



DEVELOPMENT OF A REMOTELY OPERATED AUTONOMOUS SATELLITE

TRACKING SYSTEM

THESIS

Michael E. Graff, Captain, USAF

AFIT/GSS/ENY/10-M03

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT/GSS/ENY/10-M03

DEVELOPMENT OF A REMOTELY OPERATED AUTONOMOUS SATELLITE
TRACKING SYSTEM

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science (Space Systems)

Michael E. Graff, BS

Captain, USAF

March 2010

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GSS/ENY/10-M03

DEVELOPMENT OF A REMOTELY OPERATED AUTONOMOUS SATELLITE
TRACKING SYSTEM

Michael E. Graff, BS
Captain, USAF

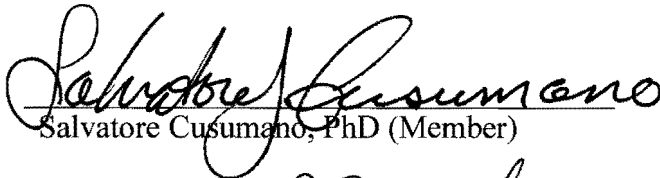
Approved:



Richard Cobb, PhD (Chairman)

15 MAR 2010

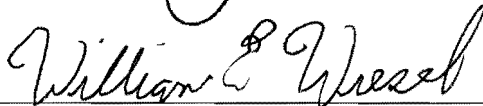
Date



Salvatore Cusumano, PhD (Member)

15 Mar 2010

Date



William Wiesel, PhD (Member)

15 Mar 2010

Date

Abstract

AFIT is currently developing a capability to remotely and autonomously track LEO satellites using commercial telescopes. Currently, the system is capable of open-loop tracking based on Two-Line Element sets (TLEs) downloaded from NORAD's space object catalog. The ability to actively track using a closed-loop control system would allow tracking of satellites which deviated from the published TLEs along with providing some information about the object's new orbital elements. To accomplish closed-loop tracking, the object is imaged by a digital camera connected to a wide field-of-view (WFOV) spotting scope. Software was developed to provide azimuth and elevation inputs in order to center the object within the WFOV. Pixel centroid location along with telescope azimuth and elevation commands can be recorded for use in estimating updated orbital elements. This thesis documents the current efforts towards achieving a remotely operated autonomous tracking optical system. Future application could include networking to other geographically-separated telescopes to allow simultaneous observation of the same space objects to accurately document orbital maneuvers.

Acknowledgements

I would like to express my sincere appreciation to Capt Matthew Schmunk for his continued enthusiasm and unflagging interest in a project he began over two years ago.

I would also like to express my sincere appreciation to Mr. Christopher Zickefoose for his superb support as a laboratory technician. His prompt assistance and willingness to learn more about my research was indispensable.

Michael E. Graff

Table of Contents

	Page
Abstract.....	iv
Acknowledgements.....	v
List of Figures.....	xii
I. Introduction	1
Background.....	1
Problem Statement.....	1
Research Objectives/Questions/Hypotheses.....	2
Research Focus	2
Investigative Questions.....	3
Closed-Loop Control	3
Remote Operation.....	3
Methodology.....	3
Closed-Loop Control	3
Remote Operation.....	4
Assumptions/Limitations	4
Implications.....	5
Preview	5
II. Literature Review.....	7

Chapter Overview	7
Description	7
Online Research	7
Previous Research	11
Relevant Research	12
Image Processing and Motion Tracking	12
Summary	12
III. Background & Methodology	13
Chapter Overview	13
Background Description	13
Closed-Loop Control Overview	13
Remote Control Overview	19
Methodology	21
Closed-Loop Tracking	21
Determining Angular Conversions	22
Collimating the Main Optics to the Guide Scope Optics	24
Laboratory Simulations	26
Simulating Satellites for Image Processing Techniques	26
Hardware Configuration	27

Test Subjects	28
Refining Targeting Logic.....	33
Converting Test Software for Real-time Use.....	38
Remote Control.....	39
Automated Shelters.....	39
Data Connectivity	43
Summary.....	47
IV. Analysis and Results.....	48
Chapter Overview	48
Closed-Loop Control	48
Results of Simulation Scenarios	48
Figure of Merit.....	48
testvid1.avi	49
testvid2.avi	50
testvid3.avi	52
testvid4.avi	55
testvid5.avi	57
testvid6.avi	59
testvid7.avi	61

testvid8.avi	62
Real-time Software Conversion	65
Remote Control	66
Automated Shelter	66
Shelter Modifications	70
Shelter Malfunction	72
Noted Observatory Location Issues	74
Data Connectivity	75
Concept 2: Distributed Control	77
Equipment Upgrades/Modifications	80
Near Infrared Camera	80
Frame grabber	81
TV to PC Converter	82
Portable LCD TV	82
Internet Router	83
Autostar Controller Extension Cord	84
Carrying Cases	84
Investigative Questions Answered	86
Closed-Loop Tracking	86

Remote Operation	86
Summary	87
Closed-Loop Tracking	87
Remote Operation	87
V. Conclusions and Recommendations	88
Chapter Overview	88
Conclusions of Research.....	88
Closed-Loop Tracking	88
Remote Operation	88
Significance of Research.....	89
Recommendations for Action	89
Recommendations for Future Research	90
Networked Telescopes	90
Mobility.....	90
Improvements in Accuracy	90
Summary	91
Appendix I: Formula for Mobility	92
Formula for mobility.....	92
Power consideration.....	92

Portability.....	95
Internet access.....	96
Miscellaneous consideration.....	97
Appendix II: Possible improvement in closed-loop tracking	99
Appendix III: Test MATLAB Code	102
Appendix IV: Real-Time MATLAB Code.....	106
begin_track.m.....	106
frame_centroid_cb.m	107
frame_centroid.m.....	109
closest_centroid.m	109
velocity_approximation.m	110
update_history.m.....	111
end_track.m.....	111
collimation.m	111
seeker_trackgui_v2.m excerpt	112
Appendix V: Instructions for Use	114
Bibliography	117
Vita.....	120

List of Figures

	Page
Figure 1: ISS photographed by the Toyama Astronomical Observatory (<i>photo credit: Toyama Astronomical Observatory 2005</i>).....	8
Figure 2: 1 m Reflector Telescope (<i>photo credit: Toyama Astronomical Observatory 2005</i>).....	8
Figure 3: ISS as captured by www.iss-tracking.de (<i>photo credit: ISS-Tracking 2007</i>) ...	9
Figure 4: Tracking Installation Page (<i>photo credit: ISS-Tracking 2007</i>)	10
Figure 5: Open-loop Control.....	14
Figure 6: Closed-loop Control	15
Figure 7: Field of View (WFOV Guide Scope).....	16
Figure 8: Binary Image Field of View	17
Figure 9: Networked Telescope Depiction (<i>AGI Satellite Tool Kit</i>).....	21
Figure 10: Small Angle Approximation	23
Figure 11: Reference Point in Main Optics Field of View	25
Figure 12: Reference Point in Wide Field of View	26
Figure 13: Laboratory Simulation Hardware Configuration	28
Figure 14: testvid1.avi	29
Figure 15: testvid2.avi	29
Figure 16: testvid3.avi	30
Figure 17: testvid4.avi	31
Figure 18: testvid5.avi	31
Figure 19: testvid6.avi	32
Figure 20: testvid7.avi	32
Figure 21: testvid8.avi	33
Figure 22: Pixel Velocity Approximation.....	35
Figure 23: Multiple Centroids.....	36
Figure 24: Flashing Target with Solid Background Object	38
Figure 25: Dome Enclosure (<i>photo credit: Candomes Observatories Ltd. 2008</i>).....	40
Figure 26: Clam-shell Enclosure (<i>photo credit: AstroHaven Enterprises 2008</i>)	41
Figure 27: Roll-off Roof Enclosure (<i>Pier-Tech, Inc 2007</i>)	42

Figure 28: Centralized Control Concept	44
Figure 29: Observation Station Configuration for Centralized Control	45
Figure 30: Distributed Control Concept	46
Figure 31: Observation Station Configuration for Distributed Control.....	47
Figure 32: testvid1.avi Calculated Centroid Path	49
Figure 33: testvid1.avi Centroid Offset by Frame Number	50
Figure 34: testvid1.avi Absolute Change in Position	50
Figure 35: testvid2.avi Calculated Centroid Path	51
Figure 36: testvid2.avi Centroid Offset by Frame Number	52
Figure 37: testvid2.avi Absolute Change in Position	52
Figure 38: testvid3.avi Calculated Centroid Path	53
Figure 39: testvid3.avi Centroid Offset by Frame Number	54
Figure 40: testvid3.avi Absolute Change in Position	54
Figure 41: testvid4.avi Calculated Centroid Path	55
Figure 42: testvid4.avi Centroid Offset by Frame Number	56
Figure 43: testvid4.avi Absolute Change in Position	56
Figure 44: testvid5.avi Calculated Centroid Path	57
Figure 45: testvid5.avi Centroid Offset by Frame Number	58
Figure 46: testvid5.avi Absolute Change in Position	58
Figure 47: testvid6.avi Calculated Centroid Path	59
Figure 48: testvid6.avi Centroid Offset by Frame Number	60
Figure 49: testvid6.avi Absolute Change in Position	60
Figure 50: testvid7.avi Calculated Centroid Path	61
Figure 51: testvid7.avi Centroid Offset by Frame Number	62
Figure 52: testvid7.avi Absolute Change in Position	62
Figure 53: testvid8.avi Calculated Centroid Path	63
Figure 54: testvid8.avi Centroid Offset by Frame Number	64
Figure 55: testvid8.avi Absolute Change in Position	64
Figure 56: realtime2.csv Absolute Change in Position.....	65
Figure 57: realtime2.csv Centroid Computation Time by Frame	66
Figure 58: AFIT Roof Top Observatory	67

Figure 59: Observatory with Roof Open and Pier Actuated.....	68
Figure 60: Observatory Internal Safety Switch (<i>photo credit: Pier-Tech, Inc. 2007</i>)....	68
Figure 61: Pier with Vibration Isolation System	69
Figure 62: Telescope Mated to Pier	70
Figure 63: Roof Motor with Worm Gear (<i>photo credit: Pier-Tech, Inc 2007</i>)	73
Figure 64: Rack and Pinion (<i>photo credit: Pier-Tech, Inc 2007</i>)	73
Figure 65: Telescope Network Control Test Configuration	76
Figure 66: NIR Camera with Wireless Controller	80
Figure 67: Epiphan VGA2USB LR Frame grabber.....	81
Figure 68: Impact Acoustics TV to PC Converter.....	82
Figure 69: Sylvania 7" LCD TV	83
Figure 70: D-Link 4-port Internet Router	83
Figure 71: Pelican 1520 Cases with Equipment Deployed.....	85
Figure 72: Pelican 1520 Case with Equipment Stowed.....	85
Figure 73: Commercial Gas Powered Generator (<i>photo credit: American Honda Motor Co., Inc. 2009</i>)	93
Figure 74: Alternative energy solution (<i>photo credit: Solar Stik 2009</i>).....	95
Figure 75: Iridium Satellite Data Modem (<i>photo credit: Outfitter Satellite Phones 2009</i>)	97
Figure 76: Stars in WFOV	100
Figure 77: Stars and Intended Target in WFOV	100
Figure 78: Stars subtracted from WFOV	101

DEVELOPMENT OF A REMOTELY OPERATED AUTONOMOUS SATELLITE TRACKING SYSTEM

I. Introduction

Background

AFIT's Satellite Tracking Telescope is a proof-of-concept research project. It demonstrates the ability of Commercial-Off-The-Shelf (COTS) optical observation equipment to track and image Low Earth Orbiting (LEO) satellites. Using radar data in the form of Two-line Element Sets (TLEs) generated by NORAD that is published to the worldwide web (WWW), satellite orbits are filtered for observability then converted to angles and angular velocity commands for use by the telescope.

The current configuration uses a Meade 10" LX200GPS for the main optics and motorized mount. In addition, a Wide Field of View (WFOV) is provided by an Orion Refractor telescope. A small USB webcam provides the imaging for the WFOV while a near-infrared (NIR) camera provides the main optics imaging.

Problem Statement

Radar information from NORAD is published every 12 hours. During this time a satellite may have maneuvered. Since the current software provides only open-loop control (with operator input) a satellite may have shifted from an expected position, decreasing the likelihood of accurately imaging a satellite. Telescope alignment errors will also result in decreased accuracy.

Research Objectives/Questions/Hypotheses

Objective 1: Use closed-loop control methods to increase the telescope's tracking accuracy and imaging capabilities.

Objective 2: Formulate requirements and outline solutions for remote control and accessibility to the tracking system.

Hypothesis 1: Using small angle approximations and optical feedback from the WFOV camera, corrections can be made to the telescope's open-loop control system to re-center the object of interest.

Hypothesis 2: A remotely operated tracking system can be developed by selecting an appropriate automated shelter and developing data connectivity models to contribute to a surveillance network.

Research Focus

The research focus for the closed-loop control portion of this thesis revolves around frame-by-frame image processing from the WFOV camera, as well as logic routines to reject erroneous inputs (background clutter, such as stars or other satellites) or to estimate a satellite's position if its brightness dims below the image processing threshold.

The research focus for the remote control portion of this thesis is conceptual and theoretical in nature, but demonstrates some practical aspects for remote control via a data network.

Investigative Questions

Closed-Loop Control

1. What sources of feedback are available for a closed-loop control system?
2. Once feedback is determined, how does it relate to the commands sent to the telescope and ultimately the angles generated?

Remote Operation

1. What hardware is required to develop a remotely operated tracking system?
2. What methods of data connectivity can be used to enable autonomy in a remotely operated tracking system?

Methodology

Closed-Loop Control

Optical feedback will be used to generate a closed-loop tracking system for the telescope. The video feed from the USB webcam is employed to collect images from the WFOV. Each image frame is used to estimate the centroid of each pixel cluster in the frame. The centroid is used to determine the angular pointing error in the telescope's position. Once the correct centroid is identified the telescope's pointing error can be determined; if no centroid is identified, an estimate of the error can be determined.

A series of eight digital videos were recorded to simulate various situations that can be encountered during the course of a satellite's observation. These videos were used to generate image processing routines and logic routines to be employed real-time in conjunction with the existing tracking software (written by Capt Matt Schmunk and Chris Carlton) which has been slightly modified to accommodate the corrections.

Remote Operation

Commercially available automated shelters were researched in order to select a suitable enclosure for a remotely operated tracking system. Expense, simplicity, and interference with the telescope's field of view (FOV) were considered in the selection process.

Data connectivity research was mostly conceptual in nature, but network control models were tested with commercially available hardware.

Assumptions/Limitations

The closed-loop tracking system assumes that angles are small enough to approximate a linear value. That is, the position of an object's image on the focal plane of the webcam can be directly related to an angular offset from the main optics FOV within the WFOV.

For the purpose of developing initial closed-loop feedback, it is also assumed that the object of interest is within the WFOV and free of background clutter (mainly stars but other celestial or airborne objects could pose a problem) when the closed-loop routine is started. Later research can be focused on automated initiation techniques and target identification in clutter.

The closed-loop tracking system is limited by object brightness as well as its position. If the object is not within the WFOV or an operator cannot place it within the WFOV, then it is highly unlikely that the closed-loop tracking routine will generate usable feedback. This situation may even further degrade the ability of the telescope to track an object during the course of its orbit.

The entire optical system is also weather and time dependent. Clear skies are necessary to view a satellite. Daytime viewing is not possible either due to excessive scattering of visible light. A blue sky background is significantly brighter than the light reflected from satellites.

Finally, the existing open-loop control system only generates commands every 0.5 seconds, which is the maximum speed at which the telescope will respond to commands. Therefore, corrections from a closed-loop routine would only be able to provide error corrections at a maximum rate of 0.5 seconds.

Implications

If an object of interest can be accurately tracked with a closed-loop control system, resulting angles recorded from the telescope can be used to estimate the position of the satellite, as well as any maneuvers that have been made. Accurate tracking with optical feedback also implies that imaging gained from the main optics of the telescope can provide further information on a satellite's structure, orientation, and stability (i.e. if a satellite is tumbling).

Development of a remotely controlled, autonomous tracking station can contribute to the overall development of a network of telescope-based observation stations. This network can be used to track high-interest satellites, augmenting information gained from active radar space surveillance systems.

Preview

Chapter I presents an introduction to the satellite tracking telescope system, as well as the focus of research for developing a closed-loop control system. Research

focus is also introduced in the area of remote operation for the satellite tracking system. Chapter II provides a review of available literature in the field of satellite tracking and image processing. Next, Chapter III provides a more in-depth look at the control concepts used in the satellite tracking system, as well as motivation to develop a remote operation concept. Chapter III also outlines the methodology used to develop targeting logic routines and image processing techniques and explores requirements and possible hardware configurations for remote operation. Chapter IV details test results that were demonstrated in a laboratory environment; final shelter selection to function as a rooftop observatory at AFIT; and data connectivity trials. Chapter V summarizes the research and test results and recommends action and future areas of research to continue development.

II. Literature Review

Chapter Overview

The purpose of this chapter is to outline available research and capabilities in the field of satellite tracking, and what methods and equipment are being employed to accomplish this. This literature is not exhaustive in nature since the goal of this research was to develop a proof-of-concept. While many different technical resources regarding motion tracking and image processing are available, specific details of a tracking algorithm can be developed in future research once the concept is demonstrated.

Description

Online Research

In order to get an idea of what is possible with different types of observation equipment, research was pursued on publically available webpages. Satellite tracking seems to fall into two categories: professional and amateur.

Professional satellite tracking techniques employ very expensive equipment at well-funded observatories. A good example of this type of this type of observatory is the Toyama Astronomical Observatory in Japan. The observatory's website has readily available images that show impressive amounts of detail in space objects, such as the International Space Station (ISS). An idea of this observatory's imaging capabilities can be interpreted from Figure 1 (Toyama Astronomical Observatory 2005). In this image, the solar panels are most apparent, and individual modules can be resolved.



Figure 1: ISS photographed by the Toyama Astronomical Observatory (*photo credit: Toyama Astronomical Observatory 2005*)

This level of detail immediately instills curiosity into the type of equipment used to capture this image. More research on the website revealed that the telescope used was a 1 m reflector, manufactured by Contraves Brasher Systems, as depicted in Figure 2 (Toyama Astronomical Observatory 2005).

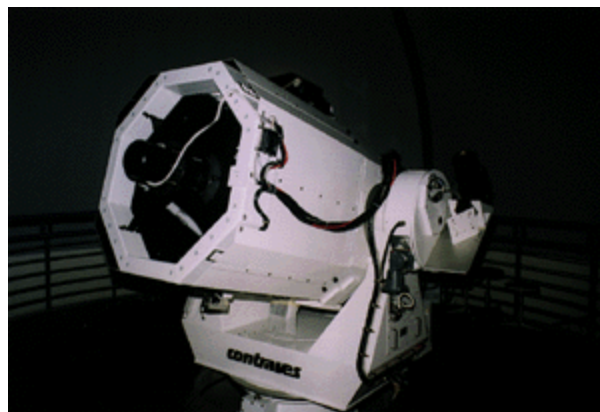


Figure 2: 1 m Reflector Telescope (*photo credit: Toyama Astronomical Observatory 2005*)

Another example of a well-funded observatory would be the Air Force's own Starfire Optical Range (SOR). According to the SOR factsheet listed on the Kirtland AFB website:

The SOR operates one of the world's premier adaptive-optics telescopes capable of tracking low-earth orbiting satellites. The telescope has a 3.5-meter (11.5 feet) diameter primary mirror and is protected by a retracting cylindrical enclosure that allows the telescope to operate in the open air. Using adaptive optics, the telescope distinguishes basketball-sized objects at a distance of 1,000 miles into space. (Air Force Research Laboratory Directed Energy Directorate 2009)

While an SOR sample image of an orbiting object could not be readily attained, it is assumed that a system of that sophistication, magnitude, and expense could generate an even higher level of detail over the Toyama Observatory's capabilities.

So as to focus research towards the capabilities currently maintained at AFIT, amateur satellite tracking websites were explored. Many of these websites focused on the basics of satellite tracking, and were too elementary in content. A Danish website that posted another highly detailed image of the ISS, as shown in Figure 3 (ISS-Tracking 2007).



Figure 3: ISS as captured by www.iss-tracking.de (photo credit: ISS-Tracking 2007)

As mentioned, the website is based in Denmark but more information can be obtained about the type of equipment that was used to obtain this image. Clicking on “tracking installation” appears to direct the viewer to a portion of the site that details the equipment used for viewing satellites. A screen capture of the tracking installation page can be seen in Figure 4 (ISS-Tracking 2007). Close inspection of the telescope in the

picture reveals that is an older model Meade LX200 Telescope which is readily available for purchase. From the size of the motorized mount, this appears to be a 14” or 16” telescope.

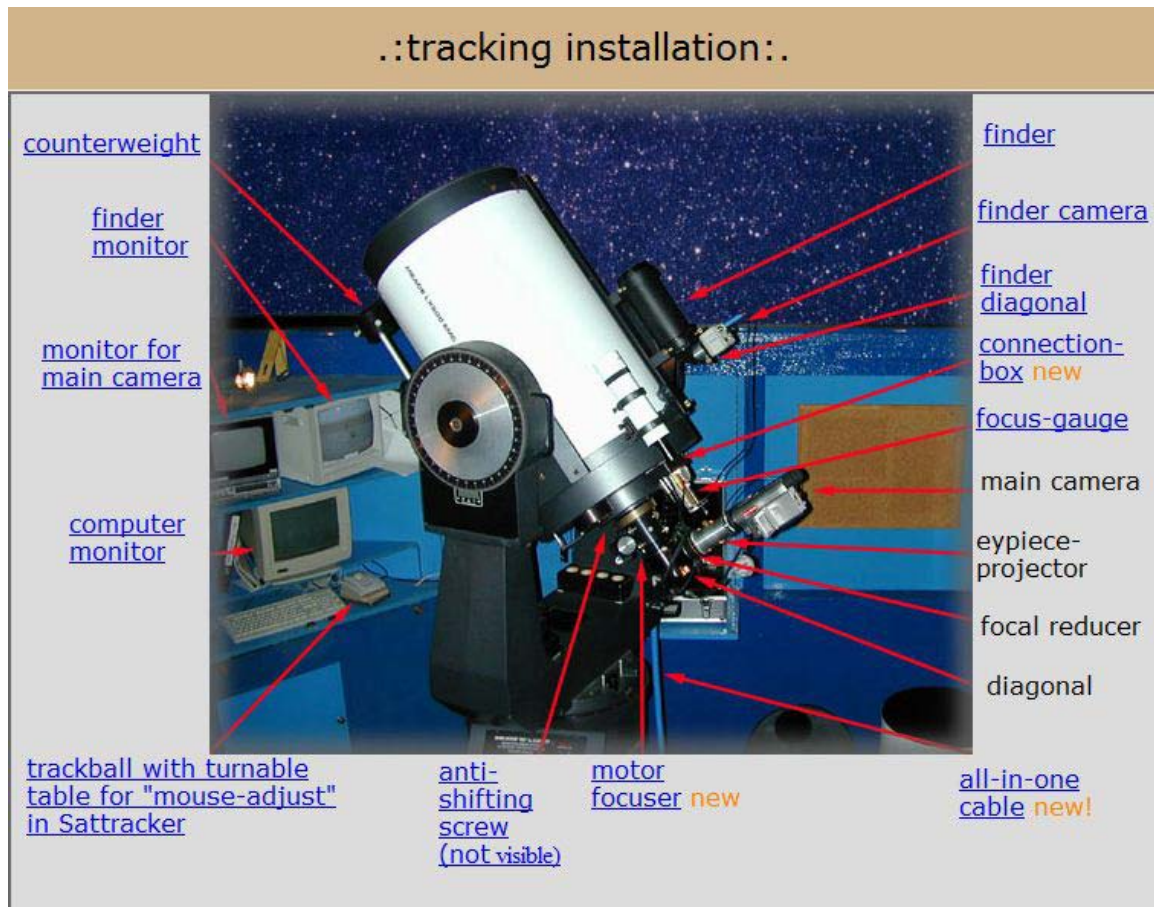


Figure 4: Tracking Installation Page (photo credit: ISS-Tracking 2007)

The rest of the information contained on the page fails to detail the type of camera or recording equipment used for night sky imaging. However, with the configuration of equipment shown, it seems that a high level of detail can be obtained with Commercial Off the Shelf (COTS) components.

