, range, and imager state is known for each image being evaluated and that each image being evaluated has enough resolution to perform the analysis. With an imager that has a relatively high frame rate not all images may be necessary to analyze depending on the application (attitude dynamics determination obviously requires more images than a single "attitude confirmation" image).

The second imagery assumption is that at least one of the edges detected by the edge-detecting algorithm represents a complete "feature definition" in the model. Full "feature definitions" are the basis for the analysis-specific spacecraft-fixed coordinate systems (described below) that are used, so at least one cannot be affected by shadowing or other physical effects.

3.1.2: Definitions

Several of the terms that are used throughout this thesis were developed to refer to specific parts of this project and may or may not be used in other applications. The following are definitions of these terms:

Manufacturer-defined spacecraft-fixed coordinate (MDSFC) system

This is the coordinate system that is fixed to the satellite body and is defined either by the manufacturer or the operations manuals. This coordinate system is the spacecraft-fixed coordinate system described in Chapter 2 that is specific to a particular satellite.

Analysis-specific spacecraft-fixed coordinate (ASSFC) system

This coordinate system is used in addition to the coordinate systems discussed in Chapter 2. This coordinate system is fixed to the satellite much like the MDSFC system discussed above but is used only in the analysis as an intermediate conversion step between the IFC system and the MDSFC system. The x-axis is defined by the line being investigated and the other two axes are arbitrary. The reason for this coordinate system is described below in further detail.

"Feature definitions"

"Feature definitions" are physical properties of the satellite that may trigger sharp intensity gradients in images. These include edges, material changes, and relatively small but contrasting parts of the satellite like antennas.

3.1.3: Description

The attitude of the satellite can be described by a direction cosine matrix that converts a vector from the ECI coordinate system to the MDSFC system. To do this the process as used by this thesis is broken up into the steps shown in Figure 3.

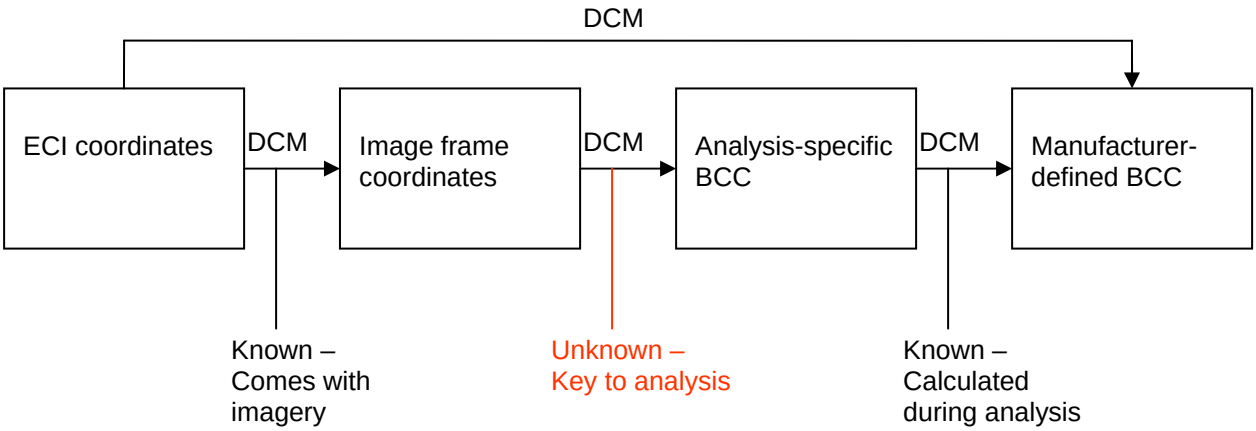

Figure 3: Coordinate System Conversion Process

In this process there are several coordinate system transformations to go through to build the final direction cosine matrix, but the key step that this project addresses is the transformation from the IFC system to the ASSFC system.

Ideally, before any satellite is built, a CAD model is created to help with manufacturing, visualization, testing, etc. Once the satellite is launched, no further work is done with that model. This approach uses that CAD model as a basis with the addition of "feature definitions" (as defined above) to determine the attitude of the satellite from imagery. These sharp intensity gradients in images are referred to as edges. In an image edges may occur because of the edges of the physical structure, changes in material properties, or shadowing effects. Additionally for long wave infrared (LWIR) imaging, thermal properties and parameters can produce edges as well, but LWIR imaging analysis and/or testing is outside the scope of this project. The model can account for the physical structure and the changes in material properties but not shadowing effects (without timing, location, and attitude information). Shadowing can produce edges that would not normally appear under direct lighting and can distort edges that would normally

appear under direct lighting. The physical structure can block the projection of a feature onto the image plane. A simple example of this is when an antenna happens to be behind a solar panel when the satellite is imaged.

There are two types of "feature definitions," "primary feature definitions" and "secondary feature definitions." "Primary feature definitions" are features that can be used as the reference x-axis in the ASSFC systems. For this project the only "primary feature definition" is a line. "Secondary feature definitions" are the kinds of shapes that create distinctive edges in images but cannot be used as reference axes. These can be shapes such as cylinders, cones, or spheres. With some additional research, the cylinder and/or cone could also be used as a "primary feature definition."

Once the physical "feature definitions" are defined in the model and the model is read by the program, the program knows what the possible sources are for edges in the imagery. The next step is to read in the imagery. Once the imagery is read into the program, there are several image-processing steps that need to occur. The first may be to sharpen the image in order to distinguish information in the image. The next is to run an edge-detecting algorithm on the image. The final step is to register the edges and read them into the program. Additionally the user may register the edges through some GUI, possibly by clicking on the two end points of an edge. At this point all edge information is in pixels, so there must be a conversion from pixels to the length units of the model. The easiest way, if all data is known, is to use the horizontal and vertical instantaneous field of view of the pixels and the range to the satellite.

Now each edge read in from the image will be compared to each "primary feature definition," which for this project will be all lines. The purpose for this is to find a possible line for the reference x-axis of an ASSFC system. If the length of the edge is equal to or less than the

length of the line being evaluated, that line will be evaluated and an ASSFC system will be established using the line as the x-axis.

A 3-2-1 Euler rotation sequence will be used to build the IFC to ASSFC DCM. Beginning with the ASSFC system aligned with the IFC system. The first rotation is a rotation about the z-axis to get the correct angle orientation of the x-axis (line) with the edge of interest in the image. There are two possible correct rotations (θ_1 and θ_1+180 deg). The next rotation is a rotation about the new y-axis to get the correct line length projected onto the image plane. There are two possible correct rotations for this rotation as well (one end of the line into the image/one end out of the image and the reverse). The model line will now be aligned suitably to project the same size and orientation onto the image plane as the edge of interest. Now there are four possible orientations for the model line being studied that must be examined. The next step for each of the four situations is to rotate about the new x-axis (the model line) to match the rest of the features and offer a possible attitude solution for the satellite (if the model line being studied is in fact the edge under investigation).

To determine the final rotation another model "feature definition" (in this case either a primary "feature definition" or a secondary "feature definition" can be used) must be compared to other edges in the image. The same type of comparison as discussed above could be done again to compare the remaining lines, but the actual modeled feature could be affected in the image by physical or shadowing effects and would then incorrectly not be considered a fit. If a line does fit an edge exactly, this would be a strong piece of evidence that there is a match. An exact fit should have a higher weight than the possible shadow/physically affected fit. In the end each primary "feature definition" will be compared to each edge in the image, and there will be several matches. Ideally for every ASSFC system developed there should be a correct solution

for almost every other registered edge (excluding those produced by shadowing/physical effects), and ideally the DCM for the correct attitude would be the most common solution. This assumes that initially there is absolutely no input from the user, which could increase the accuracy of the algorithm.

To find the value of θ_3 for each separate analysis there are three different possible types of evaluation. The first is an exact fit evaluation, which assumes that the entire second "feature definition" being used for the θ_3 calculations is projected onto the image plane. There are no shadowing effects or physical blocking of that "feature definition's" representative edge in the image. The second is a derivative of the first. The second type of evaluation is an exact fit evaluation where θ_2 is a multiple of π. The reason why this becomes a separate evaluation is because when the first "feature definition" (ASSFC x-axis) is parallel to the image plane, there is an ambiguity. The third type of evaluation is the shadow/physical effect evaluation, where it is assumed that only a portion of the second "feature definition" is projected onto the image plane. The theory behind all of these evaluations is presented in sections 3.1.3.1 – 3.1.3.3 below.

There is a question about which vector(s) should be used to perform the three different evaluations that are described below, and this is discussed later in the Applications section. Because many of the vectors being compared are relative to specific points, the ability to be accurate while relating a three dimensional coordinate system based on a two dimensional image to a three dimensional coordinate system with a different origin is questionable, and the best technique to apply becomes a matter of testing.

3.1.3.1: Exact Fit Evaluation

The full rotation sequence from the IFC system to the ASSFC system of a vector can be defined by the following equation:

$$\bar{v}_b = \bar{\bar{R}}_1(\theta_3)\bar{\bar{R}}_2(\theta_2)\bar{\bar{R}}_3(\theta_1)\bar{v}_i \tag{3.1}$$

where $\bar{v}_b$ is the vector in the ASSFC system and $\bar{v}_i$ is the vector in the IFC system.

Equation 3.1 can be broken down into

$$\begin{bmatrix} v_{b_x} \\ v_{b_y} \\ v_{b_z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{i_x} \\ v_{i_y} \\ v_{i_z} \end{bmatrix} \tag{3.2}$$

The values of θ_1, θ_2, v_{bx}, v_{by}, v_{bz}, v_{ix}, and v_{iy} are all known at this point. The values of θ_3 and v_{iz} are unknown. The matrix equation above breaks down into the following three equations:

$$v_{b_x} = (v_{i_x}\cos\theta_1 + v_{i_y}\sin\theta_1)\cos\theta_2 - v_{i_z}\sin\theta_2 \tag{3.3}$$

$$v_{b_y} = (-v_{i_x}\sin\theta_1 + v_{i_y}\cos\theta_1)\cos\theta_3 + ((v_{i_x}\cos\theta_1 + v_{i_y}\sin\theta_1)\sin\theta_2 + v_{i_z}\cos\theta_2)\sin\theta_3 \tag{3.4}$$

$$v_{b_z} = (v_{i_x}\sin\theta_1 - v_{i_y}\cos\theta_1)\sin\theta_3 + ((v_{i_x}\cos\theta_1 + v_{i_y}\sin\theta_1)\sin\theta_2 + v_{i_z}\cos\theta_2)\cos\theta_3 \tag{3.5}$$

The value of v_{iz} can be determined from Equation 3.3:

$$v_{i_z} = \frac{v_{i_x} \cos\theta_1 + v_{i_y} \sin\theta_1}{\tan\theta_2} - \frac{v_{b_x}}{\sin\theta_2} \tag{3.6}$$

(The case where θ_2 is a multiple of π will be discussed in Chapter 3.3.2)

Then, the value for θ_3 can be found by the following steps:

From Equation 3.4

$$\frac{v_{b_y}}{\cos\theta_3} = -v_{i_x} \sin\theta_1 + v_{i_y} \cos\theta_1 + \tan\theta_3 (\sin\theta_2 (v_{i_x} \cos\theta_1 + v_{i_y} \sin\theta_1) + v_{i_z} \cos\theta_2) \tag{3.7}$$

and from Equation 3.5

$$\frac{v_{b_z}}{\cos\theta_3} = -\tan\theta_3 (-v_{i_x} \sin\theta_1 + v_{i_y} \cos\theta_1) + \sin\theta_2 (v_{i_x} \cos\theta_1 + v_{i_y} \sin\theta_1) + v_{i_z} \cos\theta_2 \tag{3.8}$$

so Equation 3.7 divided by Equation 3.8 becomes

$$\frac{v_{b_y}}{v_{b_z}} = \frac{-v_{i_x} \sin\theta_1 + v_{i_y} \cos\theta_1 + \tan\theta_3 (\sin\theta_2 (v_{i_x} \cos\theta_1 + v_{i_y} \sin\theta_1) + v_{i_z} \cos\theta_2)}{-\tan\theta_3 (-v_{i_x} \sin\theta_1 + v_{i_y} \cos\theta_1) + \sin\theta_2 (v_{i_x} \cos\theta_1 + v_{i_y} \sin\theta_1) + v_{i_z} \cos\theta_2} \tag{3.9}$$