Previous Research

In March 2008, Matthew Schmunk (AFIT graduate student) published his thesis *Initial Determination of Low Earth Orbits Using Commercial Telescopes*. According to this document's Problem Statement, primary motivations for his research include:

Space Situational Awareness (SSA): Commercial systems are inexpensive, mobile and easily supported: all factors that compensate for limitations in capability. There are always new opportunities to use them for surveillance and debris monitoring.

State-of-the-Art Survey: This study offers a baseline explanation of methods used by semi-professional satellite observers today. Hopefully, it will serve as a reference for future researchers and motivate them to pursue additional work in this field.

Research Testbed: By its conclusion, this project integrated with the hardware and software required to operate a basic optical satellite tracking program. Now, students may use it to support work in sensors, image processing, orbit determination, and many other fields. It also allows AFIT students to gain hands-on experience with classroom concepts. (Schmunk 2008, 1)

This research pursued what is known as "Angles Only Orbit Determination" which is defined as follows:

Orbit solutions that do not use target range data collectively called angles only methods (Schmunk 2008, 8)

This means that an active ranging system, such as radar, is not directly used to determine a satellite's orbit. Instead a commercial Meade 10" LX200GPS telescope mount was employed to visually intercept satellites in their orbits and determine orbital characteristics. (Schmunk 2008, 59-63) Various MATLAB scripts were written to control the telescope via serial command, all operating in unison to generate accurate angular positions so as to accomplish orbit determination Low Earth Orbit (LEO)

satellites. These MATLAB scripts formed the foundation for future research efforts at AFIT.

Relevant Research

Image Processing and Motion Tracking

Using an internet search for “target tracking using MATLAB”, a website was discovered that outlined some basic points to develop a motion tracking routine. This website, developed Siamak Faridani, detailed four main “tenets” for targeting a moving object in a video feed: motion estimation, detection, association, and calibration/extraction (Orofino and Faridani 2008).

While helpful, the above site only outlined the procedure to perform motion tracking, and did not provide much in the way of actual image processing pointers. Searching the MATLAB 2007a help files posted on www.mathworks.com revealed two demonstrations of particular use: “Determining Pendulum Length” (The MathWorks, Inc. 2010) and “Laser Tracking.” (The MathWorks, Inc. 2010). These two examples provided information for image processing and centroid calculation.

Summary

This chapter contained a short literature review that demonstrated some existing capabilities in the areas of satellite observation and image processing for motion tracking. While this literature was not exhaustive in nature it allowed research to be conducted towards a proof-of concept. The next chapter gives a background into current open-loop control, a methodology to generate optical feedback, and remote operation concepts.

III. Background & Methodology

Chapter Overview

The purpose of this chapter is to describe the equipment configuration and the feedback available to the closed-loop tracking system. It also details concepts for remote operation and requirements to build a remote facility. A methodology is also presented to determine angular data in laboratory simulations in centroid tracking.

Background Description

Closed-Loop Control Overview

The open-loop control system that was written for the satellite tracking telescope is a collection of sophisticated MATLAB scripts. In order to function properly, radar tracking data is obtained from NORAD. NORAD publishes this information twice daily, and is available from organizations like Space Track (Space-Track 2004). Typically a satellite's orbit is described through six orbital elements that are then organized into a Two Line Element Set, or TLE. Space Track publishes these TLEs either as a Two Line Element Set or Three Line Element Set; the open-loop software uses the Three Line Element Set, with the additional line containing the name of the satellite as listed in the Satellite Catalog.

Once downloaded, the TLE file must be uncompressed using a program like 7zip to extract the text file (Pavlov 2009). The extracted text file must be placed in the same folder as the control software in order to for it to be opened. A pre-calculation script in MATLAB will then take that data and compute a list of available satellites to be viewed. When the pre-calculation is complete, another script that contains the Graphical User

Interface (GUI) is run. The GUI script allows the user to select satellites for viewing, a chart of visible stars, and a preview window as seen by the USB webcam.

While running, the operator can select satellites which are organized according to their brightness and their time in view (satellites may not be viewable for a variety of reasons, such as if they have not risen yet, have passed into the terminator or are too far away). Prior to the ability to incorporate closed-loop tracking, the operator could make some adjustments through the GUI to attempt to center an object if it is drifting out of view. Figure 5 shows a basic configuration of the open-loop control method.

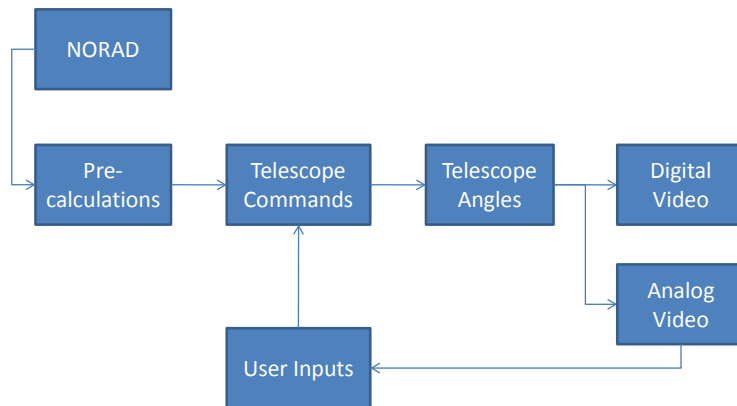


Figure 5: Open-loop Control

Normally an operator could only make imprecise or infrequent corrections to the viewing angles sent to the telescope. In order to achieve a greater degree of accuracy, the

digital output from the USB webcam is processed, filtered, and made available to the GUI software to be used as a correction to the next command update sent to the telescope. Figure 6 shows a closed-loop configuration for the telescope system. It is in these command updates that corrective information gained from angular offset data may be applied.

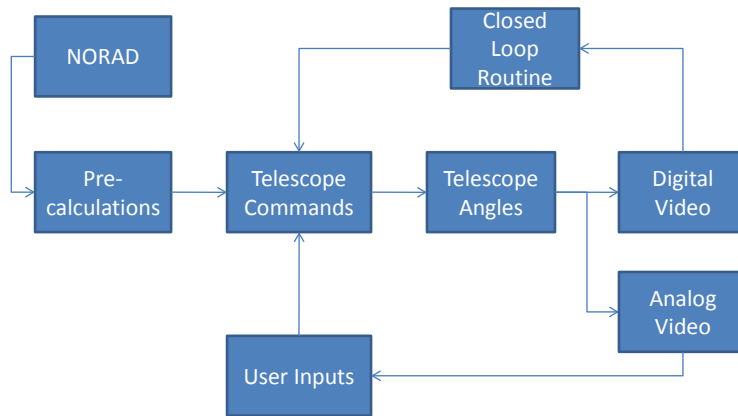


Figure 6: Closed-loop Control

The USB webcam provides an RGB 640x480 image from each frame it captures (Philips Consumer Electronics 2003). As light enters the webcam, it is focused on a focal plane array within the camera. Each element of the focal plane array corresponds to an image pixel's address. Horizontal pixel addresses equate to azimuthal position, while vertical pixel addresses equate to elevation position (a function of how the webcam is

mounted to the WFOV and to the mount). If the guide scope is perfectly collimated to the main optics field of view, then the center of the webcam's images would represent the center of the FOV of the main optics. This location is represented by " x^*,y^* " in Figure 7, is a reference point used in calculation of angular errors. The FOV of the main optics is much narrower than the FOV of the guide scope so it is desirable to align the object to the main optics FOV.

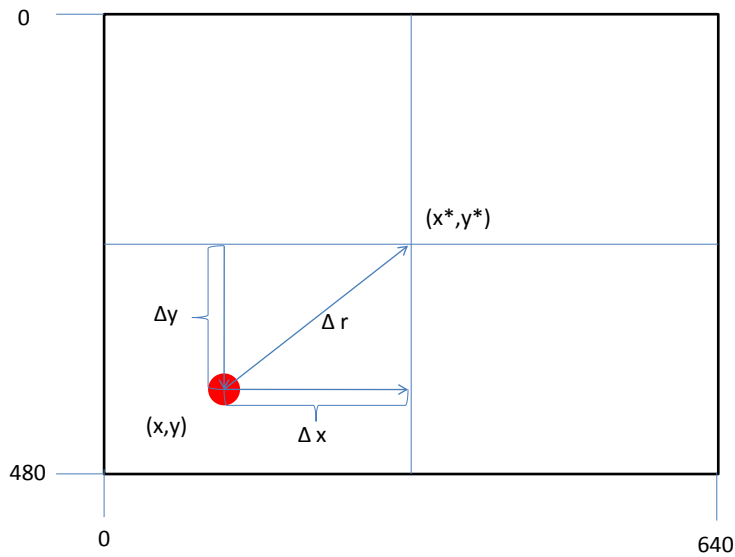


Figure 7: Field of View (WFOV Guide Scope)

The red dot in the lower left portion of the field represents an object of interest within the guide scope's FOV. In order to determine the position of the object in relation to the center of the FOV, the image must be processed. As frames are obtained from the webcam, they are immediately converted to black-and-white images (also known as

binary images). Grey scale images were not used because existing centroid calculation functions within MATLAB used only binary images.

Each frame can be visualized as an array of values, or a matrix. Pixel addresses are represented by row and column designations in a matrix, with each element containing color and brightness information. When converting to binary, each element is analyzed for brightness. Those elements that do not meet a specified threshold (between zero and 1) are converted to zeros, while those that do meet the threshold are converted to ones. If re-displayed, this new binary image would show only black and white regions, black representing zeros, white representing ones. All other information from the original frame has been filtered allowing the binary image to be further processed. Figure 8 depicts the FOV in Figure 7 after it has been processed into a binary image.

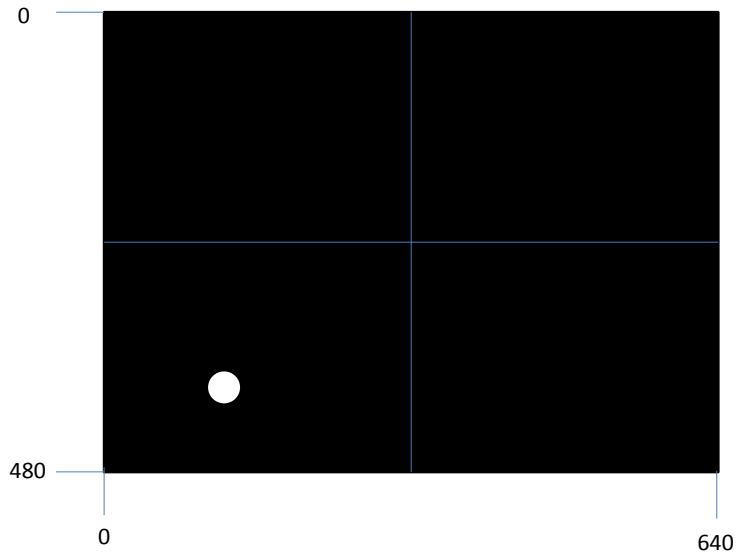


Figure 8: Binary Image Field of View

Now that the image is represented by ones and zeros only, the matrix can be analyzed to determine the object's offset from the center of the FOV. Using a routine that automatically computes the "center of mass" or "centroid" of a cluster of pixels in an image, a pixel address is then determined for the center of the object. According to *Mathematical Methods in the Physical Sciences*, a centroid is defined as follows:

The centroid of a body is the center of mass when we assume constant density (Boas 2006, 251).

In a two-dimensional example, a center of mass is normally calculated using the following equations (Boas 2006, 251):

$$x_{cm} = (\int x dm)/M \quad (1)$$

$$y_{cm} = (\int y dm)/M \quad (2)$$

where dm is an element of mass, x_{cm} and y_{cm} are the coordinate locations of the elements, and the integrals are over the whole body of mass M .

In order to calculate the centroid position, MATLAB's "bwlabel" function is used. This function determines "blobs" of neighboring pixels and labels them according to cluster. Once labeled, MATLAB's "regionprops.Centroid" is used. This function generates an $m \times 2$ array containing the pixel addresses of the centroids for each of m pixel blobs detected. Since each pixel in the binary image is assigned a value of either one or zero, any pixel blob is considered to have a uniform "density" in the calculation (see definition of a centroid, above). Subtracting the centerpoint's row and column values from those of a singular centroid yields an offset value, assuming the centerpoint of the WFOV coincides with the main optics FOV.

The offset value, saved in a 1x2 matrix, is used by the GUI to make corrections. During one control loop iteration, the GUI calculates the next incremental position of the satellite relative to the telescope's current position. Since the telescope's angular position is available to the script, the script generates commands based on the current position of the telescope and necessary position to maintain tracking. By adding the small angular corrections from the feedback loop, the telescope can theoretically move the object into the main optics field of view.

However, the offset values still need to be converted from pixel values to angular values. The conversion factor can be determined experimentally but also depends on the hardware configuration (i.e., type of guide scope, focal reducer, etc). Once converted, it is a simple matter of arithmetic in the script to provide correction to the telescope's next position.

Remote Control Overview

Prior to research into an autonomous tracking system, only one telescope was being considered. An optical tracking system has its advantages and disadvantages alongside an active radar space surveillance system. The advantages of an optical system include precision angular data, attainably higher resolution imaging, passive operation, and low cost. The disadvantages include weather limitations, sun angles, the lack of range data, and the ability to track a single object at a time. These limitations seem rather severe when considering only one telescope.

Many of these limitations can be overcome by operating a network of telescopes. Weather limitations can be mitigated through the geographic separation of observation

sites, as well as careful selection of higher-altitude, low humidity areas. This will allow a network to “peek around” cloud cover, rather than through it. Sun angle limitations can vary based on object velocity and time of day. Passive observation requires illumination from sunlight in order to be detected while an observation site passes into the terminator. Therefore, it would be advantageous for multiple observation sites to be longitudinally dispersed over a wide/global area. This provides multiple opportunities for observation throughout the satellite’s orbital period. Finally, more telescopes mean more objects could be tracked. Simultaneous tracking of a single object by two geographically separated could provide range data through the correlation of azimuth/elevation angles in a common reference frame. It should be noted that the current radar-based space surveillance network employed by the United States is capable of tracking thousands of objects with far fewer observation sites that would be required by a similarly-capable optical observation network. It is safe to say that an optical network would be applied to satellites of particular interest rather than the tracking of space debris. A networked telescope system is shown in Figure 9.

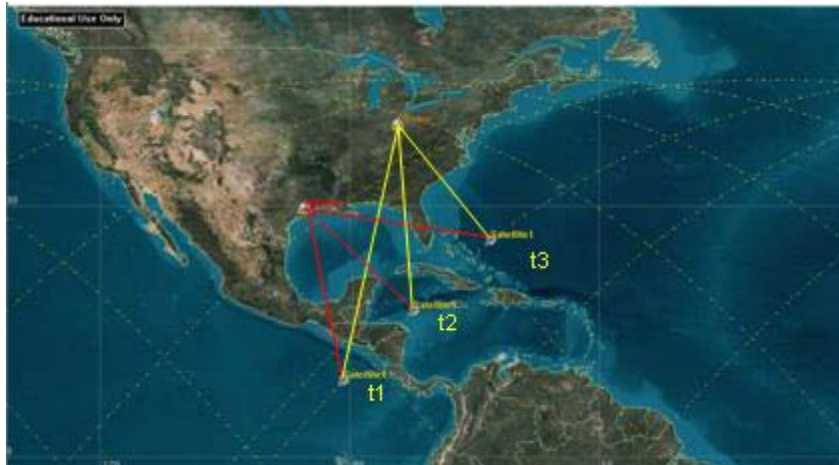


Figure 9: Networked Telescope Depiction (*AGI Satellite Tool Kit*)

A networked system of unmanned stations required research into remote operation. Two areas of research were explored: protective automated shelters and connectivity methods to a central control station.

Methodology

This section describes the methodology in developing a closed-loop tracking system, simulating video inputs to the tracking system, and also details possible methods and equipment to enable remote operation of the tracking system.

Closed-Loop Tracking

In order to track a satellite in an open loop fashion, the script “seeker_trackgui_v2.m” calculates the next angular position of a satellite during each one of its 0.5 second iterations. This calculated position is then compared to the current position of the telescope and angular velocities are determined. Using the WFOV video input, additional corrective angular data can be determined from pixel positions.

Determining Angular Conversions

In order to convert pixel addresses to angles, the relationship between angles and pixels had to be determined. The GUI, once initialized, reports a telescope's raw angular position in degrees. The telescope was then manually pointed at a distant object that had a small, distinct, and recognizable feature as a reference location. The scope's azimuth and elevation were recorded in this position, and then a snapshot of the webcam's FOV was recorded. Displaying the image in MATLAB and then using "ginput" allows the user to click on the reference feature. This command then returns a 1x2 matrix containing the "X-Y" location of the feature.

The next step of the process was to adjust the telescope's position again, while still keeping the reference feature in the webcam's FOV. The new azimuth and elevation were recorded, and another snapshot was taken. The image was displayed again and ginput returned another 1x2 matrix containing the "X-Y" address of the reference object. Subtracting the first matrix from the second matrix yielded another 1x2 matrix containing the number of pixels that the reference point travelled when the azimuth and elevation were adjusted. Figure 10 demonstrates the relationship between pixel locations and angular measurements.

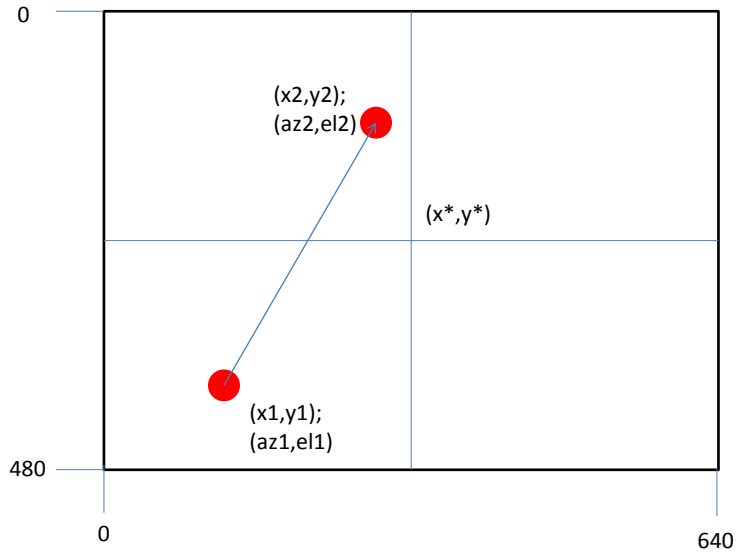


Figure 10: Small Angle Approximation

Next, the first azimuth/elevation recording was subtracted from the second azimuth/elevation recording. This yielded an angular difference that could be equated to the difference in pixels computed above. With the .5x focal reducer in place on the webcam attached to the guide scope, the angular conversion resulted in approximately 0.0015 /pixel. This value is not used until a pixel-based error offset is determined in a frame. This calculation is shown in Equations (3) and (4) below:

$$\frac{(az2 - az1)}{(x2 - x1)} = \frac{az \text{ deg}}{\text{pixel}} \quad (3)$$

$$\frac{(el2 - el1)}{(y2 - y1)} = \frac{el \text{ deg}}{\text{pixel}} \quad (4)$$

Collimating the Main Optics to the Guide Scope Optics

The guide scope provided a WFOV but it had to be aligned physically with the main optics. This was done through an “eyeball” method. The main optics had to be focused on a distant, known object such as a star. Once focused, the WFOV from the USB webcam on the guide scope was checked to see how far off center the object was. The collimation screws were then adjusted to move the object to the center of the WFOV.

However, this was not easily accomplished. The act of adjusting the collimation screws required the operator to be away from the preview window for the USB webcam, and it also caused significant shaking of the main optics, making it extremely difficult to get the reference object focused. To solve this issue, a “shake free” collimation script was written to determine the location of the main optics narrow field of view (NFOV) within the WFOV. Figure 11 and Figure 12 show the relationship between the NFOV and WFOV. This method provided an accurate measure of the collimation error, necessary for tracking.

The telescope was first aligned using its automatic alignment routine in its proprietary hand controller, the AutoStar II (Meade Instruments Corporation 2003, 18-19). After aligned, the telescope was then directed to Polaris. Any additional adjustments necessary to center Polaris were made via the hand controller. The MATLAB script, “collimation.m”, can be found in Appendix IV: Real-Time MATLAB Code.