Take the following steps,

$$\begin{aligned} &v_{b_y} (-\tan\theta_3 (-v_{i_x} \sin\theta_1 + v_{i_y} \cos\theta_1) + \sin\theta_2 (v_{i_x} \cos\theta_1 + v_{i_y} \sin\theta_1) + v_{i_z} \cos\theta_2) = \\ &v_{b_z} (-v_{i_x} \sin\theta_1 + v_{i_y} \cos\theta_1 + \tan\theta_3 (\sin\theta_2 (v_{i_x} \cos\theta_1 + v_{i_y} \sin\theta_1) + v_{i_z} \cos\theta_2)) \end{aligned} \tag{3.10}$$

$$v_{b_z}\tan\theta_3(\sin\theta_2(v_{i_x}\cos\theta_1+v_{i_y}\sin\theta_1)+v_{i_z}\cos\theta_2)+v_{b_y}\tan\theta_3(-v_{i_x}\sin\theta_1+v_{i_y}\cos\theta_1)=$$
$$v_{b_y}(\sin\theta_2(v_{i_x}\cos\theta_1+v_{i_y}\sin\theta_1)+v_{i_z}\cos\theta_2)+v_{b_z}(v_{i_x}\sin\theta_1-v_{i_y}\cos\theta_1) \quad (3.11)$$

to get

$$\theta_3=\tan^{-1}\left(\frac{v_{b_y}(\sin\theta_2(v_{i_x}\cos\theta_1+v_{i_y}\sin\theta_1)+v_{i_z}\cos\theta_2)+v_{b_z}(v_{i_x}\sin\theta_1-v_{i_y}\cos\theta_1)}{v_{b_z}(\sin\theta_2(v_{i_x}\cos\theta_1+v_{i_y}\sin\theta_1)+v_{i_z}\cos\theta_2)+v_{b_y}(-v_{i_x}\sin\theta_1+v_{i_y}\cos\theta_1)}\right) \quad (3.12)$$

3.1.3.2: Exact Fit Evaluation, $\theta_2 = n*\pi$

For the type of evaluation done above, it was shown that v_{iz} can be calculated by

$$v_{i_z}=\frac{v_{i_x}\cos\theta_1+v_{i_y}\sin\theta_1}{\tan\theta_2}-\frac{v_{b_x}}{\sin\theta_2}$$

which is undefined when θ_2 is a multiple of π. To perform this evaluation for $\theta_2 = n*\pi$

$$\bar{v}_b=\bar{\bar{R}}_1(\theta_3)\bar{\bar{R}}_2(\theta_2)\bar{\bar{R}}_3(\theta_1)\bar{v}_i$$

can be expressed as

$$\bar{v}_b=\bar{\bar{R}}_1(\theta_3)\bar{\bar{R}}_2(n\pi)\bar{\bar{R}}_3(\theta_1)\bar{v}_i \quad (3.13)$$

for n=0,1,2…

which breaks down into

$$\begin{bmatrix} v_{b_x} \\ v_{b_y} \\ v_{b_z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix} \begin{bmatrix} \cos n\pi & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \cos n\pi \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{i_x} \\ v_{i_y} \\ v_{i_z} \end{bmatrix} \tag{3.14}$$

for n=0,1,2…

when combined is

$$\bar{v}_b = \bar{\bar{R}}_1(\theta_3)\bar{\bar{R}}_2(n\pi)\bar{\bar{R}}_3(\theta_1)\bar{v}_i = \begin{bmatrix} \cos\theta_1\cos n\pi & \sin\theta_1\cos n\pi & 0 \\ -\cos\theta_3\sin\theta_1 & \cos\theta_3\cos\theta_1 & \sin\theta_3\cos n\pi \\ \sin\theta_3\sin\theta_1 & -\sin\theta_3\cos\theta_1 & \cos\theta_3\cos n\pi \end{bmatrix} \begin{bmatrix} v_{i_x} \\ v_{i_y} \\ v_{i_z} \end{bmatrix} \tag{3.15}$$

for n=0,1,2…

The values of θ_1, v_{bx}, v_{by}, v_{bz}, v_{ix}, and v_{iy} are all known. The values of θ_3 and v_{iz} are unknown. The matrix equation above breaks down into the following three equations:

$$v_{b_x} = v_{i_x}\cos\theta_1\cos n\pi + v_{i_y}\sin\theta_1\cos n\pi \tag{3.16}$$

$$v_{b_y} = (-v_{i_x}\sin\theta_1 + v_{i_y}\cos\theta_1)\cos\theta_3 + v_{i_z}\sin\theta_3\cos n\pi \tag{3.17}$$

$$v_{b_z} = (v_{i_x}\sin\theta_1 - v_{i_y}\cos\theta_1)\sin\theta_3 + v_{i_z}\cos\theta_3\cos n\pi \tag{3.18}$$

for n=0,1,2…

Rearrange Equation 3.18 to get

$$v_{i_z} = \frac{\left(-v_{i_x}\sin\theta_1 + v_{i_y}\cos\theta_1\right)\tan\theta_3}{\cos n\pi} + \frac{v_{b_z}}{\cos\theta_3\cos n\pi} \tag{3.19}$$

And substitute into Equation 3.17

$$v_{b_y} = \left(-v_{i_x}\sin\theta_1 + v_{i_y}\cos\theta_1\right)\cos\theta_3 + \left(\left(-v_{i_x}\sin\theta_1 + v_{i_y}\cos\theta_1\right)\tan\theta_3 + \frac{v_{b_z}}{\cos\theta_3}\right)\sin\theta_3 \tag{3.20}$$

$$v_{b_y}\cos\theta_3 = (-v_{i_x}\sin\theta_1 + v_{i_y}\cos\theta_1)\cos^2\theta_3 + v_{b_z}\sin\theta_3 + \left(-v_{i_x}\sin\theta_1 + v_{i_y}\cos\theta_1\right)\sin^2\theta_3 \tag{3.21}$$

Use the trigonometric identity

$$\sin^2\theta + \cos^2\theta = 1 \tag{3.22}$$

$$v_{b_y}\cos\theta_3 = -v_{i_x}\sin\theta_1 + v_{i_y}\cos\theta_1 + v_{b_z}\sin\theta_3 \tag{3.23}$$

$$v_{b_z}\sin\theta_3 - v_{b_y}\cos\theta_3 = v_{i_x}\sin\theta_1 - v_{i_y}\cos\theta_1 \tag{3.24}$$

Use the following trigonometric identities:

$$\sin\theta = \frac{2\tan\frac{\theta}{2}}{1+\tan^2\frac{\theta}{2}} \tag{3.25}$$

$$\cos\theta = \frac{1-\tan^2\frac{\theta}{2}}{1+\tan^2\frac{\theta}{2}} \tag{3.26}$$

Additionally, use the following substitutions

$$U = v_{i_x}\sin\theta_1 - v_{i_y}\cos\theta_1 \tag{3.27}$$

$$x = \tan\frac{\theta_3}{2} \tag{3.28}$$

so that

$$v_{b_z}\frac{2x}{1+x^2} - v_{b_y}\frac{1-x^2}{1+x^2} = U \tag{3.29}$$

$$2v_{b_z}x - v_{b_y} + v_{b_y}x^2 = U(1+x^2) \tag{3.30}$$

$$(v_{b_y} - U)x^2 + 2v_{b_z}x - (v_{b_y} + U) = 0 \tag{3.31}$$

In addition the solution for an equation in quadratic form,

$$ax^2 + bx + c = 0 \tag{3.32}$$

is known to be

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{3.33}$$

Use Equation 3.33 to solve Equation 3.31

$$x = \frac{-2v_{b_z} \pm \sqrt{4{v_{b_z}}^2 + 4(v_{b_y} - U)(v_{b_y} + U)}}{2(v_{b_y} - U)} \tag{3.34}$$

$$x = \frac{-v_{b_z} \pm \sqrt{{v_{b_z}}^2 + {v_{b_y}}^2 - U^2}}{v_{b_y} - U} \tag{3.35}$$

Now substitute for x

$$\tan\frac{\theta_3}{2} = \frac{-v_{b_z} \pm \sqrt{{v_{b_z}}^2 + {v_{b_y}}^2 - U^2}}{v_{b_y} - U} \tag{3.36}$$

$$\theta_3 = 2\tan^{-1}\left(\frac{-v_{b_z} \pm \sqrt{{v_{b_z}}^2 + {v_{b_y}}^2 - U^2}}{v_{b_y} - U}\right) \tag{3.37}$$

Now substitute for U

$$\theta_3 = 2\tan^{-1}\frac{-v_{b_z} \pm \sqrt{{v_{b_z}}^2 + {v_{b_y}}^2 - (v_{i_x}\sin\theta_1 - v_{i_y}\cos\theta_1)^2}}{v_{b_y} - v_{i_x}\sin\theta_1 + v_{i_y}\cos\theta_1} \tag{3.38}$$

What does this physically mean? U is the size of the image-frame-projected length perpendicular to the ASSFC x-axis. This means the following:

1. In order for the vector represented in the image frame to even possibly be equivalent to the vector represented in the ASSFC system, the size of the non-x portion of the vector in the ASSFC system must be larger than the projected length perpendicular to the ASSFC x-axis. This describes the limitation of the portion of the equation under the radical.
2. The ± comes from the fact that the θ_2 rotation gives no depth to the image so two options for the θ_3 rotation (into and out of the image plane) are possibilities. If the non-x portion of the vector in the ASSFC system is equal to the projected length in the IFC system perpendicular to the ASSFC x-axis, then there is only one possibility.
3. If the y value of the ASSFC vector is equal to the projected length in the IFC system perpendicular to the ASSFC x-axis, then the rotation angle is a multiple of π. The size of the projection onto the image plane at the attitude given is equivalent to what is seen in the image.

An example where $\theta_1 = 0$ (x-axis of ASSFC system parallel to x-axis of IFC system), may be easier to visualize:

$$\theta_3 = 2\tan^{-1}\frac{-v_{b_z} \pm \sqrt{{v_{b_z}}^2 + {v_{b_y}}^2 - {v_{i_y}}^2}}{v_{b_y} + v_{i_y}} \tag{3.39}$$

In the radical, the non x components of a vector in the ASSFC system must be equal to or larger than the y component of the vector in the IFC system to be able to match. If $v_{by} + v_{iy} = 0$, the rotation will be a multiple of π because either the size of the non-x projection onto the image plane is already equal to but opposite of the y portion of the ASSFC vector or both are parallel to their respective x-axes. Now if the non x components of a vector in the body are equal to the y component of the vector in the image, then there can only be one possible θ_3 rotation to line up the vectors.

3.1.3.3: Shadow/Physical Effect Evaluation

Many of the edges found in imagery can be either effects of or created by shadowing. Others can be affected by the physical properties of the satellite. These phenomena need to be addressed to obtain accurate results. This algorithm will indirectly ignore those edges created by shadowing effects by ruling them out through evaluation and comparison of known modeled "feature definitions" with those edges. If there is no match among any of the "feature definitions", the edge is assumed to be a shadowing effect. On the other hand some edges found in imagery do correspond to modeled "feature definitions," but are affected by either shadowing or physical effects. An example of a physical effect is the blocking of the light projection of an

antenna onto the image plane by a solar array. A portion of the antenna (a “feature definition” that would most likely be modeled) may be showing but not the entire length. In Figure 4 on the left the entire length of the antenna and all edges of the labeled face can be seen, but on the right the antenna and three of four sides are affected by the presence of the solar panel. Shadowing effects are related to physical effects. An example of a shadowing effect would be where the light from the sun does not reach the antenna because the light is blocked by the solar panel. Though the entire antenna profile projection may not be physically blocked from the image plane, there is only light reflecting off of a portion of it.