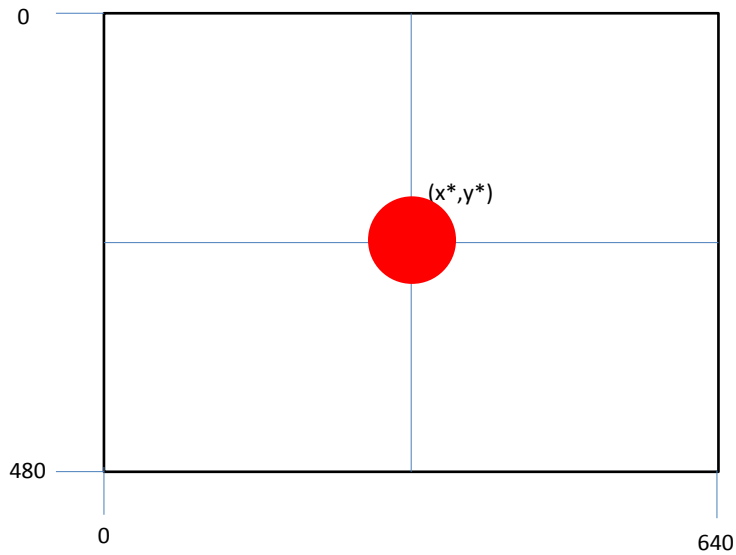


Figure 11: Reference Point in Main Optics Field of View

The WFOV was then referenced to ensure that Polaris was indeed in view. A digital snapshot was then taken of the WFOV, and then displayed as an image. The MATLAB command “ginput” was employed again, allowing the operator to manually select Polaris’ location in the WFOV. The 1x2 matrix generated by “ginput” was saved under the variable name “sweetspot” in the collimation_results.mat file.

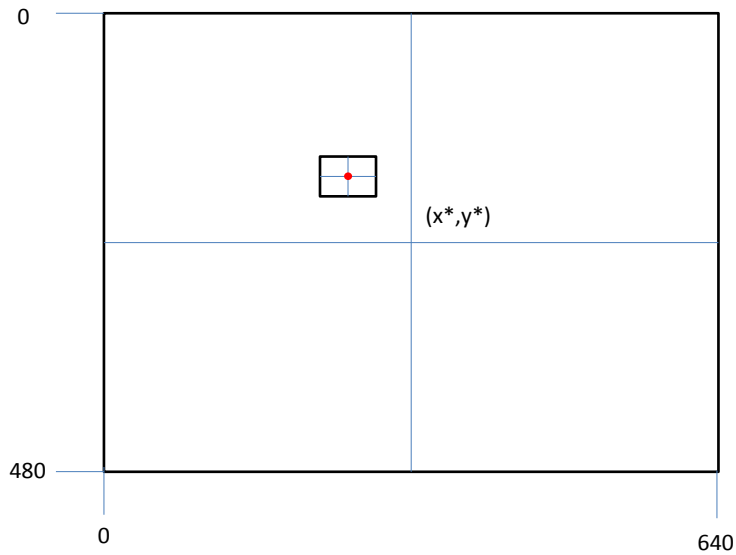


Figure 12: Reference Point in Wide Field of View

The “sweetspot” now gives the closed-loop routine a definite reference position to determine pixel offsets, rather than simply using the center of the WFOV. Without perfect physical collimation, the NFOV could be far enough away from the center of the WFOV to never have an object centered within it; generating a “sweetspot” provided an accurate bias to angular computations for increased accuracy.

Laboratory Simulations

Simulating Satellites for Image Processing Techniques

In order to determine the effectiveness of different image processing techniques, a series of eight videos were recorded, simulating satellites in various scenarios. These scenarios were based on data collected from previous research in the development of the open-loop control method. Real videos of satellites were not used because previous

research did not record satellites in a usable video format. In order to convert .avi files into a MATLAB object, they must be 8-bit Indexed or grayscale images, 16-bit grayscale, or 24-bit TrueColor images (The MathWorks, Inc. 2010). The satellite videos were recorded as 12-bit Indexed images. Without appropriate conversion software, these videos could not be used in the test script.

The simulation data was recorded in a laboratory environment. The backup USB webcam was employed with its original optics installed. The camera was then pointed at a blank wall in a darkened room, recording laser pointers that were shined within the FOV. Video data was then recorded at a rate of 10 frames per second, generating 200 frames. The frames were then compiled into .avi files for use later.

As mentioned, the simulated satellites were represented by laser pointers. Earlier videos contained only singular, stationary laser dots; later videos employed moving dots, flashing dots, solid dots (representing the star field background), all in varying combinations. Each combination allowed software to be developed further to solve different situations.

Hardware Configuration

In order to conduct these laboratory simulations, only a Philips SPC-900NC and the control laptop were required. The webcam had its factory-included lens installed to allow normal focus while it was not connected to the guide scope. This configuration is depicted in Figure 13.

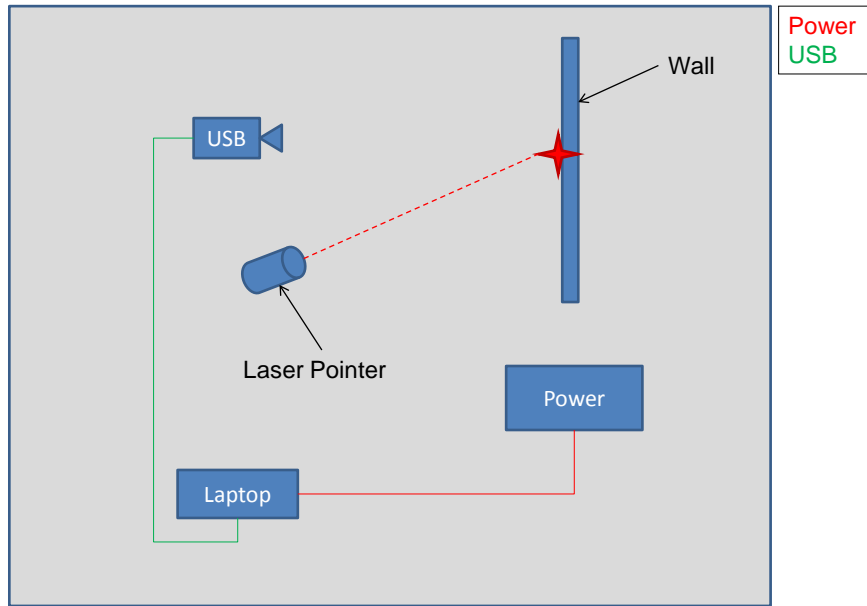


Figure 13: Laboratory Simulation Hardware Configuration

Test Subjects

Test videos were recorded using the Philips SPC-900NC webcam. These videos simulated varying scenarios over the duration of an observation period. Figure 14 through Figure 21 depict compiled images of each video. These images were constructed using the relevant areas from every tenth frame of videos. The object paths and locales labeled on the images do not represent the outcome of the centroid tracking routine and are only used to give the reader an impression of the object’s movement. These videos are not necessarily representative of typical situations; instead, they are to be considered “worst case” scenarios.

The first video recorded was “testvid1.avi.” It simulates a single object within the WFOV, drifting in a random pattern. This type of movement would be indicative of a telescope mount vibration issue. The object path is depicted in Figure 14.

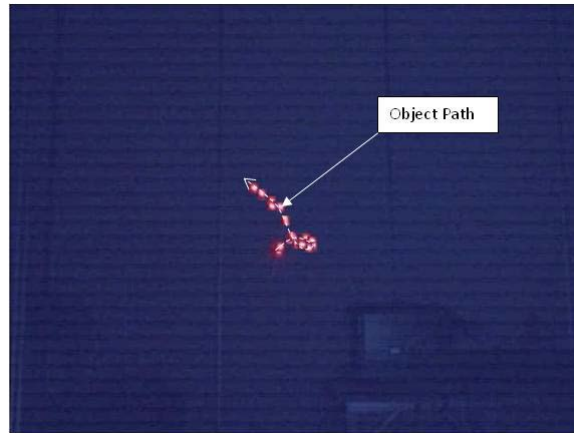


Figure 14: testvid1.avi

The second video recorded was “testvid2.avi.” This video also simulates a single object within the WFOV but with more motion, simulating a wider degree of telescope mount vibration. The object path is depicted in Figure 15.

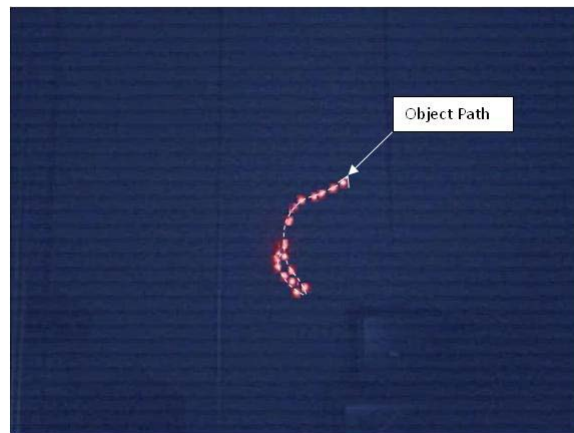


Figure 15: testvid2.avi

The third video recorded was “testvid3.avi.” This video simulates a tumbling satellite. The tumbling satellite aspect is recreated by using a flashing laser dot. The flashing was accomplished by rotating a disk in front of the laser pointer. The disk was half-transparent and half-opaque, and affixed to a cordless drill. A laser pointer was then moved through the field of view while the disk rotated in front of it. The object path is depicted in Figure 16.

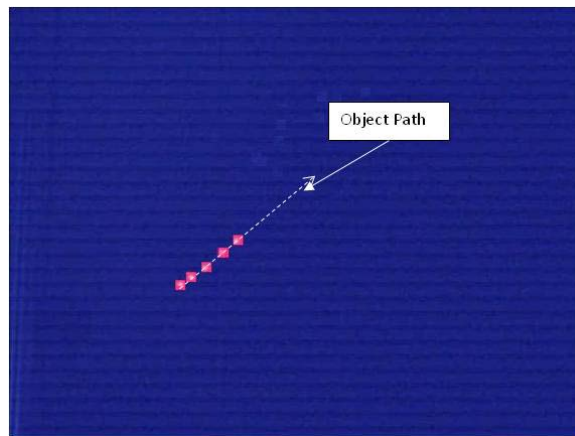


Figure 16: testvid3.avi

The fourth video recorded was “testvid4.avi.” This video simulates a relatively stationary object within the WFOV (meaning that the telescope system is tracking continuously and with a continual bias), that is also tumbling. The object locale is depicted in Figure 17.

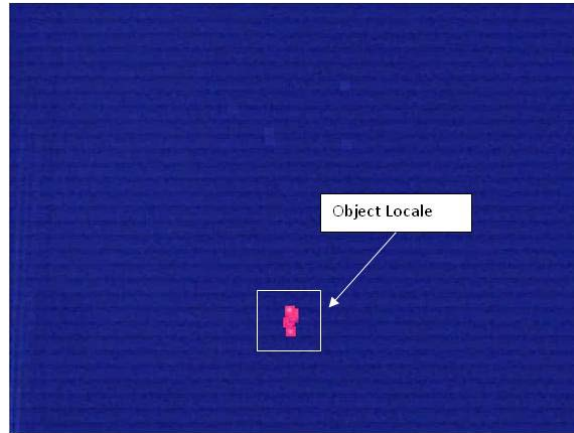


Figure 17: testvid4.avi

The fifth video record was “testvid5.avi.” This video has a relatively stationary target within the WFOV but has a flashing background object moving through the FOV. The scenario is depicted in Figure 18.

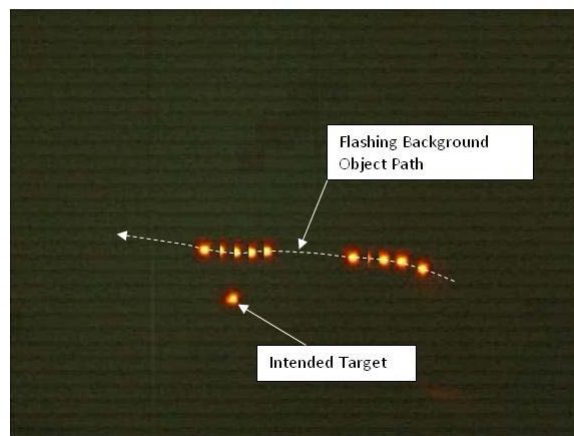


Figure 18: testvid5.avi

The sixth video recorded was “testvid6.avi.” This recording simulates a solid moving target with a flashing background object moving through the WFOV. The scenario is depicted in Figure 19.

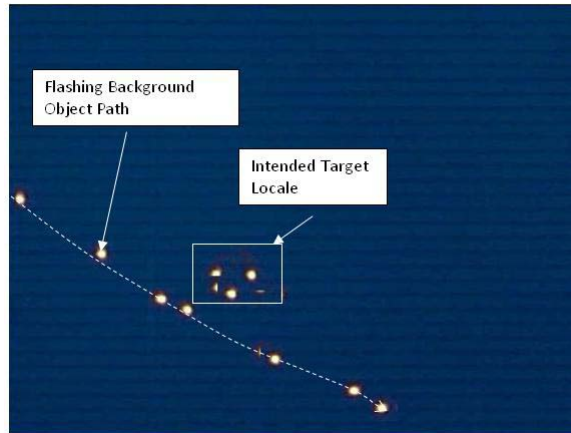


Figure 19: testvid6.avi

The seventh video recorded was “testvid7.avi.” This video simulates a tumbling target that is passed closely by a solid background object. The scenario is depicted in Figure 20.

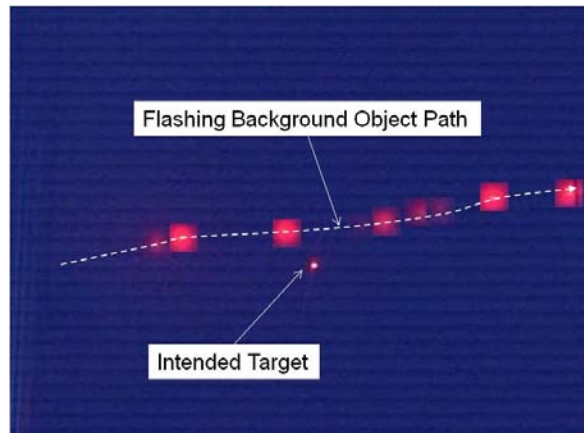


Figure 20: testvid7.avi

The eighth and final video recorded in the laboratory was “testvid8.avi.” This video simulates a tumbling, moving object that is closely passed by a solid background object. The scenario is depicted in Figure 21.

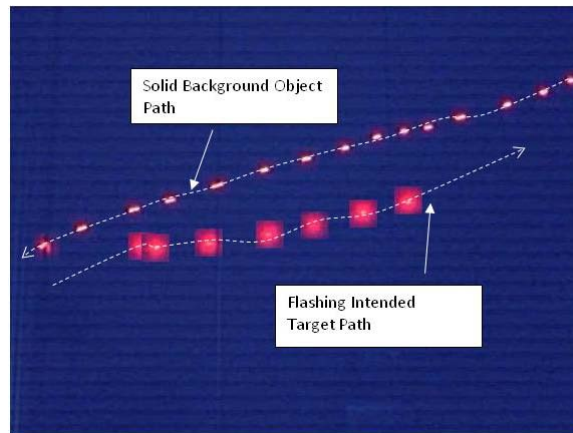


Figure 21: testvid8.avi

Refining Targeting Logic

During the laser dot experiments, it was observed on many occasions that primitive versions of the tracking software would end up tracking the wrong object, most often if the intended target was flashing while a steadily illuminated object passed through the WFOV. This called for some advanced techniques and logic loops to ensure the tracking software was “locked on” to the correct object, or could interpolate its next position to some acceptable degree of accuracy until an actual target is re-acquired. In order to employ this logic, a centroid history needed to be used. The centroid history is a 10x2 matrix containing the last ten centroids computed at any given time. The centroid history is initialized at the “sweet spot” and is updated as centroids are computed.

The first piece of logic that was written dealt with a singular flashing object, simulating a tumbling satellite with a fluctuating amount of reflectivity. In this case, only one centroid or less is being considered in a frame. Should the software be tracking a centroid then lose sight of it, a first-order approximation of position is accomplished. Using the last two values in the centroid history allows the software to develop a “pixel velocity” then applies that velocity to the subsequent frames until the object is re-acquired. This is depicted in Figure 22. In order to reduce the possibility that an estimated velocity will “walk” the telescope off target, a velocity limit is set. Equation (5) illustrates how the horizontal velocity, v_x , is computed:

$$v_x = \frac{(x_2 - x_1)}{\Delta t_1} \quad (5)$$

where x_1 is the horizontal position of the second-to-last centroid computed, x_2 is the horizontal position of the last centroid computed, and Δt_1 is the elapsed time between recording the last and second-to-last frames. Equation (6) illustrates how the vertical velocity, v_y , is computed:

$$v_y = \frac{(y_2 - y_1)}{\Delta t_1} \quad (6)$$

where y_1 is the vertical position of the second-to-last centroid computed, y_2 is the vertical position of the last centroid computed, and Δt_1 is the elapsed time between recording the last and second-to-last frames. The next estimated horizontal position, x_3 , is given by Equation (7):

$$x_3 = v_x \Delta t_2 \quad (7)$$

where v_x is the horizontal pixel velocity computed in Equation (5), and Δt_2 is the elapsed time between recording the last frame and the current frame for which a centroid position is being approximated. The next estimated vertical position, y_3 , is given by Equation (8):

$$y_3 = v_y \Delta t_2 \quad (8)$$

where v_y is the vertical pixel velocity computed in Equation (6), and Δt_2 is the elapsed time between recording the last frame and the current frame for which a centroid position is being approximated. Figure 22 shows a graphical representation of the estimated position of a flashing target.

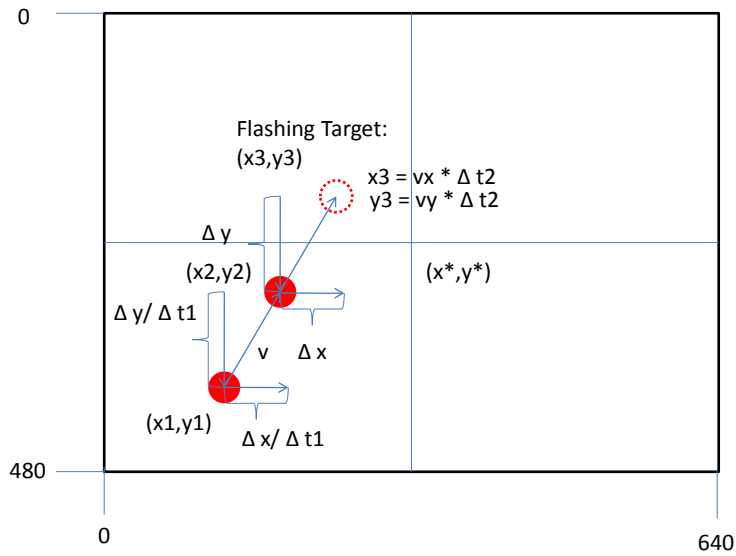


Figure 22: Pixel Velocity Approximation

The next piece of logic needing development was tracking the correct centroid against a background containing other centroids. This simulates a satellite moving against the star field. When two or more centroids are encountered in a frame, as in Figure 23, the software can calculate the centroid positions for each object, but does not automatically know which one should be tracked. Therefore, it was necessary to develop an array of existing centroids in a frame as a list of candidates, and then select the one that was the minimal distance from the last entry in the centroid history.

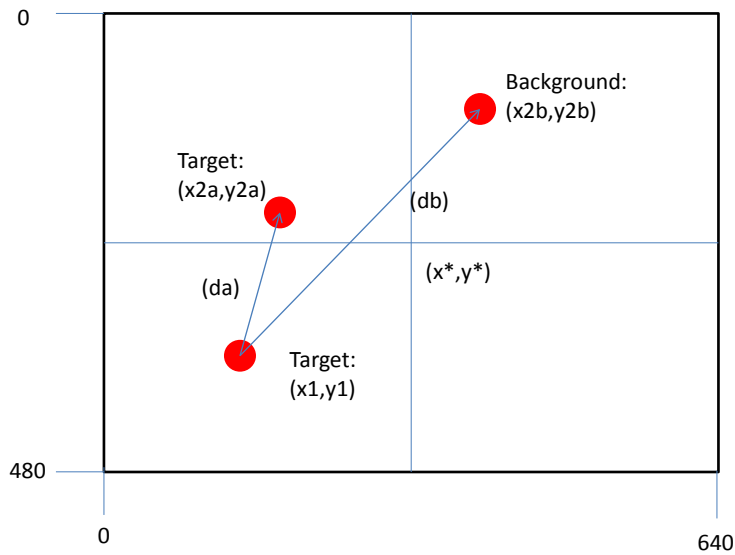


Figure 23: Multiple Centroids

As described earlier, if two or more centroids are encountered in one frame the “regionprops.Centroid” function will generate centroid values for all pixel blobs in the WFOV. In Figure 23, the intended target is last detected at position (x_1, y_1) , where x_1 and y_1 respectively represent the horizontal and vertical positions of the target. The next

frame provides generates two centroids from the intended target and a background object. The target and background object are represented by (x_{2a}, y_{2a}) and (x_{2b}, y_{2b}) , respectively. The absolute pixel distance between the intended target and the last calculated centroid position, d_a , is given by Equation (9):

$$d_a = \sqrt{(x_{2a} - x_1)^2 - (y_{2a} - y_1)^2} \quad (9)$$

The absolute pixel distance between the background object and the last calculated centroid position, d_b , is given by Equation (10):

$$d_b = \sqrt{(x_{2b} - x_1)^2 - (y_{2b} - y_1)^2} \quad (10)$$

When a list of candidate centroids is compiled, the candidate with the minimum distance to the last centroid position is used as the next centroid.

Finally, a method needed to be developed that would allow a flashing object to be tracked against a star field. In this case, only one centroid would be detected, but it would be the incorrect object, as depicted in Figure 24. Without this logic in the software, it would track the new object and lose lock on the target object even if it is re-illuminated. Since a satellite is presumed to be relatively stationary in the field of view while the background object moves through it, it is an acceptable assumption to place a limit on the distance an object may have traveled in the field of view. If a centroid is detected beyond this limit, the linear approximation for position using velocity is employed.

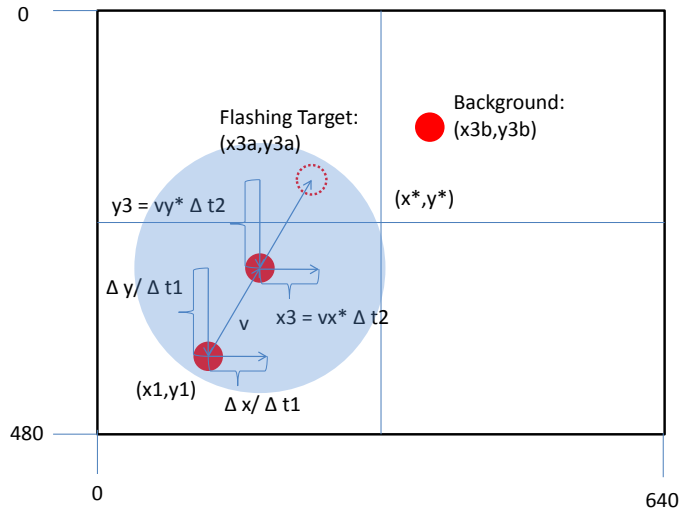


Figure 24: Flashing Target with Solid Background Object

Converting Test Software for Real-time Use

Converting the test script to a real time version resulted in the development of two control scripts and six modular function scripts: “begin_track.m”, “frame_centroid_cb.m”, “frame_centroid.m”, “closest_centroid.m”, “velocity_approximation.m”, “compute_offsets.m”, “update_history.m”, and “end_track.m”. A description of each script follows:

begin_track.m: this script initiates the optical centroid tracking function

frame_centroid_cb.m: this script is a timer callback function; it periodically reads video data from the USB webcam for processing

frame_centroid.m: this script processes each image as it is attained by *frame_centroid_cb.m*; it outputs an array of centroids detected in an image

closest_centroid.m: in the event that multiple centroids are detected, this function returns the centroid that was closest to the last centroid detected/approximated

velocity_approximation.m: in the event that a computed/estimated centroid is outside of set tolerances, this script generates a first-order approximation of centroid position

compute_offsets.m: this function takes the computed/estimated centroid, determines its offset from the sweetspot, and returns angular offset data

update_history.m: this script simply updates the centroid history array with the latest centroid data and deletes the oldest data

end_track.m: this script terminates the optical centroid tracking function and resets parameters for re-initiation

Some scripts were slightly modified to allow logging of real-time data for analysis in Chapter IV; actual versions do not need to log data. The real-time experiment used a laser pointer moved in a random pattern in order to determine the average time between centroid computations during real-time use. The real-time MATLAB scripts can be found in Appendix IV: Real-Time MATLAB Code. In addition to the above scripts, an excerpt of code from the main GUI script, “*seeker_trackgui_v2.m*”, can be found in Appendix IV.

Remote Control

In order to contribute to the concept of a remotely operated autonomous tracking system, two areas of research needed to be explored: automated shelters and data connectivity.

Automated Shelters

A remotely operated observation station would most likely be unmanned; therefore facility housing a telescope would have to be automated in its design. To this end, common shelter types were reviewed for their advantages and disadvantages. The three types of shelters researched were domes, clam-shell, and roll-off roofs.

Domes

Description: Dome-style observatories are very common for night sky viewing, and are available in many a scale of sizes. Domes of the size necessary to enclose a 10” telescope are usually constructed from plastic, but are also available in sheet metal designs. In order to allow a clear view of the sky, the domes have an opening that partially opens a slit on one side of the shelter. Automated domes can sometimes interface with a motorized telescope mount to align the slit with the FOV of the telescope. An example of a dome shelter is illustrated in Figure 25 (Candomes Observatories Ltd. 2008).



Figure 25: Dome Enclosure (*photo credit: Candomes Observatories Ltd. 2008*)

Advantages: Domes are readily available and can be installed rather easily. Most models offer excellent weather proofing due to the low number of exposed seams. Additionally, only the slit needs to be closed in order to seal the enclosure; no other equipment is required to move the telescope for storage.

Disadvantages: As mentioned, the entire dome needs to rotate to keep the telescope's FOV unobstructed. This means that the dome's rotational drive would be experiencing many more cycles during an observation period, and would have to be fast enough to rotate the structure with the telescope.

Clam-shells

Clam-shell designs are a type of dome, but open and operate in a different fashion from the domes described above. When closed these designs resemble domes, but they open by splitting down the center in segments. An example of a clam-shell follows in Figure 26 (AstroHaven Enterprises 2008).



Figure 26: Clam-shell Enclosure (*photo credit: AstroHaven Enterprises 2008*)

Advantages: The clam-shell design benefits in a few key areas. When the enclosure is open, it has a completely unobstructed view of the sky. The clam-shell dome does not need to rotate with the telescope so the mechanism is simplified. The clam-shell also requires no additional equipment to move the telescope for clearance issues.

Disadvantages: The chief disadvantage to a clam-shell design is its expense. These shelters are not common among backyard astronomers, and they are not produced in quantity. While mechanically uncomplicated, this design's cost could be more than the cost of the telescope enclosed within.

Roll-off Roof

The roll-off roof designs considered were shed-like structures. The roof is mechanically actuated to slide horizontally away, opening the top of the structure. A mechanical pier in the center of the structure raises the telescope into position. An example of this type of enclosure is illustrated in Figure 27 (Pier-Tech, Inc. 2007).

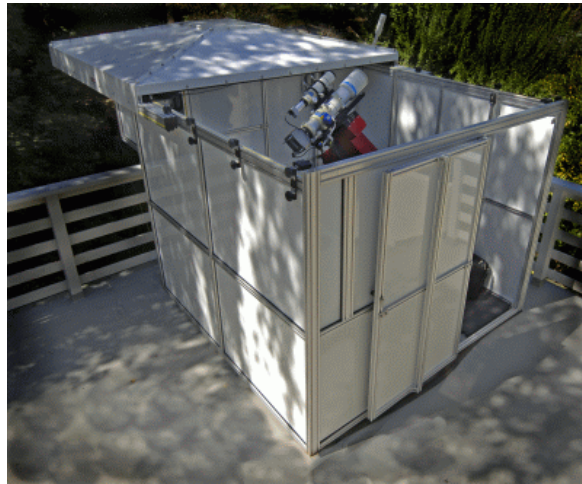


Figure 27: Roll-off Roof Enclosure (*Pier-Tech, Inc 2007*)

Advantages: The roll-off design allows a mostly unobstructed view of the sky. The design is easily constructed, and some designs can be taken apart and moved as necessary. This option can also be rather inexpensive, depending on the design chosen.

Disadvantages: The roll-off design requires the addition of a vertically actuated pier to move the telescope into position. This is also required to provide clearance for the

roof to close. Telescope size is limited to the weight capacity of the pier. Due to the linear movements of the components, safety switches are required to ensure damage to the structure does not occur during opening or closing.

Data Connectivity

Figure 9 illustrated an example of a satellite being observed by two geographically separated tracking stations. This is an example of a simultaneous observation; a network of telescopes does not necessarily need to have two telescopes observing one satellite simultaneously. In order to coordinate the operation of these telescopes, remote control concepts must be explored.

Concept 1: Centralized Control

One method of controlling a network of telescopes would have multiple optical devices being controlled by a central computer. This network would require only one computer to compute azimuth/elevation angles for each site, controlling each telescope in real-time. This network would have some cost-saving advantages with only one computer being employed for the entire network, but as a network expands bandwidth and computation requirements would likewise continue to rise. A significant disadvantage of this configuration comes from network delays and the necessity for continual contact between the control computer and multiple telescopes. This control concept is depicted in Figure 28.

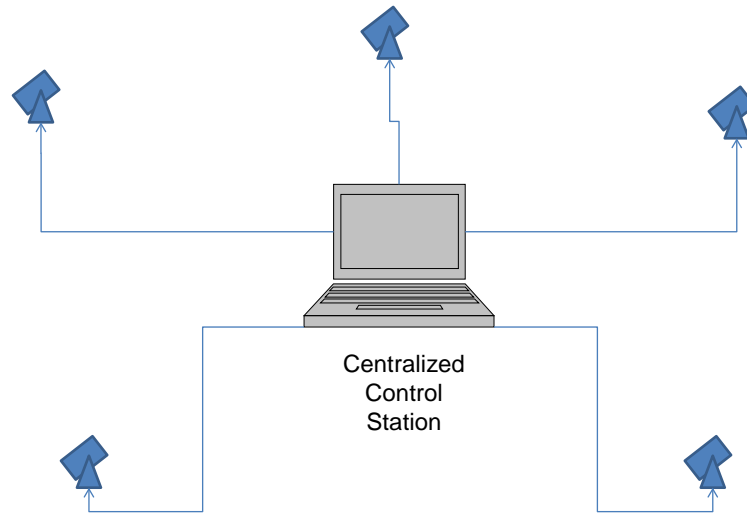


Figure 28: Centralized Control Concept

Each of the observation stations would be configured as detailed in Figure 29. This configuration requires controller connectivity through a network USB device, such as the Belkin Network USB Hub (Belkin International, Inc. 2010). This device theoretically allows a computer to access a USB device via an Ethernet connection, enabling remote control. Also, no control computer would be required at the observation site.

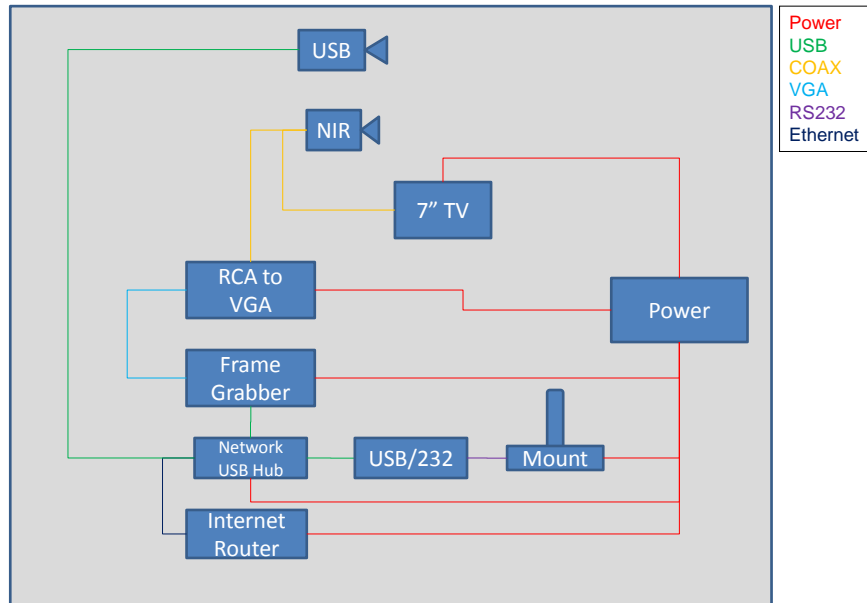


Figure 29: Observation Station Configuration for Centralized Control

Concept 2: Distributed Control

Another approach to remote control of a telescope surveillance network would employ standalone observation sites, each with dedicated control computers. Each computer would run the required pre-calculations for a given observation period, as well as the azimuth/elevation angle calculations. Taskings would be received from a command computer; the observation sites would be “slaved” to this command computer only for direction to follow selected satellites. Recorded data could be collected, correlated, and analyzed later at the command site. This control concept is depicted in Figure 30.

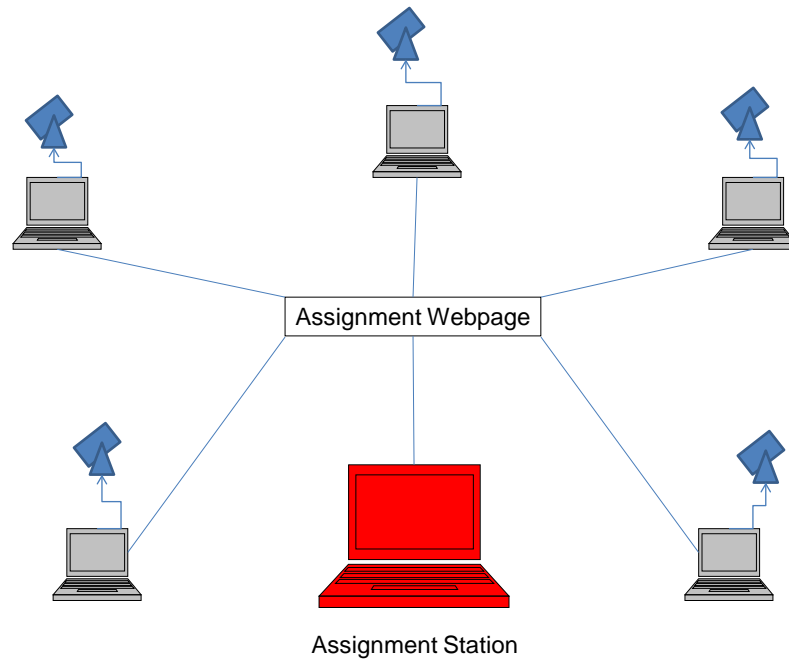


Figure 30: Distributed Control Concept

In this configuration, each observation site is a “stand alone” installation. Their hardware configuration is depicted in Figure 31. This configuration differs from the Centralized Control Concept’s configuration in that it does not need additional hardware to obtain connectivity with the USB devices.

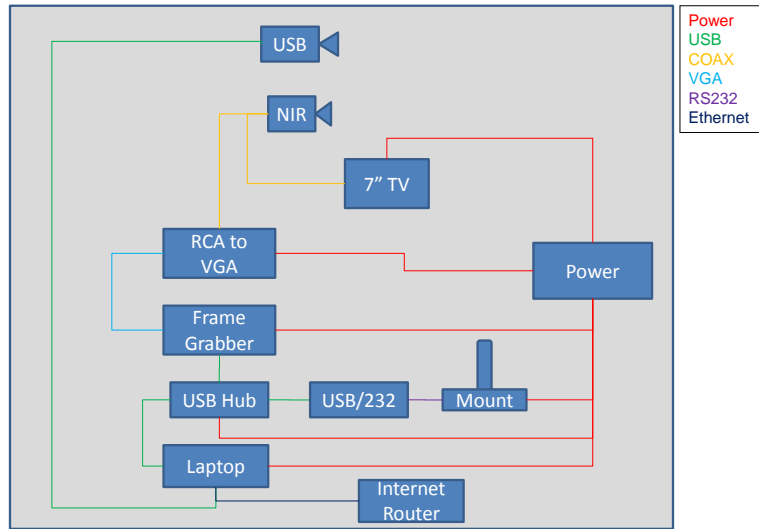


Figure 31: Observation Station Configuration for Distributed Control

Summary

Background information about AFIT's current satellite observation system was detailed at the beginning of the chapter. A method was presented to simulate the viewing of satellites through the WFOV. Laser pointers in a laboratory environment were used to demonstrate different scenarios that might be encountered during an observation period. The eight test videos recorded in the laboratory were intended to simulate various situations that might be encountered during the course of an observation. These videos were used to develop sound logic decisions once an image is processed in order to provide closed-loop feedback to the control software. In addition to the laboratory tracking simulations, different methods of remote control and automation were explored. The next chapter shows the results of laboratory tests, shelter selection, and connectivity trials.

IV. Analysis and Results

Chapter Overview

The purpose of this chapter is to display test results from the laboratory simulations outlined in Chapter III, and to detail final shelter selection. Hardware component upgrades are also listed.

Closed-Loop Control

Each of the eight simulation videos were processed using “frame_centroid_test.m.” which is contained in Appendix III: Test MATLAB Code. This script produces a frame-by-frame plot with a calculated centroid over the image. Visual observation allowed for confirmation of “truth.” The resulting centroids were saved in comma-separated value (CSV) and plotted in Microsoft Excel. Each simulation generated three plots: Calculated Centroid Path, Centroid Offset by Frame Number, and Absolute Change in Position.

Results of Simulation Scenarios

Figure of Merit

In order to properly gauge the results of the simulations, a figure of merit is needed. The purpose of tracking the centroid in the WFOV is to generate angular offset data to correct the telescope’s movements. Since the WFOV is not square, the shortest dimension of 480 pixels is used to calculate the figure of merit. With the .5x focal reducer in place on the guide scope, the relationship between angular measure and pixels was determined to be $0.0015^\circ/\text{pixel}$. Multiplying this value by 480 pixels yields a figure of merit equal to 0.72° . This figure of merit implies that if a centroid is estimated to be

more than 0.72° away from the object's actual position, then the object may have drifted out of the WFOV and will not be reacquired.

testvid1.avi

Results: The plots were as expected and showed a smooth track of the laser dot. The calculated centroid path is plotted in Figure 32. Figure 33 shows a “jump” in position; this is because the tracking logic will engage a linear approximation if no centroid can be determined. Once reacquired, the third plot, Figure 34 will likely show a “spike” in position change as the difference between the last approximated position is compared to the next real position of the centroid. The spikes in position seem steep, however the average change in position from frame to frame is 0.0016° , as represented by the horizontal dashed line in Figure 34. All reacquisition spikes fall well below the figure of merit, 0.72° .

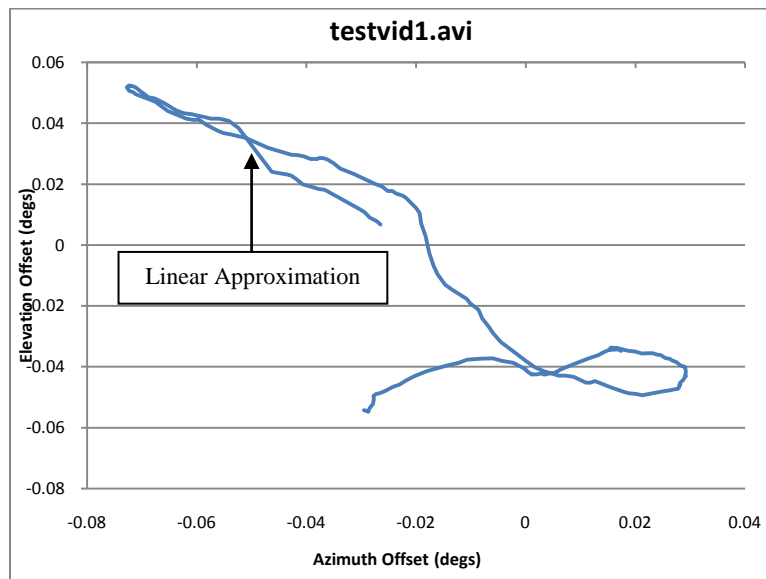


Figure 32: testvid1.avi Calculated Centroid Path

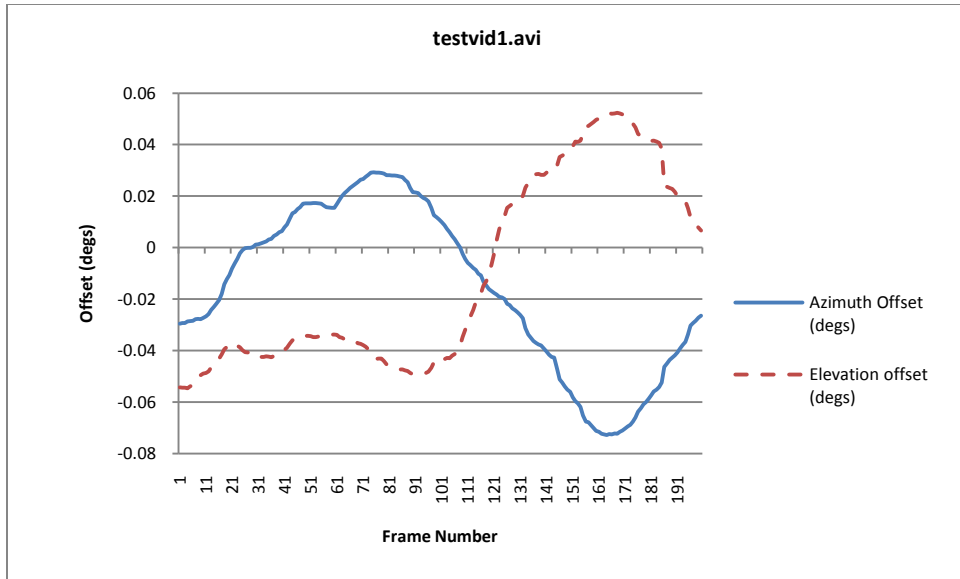


Figure 33: testvid1.avi Centroid Offset by Frame Number

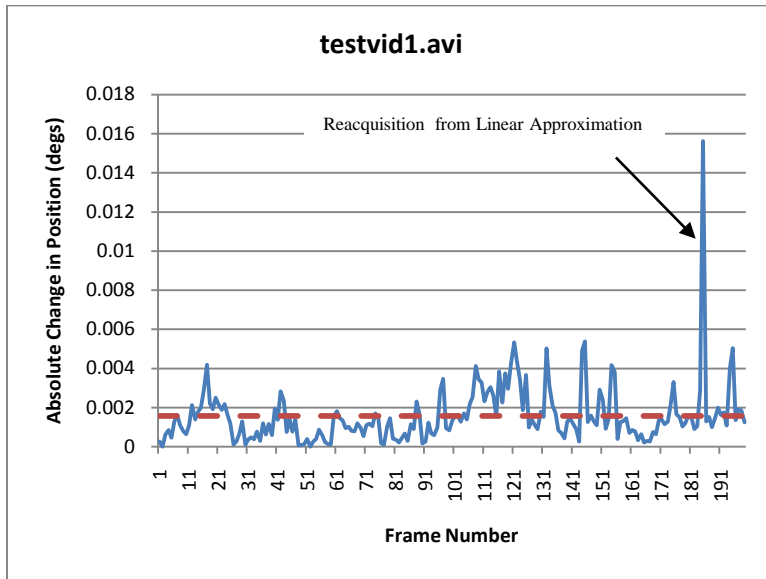


Figure 34: testvid1.avi Absolute Change in Position

testvid2.avi

Results: testvid2.avi was another simple operational check of “frame_centroid_test.m”. Figure 35 showed a smooth track of the laser dot. Figure 36

did not appear to show any linear approximations. Spikes in position change averaged 0.0019° in magnitude, as indicated in Figure 37. All reacquisition spikes fall well below the figure of merit, 0.72° .