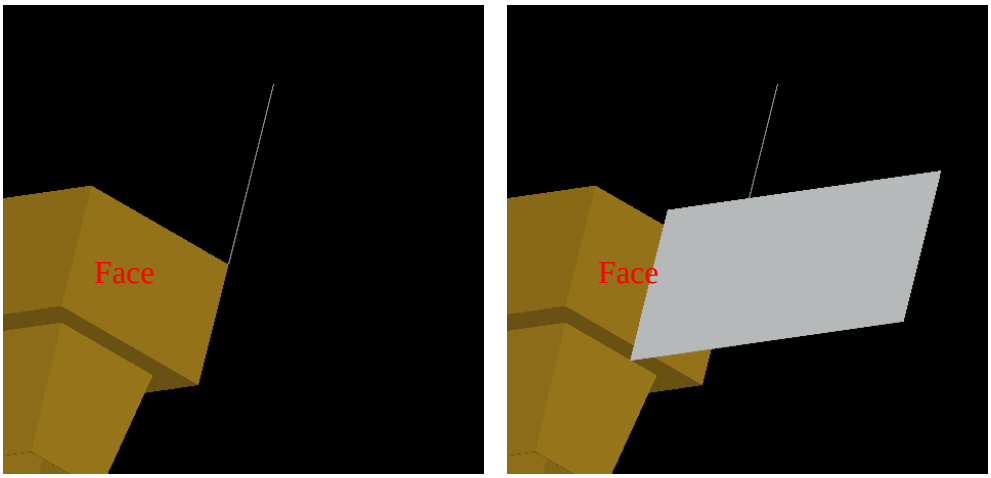

Figure 4: Demonstration of Physical Effects

In both cases, a portion of the “feature definition” is showing and for a line, that portion could be considered a fraction of the whole so that the length of the vector in the ASSFC system is *m* times the length of the vector in the image frame coordinate system.

To address this, an assumption can be made that a portion of the "feature definition" appears in the image but is only a fraction of the true length. Let $\frac{1}{m}$ be the fraction of the "feature definition" that is showing so that

$$\bar{v}_b = \bar{\bar{R}}_1(\theta_3)\bar{\bar{R}}_2(\theta_2)\bar{\bar{R}}_3(\theta_1)m\bar{v}_i \tag{3.40}$$

which breaks down into

$$\begin{bmatrix} v_{b_x} \\ v_{b_y} \\ v_{b_z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & \sin\theta_3 \\ 0 & -\sin\theta_3 & \cos\theta_3 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} mv_{i_x} \\ mv_{i_y} \\ mv_{i_z} \end{bmatrix} \tag{3.41}$$

The values of θ_1, θ_2, v_{bx}, v_{by}, v_{bz}, v_{ix}, and v_{iy} are all known. The values of θ_3, v_{iz}, and m are unknown. The matrix equation above breaks down into the following three equations:

$$v_{b_x} = (mv_{i_x}\cos\theta_1 + mv_{i_y}\sin\theta_1)\cos\theta_2 - mv_{i_z}\sin\theta_2 \tag{3.42}$$

$$\begin{aligned} v_{b_y} &= (-mv_{i_x}\sin\theta_1 + mv_{i_y}\cos\theta_1)\cos\theta_3 + \\ &((mv_{i_x}\cos\theta_1 + mv_{i_y}\sin\theta_1)\sin\theta_2 + mv_{i_z}\cos\theta_2)\sin\theta_3 \end{aligned} \tag{3.43}$$

$$\begin{aligned} v_{b_z} &= -(-mv_{i_x}\sin\theta_1 + mv_{i_y}\cos\theta_1)\sin\theta_3 + \\ &((mv_{i_x}\cos\theta_1 + mv_{i_y}\sin\theta_1)\sin\theta_2 + mv_{i_z}\cos\theta_2)\cos\theta_3 \end{aligned} \tag{3.44}$$

Let

$$u = v_{i_x} \cos\theta_1 + v_{i_y} \sin\theta_1 \tag{3.45}$$

which is known and

$$w = -v_{i_x} \sin\theta_1 + v_{i_y} \cos\theta_1 \tag{3.46}$$

which is also known. Then

$$v_{b_x} = mu\cos\theta_2 - mv_{i_z} \sin\theta_2 \tag{3.47}$$

$$v_{b_y} = mw\cos\theta_3 + (mu\sin\theta_2 + mv_{i_z} \cos\theta_2)\sin\theta_3 \tag{3.48}$$

$$v_{b_z} = -mw\sin\theta_3 + (mu\sin\theta_2 + mv_{i_z} \cos\theta_2)\cos\theta_3 \tag{3.49}$$

Rearrange Equation 3.47 to solve for v_{iz}

$$v_{i_z} = \frac{u}{\tan\theta_2} - \frac{v_{b_x}}{m\sin\theta_2} \tag{3.50}$$

and substitute into Equation 3.48

$$v_{b_y} = mw\cos\theta_3 + \left(mu\sin\theta_2 + m\left(\frac{u}{\tan\theta_2} - \frac{v_{b_x}}{m\sin\theta_2} \right)\cos\theta_2 \right)\sin\theta_3 \tag{3.51}$$

Rearrange to find *m*

$$v_{b_y} = mw\cos\theta_3 + \left(mu\sin\theta_2 + \frac{mu\cos\theta_2}{\tan\theta_2} - \frac{v_{b_x}}{\tan\theta_2} \right)\sin\theta_3 \tag{3.52}$$

$$v_{b_y}\tan\theta_2 + v_{b_x}\sin\theta_3 = mw\tan\theta_2\cos\theta_3 + mu\sin\theta_2\tan\theta_2\sin\theta_3 + mu\cos\theta_2\sin\theta_3 \tag{3.53}$$

$$m = \frac{v_{b_y}\tan\theta_2 + v_{b_x}\sin\theta_3}{w\tan\theta_2\cos\theta_3 + u\sin\theta_2\tan\theta_2\sin\theta_3 + u\cos\theta_2\sin\theta_3} \tag{3.54}$$

Additionally, *m* can be solved for using Equation 3.49

$$v_{b_z} = -mw\sin\theta_3 + \left(mu\sin\theta_2 + m\left(\frac{u}{\tan\theta_2} - \frac{v_{b_x}}{n\sin\theta_2} \right)\cos\theta_2 \right)\cos\theta_3 \tag{3.55}$$

$$v_{b_z} = -mw\sin\theta_3 + \left(mu\sin\theta_2 + \frac{mu\cos\theta_2}{\tan\theta_2} - \frac{v_{b_x}}{\tan\theta_2} \right)\cos\theta_3 \tag{3.56}$$

$$v_{b_z} \tan\theta_2 + v_{b_x} \cos\theta_3 = -mw \tan\theta_2 \sin\theta_3 + mu \sin\theta_2 \tan\theta_2 \cos\theta_3 + mu \cos\theta_2 \cos\theta_3 \tag{3.57}$$

$$m = \frac{v_{b_z} \tan\theta_2 + v_{b_x} \cos\theta_3}{-w \tan\theta_2 \sin\theta_3 + u \sin\theta_2 \tan\theta_2 \cos\theta_3 + u \cos\theta_2 \cos\theta_3} \tag{3.58}$$

These are equal so

$$\frac{v_{b_y} \tan\theta_2 + v_{b_x} \sin\theta_3}{w \tan\theta_2 \cos\theta_3 + u \sin\theta_2 \tan\theta_2 \sin\theta_3 + u \cos\theta_2 \sin\theta_3} = \frac{v_{b_z} \tan\theta_2 + v_{b_x} \cos\theta_3}{-w \tan\theta_2 \sin\theta_3 + u \sin\theta_2 \tan\theta_2 \cos\theta_3 + u \cos\theta_2 \cos\theta_3} \tag{3.59}$$

Rearrange Equation 3.59 and let

$$LHS = (v_{b_y} \tan\theta_2 + v_{b_x} \sin\theta_3)(-w \tan\theta_2 \sin\theta_3 + u \sin\theta_2 \tan\theta_2 \cos\theta_3 + u \cos\theta_2 \cos\theta_3) \tag{3.60}$$

and

$$RHS = (v_{b_z} \tan\theta_2 + v_{b_x} \cos\theta_3)(w \tan\theta_2 \cos\theta_3 + u \sin\theta_2 \tan\theta_2 \sin\theta_3 + u \cos\theta_2 \sin\theta_3) \tag{3.61}$$

so that

$$LHS = RHS \tag{3.62}$$

Expand Equation 3.62 using Equation 3.60

$$-wv_{b_y}\tan^2\theta_2\sin\theta_3+uv_{b_y}\sin\theta_2\tan^2\theta_2\cos\theta_3+uv_{b_y}\sin\theta_2\cos\theta_3-wv_{b_x}\tan\theta_2\sin^2\theta_3+$$
$$uv_{b_x}\sin\theta_2\tan\theta_2\sin\theta_3\cos\theta_3+uv_{b_x}\cos\theta_2\sin\theta_3\cos\theta_3=RHS \quad (3.63)$$

Expand Equation 3.62 using Equation 3.61

$$wv_{b_z}\tan^2\theta_2\cos\theta_3+uv_{b_z}\sin\theta_2\tan^2\theta_2\sin\theta_3+uv_{b_z}\sin\theta_2\sin\theta_3+wv_{b_x}\tan\theta_2\cos^2\theta_3+$$
$$uv_{b_x}\sin\theta_2\tan\theta_2\sin\theta_3\cos\theta_3+uv_{b_x}\cos\theta_2\sin\theta_3\cos\theta_3=LHS \quad (3.64)$$

Combine Equation 3.63 and 3.64 to create

$$wv_{b_z}\tan^2\theta_2\cos\theta_3+uv_{b_z}\sin\theta_2\tan^2\theta_2\sin\theta_3+uv_{b_z}\sin\theta_2\sin\theta_3+wv_{b_x}\tan\theta_2\cos^2\theta_3=$$
$$-wv_{b_y}\tan^2\theta_2\sin\theta_3+uv_{b_y}\sin\theta_2\tan^2\theta_2\cos\theta_3+uv_{b_y}\sin\theta_2\cos\theta_3-wv_{b_x}\tan\theta_2\sin^2\theta_3 \quad (3.65)$$

Use the trigonometric identity in Equation 3.22 and reduce to

$$-wv_{b_y}\tan^2\theta_2\sin\theta_3+uv_{b_y}\sin\theta_2\tan^2\theta_2\cos\theta_3+uv_{b_y}\sin\theta_2\cos\theta_3=$$
$$wv_{b_z}\tan^2\theta_2\cos\theta_3+uv_{b_z}\sin\theta_2\tan^2\theta_2\sin\theta_3+uv_{b_z}\sin\theta_2\sin\theta_3+wv_{b_x}\tan\theta_2 \quad (3.66)$$

Divide Equation 3.66 by $\tan\theta_2$

$$\cos\theta_3(uv_{b_y}\ \tan\theta_2\ \sin\theta_2 + uv_{b_y}\ \cos\theta_2 - wv_{b_z}\ \tan\theta_2) -$$
$$\sin\theta_3(wv_{b_y}\ \tan\theta_2 + uv_{b_z}\ \tan\theta_2\ \sin\theta_2 + uv_{b_z}\ \cos\theta_2) = wv_{b_x} \tag{3.67}$$