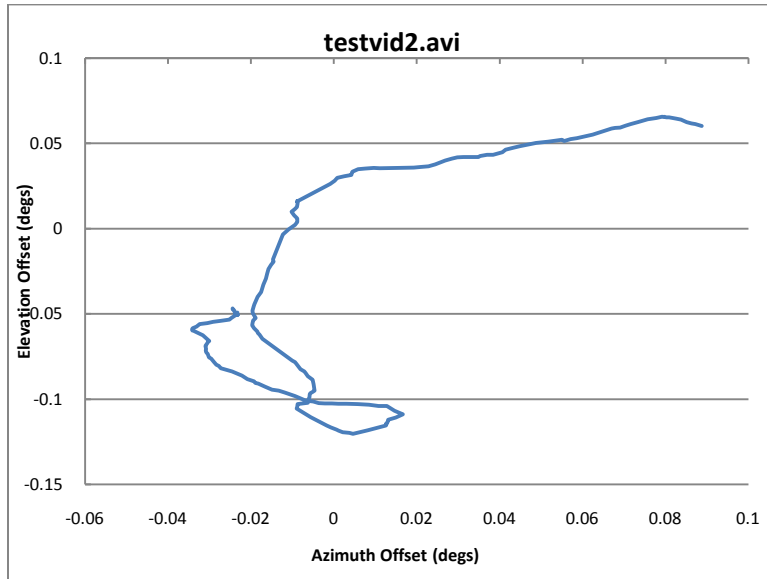


Figure 35: testvid2.avi Calculated Centroid Path

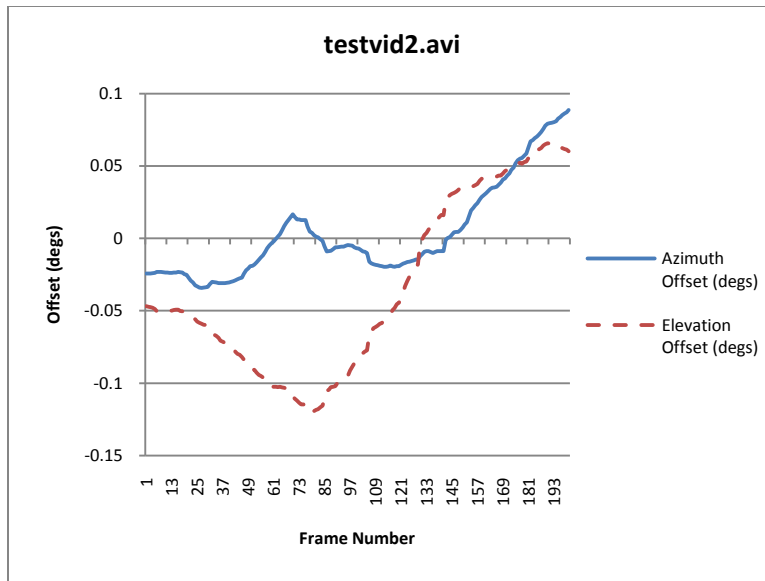


Figure 36: testvid2.avi Centroid Offset by Frame Number

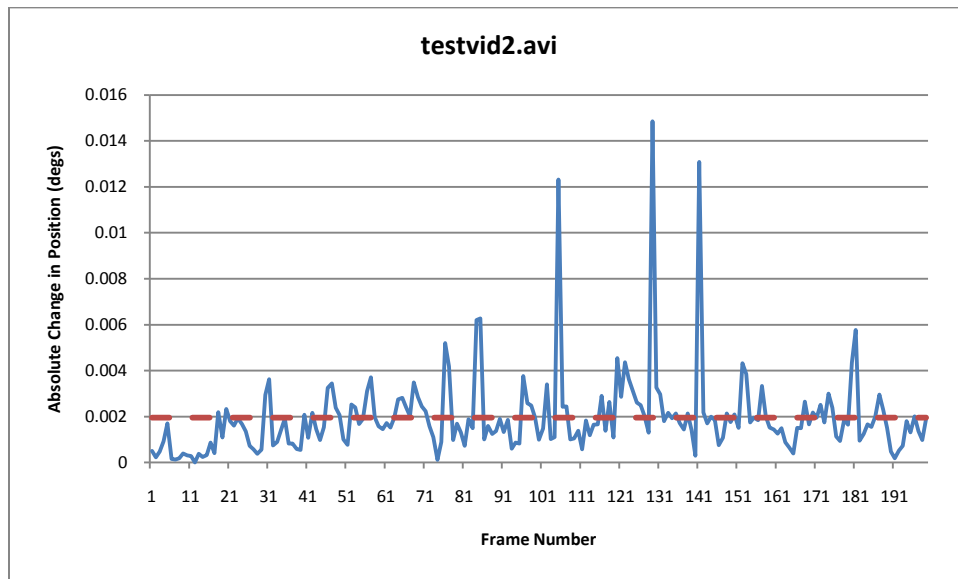


Figure 37: testvid2.avi Absolute Change in Position

testvid3.avi

Results: testvid3.avi was a demonstration “frame_centroid_test.m”’s ability to track a moving inconsistent target. Figure 38 shows a large displacement from the center

of the WFOV; this is because the laser dot was not present at the beginning of the recording. Figure 39 and Figure 40 show telltale signs of linear approximations. In Figure 39, some portions of azimuth and elevation show a steady change in position, while Figure 40 has contains spikes indicative of centroid reacquisition. The average magnitude in change in position was 0.0019° , as indicated by the dashed red line in Figure 40. All reacquisition spikes fall well below the figure of merit, 0.72° .

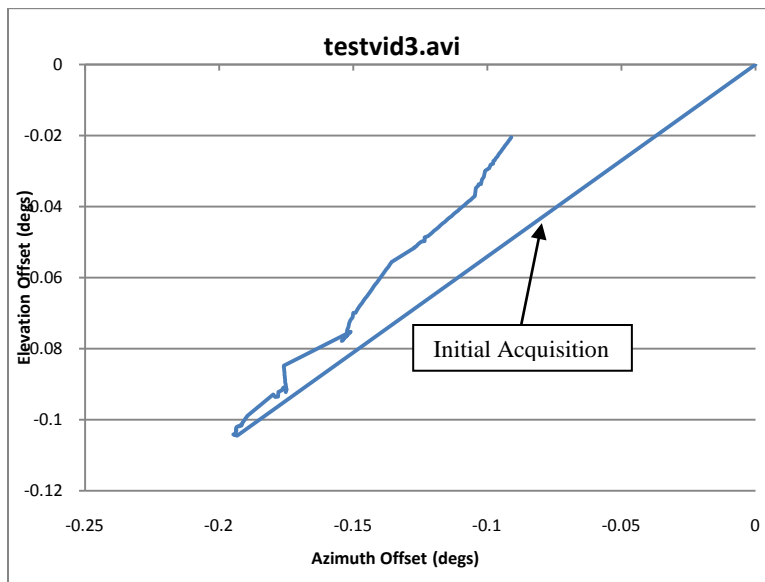


Figure 38: testvid3.avi Calculated Centroid Path

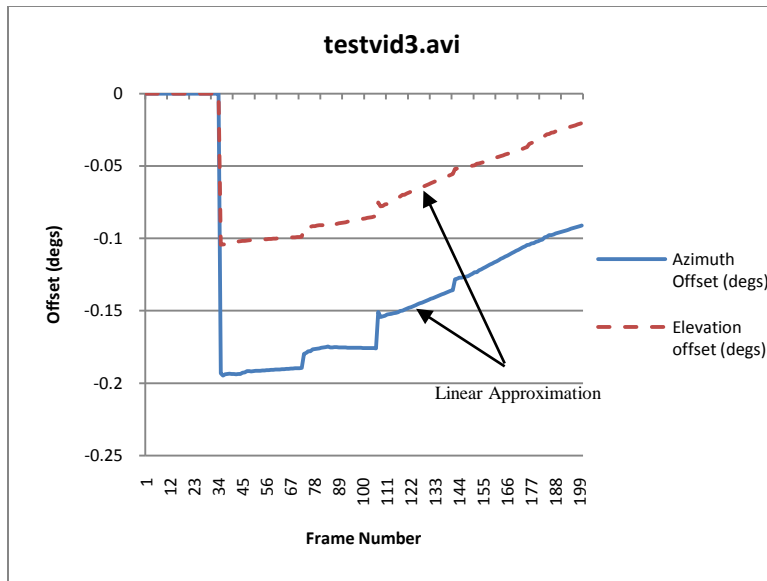


Figure 39: testvid3.avi Centroid Offset by Frame Number

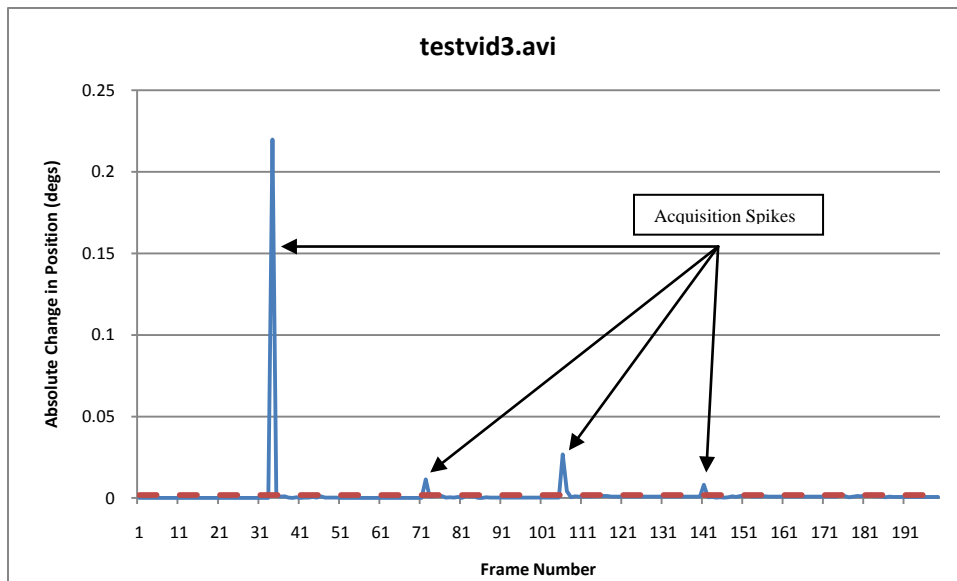


Figure 40: testvid3.avi Absolute Change in Position

testvid4.avi

Results: testvid4.avi was a demonstration of frame_centroid_test.m's ability to track a relatively stationary but inconsistent target. Since this was a relatively stationary target, but inconsistent, the linear approximation caused some displacements away from "truth" as seen in Figure 41. Figure 42 shows some linear approximation periods and Figure 43 shows the large re-acquisition spikes in absolute position; the average change in position was 0.0021° . This information can be interpreted as follows: if a satellite is tumbling and dims from view for a longer period, the script is more likely to "walk off" target. All reacquisition spikes fall well below the figure of merit, 0.72° .

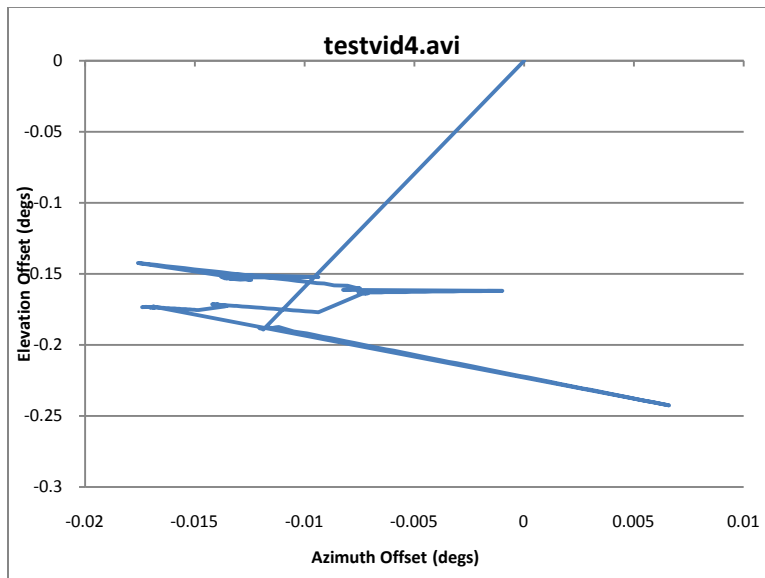


Figure 41: testvid4.avi Calculated Centroid Path

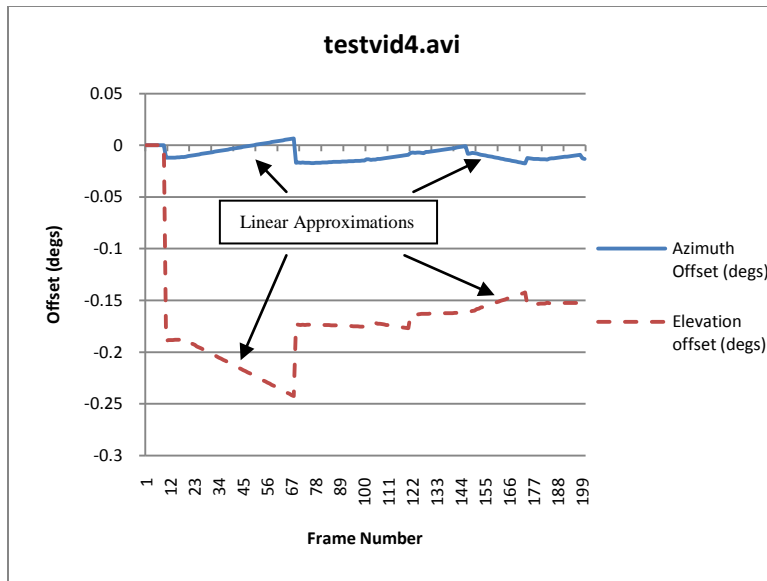


Figure 42: testvid4.avi Centroid Offset by Frame Number

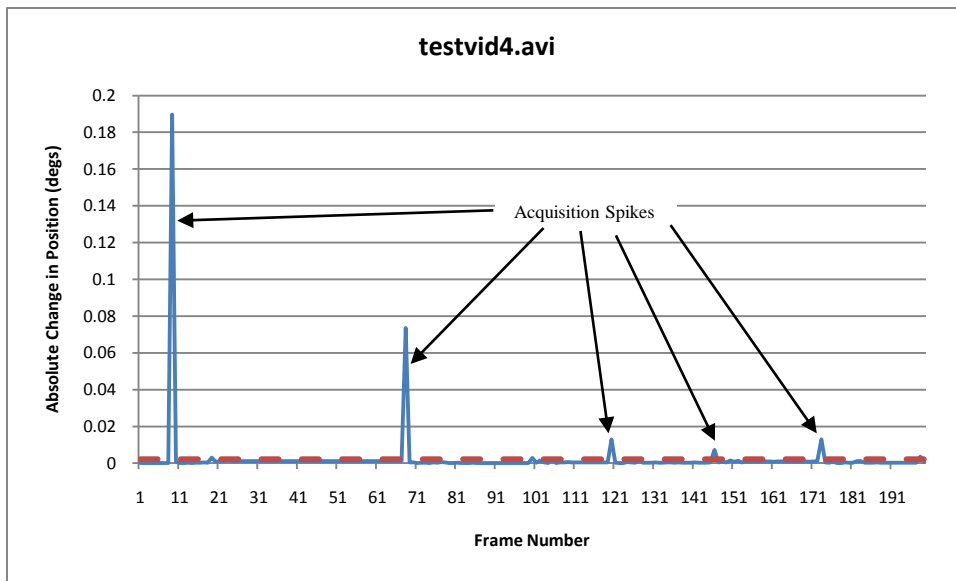


Figure 43: testvid4.avi Absolute Change in Position

testvid5.avi

Results: testvid5.avi was a demonstration of frame_centroid_test.m's ability to track a relatively stationary but consistent target with an inconsistent, moving background object within the FOV. Figure 44 shows that the script locked onto the target; the plot seems like an erratic scribble, however the position of the centroid varies very little. Figure 45 shows a clean lock with no velocity approximations. Figure 46 also seems erratic in nature, but actually averages only 0.0001° over the plot. No reacquisition spikes were recorded; therefore all changes in position fall well below the figure of merit.

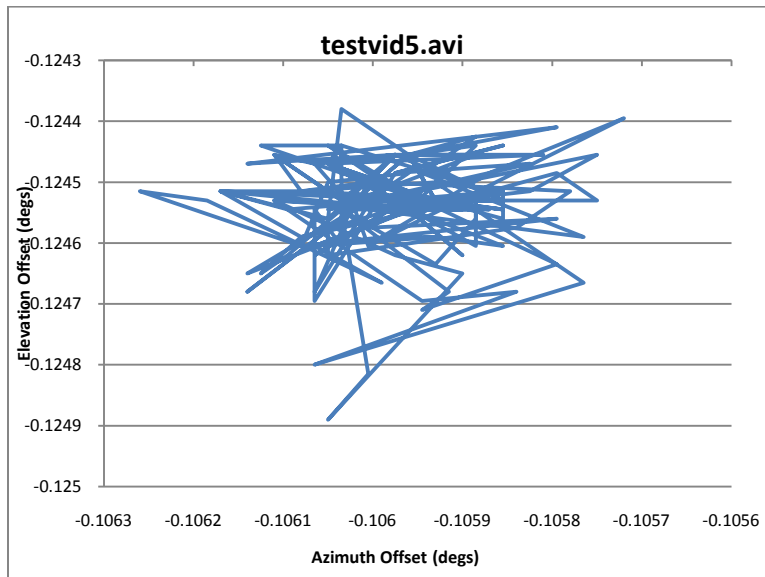


Figure 44: testvid5.avi Calculated Centroid Path

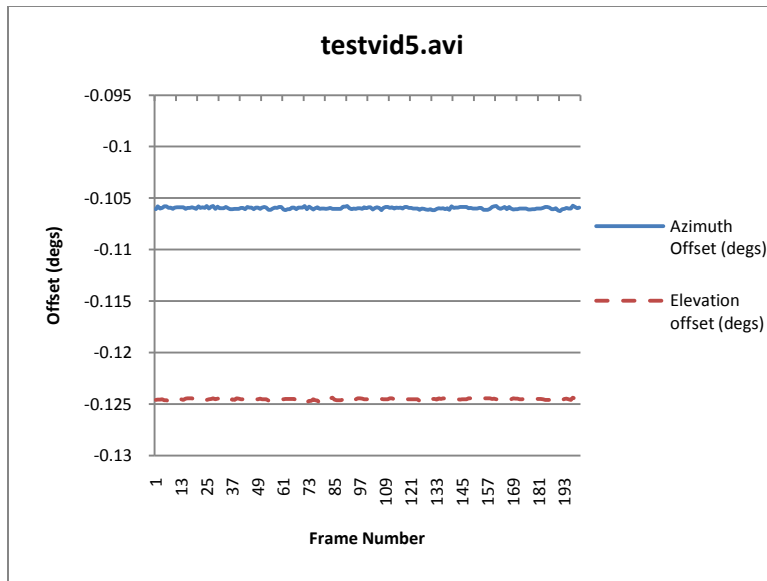


Figure 45: testvid5.avi Centroid Offset by Frame Number

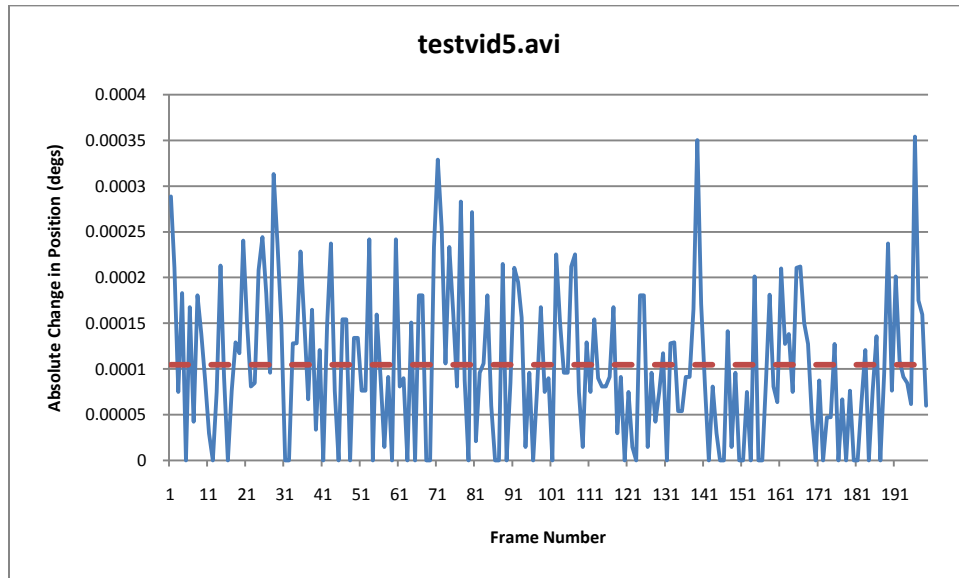


Figure 46: testvid5.avi Absolute Change in Position

testvid6.avi

Results: testvid6.avi was a demonstration of frame_centroid_test.m's ability to track a moving but consistent target with an inconsistent, moving background object within the FOV. Figure 47 shows a consistent track of the target in the WFOV. Figure 48 shows no obvious linear approximation periods. Figure 49 depicts some smaller reacquisition spikes in position, but the average change in position was 0.0012° . All reacquisition spikes fall well below the figure of merit, 0.72° .

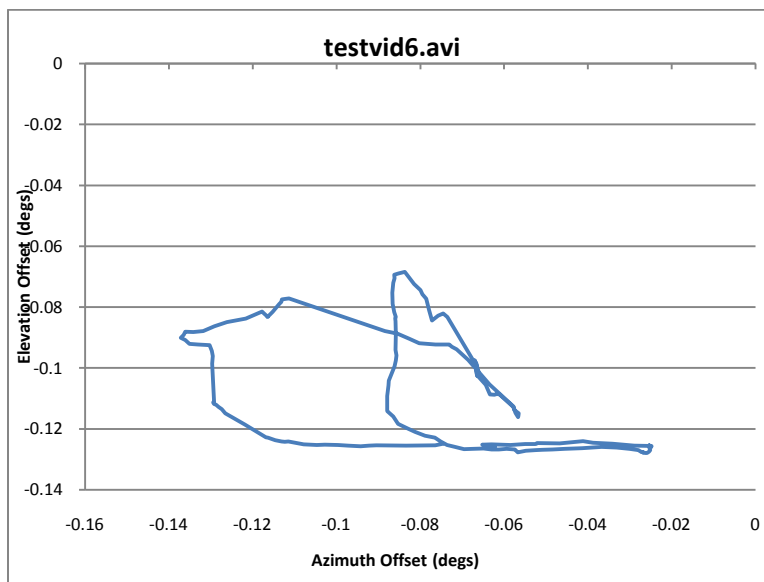


Figure 47: testvid6.avi Calculated Centroid Path

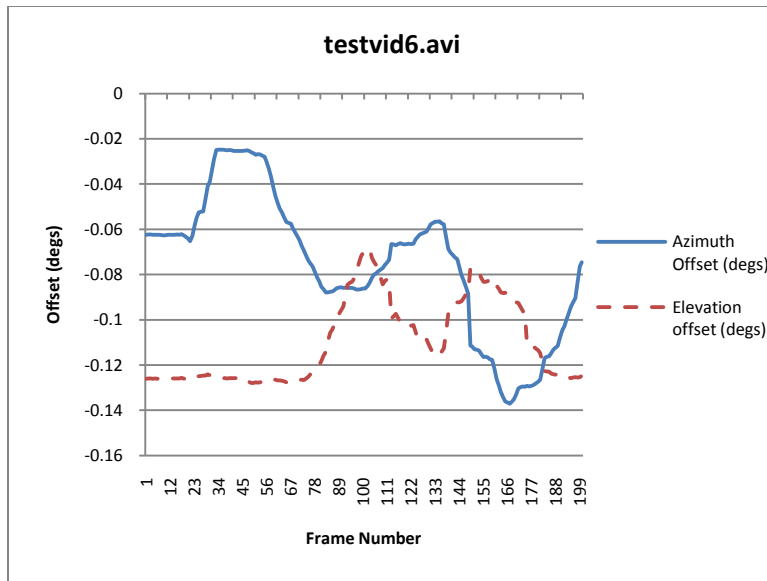


Figure 48: testvid6.avi Centroid Offset by Frame Number

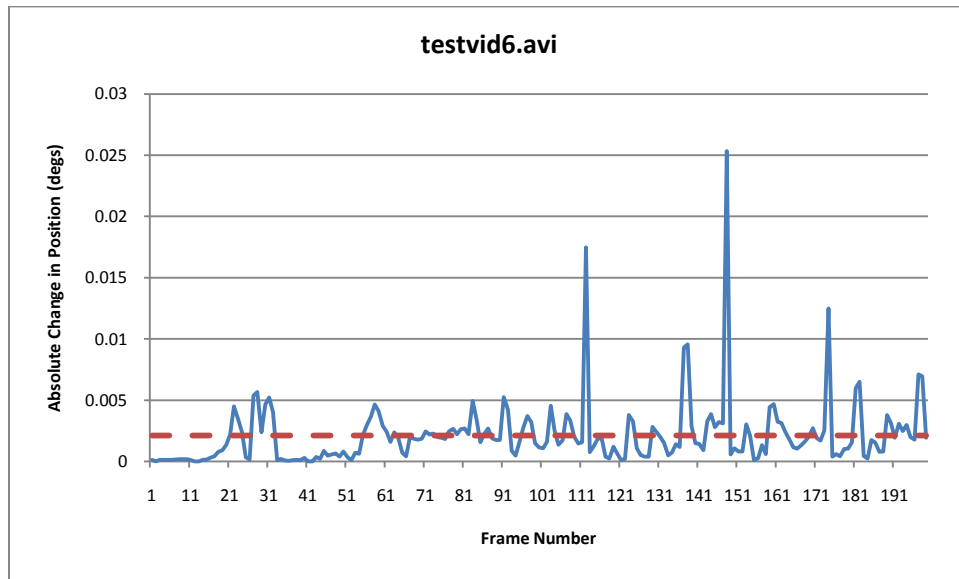


Figure 49: testvid6.avi Absolute Change in Position

testvid7.avi

Results: testvid7.avi was a demonstration of frame_centroid_test.m's ability to track a stationary, consistent target with an inconsistent, moving background object passing close to the intended target. Figure 50: testvid7.avi Calculated Centroid seems to show an erratic movement, when in fact it was a steady target varying very little in movement. Figure 51: testvid7.avi Centroid Offset by Frame Number depicts a very steady lock on the target, as the angular values remain flat. Figure 52 shows that the average change in position was less than 0.0001° . No reacquisition spikes were recorded; changes in position were therefore below the figure of merit.