Equation 3.67 can be solved using the property

$$a\sin\theta + b\cos\theta = \sqrt{a^2 + b^2}\ \sin(\theta + \varphi) \tag{3.68}$$

where

$$\varphi = \tan^{-1}\left(\frac{b}{a}\right) \tag{3.69}$$

In this case

$$a = -(wv_{b_y}\ \tan\theta_2 + uv_{b_z}\ \tan\theta_2\ \sin\theta_2 + uv_{b_z}\ \cos\theta_2) \tag{3.70}$$

which can be reduced to

$$a = -wv_{b_y}\ \tan\theta_2 - \frac{uv_{b_z}}{\cos\theta_2} \tag{3.71}$$

and

$$b = uv_{b_y}\ \tan\theta_2\ \sin\theta_2 + uv_{b_y}\ \cos\theta_2 - wv_{b_z}\ \tan\theta_2 \tag{3.72}$$

which can be reduced to

$$b = -wv_{b_z} \tan\theta_2 + \frac{uv_{b_y}}{\cos\theta_2} \tag{3.73}$$

From Equations 3.68 and 3.69

$$\theta_3 = \sin^{-1}\left(\frac{wv_{b_x}}{\sqrt{a^2+b^2}}\right) - \tan^{-1}\left(\frac{b}{a}\right) \tag{3.74}$$

and substitute in from Equations 3.71 and 3.73 and simplify

$$\theta_3 = \sin^{-1}\left(\frac{wv_{b_x}\cos\theta_2}{\sqrt{2w^2{v_{b_z}}^2\sin^2\theta_2 + 2u^2{v_{b_y}}^2}}\right) - \tan^{-1}\left(\frac{wv_{b_z}\sin\theta_2 - uv_{b_y}}{wv_{b_z}\sin\theta_2 + uv_{b_y}}\right) \tag{3.75}$$

Equation 3.75 shows that, as expected, there are also limitations to the shadow/physical effect evaluation. The value of $\sqrt{2w^2{v_{b_z}}^2\sin^2\theta_2 + 2u^2{v_{b_y}}^2}$ must be larger than the value of ($wv_{b_x}\cos\theta_2$) and the value of ($2w^2{v_{b_z}}^2\sin^2\theta_2 + 2u^2{v_{b_y}}^2$) must be greater than zero. Also, the value of *m* must be greater than or equal to 1.

3.4: Application

The theory described above suggests that this study's approach is feasible. The next step is to develop software code that will implement and test it. Matlab was chosen to do this. Matlab was specifically developed for handling matrix and vector formulations. It is a very versatile language that is spreading as an industry standard. In addition there are several software packages, or "toolboxes," available that include functions used for specific applications. In this case the "Image Processing" toolbox is particularly useful because it includes several functions that are required to perform this analysis, including edge detection and edge registration functions. The scope of this thesis does not include improving or developing any image processing techniques that are not already available, so Matlab with its "Image Processing" toolbox was ideal.

The code that was developed has five sections that are described below.

3.4.1: Read Model File

This section reads the file that defines all of the satellite's "feature definitions." As described above "feature definitions" refer to features on the satellite that are likely to produce intensity gradients, or edges, in imaging focal planes. These "feature definitions" are defined in this file in the MDSFC system. Geometric lighting effects are not taken into account in the model file, but structural edges and changes in material properties are. Ideally, this file would

have already been created along with the CAD model and verified by potential users before launch to ensure that it contained the most accurate information on physical dimensions and configuration. In this case configuration may refer to anything from correct blanketing to correct placement of articulating components such as solar panels and antennas. For this project the only features that were investigated were primary "feature definitions," which are made up of straight lines only. Because of this, the model file for this project will be made up of a text document containing lines like the one in Figure 5. The first column tells the program that the feature is a line and, therefore, a primary feature. The next three columns tell the program the x, y, and z values in the MDSFC system of the first endpoint of the line, and the last three columns tell the program the x, y, and z values in the MDSFC system of the second endpoint of the line.

```
LINE  1.00000E+00  2.00000E+00  3.00000E+00  1.00000E+00  2.00000E+00 -3.00000E+00
```

Figure 5: Example of "Feature Definition" Input

Ideally in the future there would also be secondary "feature definitions" like cylinders, spheres, and cones. Additionally this is where the number of each feature is counted. For this project the model file is simply called "Model.txt," but obviously the different ways that this file can be input (or even written) are nearly limitless.

3.4.2: Read Imagery Parameters and Imagery

In an operational environment this section would be a function that would constantly read in imagery for analysis and imager parameters associated with that imagery, and that data would

be queued for analysis. For this project the imager parameters are input as constants with only one image that must be named “Image.tif.”

The first parameter that must be input is the horizontal instantaneous field of view. This value is given in radians. Ideally this would be a known value and along with the range to the satellite would provide a pixel to length conversion for the image frame coordinate system x-axis. This approach assumes that the entire field of view of the imager is collected by the focal plane array so that spacing between the pixels can be neglected.

The second parameter that must be input is the vertical instantaneous field of view. This value is also given in radians. Like the horizontal instantaneous field of view, ideally this would also be a known value and along with the range to the satellite would provide a pixel to length conversion for the image frame coordinate system y-axis. As stated above, this approach assumes that the entire field of view of the imager is collected by the focal plane array so that spacing between the pixels can be neglected.

The third parameter is the range to the satellite. This value is given in the same units as the features given in the model file and is the slant range from the imager to the satellite. This would also be a known value that would change with each image. As stated above, the range helps calculate a pixel to length conversion for both the x- and y- axes of the image frame coordinate system.

The final parameter is the ECI to IFC system direction cosine matrix. This is a 3 x 3 matrix that converts vectors defined in ECI to the IFC system. This parameter would be a known value that would change with every image due to Earth rotation, gimbal rotation, and satellite tracking.

3.4.3: Image Processing and Edge Registration

This section performs the image processing that is required to detect edges in the images and register those edges as satellite feature projections onto the x-y plane of the IFC system. The image processing techniques used in this project are all well-established and used universally. All of the image processing functions used here are actually functions that come with the Matlab Image Processing Toolbox.

There are a few variables defined at the beginning of this section that can be changed by the user to refine the analysis. Normally these would be adjustable through the user interface. These variables include various resolutions and thresholds for edge registration.

3.4.4: Calculations of Possible DCM Solutions

This is the section that incorporates the theory discussed earlier in this chapter. The section begins by initializing two counter variables – "poscount" and "shadcount." The "poscount" counter variable is used to count all possible correct DCMs based on complete "feature definition" matches. The "shadcount" counter variable is used to count all possible correct DCMs based on partial matches (different lengths but same directions).

The entire goal of this code is to determine the attitude of the satellite by calculating possible DCMs. To do this, only relative positions can be used so all of the absolute position values (pix, piy, pmx, pmy, pmz) will be converted into relative position vector values that are referenced to the image edge end points and model line end points.

The DCM that converts the original MDSFC system to the ASSFC system using the current model line of interest (model line aligned with x-axis, others are arbitrary) is calculated. This is done using two rotations. The only important feature that must be transformed is the model line of interest into the new body x axis. The remaining features can remain in arbitrary positions. The first rotation in the transformation from MDSFC to ASSFC is a rotation about the body z axis. This is done by using the model line projection onto the x-y plane and rotating to align that projection to the x axis. The second rotation is a rotation about the new y-axis to move the model line axis into the x-y plane and therefore along the x-axis. The signs of the rotations correspond to a standard "right-hand rule" rotation.

This approach assumes at least one full line "feature definition" makes a complete edge in the image. This is because the second rotation in the IFC to ASSFC transformation requires a full length comparison when rotated into or out of the image plane. Because all edges will be compared with all lines, at least one edge must be completely well-lit and physically unblocked. This is good for solar panels.

For the calculation of the IFC to ASSFC DCM a loop that runs for each registered linear edge is begun. For each registered linear edge there are only two unique rotations about the image frame's z axis, which are both independent of the model line being used for comparison. This rotation aligns the x axis with the edge's direction – one rotation is a rotation between -90 deg and 90 deg and the other is 180 deg more than the first. Using the standard right-hand rule, the rotation about the z axis will be in the same direction as the angle from the x axis to the edge of interest.

The second rotation in the IFC to ASSFC calculation (a rotation about the new y axis) is dependent upon the model line being used for comparison. The edge in the image frame is a

projection of the satellite feature that the model line represents so the second rotation depends on the length of that feature (and therefore the model line representing that feature). Any model line that is at least the length of the image edge will be considered.

Just like above there are two possible rotations that can align the model line with the edge – into the image frame and out of the image frame. To determine the two possible angles of rotation the inverse cosine of the ratio of the edge length to the model line length and the negative of that angle are used. We now have four different evaluation scenarios with each model-line-to-edge-of-interest situation.

If the edge of interest and model line of interest are aligned and correctly correlated, the last possible rotation ideally will align all remaining edges to remaining model features assuming the features that make the edges in the image are modeled (and are not affected by shadowing/physical effects). This is done by rotating about the newest x-axis (the model line of interest) using the four previously developed situations. Each of those four is separately used as a basis for the third rotation, and other feature comparisons will be done for each case.

For this project two different ways to determine the third rotation were developed. One way uses the comparison of the vectors between all endpoints (referred to in Chapter 4 as Version 2). There are four combinations:

1. Endpoint 1 of line 1 to endpoint 1 of line 2
2. Endpoint 1 of line 1 to endpoint 2 of line 2
3. Endpoint 2 of line 1 to endpoint 1 of line 2
4. Endpoint 2 of line 1 to endpoint 2 of line 2

One of these vectors could be zero because two lines could share one of their endpoints, so this is taken into account logically by using a flag. The θ_3 calculation is performed as a separate

function where the inputs are the known vector values (θ_1, θ_2, v_{bx}, v_{by}, v_{bz}, v_{ix}, and v_{iy}) and a flag saying whether this is a shadow effect evaluation or an exact fit evaluation. Within that function the appropriate calculations are performed and the output is the calculated θ_3 and a flag that says whether the function was able to do the calculations or not. An example of a type of calculation that would give an error flag would be if two endpoints met at a corner or if the projected vector is larger than the body vector.

The second way uses the same separate function as described above and the fact that a three dimensional line is projected onto the two dimensional image plane (referred to in Chapter 4 as Version 1). Using the projected line of interest as the x-axis in the equation of a line, Equation 3.76 and Equation 3.77, for both coordinate systems allows the program to simply calculate and compare one θ_3.

$$y = mx + b \tag{3.76}$$

or

$$x = b \tag{3.77}$$

The results found by using each of these two approaches are compared in the next chapter. As more are compared and more match, for each direction cosine matrix that is calculated and matches within a tolerance there is a higher possibility that the edge/line of interest are correct and that it is the correct attitude solution. There may be several possibilities that come up correct or there may be none, but the correct DCM should occur more often.

Ideally as every pair of “feature definitions” is compared the only ones that match would be those in the correct attitude and various random pairs that simply match coincidentally. Foreseeable exceptions include highly symmetric satellites and not well-resolved imagery. The shadow evaluation answers are used as tie-breakers for times when different DCMs have the same number of occurrences. Once the algorithm is complete, the list of matched DCMs is presented to the user in order of number of occurrences.

CHAPTER 4: RESULTS

Because of availability and controllability, the only way to test this algorithm is through simulation. At the current time there are a few different codes that simulate imagery and focal plane arrays. This analysis uses a code named TASAT (Time-domain Analysis and Simulation for Advanced Tracking). TASAT can provide a medium-to-high fidelity, end-to-end simulation of an electro-optical system (Riker, et al., 1992). Included in the simulation are atmospheric effects as well as other physical effects of the electro-optical system. This code allows a user to basically build an electro-optical site and simulate what it can see given a full physical model and a trajectory of the user's object of interest. To test the basic feasibility of this algorithm three models, one very simple box, one very simple box with an antenna, and one very simple box with four antennas, were created.

4.1: Simple Box Simulations Aligned with IFC System

The first model is the simple box shown in Figure 6 with an aluminized kapton blanket. This simple box presents the first weakness of this approach. Symmetric objects will naturally have multiple solutions. In the case of this box which is symmetric about the x-y plane, the x-z plane, and y-z plane the user should expect several strong outputs.

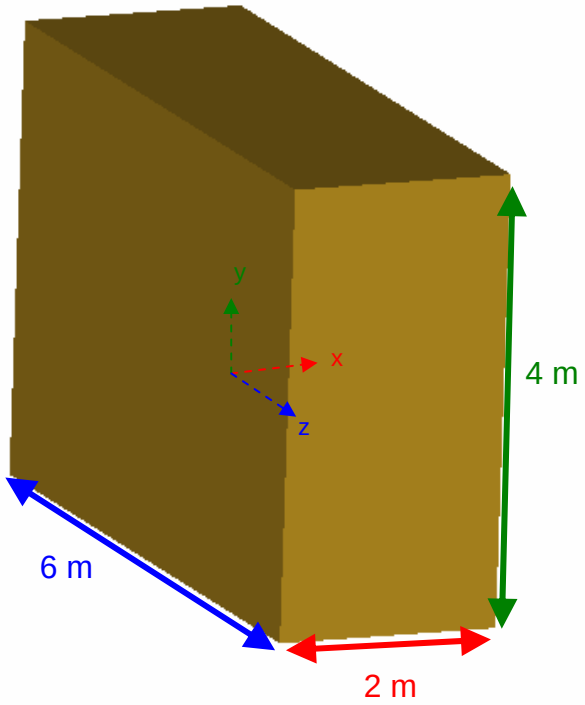

Figure 6: Model of Simple Box

This model has 12 “feature definitions” (each of the edges), so the model file will look like Figure 7.

```
LINE  1.00000E+00  2.00000E+00  3.00000E+00  1.00000E+00  2.00000E+00 -3.00000E+00
LINE  1.00000E+00  2.00000E+00  3.00000E+00  1.00000E+00 -2.00000E+00  3.00000E+00
LINE  1.00000E+00  2.00000E+00  3.00000E+00 -1.00000E+00  2.00000E+00  3.00000E+00
LINE -1.00000E+00 -2.00000E+00 -3.00000E+00 -1.00000E+00 -2.00000E+00  3.00000E+00
LINE -1.00000E+00 -2.00000E+00 -3.00000E+00 -1.00000E+00  2.00000E+00 -3.00000E+00
LINE -1.00000E+00 -2.00000E+00 -3.00000E+00  1.00000E+00 -2.00000E+00 -3.00000E+00
LINE  1.00000E+00  2.00000E+00 -3.00000E+00  1.00000E+00 -2.00000E+00 -3.00000E+00
LINE  1.00000E+00  2.00000E+00 -3.00000E+00 -1.00000E+00  2.00000E+00 -3.00000E+00
LINE  1.00000E+00 -2.00000E+00 -3.00000E+00  1.00000E+00 -2.00000E+00  3.00000E+00
LINE  1.00000E+00 -2.00000E+00  3.00000E+00 -1.00000E+00 -2.00000E+00  3.00000E+00
LINE -1.00000E+00  2.00000E+00  3.00000E+00 -1.00000E+00  2.00000E+00 -3.00000E+00
LINE -1.00000E+00  2.00000E+00  3.00000E+00 -1.00000E+00 -2.00000E+00  3.00000E+00
```

Figure 7: “Feature Definition” Model Input

The box was put through two simulations. In both simulations the optics and range were set up so that the imagery would produce 1 cm resolution. In some cases this may be unrealistic,

but it helps determine the basic feasibility of the approach. The first scenario was set up with the box's MDSFC system aligned with the ECI coordinate system so that the correct answer for the value of the ECI to MDSFC DCM would be

$$DCM_{ECI \to body} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the simulation looked like Figure 8.

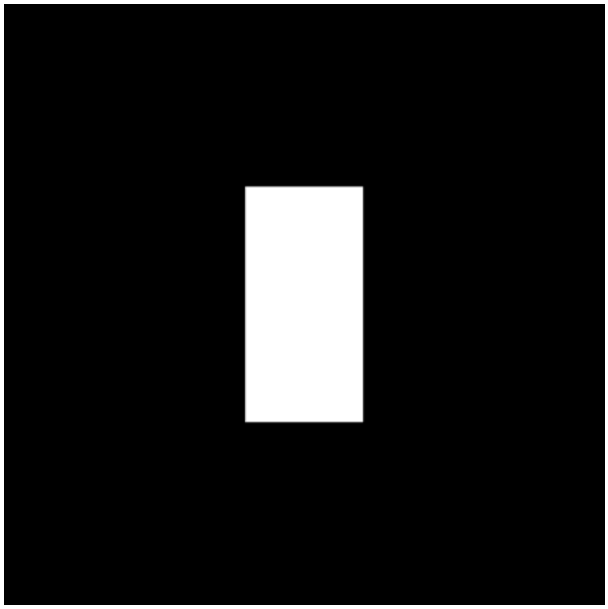

Figure 8: Simple Box Simulation Aligned with ECI

This image was run through the image processing. The algorithm was able to detect all edges as shown in Figure 9, but was only able to register one edge. In this case a human in the loop was necessary to ensure all edges (or at least all appropriate edges) were registered.

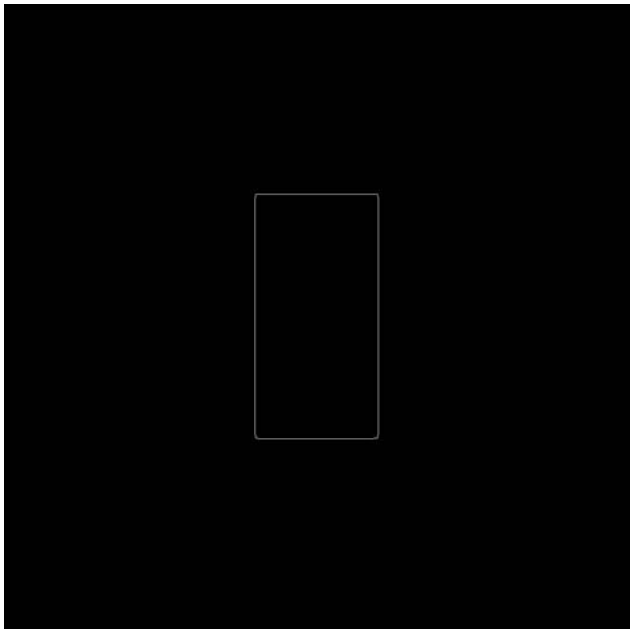

Figure 9: Edge Detection Result of Simple Box Simulation Aligned with ECI

The data was then run through both versions of the code, the analysis of the line vector which will be called Version 1 and the analysis of endpoint vectors which will be called Version 2. It should be noted that the term tier will be used to describe the results. After examining the results, for the following examples, a phenomenon kept reoccurring. The results for matched solutions had a range from tens to hundreds of different DCMs, but there always seemed to be tiers. The most matched DCMs could have an average of 50 matches, but at some point there was always a sudden large decrease in the number of matches per DCM. In other words at some point the average number of matches could suddenly have dropped to 25 matches per DCM and remain there for a span. The exact point does not appear to easily be predictable. Because of this characteristic, the group of most-matched DCMs is considered the top tier, and other groups' tiers follow.

The top four outputs of Version 1 (line vector version) were the four symmetrically equivalent results with each obtaining between 5% and 10% of the total matched solutions:

1. $DCM_{ECI\rightarrow body} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & -1.0000 & 0.0000 \\ 0.0000 & 0.0000 & -1.0000 \end{bmatrix}$ with 10.04% of the matched solutions

2. $DCM_{ECI\rightarrow body} = \begin{bmatrix} -1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & -1.0000 \end{bmatrix}$ with 10.04% of the matched solutions

3. $DCM_{ECI\rightarrow body} = \begin{bmatrix} -1.0000 & 0.0000 & 0.0000 \\ 0.0000 & -1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$ with 6.69% of the matched solutions

4. $DCM_{ECI\rightarrow body} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$ with 5.44% of the matched solutions

The highest percentage of the next tier of solutions was 2.93% of the matched solutions.

Although Version 2 (endpoint vector version) did output the correct answer within the first two outputs, only two of the expected four symmetrically equivalent results were in the top 10:

1. $DCM_{ECI\rightarrow body} = \begin{bmatrix} -1.0000 & 0.0000 & 0.0000 \\ 0.0000 & -1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$ with 11.77% of the matched solutions

2. $DCM_{ECI \to body} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$ with 11.31% of the matched solutions

The next closest tier produced images similar to Figure 10. In this case the gray area is not illuminated by the sun, but the projection onto the image plane produces the equivalent edges that are in the actual image.

$$DCM_{ECI \to body} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.7454 & -0.6667 \\ 0.0000 & 0.6667 & 0.7454 \end{bmatrix}$$ with 3.66% of the matched solutions

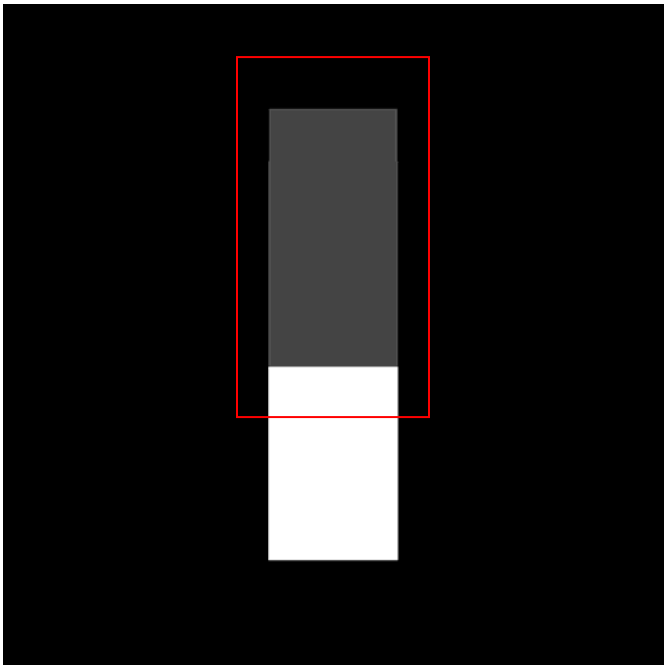

Figure 10: Second Tier Result of Simple Box Aligned with ECI

4.2: Simple Box Simulations not Aligned with IFC System

In this section the simple box will not be aligned with the IFC system. Given in a 3-2-1 Euler rotation, θ_1 will be 30 deg, θ_2 will be 45 deg, and θ_3 will be 20 deg so that the correct DCM is:

$$DCM_{ECI \to body} = \begin{bmatrix} 0.6124 & 0.3536 & -0.7071 \\ -0.2604 & 0.9347 & 0.2418 \\ 0.7465 & 0.0360 & 0.6645 \end{bmatrix}$$

The correct image looks like Figure 11:

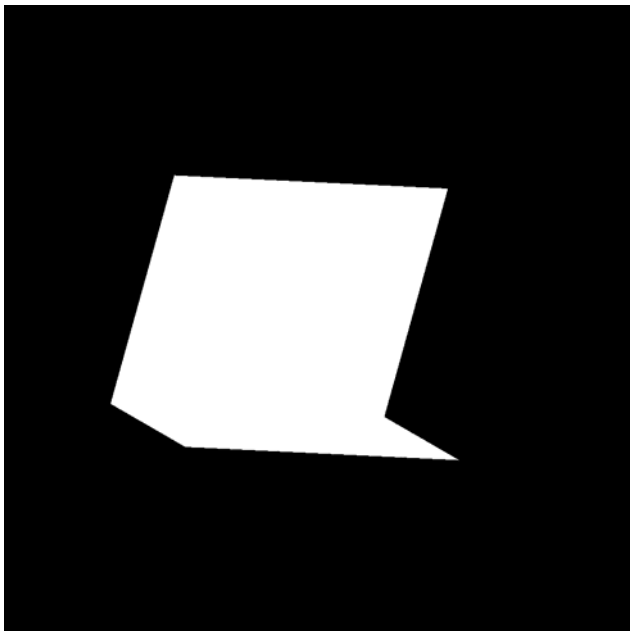

Figure 11: Simple Box Simulation with 30-45-20 Rotation

Version 1 (line vector version) produced one of the symmetrically equivalent results, but the actual answer although still in the highest tier was the seventh strongest output with only 3.11% of the matched solutions. The strongest output was

$$DCM_{ECI\to body} = \begin{bmatrix} 0.6149 & 0.3493 & -0.7070 \\ 0.2586 & -0.9363 & -0.2377 \\ -0.7450 & -0.0367 & -0.6661 \end{bmatrix}$$ with 4.56% of the matched solutions.