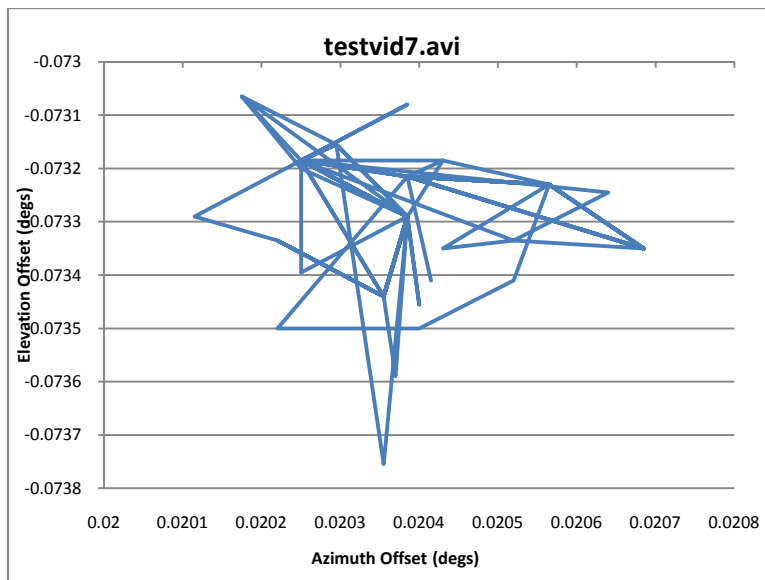


Figure 50: testvid7.avi Calculated Centroid Path

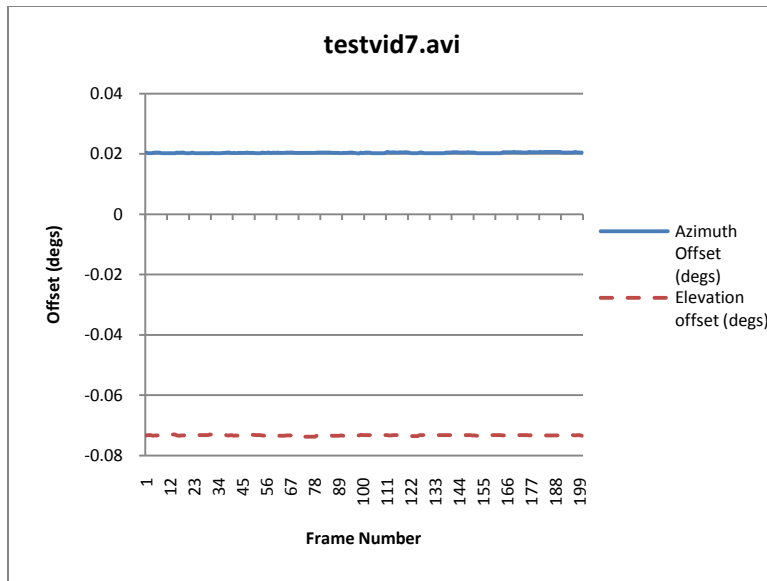


Figure 51: testvid7.avi Centroid Offset by Frame Number

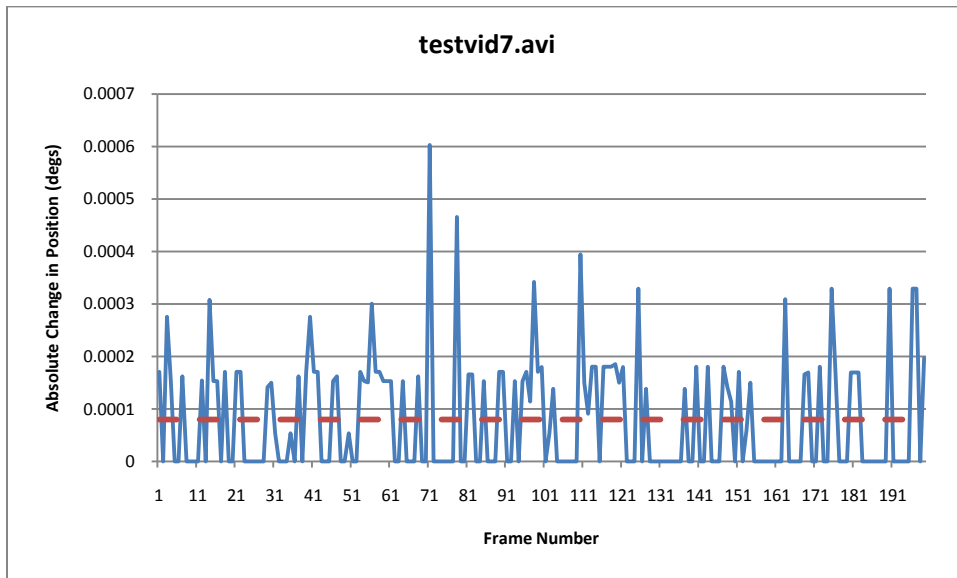


Figure 52: testvid7.avi Absolute Change in Position

testvid8.avi

Results: testvid8.avi was a demonstration of frame_centroid_test.m's ability to track a moving but inconsistent target with a consistent, moving background object

passing close to the intended target. Figure 53 depicts a track of the centroid with some linear approximation zones. Figure 54 displays the linear approximation zones more clearly. Figure 55 indicates the centroid acquisition spikes, and the average position change of 0.0052° . All reacquisition spikes fall well below the figure of merit, 0.72° .

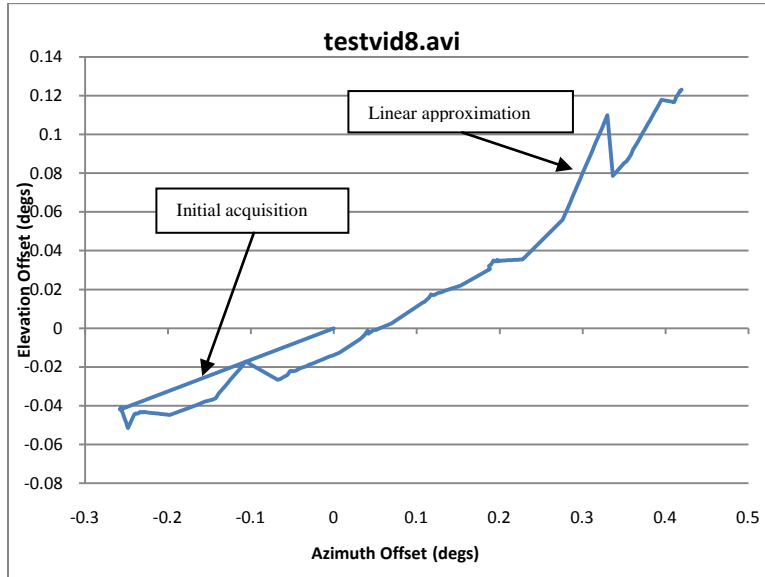


Figure 53: testvid8.avi Calculated Centroid Path

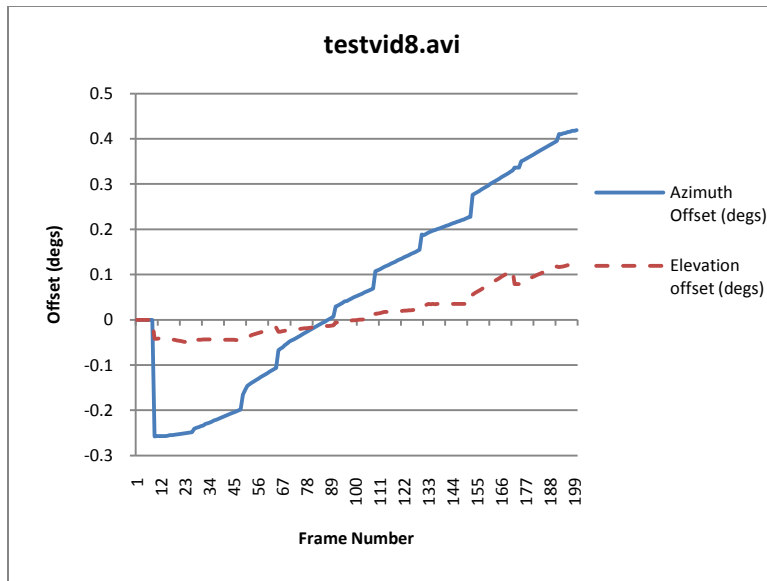


Figure 54: testvid8.avi Centroid Offset by Frame Number

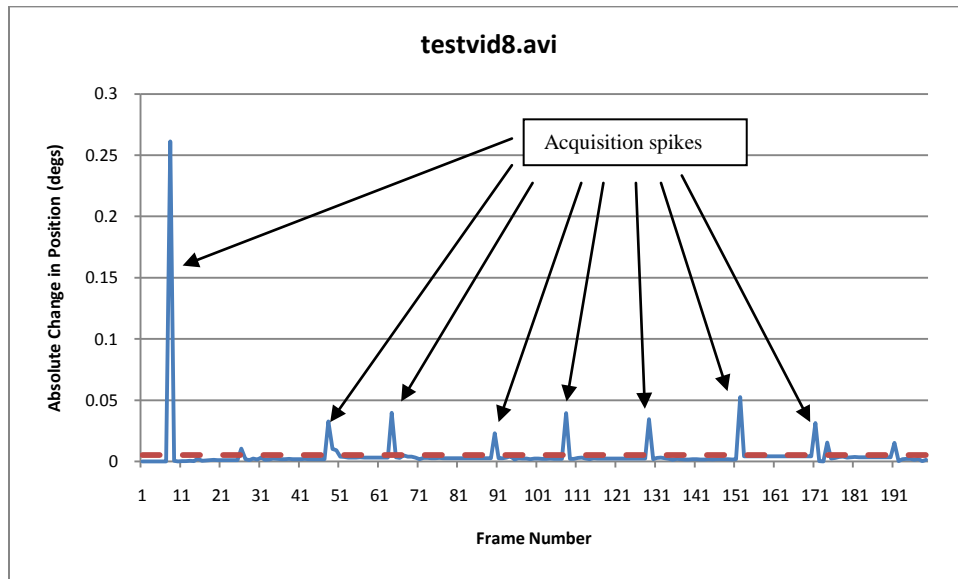


Figure 55: testvid8.avi Absolute Change in Position

Real-time Software Conversion

Real-time software conversion successfully allowed centroid tracking. Videos were not recorded, however angular offset data was analyzed for absolute change in centroid position, time between computations, and average time between computations. Figure 56 shows the change in position between computations, with an average change in position of 0.1071° as indicated by the red dashed line. Figure 57 shows the centroid computation times through the course of experiment, averaging 0.1000 seconds over the course of 140 frames. This shows that the real-time centroid tracking software can update the angular offset faster than the 0.5 seconds it takes to iterate through one loop in the GUI script, providing timely feedback. The real-time software successfully interacted with the GUI script as well.

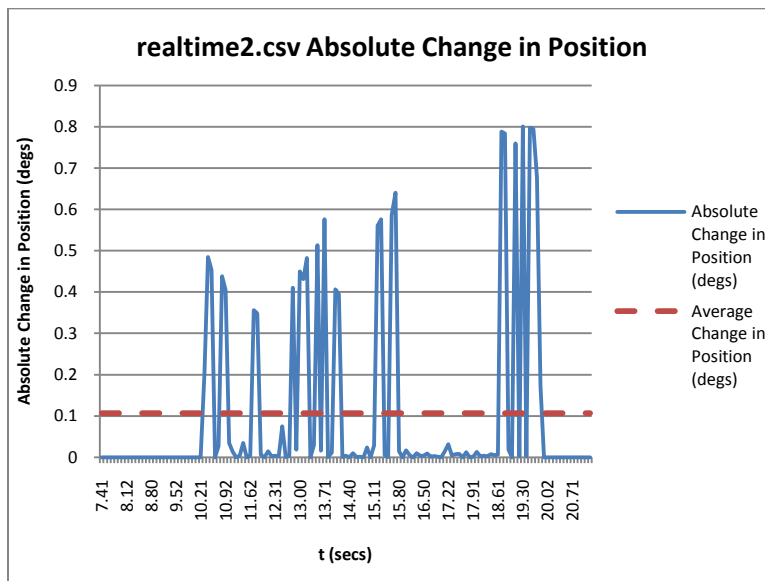


Figure 56: realtime2.csv Absolute Change in Position

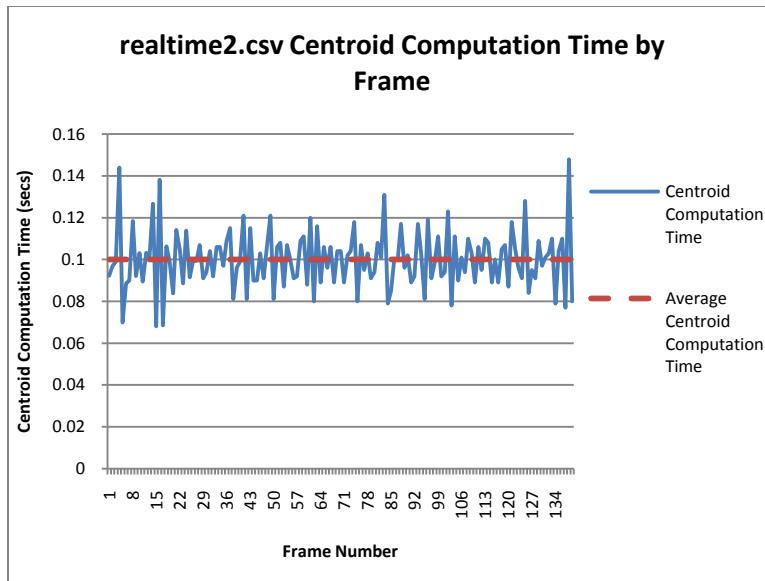


Figure 57: realtime2.csv Centroid Computation Time by Frame

Remote Control

Automated Shelter

A Pier-Tech Tele-Station 2 (Pier-Tech, Inc 2007), a commercially manufactured observatory shelter was constructed on top of AFIT's Building 640 at Wright-Patterson AFB. An existing I-beam structure on top of Building 640 provided the foundation for the observatory shelter. A lumber deck was built to form the platform underneath the shelter. The deck features include stairway access and 120v AC power, and safety features include handrails, glow-in-the-dark tape on the stairs, and non-slip rubber mats on the roof's walkway.

Shelter Description

The observatory shelter is built atop the wooden deck described above. The observatory shelter measures 10' x 10' and is approximately 6.5' tall at the peak of its roof, as shown in Figure 58. Significant features for the rooftop observatory include a

motorized, retractable roof and actuated pier. The pier also has an optional vibration isolation system.



Figure 58: AFIT Roof Top Observatory

Retractable Roof

The roof of the observatory is retractable by sliding the roof along rails, as illustrated in Figure 59. This process is facilitated by a torque motor that drives a worm gear. The worm gear is then connected to a rack and pinion system, with the rack being attached along the axis of motion for the roof. The torque motor voltage is specified by a digital control box. The digital control box takes user inputs (such as ‘open’ or ‘close’) as well as from electrical safety switches that are intended to prevent damage to equipment. An example of this switch is shown in Figure 60 (Pier-Tech, Inc 2007).



Figure 59: Observatory with Roof Open and Pier Actuated



Figure 60: Observatory Internal Safety Switch (*photo credit: Pier-Tech, Inc. 2007*)

Actuated Pier

An electrically actuated pier is located in the center of the shelter. This pier can support up to 215 lbs in weight and is used to raise or lower a mounted telescope (Pier-Tech, Inc. 2007). The pier also has a dedicated position switch that completes an electrical circuit for the digital control box mentioned above. This switch ensures that the roof will only open or close when the pier is in the fully-collapsed position. The pier is operated by a simple hand box controller which only has “up” and “down” buttons. The vibration isolation system is a manufacturer-tuned suspension system. This system is meant to be mounted between the actuated pier and the telescope base, as shown in Figure 61. The pier is shown with a Meade 10” LX200GPS is shown in Figure 62



Figure 61: Pier with Vibration Isolation System



Figure 62: Telescope Mated to Pier

Shelter Modifications

After the shelter was constructed, a few design deficiencies were noted. While the manufacturer claimed to have provided parts that would fit the Meade 10" LX200GPS, this was not the case. Among the deficiencies were incorrectly matched bolt holes mounting plates on the pier; non-standardized bolts and fasteners of improper length for use in assembly; and a lower-than-expected telescope height once the pier was in the fully-extended position.

Bolt Holes

In order to correctly mount the telescope, the AFIT machine shop was able to take the existing parts and re-drill them to accommodate proper mounting. This was a simple fix and did not require manufacturer involvement.

Bolts

The shelter manufacturer included bolts for assembly; however bolts were of two different sizes. While this did not inhibit assembly, it was simplified to have the machine shop drill and tap standardized holes in the mounting plate so only one variety of bolt was required. Bolts were also found to be of improper length; these bolts would have interfered with proper assembly when the mounting plates were intended to be “sandwiched” against each other. These fixes did not require manufacturer involvement.

Telescope Height

Once the telescope was mounted to the pier, the roof was retracted and the pier was actuated into its fully-extended position. Once in position, it was noted that the centerline of the main optics (when in the horizontal position) did not crest the shelter’s walls. Also, when retracted, the peak of the roof partially obstructed the eastern sky. The wall height was not as large an issue as the roof height, since the script limits the telescope’s elevation to a minimum of 10° which is enough point over the walls. The roof blocked approximately 35° of elevation at its peak, which could result in a significant obstruction.

In order to reduce this obstruction, the machine shop was able to fabricate a “booster box.” This booster box was constructed of aluminum pylons and plates that

matched the bolt holes for the mounting plates on the pier and telescope. This raised the height of the telescope approximately six inches while on top of the pier. This reduced the roof's obstruction to approximately 20° at its peak. While more height was desirable, clearance to retract the roof was considered during the booster box's design.

Shelter Malfunction

At one point during research, the shelter's retractable roof malfunctioned. An out-of-tolerance electrical condition was causing the Ground Fault Circuit Interrupt (GFCI) breaker to open. An electrical arc was observed at the 120v AC socket that was being used to power the roof control components. The use of the worm gear made the roof un-retractable, or jammed in its closed position. Due to the opaque nature of the shelter's roof, this made viewing the sky from the shelter impossible. While a permanent repair solution was pursued, a temporary solution was employed to regain an unobstructed view of the sky.

Temporary Solution

While a worm gear is noted for its ability to gain a high gear ratio in a compact space, the self-locking attribute of screws makes it unlikely the pinion gear can be turned without disengaging the screw. Since the worm gear was housed in an enclosure (see Figure 63 (Pier-Tech, Inc. 2007)), it was a more logical course of action to disengage the pinion gear from the rack (see Figure 64, (Pier-Tech, Inc 2007)). Once the torque motor, worm gear box, and pinion were slid away from the rack, the roof could be opened and closed manually. Manual operation requires significant physical exertion, especially while closing the roof.



Figure 63: Roof Motor with Worm Gear (*photo credit: Pier-Tech, Inc 2007*)



Figure 64: Rack and Pinion (*photo credit: Pier-Tech, Inc 2007*)

Permanent Repair Solution

AFIT Laboratory Technician, Christopher Zickefoose, assisted with repairs. Referring to the shelter manufacturer for assistance, an immediate solution was not

discovered, nor was any damage noted to any electronic control or power electronic components. The shelter manufacturer replaced the digital controller, but reconnecting the device did not yield a successful repair. Later it was determined that a faulty power cable seemed to be shorting at the outlet. This cable was replaced, but this only fixed the out-of-tolerance electrical condition.

Mr. Zickefoose made further inquiries with the manufacturer and discovered that digital settings for the torque controller were incorrect. The cause of this is undetermined, though it is possible that the faulty power cable caused the digital controller to default to factory settings. Continued aid from the shelter manufacturer allowed Mr. Zickefoose to obtain correct digital settings and restore full operation to the roof's actuation drive.

Noted Observatory Location Issues

As mentioned previously AFIT's observatory provides a high degree of convenience to a researcher, but its location may detract from optimal viewing conditions. Several attempts were made to operate the script in an open-loop mode but provided no usable data regarding satellite tracking. Some factors have been identified as possible contributors to observation failures:

Light pollution: this could be a limiting factor, since objects could not be spotted in the WFOV, or in the main optics of the telescope. The observatory's close proximity to a populated area increases the probability of light pollution.

Electromagnetic interference (EMI): Power cables that run through the roof of Building 640 may be creating enough to affect the magnetic compass integrated into the

base of the Meade 10" LX200GPS telescope (Meade Instruments Corporation 2003, 21). This may contribute to misalignments in the telescope's mount.