Version 2 (endpoint vector version) on the other hand also produced one of the symmetrically equivalent results as one of the strongest outputs (second output below), but the actual answer was the 11th strongest answer with 3.00% of the matched solutions. In addition the shadow tie-breaker placed the first output below as the strongest rather than the second output below. By looking at the image produced by the first output, it is understandable why there would be more shadow analysis, so the algorithm in fact worked correctly. The strongest outputs were

1. $$DCM_{ECI\to body} = \begin{bmatrix} 0.6149 & 0.3493 & -0.7070 \\ 0.2586 & -0.9363 & -0.2377 \\ -0.7450 & -0.0367 & -0.6661 \end{bmatrix}$$ with 8.00% of the matched solutions

2. $$DCM_{ECI\to body} = \begin{bmatrix} 0.6149 & 0.3493 & -0.7070 \\ 0.2586 & -0.9363 & -0.2377 \\ -0.7450 & -0.0367 & -0.6661 \end{bmatrix}$$ with 8.00% of the matched solutions

The first output above produces imagery that looks like the image on the right in Figure 12 (left is actual image) where the gray areas are not actually illuminated by the sun.

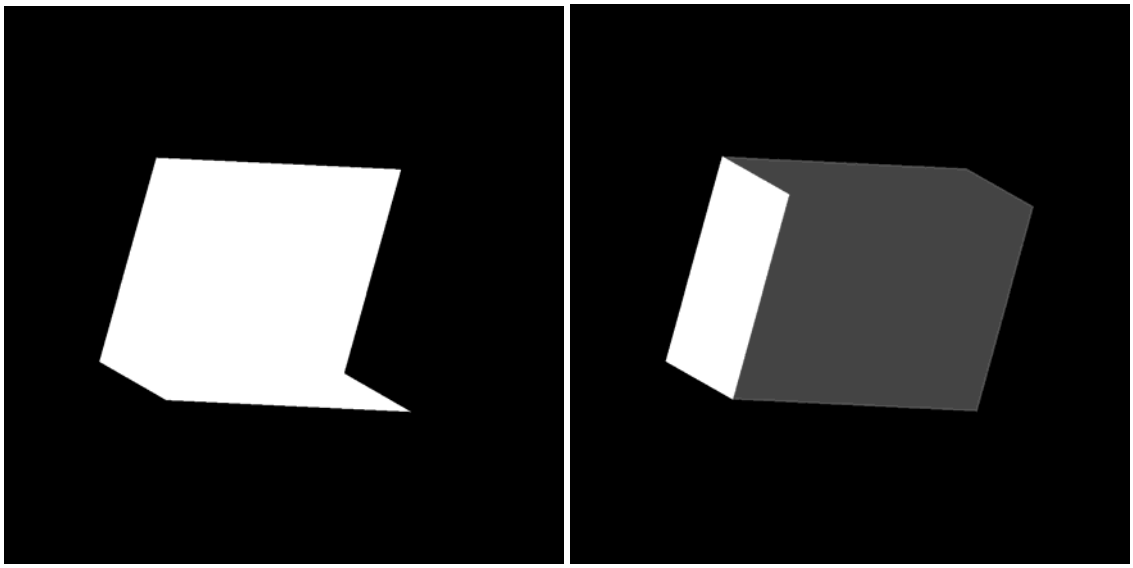

Figure 12: Comparison between Actual Image and Strongest Output of 30-45-20 Version 2

The next item that was examined was an asymmetric body. An “antenna” was added to one side of the simple box to make it asymmetric. The simple box now looks like Figure 13 with antenna circled.

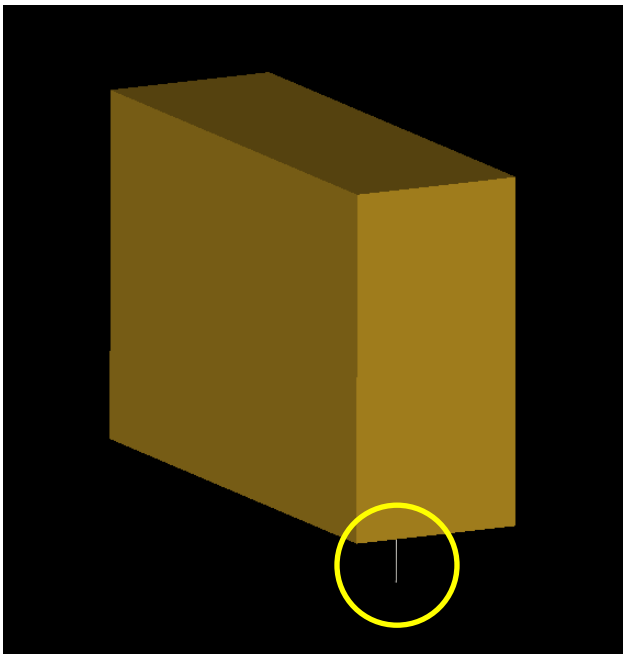

Figure 13: Simple Box with Additional Antenna

The DCM that was used was the same as the DCM for the 30-45-20 rotation shown above so the image looks like Figure 14 with the antenna circled.

$$DCM_{ECI \to body} = \begin{bmatrix} 0.6124 & 0.3536 & -0.7071 \\ -0.2604 & 0.9347 & 0.2418 \\ 0.7465 & 0.0360 & 0.6645 \end{bmatrix}$$

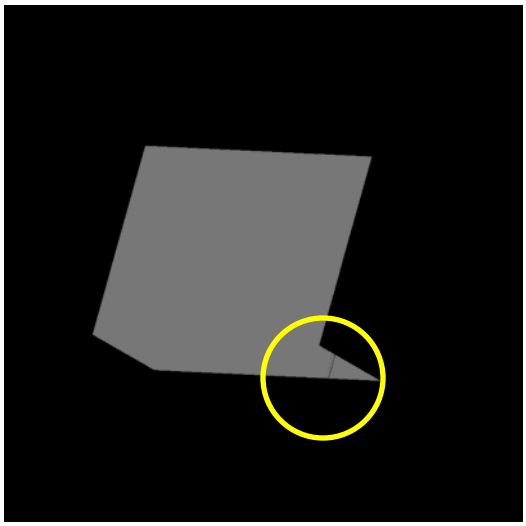

Figure 14: Simulated Image of Simple Box with Antenna in 30-45-20 Rotation

Version 1 (line vector version) produced nearly identical results as before. The actual answer although still in the highest tier was still the seventh strongest output but this time with only 2.34% rather than 3.11% of the matched solutions. The strongest output this time was

$$DCM_{ECI\rightarrow body} = \begin{bmatrix} -0.6140 & -0.3526 & 0.7062 \\ 0.2607 & -0.9351 & -0.2401 \\ 0.7450 & 0.0367 & 0.6661 \end{bmatrix}$$ with 3.43% of the matched solutions.

The strongest output produces the image shown in Figure 15 with the antenna circled.

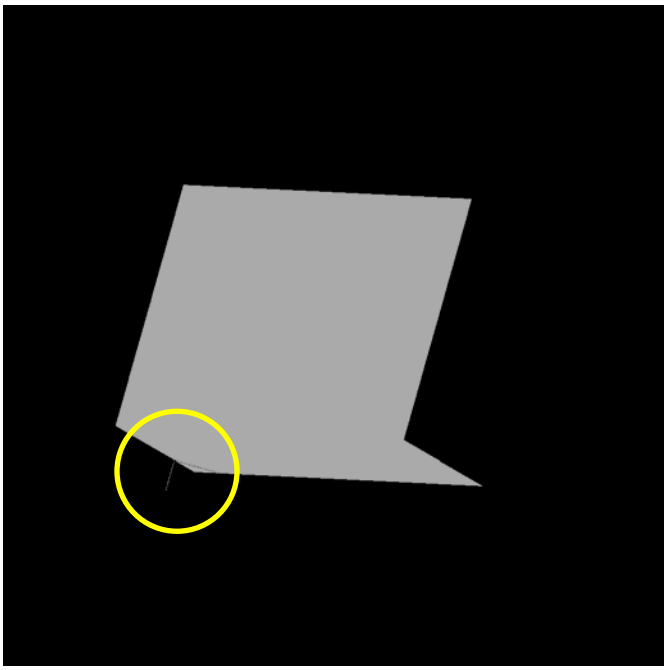

Figure 15: Strongest Version 1 Output for Simple Box with Antenna

The output of Version 2 (endpoint vector version) was also very similar to the simulation without the antenna. This time the actual answer was the 12th strongest answer with 1.70% of the matched solutions. The strongest outputs were

1. $DCM_{ECI \to body} = \begin{bmatrix} -0.6149 & -0.3493 & -0.7070 \\ -0.2586 & 0.9363 & -0.2377 \\ 0.7450 & 0.0367 & -0.6661 \end{bmatrix}$ with 4.55% of the matched solutions

2. $DCM_{ECI \to body} = \begin{bmatrix} 0.6149 & 0.3493 & -0.7070 \\ 0.2586 & -0.9363 & -0.2377 \\ -0.7450 & -0.0367 & -0.6661 \end{bmatrix}$ with 4.55% of the matched solutions

The first strongest output representation is shown in Figure 16 where the gray area in the image on the left (most of the image) shows the boundary. The image on the right shows what is

illuminated by the sun. The antenna is circled in both images. This appears to show that many asymmetric "feature definitions" are required.

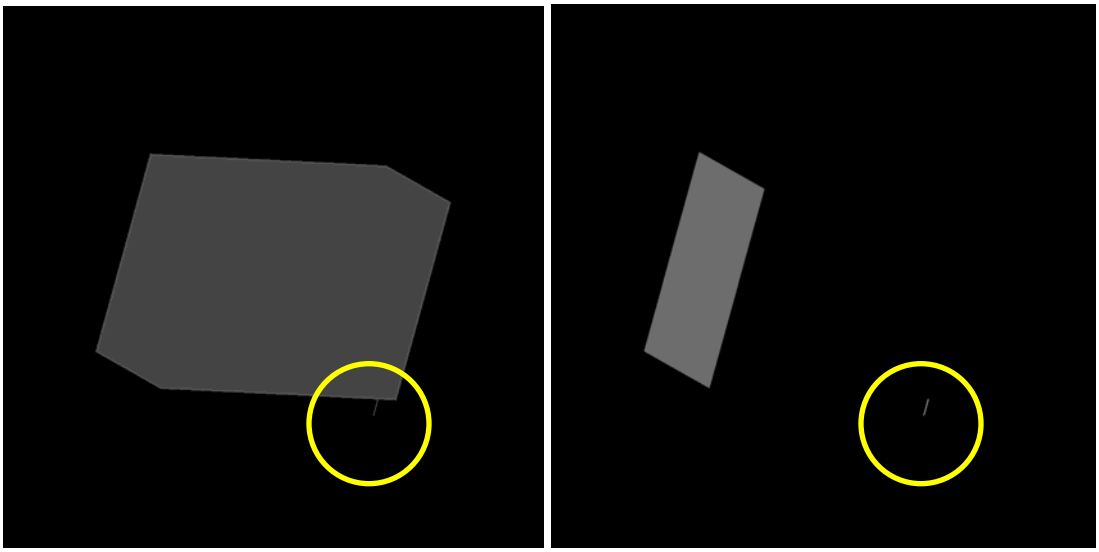

Figure 16: Boundary and Illumination Images of the Strongest Version 2 Output

4.3: Simple Box with Asymmetric Antennas in Realistic Simulation

The next step is to simulate a "realistic" ground-based telescope scenario. The scenarios above could be considered "realistic" servicing satellite scenarios. To do this TASAT allows the user to create his or her own telescope and place it anywhere in the world. In the following scenario the telescope is placed in Orlando, FL at 28.55 deg latitude, -81.38 deg longitude (negative is west of Greenwich), and 30 meters altitude. Obviously this would not be an ideal location for a telescope for various reasons (low altitude, etc), but the purpose of this exercise is to develop a "realistic" situation in which exact location and exact telescope parameters do not completely drive the results. The telescope developed for this simulation has a 1.6 m aperture

with a pixel instantaneous field of view of 3E-07 rad. It is assumed that some sort of image processing can be done to the imagery to enhance edge detection. The new "satellite" model is a modification of the simple box and is shown in Figure 17. Four different sized antenna are placed on the +X, -X, +Y, and -Y faces of the original box. This "satellite" will be traveling in the same orbit as the Hubble Space Telescope, which is at an altitude of about 560 km.

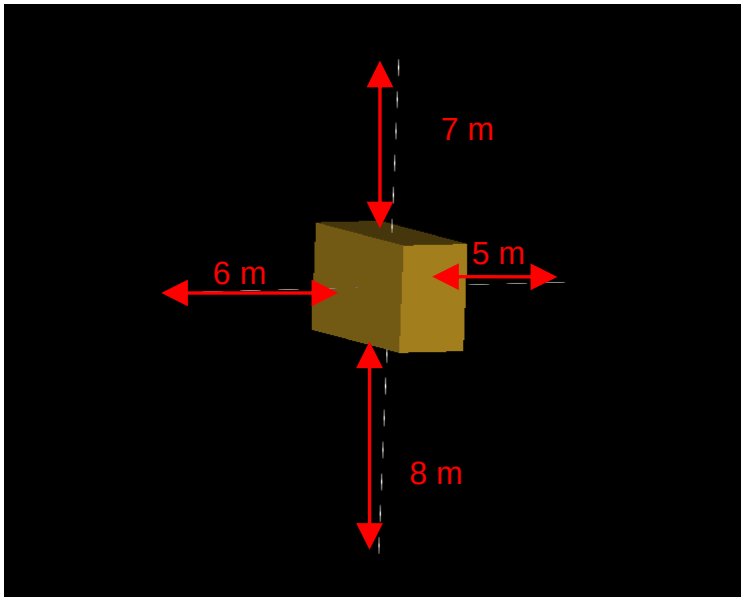

Figure 17: Model of Simple Box with Four Antennas

Figures 18 and 19 below are two simulated images from one very good satellite pass of the scenario described above. Figure 18 shows both a linear and a logarithmic intensity image of the well-lit satellite at a slant range of about 570.4 km and Figure 19 shows both a linear and a logarithmic intensity image of the well-lit satellite at a slant range of about 572.6 km.

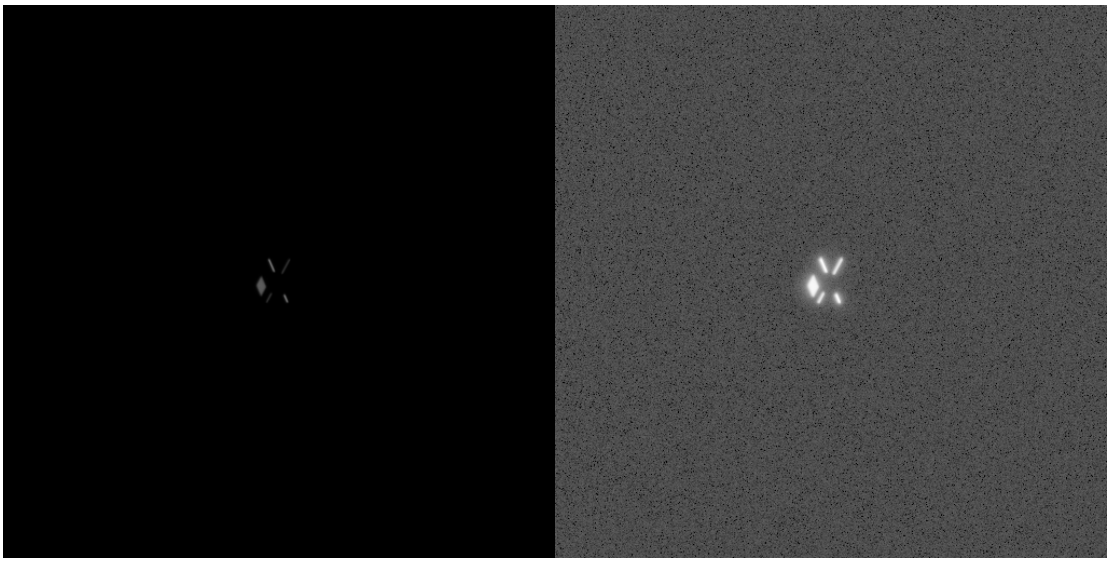

Figure 18: Linear and Logarithmic Images at 570.4 km

The correct DCM for Figure 18 is

$$DCM_{ECI \to body} = \begin{bmatrix} 0.3892 & 0.5728 & -0.7214 \\ 0.3056 & 0.6585 & 0.6877 \\ 0.8690 & -0.4881 & 0.0813 \end{bmatrix}$$

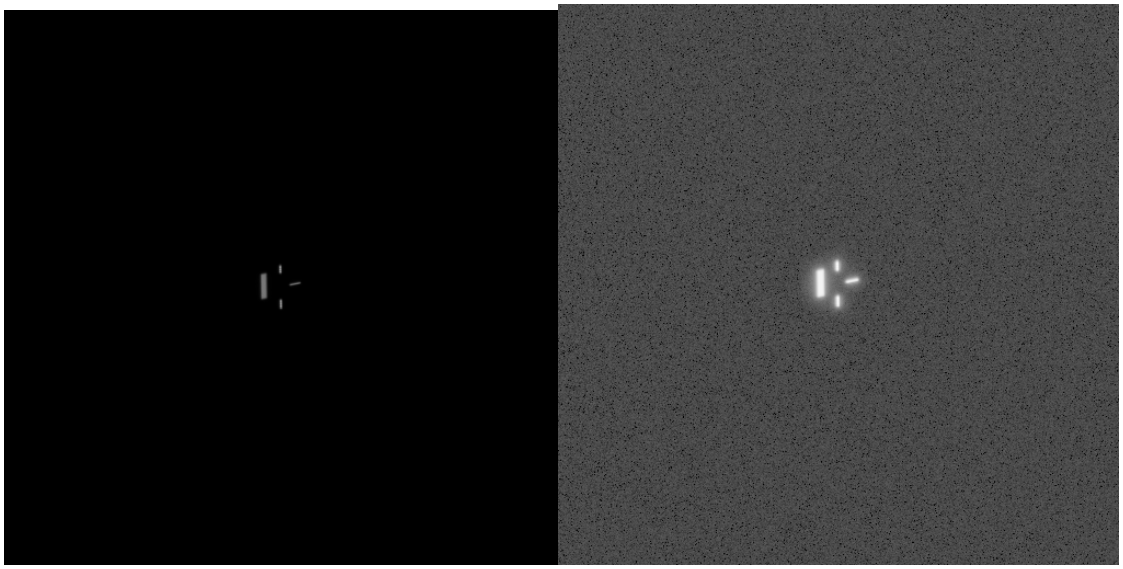

Figure 19: Linear and Logarithmic Images at 572.6 km

The following analysis will examine only Figure 18 using the two versions of the code.

Version 1 (line vector version) was able to determine a relatively close match to the correct answer as the sixth most matched answer (still in the top tier) with 0.46 % (most matched was 0.53 %). The simulated image is shown in Figure 20 and the DCM is given below:

$$DCM_{ECI \to body} = \begin{bmatrix} 0.4443 & 0.5317 & -0.7210 \\ 0.3423 & 0.6429 & 0.6852 \\ 0.8278 & -0.5513 & 0.1037 \end{bmatrix}$$

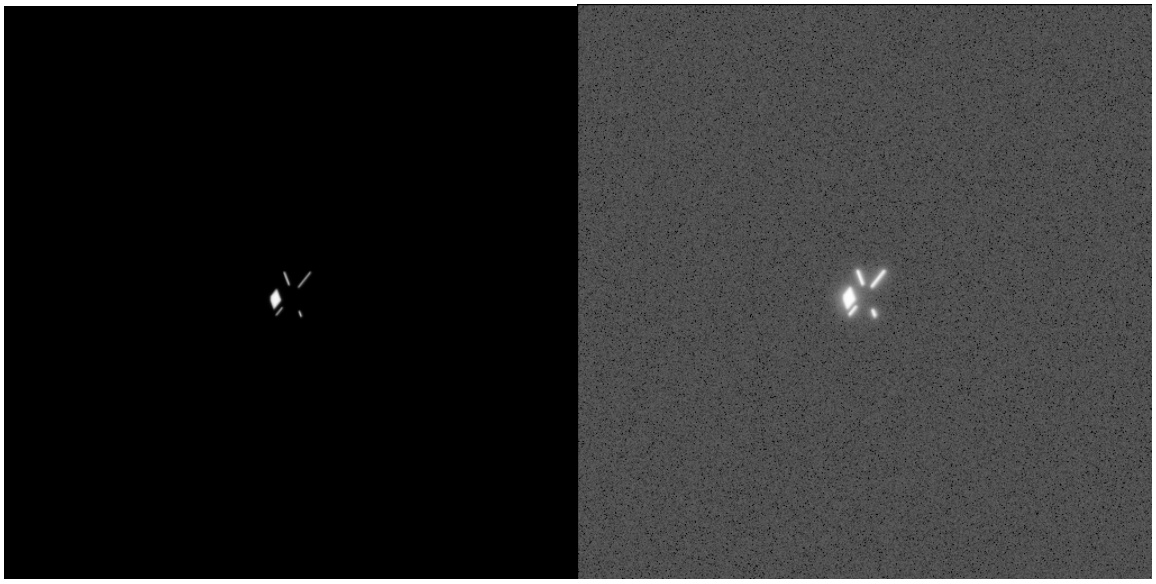

Figure 20: Simulation of Relatively Close Match from Version 1

Version 2 (end point vector version) did not come up with the correct DCM, but additionally while running Version 2, there were 778 DCMs that had a match. The most matches that any one DCM had was three. Obviously for Version 2 the resolution was not low enough.

The results show that this approach may be able to calculate attitude from realistic conditions. Although Version 2 was not able to provide an answer for the "realistic" ground-based telescope scenario, it should not be thrown out. It is possible that Version 2 could be modified to be more stable.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

The examples given have shown that this approach to attitude determination is feasible. Though full autonomy and even autonomous edge registration were not achieved, the concept presented here for attitude determination does appear to be functional and may prove valuable given a helpful user interface. A constructive graphical user interface (GUI) could make autonomy unnecessary. In an operational environment as the imagery is being processed, it would be useful to show several image panels to the user to show the processing chain. If each step of the analysis process were presented to the user in a logical and understandable way and if the user could change settings and see the results, it could make the process extremely easy. Shown below in Figure 21 is a proposed concept for what the user may see in the GUI next to other image processing tools and modeling tools. The raw image (upper left) gives a reference for the current analysis. The binary edge detection image (upper right) shows where the edges appear. The registered edge display (lower left) shows which edges are in memory. The model attitude display (lower right) shows the calculated attitude as seen from the imager. In addition this GUI could allow the user to not only define edges himself but also assign edges to known "feature definitions" and cut down on computation time. Right now the algorithm goes through each edge and compares it to each "feature definition." Most times if the image is resolved, the user can already determine which edges belong to which "feature definitions." If the user had the ability to assign edges to "feature definitions," then not only would the required computation time decrease but so should the error.