Structural vibration: as mentioned previously, the observatory is built atop a pre-existing I-beam structure mounted on Building 640. This structure can be seen in Figure 58. Since the pier is hard-mounted directly on the structure, vibrations originating from the building could translate to inaccuracies in the telescope's angular position. In order to test the NIR camera's capabilities, a track of the moon was performed and recorded as "moon.avi." This video is 1 minute and 27 seconds in length and records the moon through the main optics. Over the course of this recording, oscillations in the image show that the telescope mount may be shaking.

Additional attempts at open-loop tracking were made; while no satellites were spotted, stars moving through the field of view generated a sinusoidal streak, further indicating that vibrations were affecting the telescope mount. These videos are known as "sines.avi", "17566.avi", and "10967.avi". They were recorded with the .5x focal reducer attached to the NIR camera.

Data Connectivity

Concept 1: Centralized Control

This concept of connectivity has been proved to be somewhat feasible. A laboratory experiment was conducted to determine if appropriate connections could be made through a network. The experiment employed a basic D-Link 4-port Ethernet router (D-Link Corporation/D-Link Systems, Inc. 2010), a Belkin Network USB Hub (Belkin International, Inc. 2010), and a control laptop. The laptop and telescope mount

were connected on the Local Area Network (LAN) ports of the router. This configuration is illustrated in Figure 65.

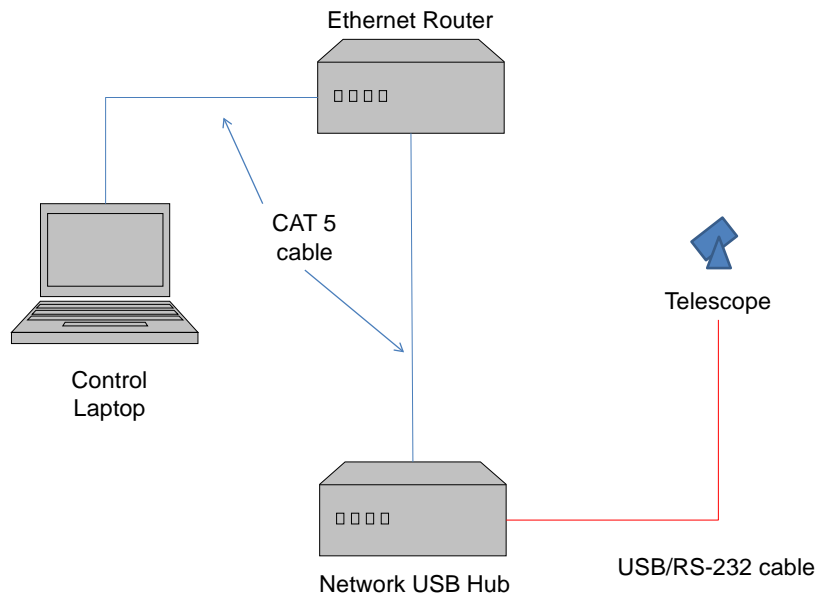


Figure 65: Telescope Network Control Test Configuration

While running the control script, a connectivity check is conducted. The script searches for USB connected devices and will display “LX200GPS telescope found on COMMXX,” where “XX” represents the USB port address for the telescope mount. When configured for network connectivity, the script detected the telescope mount and would send commands as normal; no modifications needed to be made to the script. The proprietary software included with the Network USB Hub allowed the operator to connect to any USB device connected to the hub. This configuration may require a review of the control software to see if any significant data latency is induced by the additional equipment.

This configuration has some promise in finding a way to “narrow” the number of connections to a control laptop and could be used in open-loop operation. However, it also demonstrated the Belkin Network USB Hub would not support the USB webcam needed to generate the WFOV. Similar devices were investigated, like the Digi AnywhereUSB (Digi International Inc. 2010), but still did not support the webcam connection.

During a phone conversation with engineers from Digi, it was learned that most modern webcams are “isochronal” devices. The Digi Anywhere USB and Belkin Network USB Hub devices do not support “isochronous” devices, and are meant to support “bulk transfer” devices such as printers. Isochronous devices require constant connection to the control computer in order to operate. A suitable USB-to-Ethernet device has not yet been found. If the webcam’s data could be transmitted via another medium, perhaps wirelessly, this option may be feasible. This configuration could allow the control computer to remain in a shelter away the telescope, protected from out-of-tolerance environmental conditions.

Concept 2: Distributed Control

This configuration requires far less network bandwidth and only for the purpose of delivering observation taskings and results. It does require more computers; one computer for each observation site. The cost of a capable computer would only be marginal in the installation of an observation site. An additional benefit of this configuration is that it does not require the connections to be converted to an internet protocol format; an internet connected control computer could be stored in a climate

controlled equipment closet and standard connections could be used. This eliminates the need for a USB-to-Internet device to support the USB webcam.

In order to develop this configuration for autonomous use, a few changes and augmentations would need to be made to the system. These changes are by no means exhaustive, but cover the major portions of suggested development.

First, a webpage would need to be developed to post an evening's assignments. The assignments could take the form of a basic text document that would be downloaded by each remote station prior to the observation period. This document could be singular with all telescope assignments listed within, or could be divided into several documents identified for use at specific observation sites. In order to develop this document, a linear programming (LP) model should be employed to discern the optimal viewing sites for each object to be observed. Linear programming is described as follows:

Linear programming is a powerful technique for dealing with the problem of allocating limited resources among competing activities as well as other problems having similar mathematical formulations (Hillier and Lieberman 2008, 75)

In short, this area of Operations Research (OR) allows optimization of resources in a complex task. This applies especially well to an optical surveillance network where each telescope can track only one object of interest at any one time.

Second, additional software would need to be written in order to autonomously download the evening's assignments. This need not take the form of a MATLAB script; instead it would probably be more appropriate to develop a separate program that simply downloads the text document to a specified file location to be accessed by the control software later.

Third, the control software would need to be modified to replace user input with automated inputs. Normally an operator selects specific satellites as they come into view; instead the software would need to select satellites as they are listed in the control document.

Fourth, data logging and video recording would need to be automated and organized. Current video recording is initiated via user-input through the proprietary software included with the frame grabber. Video filenames should relate to space catalog object names, dates, and times of observations. Co-located data files should record telescope azimuth/elevation values, as well as times.

Fifth, observation results would need to be delivered back to the control station, or wherever analysis is to take place. This could take the form of an automated posting to another website, or a direct file transfer.

Sixth, correlation software would need to be written in MATLAB to convert the logged angular data during observations into a new set of orbital elements. This software can compare the telescope-generated data against the radar-generated data. Once anomalies are detected, orbital maneuvers may be calculated and saved for later analysis.

Seventh, software to automate the shelter must be developed. This will allow the observation system to open the shelter and extend the telescope pier into position. After an observation period is complete, the software can retract the pier and close the roof. This function can also be tied to weather monitoring equipment to protect the station from precipitation or high winds.

Equipment Upgrades/Modifications

During the course of research, several other pieces of equipment were purchased in order to improve the operation of the satellite tracking system.

Near Infrared Camera

An Astrovid StellaCam3 Cooled Near Infrared (NIR) Camera replaced the second Panasonic SPC900NC USB webcam that had been previously used to record video through the main optics. This camera allows variable frame capture rate and adjustable gain. The camera's output is analog video via coaxial cable. A wireless controller was purchased for the StellaCam3 camera. This removed the need to connect a control cable directly to the back of the camera. Instead, this cable was replaced with a small transmitter that was included with the wireless controller (Adirondack Video Astronomy 2010). The Astrovid StellaCam3 Camera and associated control equipment are shown in Figure 66.



Figure 66: NIR Camera with Wireless Controller

While working with the StellaCam3, it was noted that the power cable was too short; once the pier was fully extended, it caused the power cord's transformer to dangle. This would add tension to the back of the telescope, and would cause the power cord to snag on the corners of the pier's mounting plates, as well as swing the transformer into the pier. To correct this, an additional four feet of cable length was spliced into the power cord, allowing the power transformer to remain stationary during telescope operation.

Frame grabber

Prior hardware configurations required a secondary laptop to record video from the main optics. An Epiphan VGA2USB LR frame grabber replaced the secondary laptop's function, allowing video from the NIR camera to be recorded via USB on the control laptop (Epiphan Systems Inc. 2009). This hardware requires a VGA input and is shown in Figure 67.



Figure 67: Epiphan VGA2USB LR Frame grabber

TV to PC Converter

Because the Epiphan VGA2USB LR frame grabber requires a VGA signal, the analog output from NIR camera needed to be converted. This was accomplished through the use of an Impact Acoustics TV to PC Converter. This device takes analog input via S-video or coaxial cable and converts it to a VGA signal which would be fed into the frame grabber (Lastar 2006). The TV to PC converter is shown in Figure 68.



Figure 68: Impact Acoustics TV to PC Converter

Portable LCD TV

During alignment activities for the telescope, it became apparent that an operator would have difficulty aligning the main optics of the telescope while the NIR camera was in place. To remedy this situation, the analog output from the NIR camera was duplicated using a common coaxial splitter. One analog feed was sent to the TV to PC Converter, while the other was connected to a Sylvania 7" LCD (Wal-Mart Stores, Inc. 2010). This LCD TV would aid the operator by providing a convenient view through the NIR camera, so alignment activities could proceed normally. The LCD TV can function

on rechargeable internal batteries or on normal 120v AC power and is shown in Figure 69.



Figure 69: Sylvania 7" LCD TV

Internet Router

A D-Link 4-Port internet router was used to test connectivity to the telescope via Ethernet cable. It is a standard 10/100 Ethernet router (D-Link Corporation/D-Link Systems, Inc. 2010) and is shown in Figure 70.



Figure 70: D-Link 4-port Internet Router

Autostar Controller Extension Cord

The Meade Autostar Controller included with the Meade 10" LX200GPS Telescope allows the operator to perform set up activities, such as alignment (Meade Instruments Corporation 2003, 22-23). The cable included with the controller is approximately 10" long and is coiled. The short length and coils of the cable can cause tangles during observation periods, so the cable was replaced with a cable fabricated by Mr. Zickefoose. The replacement is approximately 10' in length, is uncoiled, and allows the controller to be moved away from the telescope while still performing the same function.

Carrying Cases

When the observatory was constructed, the most unwieldy component of the observation system, the telescope, could be left affixed to the pier. The observatory is not climate controlled and it is best not to leave the sensitive electronic components in uncontrolled conditions. These components could be carried in a normal cardboard box, but a more convenient and organized method was pursued to allow easy portability of smaller parts. Two Pelican 1520 Cases were employed to carry all components as depicted in Figure 71 and Figure 72. The interior dimensions of these cases measure 18.06" x 12.89" x 6.72" and they are water tight and impact resistant (Pelican Products, Inc. 2010).



Figure 71: Pelican 1520 Cases with Equipment Deployed



Figure 72: Pelican 1520 Case with Equipment Stowed

Investigative Questions Answered

Closed-Loop Tracking

What sources of feedback are available for a closed-loop control system? In its current configuration, only two sources of feedback are available: telescope angular position, and video input from the USB webcam.

Once feedback is determined, how does it relate to the commands sent to the telescope and ultimately the angles generated? The telescope angular data is already used in order to compute the required positions in a dynamic environment based upon downloaded TLE data. The USB webcam video data can be used to determine small angular offsets to provide feedback to the control system.

Remote Operation

What hardware is required to develop a remotely operated tracking system? Two areas of research investigated necessary components to answer this question: automated shelters and data connection hardware. Any remotely operated observation site will need a shelter that can protect the optical and electronic equipment inside, but it needs to be automated in order to achieve this goal. Data connectivity hardware requirements depend almost solely on the control model developed.

What methods of data connectivity can be used to enable autonomy in a remotely operated tracking system? If telescope control is to be centralized, sophisticated hardware needs to be researched in order to send commands and data via a network. If control is to be distributed, only a basic network connection needs to exist at the observation site.

Summary

Closed-Loop Tracking

Successful development of logic routines showed that a centroid could be tracked in recorded videos depicting “worst case” scenarios. The real-time tracking version proved that the angular offset data could be generated in approximately 0.1 seconds, which is well below the 0.5 seconds necessary to iterate through one control loop in the GUI.

Remote Operation

AFIT’s rooftop observatory comes with advantages and disadvantages. Notable advantages include a permanent structure to house observation equipment, a convenient location for research, and a fully functioning test bed to provide a “formula” for a remotely operated tracking station. While poor viewing conditions from light pollution and weather may limit satellite visibility, the observatory will allow researchers to document requirements and test systems as an observation network concept is developed.

Data connectivity experiments show that some functions of control for the satellite tracking system can be achieved remotely. Choice of network control concepts will mandate the type of equipment needed to satisfy the goal of developing a remotely operated autonomous tracking system.

The next chapter will summarize research efforts as well as make recommendations for future research and development areas.

V. Conclusions and Recommendations

Chapter Overview

This chapter will summarize the results of research contributions to the satellite tracking telescope system.

Conclusions of Research

Closed-Loop Tracking

The satellite tracking telescope has already demonstrated some merit as an open-loop control system capable of observing satellites in the WFOV. Research into closed-loop control has furthered the possibility of accurately controlling the telescope tracking capabilities by providing robust logic for target tracking. In addition to target tracking logic, real-time software is able to generate optical feedback for angular offset data faster than the open-loop control script updates. While this requires an operator to initiate the tracking routine, it is evident that the script is capable of performing its function fast enough to provide usable feedback for use in closed-loop control.

Remote Operation

In an effort to develop a remotely operated tracking system, two areas of research were explored: automated shelters and data connectivity. A roll-off roof observatory was ultimately selected and constructed on top of AFIT's Building 640. This observatory provides a substantial benefit to the institution's pursuit of developing a remotely operated tracking station. The data connectivity trials conducted in Chapter IV demonstrated the feasibility of controlling some aspects of the satellite tracking system, but also outlined the advantages and disadvantages of two network control concepts.

Significance of Research

A closed-loop control system for satellite tracking telescope contributes in two significant areas: autonomy and accuracy. In an open-loop system, an operator can technically provide feedback by making manual adjustments in the GUI, but these adjustments are generally not precise enough to move a target within the main optics FOV. By developing a machine operated closed-loop control system, the operator can be taken out of the loop while allowing for further development of a remote, unmanned tracking system.

Recommendations for Action

Develop GUI controls to allow an operator to “seed” the centroid generation routine. This will simultaneously activate the optical tracking routine and initialize a centroid history around an operator input. It will also allow the tracking software to begin in the presence of background clutter, since the operator’s input will determine the first location of the centroid.

Upgrade the GUI to allow an operator to select a configuration of equipment to change pixel-to-angle conversions, rather than having to manually edit script information. This could also include the capability to apply changes to different sizes of telescopes so the same software can be employed on any piece of equipment as documented.

Develop a GUI to emulate the hand controller for the Meade telescope. This will allow for a simplification of equipment configuration, further narrowing the number of connections to the telescope system, and advancing the steps in developing a remotely

operated station. A recommended capability would be to use the centroid generation routine to automatically adjust telescope position during alignment procedures.

Recommendations for Future Research

Networked Telescopes

In order to achieve an autonomous tracking station model, further development is required to coordinate the actions of two or more telescopes. See “Data Connectivity” in Chapter III for more information on this topic as well as proposed control architectures.

Research is also required to determine if any significant command latency is induced by controlling the telescope through an Ethernet/USB-based connection as detailed in Chapter IV.

Mobility

While many upgrades to the research facility at AFIT have been made, more consideration for mobility should be applied. The current research facility provides an excellent testbed to develop a model for a static structure; it does not provide a perfect viewing conditions. Development in this area will allow students to more easily travel to locations with better viewing conditions and also contribute to thoughts for operational deployment. See “Appendix I: Formula for Mobility” for more information.

Improvements in Accuracy

In order to increase the reliability of the closed-loop tracking routine, some suggested improvements in accuracy can come from filtering feedback data. Methods can employ star map data or optical filters to reduce light from sources not associated

with reflected light from the sun. See “Appendix II: Possible improvement in closed-loop tracking” for more information on this topic.

Summary

In order to develop an autonomous remotely operated satellite tracking system, efforts were made to convert the open-loop control system to a closed-loop control system. Research was also conducted into data connectivity issues and permanent, automated shelters to be used as a model for remote facilities.

The focus of closed-loop control research was to take readily available video data and develop a feedback loop to correct the telescope’s position. While image processing techniques were simple to develop, the majority of research went towards targeting logic in order to maintain tracking in worst-case scenarios. Closed-loop feedback using this logic in a real-time scenario demonstrated the feasibility of developing a fully-closed loop tracking system.

An automated shelter was constructed to further AFIT’s research efforts. As development continues in networking capability and further “fine tuning” of closed-loop control occurs a truly autonomous remotely operated satellite tracking system is indeed possible.

Appendix I: Formula for Mobility

Formula for mobility

Permanent installations offer the advantages of full time operation, but they may not be in the most advantageous areas to observe specific events in low-earth orbit (e.g., the docking of two spacecraft). An event that occurs in the “blindspot” of an optical space surveillance network may still be observed given enough warning. In order to develop a mobile optical observation station, three primary factors must be considered: power availability, portability, and internet access.

Power consideration

All components in the current configuration will run on 120v AC power. While commercial power may be available in most parts of the United States, it should be noted that a given event may not be available from inside the CONUS. Considering rural/remote locations or countries that may not have available/reliable commercial power, a portable power supply must be deployed with the observation system. Many options exist to fulfill this requirement: vehicle power with AC inverters, portable gas-powered generators, battery banks, or alternative energy solutions.

Power supply choice should be tailored given the length of a mission, terrain, and the ability to resupply. An accessible observation site that is road-accessible and near a fueling station would be ideal for vehicle power. However, a site that is not road-accessible, but still near a fueling station might consider using a gas-powered generator, such as those manufactured by companies such as Honda. An example of a gas generator is depicted in Figure 73 (American Honda Motor Co., Inc. 2009). If an observation site is

to be employed for several weeks at a time, alternative energy solutions should be considered.



Figure 73: Commercial Gas Powered Generator (*photo credit: American Honda Motor Co., Inc. 2009*)

Commercially available solar and wind systems may provide the necessary power to fully sustain a longer-term observation mission. One option already being employed by the US military is the SolarStik. According to the United States Army Acquisition Support Center (USAASC) website:

The Solar Stik is a tripod system with a pair of 50-watt rigid-panel solar arrays used to capture solar energy. The tripod can also be outfitted with an optional wind turbine that is capable of producing up to an additional 200 watts of power.

Setting up the lightweight, easily transportable system is simple. The first step is identifying the sun's location in the setup area. Next, the user aligns the system's mast to bring in a maximum amount of sunlight. These masts can be redirected to pull in the maximum amount of solar energy. "Within minutes you can have the

system up, connected, and pulling in power,” Richard said. “They can probably have these systems easily set up much faster than they would all of the communications gear that is in the CP tent. So it’s quick, it’s fast, and it’s easy to set up.” (Davidson 2009)

The SolarStik is a commercially-produced, portable power system that provides solar power and optional wind power and also includes battery packs for night-time use.

While solar power does diminish with cloud cover, a temporary observation site should be chosen based on its probability of clear skies for observation; in most cases this would translate to a higher likelihood of solar power usage. Figure 74 depicts a SolarStik deployed for power generation (Solar Stik 2009)



Figure 74: Alternative energy solution (*photo credit: Solar Stik 2009*)

Portability

Beyond the power supply's portability, the selected telescope itself must be portable. The Meade 16" telescope is not an ideal choice for portability unless it is transported by truck or aircraft. It is possible that the telescope could be emplaced on a truck bed, but it may suffer from significant vibrations if the vehicle itself is fulfilling the power requirements (i.e., engine operation). The Meade 10" telescope would be a better choice since its assembled weight is approximately one-third of that of the 16" telescope

(approximately 100 pounds versus 300 pounds). Additional equipment (cameras, cables, monitor, etc.) can be hand carried in two Pelican briefcases. The telescope itself should be carried in its manufacturer-supplied case. Given the telescope's packed weight and possibly unforgiving terrain, all-terrain vehicles (ATVs) or pack animals should be used to carry this equipment.

Internet access

As noted previously, the telescope requires access to a TLE database of some sort. Current operation requires download via the internet to a portable storage medium, then transfer to the control laptop. In order to ensure the most up-to-date information is available, only brief access to the internet is required. If access can be gained before departing for a nightly observation, then internet availability is not an issue. However, LAN/WAN access are not likely to be available in a remote area, so alternatives should be developed. Commercial Broadband internet access, such as Verizon's Mobile Broadband, is an obvious solution, using commercial datalinks to obtain access to the worldwide web but depending on the location this may not be an option. Therefore, a satellite communication system should be considered. Military SATCOM networks may be available, but high-priority operational requirements could inhibit access. Commercial examples include the Globalstar or Iridium networks which offer satellite-based data services in almost all locations. An example of a commercial satellite data device is depicted in Figure 75 (Outfitter Satellite Phones 2009).



Figure 75: Iridium Satellite Data Modem (*photo credit: Outfitter Satellite Phones 2009*)

If observation data is required immediately, results can be transmitted via the same mobile data link to a collection site.

Miscellaneous consideration

Due to the necessarily outdoor nature of a portability formula, inclement weather in a location should also be noted. Precipitation can be hazardous to equipment lifetime, as can a dust storm in a desert operation. Excessive sun exposure could heat equipment to performance degradation or failure, and thermal effects can contribute to errors in optical tracking ability. If weather is an issue, different shelters could be considered ranging from tarps and tents to more rigid structures. Depending on application, a shelter of some design should be considered in any portable system. Shelters could also be

tailored for camouflage if a station is to operate clandestinely; camouflage netting would be ideal for this purpose.

Appendix II: Possible improvement in closed-loop tracking

Given a clear night and low levels of light pollution, the star field may represent the largest obstacle to closed-loop tracking. Starlight typically provides more light than a reflective object and when processed through a CCD device can cause the current centroid-generation routine to reject information from a frame. A solution may come from using an accurate star map.

Each frame generates a matrix of values corresponding to colors and intensities as they are incident on the focal plane array of the imaging device. Each element of the matrix represents an image pixel. This information is ultimately filtered and reduced to a matrix of ones and zeros, ones representing a significant level of light, zero representing insignificant value. Stars in the field of view of the sensor will generate ones unless their intensities are below a specified threshold, in which case they will be represented by zeros. By referencing a star map, the current angles and location of the telescope, current time, and the field of view of the sensor, the stars might be digitally “edited” out of the frame so that they do not contribute to the centroid processing. This would allow for fewer rejected frames, but would probably require a significant amount of processing power. The load on a processor might be reduced if a separate piece of hardware is employed, like a star sensor. The input from the star sensor could be used to immediately subtract star data from a frame, if the star sensor and imaging sensor could be properly synchronized. Synchronization may not be needed if the star sensor and imaging sensor share the same aperture.