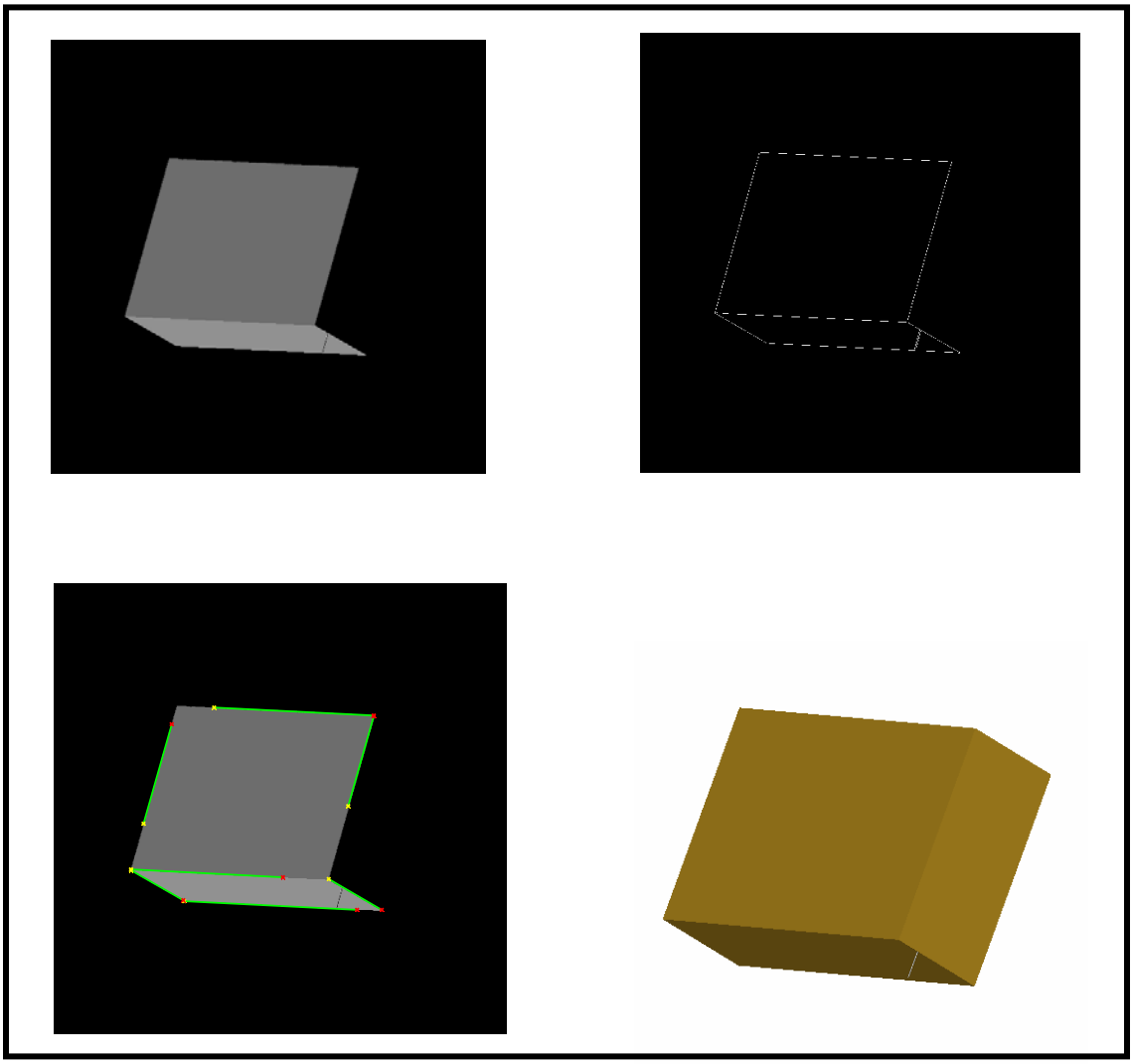

Figure 21: Proposed GUI Basis

Another topic for research would be the use of CAD models in this process. Right now the algorithm reads a completely separate text file to determine "feature definitions." If this

program could read from a “layered” type CAD model, it would be easier to transition the “manufacturing” CAD model into an “operational” CAD model. A specific layer in the model could be designated as the “feature definition” layer, and the program would only need to look at data that is in that specific layer.

In addition to the types and ways that “feature definitions” are written and loaded, the topic of secondary “feature definitions” needs to be investigated. Many satellites have cylindrical, conical, and spherical shapes. Obviously to make this as universally applicable as possible, these need to be considered, and if they can be classified as primary “feature definitions,” then more satellites can be analyzed. From the results in chapter four it appears that the ratio of asymmetric “feature definitions” to symmetric “feature definitions” may affect the accuracy of the results. In chapter four the simple box with the single antenna appeared to produce the same results as just the simple box. Obviously the addition of the single antenna had little effect on the results. The ratio of asymmetric to symmetric “feature definitions” was still only 1/12 or 8.33%. The matches for the symmetric “feature definitions” far out-numbered those involving the asymmetric matches. In certain situations the use of secondary “feature definitions” may help with this.

The final topic that should be investigated is the ability to remove shadowing and physical effects from the equation. The removal of physical effects should be relatively easy because the physical structure of the satellite is assumed to be well-known. A “face definition” may be added to the code, where each flat face on the satellite is defined. In addition the removal of shadowing effects also should not be a huge challenge. Since the program knows the positions of the sensor and the satellite as well as the date and time, the position of the sun would also be known.

Though this thesis did not find a conclusive answer for finding the attitude of a satellite using a comparison between edges detected in imagery and known "feature definitions" on the satellite, it did show that it is feasible. There are very few attitude determination algorithms that are similar. The Vis-STAR software has the same problems with multiple solutions on symmetrical bodies that the algorithm developed for this thesis has shown, but it is able to determine attitude from an image with background clutter. Some adjustments may be required for the algorithm developed from work done on this thesis to be able to automatically process an image with background clutter. Though with a man in the loop defining the satellite edges in the image, background clutter would make no difference.

In all but one of the applicable cases stated in Chapter 2, the target/client/serviceable satellites were required to have equipment or specific modifications to allow for attitude determination by the servicing satellite. The other applicable case, the use of the Vis-STAR software, required a library of reference images. This means that even though no additional equipment was required to be put onto the client satellite, a library of ground images was still needed from defined angles and views. Using edge models as opposed to reference image libraries allows for maximum flexibility with minimum requirement for precise pre-launch information (specifically formatted pictures). These edge models can be created from existing satellite models and not from ground imagery of the satellite as long as the on-orbit configuration is known.

Much like the Orbital Express prototypes it can be assumed that in the future most satellites may be fit with the correct refueling and docking ports with attitude determination aids for servicing satellites. Depending on the mission, incorporating these accessories on every new satellite may be impossible or too expensive (in not only cost but also weight and/or volume),

and obviously retrofitting all current operational satellites would be impossible. If one of these satellite customers requires servicing, they should not automatically be disregarded because they lack servicing equipment, especially if the servicing satellite has the capability. Additionally, the customer may only want an inspection of its spacecraft for anomaly resolution. This would not require docking, but would require attitude determination to ensure that all visual angles of the satellite are covered during the operation. The Vis-STAR approach could not support these requests because there most likely would not be a library of correctly formatted reference images available, and the other options would not be useful because of the lack of required hardware on the client satellite. Because of the cost of satellite launch and operation, a servicing satellite company would not want to exclude any potential customers. The algorithm developed in support of this thesis may answer that question.

With the exception of potential Earth background clutter there is very little difference between using ground-based imagery and using imagery from a servicing satellite. In addition with the potential for higher resolution from a servicing satellite the use of IR imagery is possible. From the ground, IR imagery of satellites may be extremely difficult to resolve. With proximity operations on orbit this can be overcome. As long as the "thermal source edges" are known, an IR edge model can most likely be created. These may even be simpler than the visible edge models if only the radiators or other external thermal sources need to be modeled.

In the examples that were studied it was shown that the algorithm could find the correct attitude, though the algorithm may not be able to place it as the strongest output. Several factors affect what the algorithm outputs, and it appears to be situation-dependent. For situations where the models are very complex or the imagery has many detectable edges one would assume that the correct answer would be easier to find, but in reality the models are actually too complex and

the imagery has too many minor edges (resolved too well). On the other hand if the imagery is resolved just enough to detect the edges, the resolution may be too low to determine the difference between the sizes of two edges, though on the satellite they may relatively be very different. The purpose of this work was to develop a good foundation, but it can be seen that much more work does need to be done.

LIST OF REFERENCES

Armstrong, Dennis (Ed). (May, 2006). *Overview of the DART Mishap Investigation Results*. Retrieved July 20, 2007 from http://www.nasa.gov/mission_pages/dart/main/

Bate, R.R., Mueller, D.D., & White, J.E.. (1971). *Fundamentals of Astrodynamics*. New York: Dover Publications, Inc.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6),* 679-698

Davis, Thom. (2003). *XSS-10 Microsatellite Flight Demonstration Program*. Retrieved July 21, 2007 from http://www.afrlhorizons.com/Briefs/Dec04/VS0312.html

Engineering Test Satellite #VII Homepage. (March, 2003). Retrieved July 21, 2007 from http://robotics.jaxa.jp/project/ets7-HP/index_e.html

Images in Matlab and Image Processing Toolbox. (2007). Retrieved January 12, 2007 from http://www.mathworks.com

Kelso, T. S. (2006). *Orbital Coordinate Systems, Part I*. Retrieved February 22, 2007 from http://www.celestrak.net/columns/v02n01/

Maron, Geoffrey S. (1998). *Real Time Detection Of Anomalous Satellite Behavior From Ground-Based Telescope Images*. Wright-Patterson AFB, OH: DTIC.

Prussing, J.E. & Conway, B.A. (1993). *Orbital Mechanics*. New York: Oxford University Press.

Riker, J.K., Crockett, G.A., & Brunson, R.L. (1992). The time-domain analysis simulation for advanced tracking (TASAT). *SPIE Vol 1697, Acquisition, Tracking, and Pointing VI*, 297-309.

Rotation matrix. (2006, November). Retrieved January 28, 2007 from http://mathworld.wolfram.com/RotationMatrix.html

Schultz, T.J., Stribling, B.E, & Miller, J.J. (1997). Multiframe blind deconvolution with real data: imagery of the Hubble Space Telescope. *Optics Express*, 1(11), 355-362

Space Shuttle Columbia High-Resolution Images. (2004, February). Retrieved March 17, 2007 from http://www.afrlhorizons.com/Briefs/Feb04/DE0303.html

Weismuller, T. & Leinz, M. (2006) GN&C Technology Demonstrated by the Orbital Express Autonomous Rendezvous and Capture Sensor System. *29th Annual American Astronautical Society Guidance and Control Conference*.

Wertz, J. R. & Larson, W. J. (Eds.) (1999). *Space Mission Analysis and Design* (3rd ed.). El Segundo, CA: Microcosm Press.

Wood, G. E. (1996). *Estimation of Satellite Orientation from Space Surveillance Imagery Measured with an Adaptive Optics Telescope*. Wright-Patterson AFB, OH: DTIC.

XSS-11 Microsatellite Fact Sheet. (December, 2005). Retrieved July 21, 2007 from http://www.kirtland.af.mil/shared/media/document/AFD-070404-108.pdf