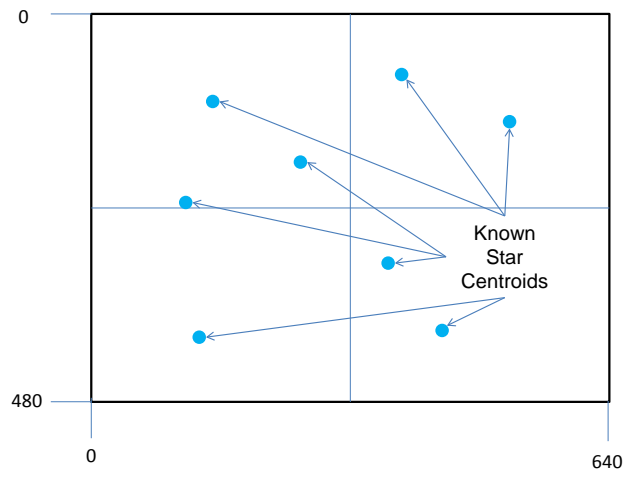


Figure 76: Stars in WFOV

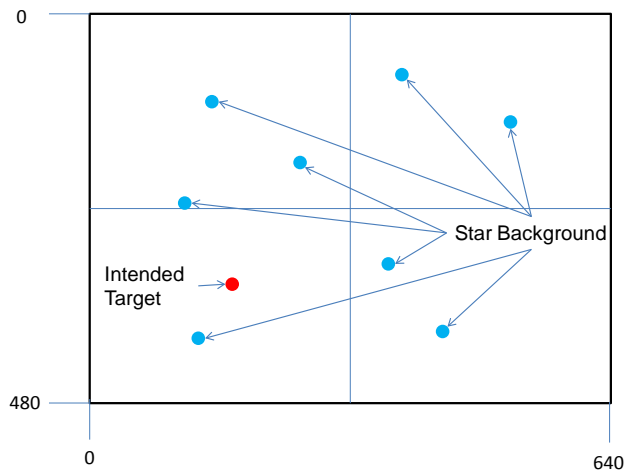


Figure 77: Stars and Intended Target in WFOV

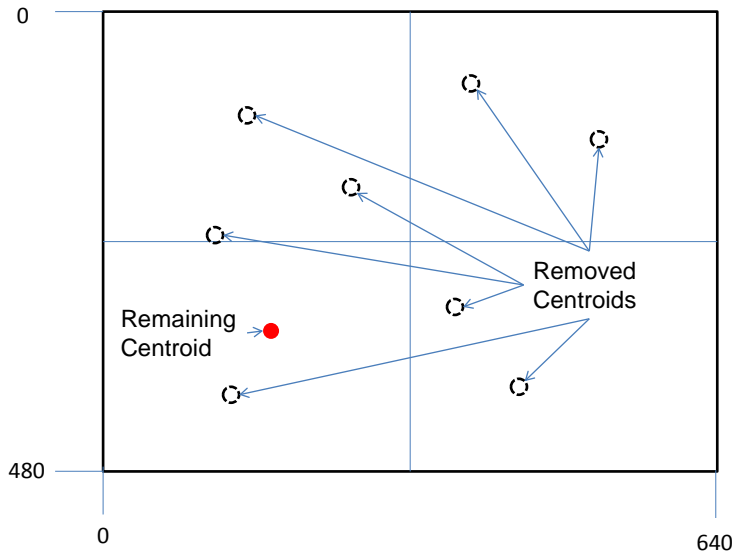


Figure 78: Stars subtracted from WFOV

Another method of reducing star light influence in centroid calculation may come from employment of an optical filter. Viewing objects in low-earth orbit requires the reflected light from the sun. Use of a filter that passes a narrow band of visible wavelengths that is characteristic to the sun's radiation may serve to reduce the intensity of a star field background and requires no digital processing at all. This increases the overall contrast of an object within the imaging sensor's field of view, allowing for more reliable calculation of a frame's centroid. Conversely, the reduction of light into the USB webcam may make it insufficient for use. The availability of such a filter would require additional research.

Appendix III: Test MATLAB Code

%This function is a test function. It takes a recorded video object in and produces a track of each frame's centroid successively. The 'sweetspot' variable must be loaded from the collimation_results.mat file prior to operation, or the sweetspot could be modified to [240 320] to represent the center of a 640x480 frame.

```
function frame_centroid_test (framesin)
A = [319.95:.01:320.04]';
B = [239.95:.01:240.04]';
centroid_history = [A B];

global sweetspot;
sweetspot = [320 240];
track_on = 0;

for i = 1:size(framesin,2)
    bw_sample = im2bw(framesin(1,i).cdata(:,1),.7);
    lab = bwlabel(bw_sample);
    property = regionprops(lab,'Centroid');
    if size(property,1) < 1 %failsafe for no centroid in a frame
        if track_on == 1
            centroid_velocity = ((centroid_history(size(centroid_history,1),:)-
centroid_history(size(centroid_history,1)-4,:))/4); %Develop a velocity of centroid over
last 4 frames
            if norm(centroid_velocity(1)) > 2 % Limit the row velocity
                centroid_velocity(1) = norm(centroid_velocity(1))*2/(centroid_velocity(1));
            end
            if norm(centroid_velocity(2)) > 2 % Limit the column velocity
                centroid_velocity(2) = norm(centroid_velocity(2))*2/(centroid_velocity(2));
            end
            centroid(i,:) = centroid_history(size(centroid_history,1),:) + centroid_velocity;
        else
            centroid(i,:) = sweetspot;
        end
    elseif size(property,1) > 1 %failsafe for more than one centroid in a frame
        for n = 1:size(property,1) %Develop array of all centroids calculated
            centroid_array(n,:) = property(n).Centroid;
        end
        for j = 1:size(centroid_array,1) %Scan all centroids and determine their distance
from last centroid
            distance(j) = norm(centroid_array(j,:)-
(centroid_history(size((centroid_history),1),:)));
```

```

        distance = distance';
    end
    [min_val, min_row] = min((distance)); %Select index of centroid with shortest
distance
    centroid(i,:) = centroid_array(min_row,:); %New centroid equals closest centroid
    if track_on == 1 %If a real centroid has been detected (hopefully the target), then the
following happens
        if norm((centroid(i,:)-(centroid_history(size((centroid_history),1),:)))) > 70
%Reject centroids over 70 pixels from last computed
            centroid_velocity = ((centroid_history(size(centroid_history,1),:)-
centroid_history(size(centroid_history,1)-4,:))/4); %Develop a velocity of centroid over
last 4 frames
                if norm(centroid_velocity(1)) > 2 % Limit the row velocity
                    centroid_velocity(1) = norm(centroid_velocity(1))*2/(centroid_velocity(1));
                end
                if norm(centroid_velocity(2)) > 2 % Limit the column velocity
                    centroid_velocity(2) = norm(centroid_velocity(2))*2/(centroid_velocity(2));
                end
                centroid(i,:) = centroid_history(size(centroid_history,1),:) + centroid_velocity;
            end
        end
        track_on = 1; %Set tracking mode on since a real centroid has been detected
    else %Easy case of one centroid. Consider cutting this if the above can be >= 1
centroid
        next_centroid = property.Centroid;
        centroid(i,:) = [next_centroid];
        if track_on == 1
            if norm((centroid(i,:)-(centroid_history(size((centroid_history),1),:)))) > 70
                centroid_velocity = ((centroid_history(size(centroid_history,1),:)-
centroid_history(size(centroid_history,1)-4,:))/4);
                if norm(centroid_velocity(1)) > 2
                    centroid_velocity(1) = norm(centroid_velocity(1))*2/(centroid_velocity(1));
                end
                if norm(centroid_velocity(2)) > 2
                    centroid_velocity(2) = norm(centroid_velocity(2))*2/(centroid_velocity(2));
                end
                centroid(i,:) = centroid_history(size(centroid_history,1),:) + centroid_velocity;
            end
        end
        track_on = 1;
    end

%Limit protection
if centroid(i,1) > 640

```

```

        centroid(i,1) = 640;
    end

    if centroid(i,1) < 0
        centroid(i,1) = 0;
    end

    if centroid(i,2) > 640
        centroid(i,2) = 640;
    end

    if centroid(i,2) < 0
        centroid(i,2) = 0;
    end

    % Update the centroid history
    A_prime = A(2:size(centroid_history));
    B_prime = B(2:size(centroid_history));
    centroid_history(1:size(centroid_history)-1,1) = A_prime;
    centroid_history(1:size(centroid_history)-1,2) = B_prime;
    centroid_history(size(centroid_history,1),:) = centroid(i,:);
    A = centroid_history(:,1);
    B = centroid_history(:,2);

    imshow(framesin(1,i).cdata);
    hold on
    plot(centroid(i,1),centroid(i,2),'ws','MarkerSize',18);
    line([sweetspot(1) centroid(i,1)],[sweetspot(2) centroid(i,2)]);
    drawnow();
    hold off
end

for n = 1:size(centroid,1)
    error_matrix(n,:) = centroid(n,:) - sweetspot;
end

% Create figure
figure2 = figure;

% Create axes
axes1 = axes('Parent',figure2);
box('on');
hold('all');

```

```

plot1 = plot(error_matrix,'Marker','*');
set(plot1(1),'DisplayName','Vertical Error','Color',[1 0 0]);
set(plot1(2),'DisplayName','Horizontal Error','LineStyle',':','...
    'Color',[0 0 1]);

% Create xlabel
xlabel('Frame Number');

% Create ylabel
ylabel('Pixels');

% Create legend
legend(axes1,'show');
set(legend,'Location','Best');

% Create title
title('testvid8.avi');

for m = 2:size(centroid,1)
    delta_position(m-1,:) = centroid(m,:) - centroid(m-1,:);
    delta_position_norm(m-1,:) = norm(delta_position(m-1,:));
end

figure3 = figure;

% Create axes
axes('Parent',figure3);
box('on');
hold('all');

% Create plot
plot(delta_position_norm,'Color',[1 0 0]);

% Create xlabel
xlabel('Frame Number');

% Create ylabel
ylabel('Pixels');
plot(delta_position_norm,'r');

% Create title
title('testvid8.avi');
%line([0 200],[.075 .075],'LineStyle','--','Color',[0 0 1]);

```

Appendix IV: Real-Time MATLAB Code

begin_track.m

%This script initializes an optical track. In order to initiate, type
%begin_track and press enter. To terminate, type end_track

tic

```
global sweetspot  
global centroid_history  
global angular_offset  
global track_on  
global offset_history
```

```
%angular_offset = [0 0]; %[azimuth elevation]  
%offset_history = [angular_offset,0]; %[azimuth offset time]  
%sweetspot = [320 240]; %[azimuth elevation]  
%track_on = 0;  
a = toc;
```

```
centroid_history = [(sweetspot(1)-.5:.1:sweetspot(1)+.4),(sweetspot(2)-  
.5:.1:sweetspot(2)+.4),(a-.0005:.0001:a+.0004)'];
```

```
% vidobj = videoinput('winvideo',1,'I420_640x480');  
% set(vidobj,'ReturnedColorSpace','rgb');  
% triggerconfig(vidobj,'manual');  
% start(vidobj)  
% preview(vidobj)
```

```
cameratimer = timer('Period',.1); %Creates cameratimer on .1 sec repeat.  
set(cameratimer,'TimerFcn',{'frame_centroid_cb',vidobj}); %Calls frame_centroid_cb as  
a callback function  
set(cameratimer,'Name','cameratimer'); %Names the timer 'cameratimer'  
set(cameratimer,'ExecutionMode','FixedRate'); %Sets execution mode to fixed rate  
set(cameratimer,'TasksToExecute',inf);  
set(cameratimer,'StartFcn','disp("Started optical track. Creating cameratimer.")');  
start(cameratimer);  
disp('optical track started...');
```

frame_centroid_cb.m

```
function frame_centroid_cb(cameratimer, event, obj)
global centroid_history;
global centroid;
global sweetspot;
global track_on;

%This function gets a snapshot from the USB camera as often as it is
%called. It will generate a single centroid array which is composed of a
%row, a column, and a toc value. The toc value is only used to make first
%order velocity-based approximations for position if a centroid isn't found
%or if detected centroids are unreasonably displaced from the last position
%listed in the centroid_history. centroid_temp is used until the function
%is finished to keep the global centroid variable from being updated with
%unfiltered information before the function is complete with it's
%iteration.

%Get snapshot from camera and log CPU time using toc.
sample_frame = struct('frame',peekdata(obj,1),'frame_time',toc);

%Convert image to binary
bw_sample = im2bw(sample_frame.frame);

%Label blobs
lab = bwlabel(bw_sample);

%Calculate centroids of each blob
property = regionprops(lab,'Centroid');

%If no centroid is detected in a given frame
if size(property,1) < 1
    %If a centroid had previously been detected but no centroid is found
    if track_on == 1
        centroid_temp = centroid_approx(centroid_history,sample_frame.frame_time);
    %If a centroid has not yet been detected
    else
        centroid_temp = [sweetspot,toc];
    end
%If more than one centroid is detected in a given frame
elseif size(property,1) > 1
    %Develop array of all centroids calculated
    for n = 1:size(property,1)
```

```

        centroid_array(n,:) = property(n).Centroid;
    end
    %Scan all centroids and determine their distance from last centroid
    for j = 1:size(centroid_array,1)
        distance(j) = norm(centroid_array(j,:)-
(centroid_history(size((centroid_history),1),1:2)));
        distance = distance';
    end
    %Select index of centroid with shortest distance
    [min_val, min_row] = min((distance));
    %New centroid equals closest centroid
    centroid_temp = [centroid_array(min_row,:),sample_frame.frame_time];
    %If a real centroid has been detected (hopefully the target)
    if track_on == 1
        %Reject centroids over 70 pixels from last computed
        if norm((centroid_temp-(centroid_history(size((centroid_history),1),1:2)))) > 70
            centroid_temp = centroid_approx(centroid_history,sample_frame.frame_time);
        end
    end
    %Set tracking mode on since a real centroid has been detected
    track_on = 1;
    %Easy case of one centroid. Consider cutting this if the above can be >= 1 centroid
    else
        next_centroid = property.Centroid;
        %Log frame_time to centroid
        centroid_temp = [next_centroid,sample_frame.frame_time];
        if track_on == 1
            %Reject centroids over 70 pixels from last computed
            if norm((centroid_temp(1:2)-(centroid_history(size((centroid_history),1),1:2)))) >
70
                centroid_temp = centroid_approx(centroid_history,sample_frame.frame_time);
            end
        end
        track_on = 1;
    end

    %Limit protection
    if centroid_temp(1) > 480
        centroid_temp(1) = 480;
    end

    if centroid_temp(1) < 0
        centroid_temp(1) = 0;
    end
end

```

```

if centroid_temp(2) > 640
    centroid_temp(2) = 640;
end

if centroid_temp(2) < 0
    centroid_temp(2) = 0;
end

centroid = centroid_temp;

%Update the centroid history
centroid_history = [centroid_history(2:size(centroid_history));centroid];

```

frame_centroid.m

```

function centroid_array = frame_centroid(frame_in)

%This function is for use during optical tracking. It takes a video frame
%and converts it to binary, using a .6 threshold. A list of centroids is
%generated if any are found.

frame_bw = im2bw(frame_in,.6);

frame_bw_label = bwlabel(frame_bw);

centroid_array = regionprops(frame_bw_label,'Centroid');

end

```

closest_centroid.m

```

function centroid_out = closest_centroid(centroids_in,centroid_history)

%This function is for use during optical tracking. It takes an array of
%centroids as well as a centroid history and determines the closest
%centroid to the last centroid computed in the history.

%disp('closest_centroid');

last_centroid = centroid_history(size(centroid_history,1),1:2);

```

```

for j = 1:size(centroids_in,1) %Scan all centroids and determine their distance from last
centroid
    distance(j) = norm(centroids_in(j,:)-(last_centroid));
    distance = distance';
end

[min_val, min_row] = min(distance); %Select index of centroid with shortest distance
centroid_out = centroids_in(min_row,:); %New centroid equals closest centroid
end

```

velocity_approximation.m

```

function [centroid_approximation] = velocity_approximation(centroid_history,time_in)

%This function is for use during optical tracking. In the event that no
%centroid or an out-of-tolerance centroid is detected, a first order
%velocity approximation is used.

time_vector = centroid_history(:,3);
position_vector = centroid_history(:,1:2);
velocity_limit = 900;

%disp('centroid_approximation');

for i = 6:(size(time_vector)-1)
    velocity_vector(i-5,:) = (position_vector(i+1,:)-position_vector(i,:))/(time_vector(i+1)-
time_vector(i)); %#ok<AGROW>
end

average_velocity = [mean(velocity_vector(:,1)),mean(velocity_vector(:,2))];

if norm(average_velocity(1)) > velocity_limit
    average_velocity(1) = sign(average_velocity(1))*velocity_limit;
end

if norm(average_velocity(2)) > velocity_limit
    average_velocity(2) = sign(average_velocity(2))*velocity_limit;
end

delta_t = time_in - time_vector(10);
centroid_approximation = [average_velocity(1)*delta_t,average_velocity(2)*delta_t];

end

```

update_history.m

```
function history_out = update_history(latest_centroid,centroid_history)

%This function is for use during optical tracking. It updates the centroid
%history by deleting the oldest data and adding the latest data.

history_out = [centroid_history(2:size(centroid_history,1),:);latest_centroid];

end
```

end_track.m

```
%This function terminates the optical tracking routine and resets
%parameters for re-initialization later. To end optical tracking type
%end_track and press enter.
```

```
stop(cameratimer)
delete(cameratimer)
```

```
tic
```

```
global sweetspot
global centroid_history
global angular_offset
global track_on
```

```
angular_offset = [0 0];
sweetspot = [240 320];
a = toc;
track_on = 0;
```

```
centroid_history = [(sweetspot(1)-.5:.1:sweetspot(1)+.4)',(sweetspot(2)-
.5:.1:sweetspot(2)+.4)',(a-.0005:.0001:a+.0004)'];
```

collimation.m

```
clear all
close all
clc
%Create camera object
camera = videoinput('winvideo',1,'I420_640x480');

triggerconfig(camera,'manual');
```

```

set(camera,'FramesPerTrigger',1);
set(camera,'TriggerFrameDelay',10);
set(camera,'ReturnedColorSpace','rgb');
start(camera);
trigger(camera);
frame = getdata(camera);
image(frame);

stop(camera);
delete(camera);
clear camera;

sweetspot = ginput(1);
%
sweetspot = [sweetspot(1), sweetspot(2)];
outfile = 'collimation_results.mat';

    save(outfile,'sweetspot');

    fprintf('Results saved to %s.\n',outfile);

```

seeker_trackgui_v2.m excerpt

```

delta_az = 0; %initialize
delta_el = 0;
azreport = [];
elreport = [];
travelrates = [];
aa = 1;
bb = 1;
nn = 1;
scheduleslider = [];
pass_start = [];
pass_end = [];
pp_az = [];
pp_azrate = [];
pp_el = [];
pp_elrate = [];
pp_range = [];
backdelay = 0.7/86400;
outdelay = 0.26/86400;
backtoave = 10;
period = 0.5;
test = [0 0];

```

```
test_lead = [];  
global angular_offset;  
global track_on;  
global sweetspot;  
angular_offset = [0 0];  
track_on = 0;
```

```
'target_az = lead_sat_az + delta_az + angular_offset(1);'...%this is the modified  
position of the sat at the *next* period
```

```
'target_el = lead_sat_el + delta_el + angular_offset(2);'...
```

Appendix V: Instructions for Use

- Step 1: Download latest TLE data from www.space-track.org
- Step 1a: On an internet-connected computer, go to <http://www.space-track.org/perl/login.pl>
- Step 1b: Enter log-in
- Step 1c: Enter password
- Step 1d: Click "Submit"
- Step 1e: Under "Two Line Element Set (TLE) Data," click on "Bulk Catalog Data Downloads"
- Step 1f: Under "Full Satellite Catalog," click on "Three-Line Format (includes object names)"
- Step 1g: Save catalog file to a known location
- Step 1h: Open file location
- Step 1i: Extract TLE text file to a known location
- Step 1j: Transfer TLE text file to transferrable media (CD, DVD, etc.)
- Step 1k: Move TLE text file to control computer and place in same directory as pre-calculation script.
- Step 2: Perform pre-calculations
- Step 2a: In the "Seeker Precalc and Tracking" directory, click on "seeker_precalcs_v2.m"
- Step 2b: Run the "seeker_precalcs_v2.m" script
- Step 2c: In the command line, enter geodetic latitude or press enter to accept displayed default value
- Step 2d: In the command line, enter geodetic longitude or press enter to accept displayed default value
- Step 2e: In the command line, enter GPS altitude (meters) or press enter to accept displayed default value
- Step 2f: In the command line, enter geoid height (meters) or press enter to accept displayed default value
- Step 2g: In the command line, enter day offset (for simulation purposes) or press enter to

- accept displayed default value
- Step 2h: Allow program to make calculations (progress line will be displayed); wait for "Choose a TLE File" GUI to appear
- Step 2i: In the "Choose a TLE File" GUI, select latest TLE file (the file that was transferred in Step 1). An earlier file can be used, however this information will not provide most recent radar data from NORAD.
- Step 2j: In the command line, input maximum satellite period to consider (minutes) or press enter to accept displayed value
- Step 2k: In the command line, input minimum satellite brightness to consider or press enter to accept displayed value
- Step 2l: In the command line, input elevation angle threshold (degrees) or press enter to accept displayed default value
- Step 2m: Allow program to make calculations (progress line will be displayed); wait for "Results saved to precalc_results.mat" to appear.
- Step 3: Open control GUI
- Step 3a: In the "Seeker Precalc and Tracking" directory, select "seeker_trackgui_v2.m"
- Step 3b: Run "seeker_trackgui_v2.m" in MATLAB
- Step 3c: In the command line, enter "1" to simulate a start time, or press enter to accept displayed default value. Note: a simulated start time will allow the telescope to run in a simulation mode, and makes calculations as if the sun has set
- Step 3d: In the command line, ensure that a webcam and telescope are both detected
- Step 3e: In the GUI, select "Sync" in order to synchronize the control computer with the GPS input from the telescope's control mount
- Step 3f: In the GUI, in the precalculated results area under the star map, select desired satellite to observe
- Step 3g: Select "Track" to begin open-loop telescope control; control will highlight red
- Step 4: Enable optical feedback
- Step 4a: Wait until desired satellite is in the webcam's WFOV and free from background clutter
- Step 4b: In the command line, type "begin_track" and press enter

- Step 4c: In the GUI, ensure that the object is being moved into the NFOV
- Step 4d: On the portable LCD display, ensure object is within view
- Step 5: Record main optics video
 - Step 5a: Open Epiphan "VGA2USB" recording software
 - Step 5b: Observe object in the preview window
 - Step 5c: Adjust main optics video settings
 - Step 5d: On the Stellacam3 wired or wireless controller, select desired frame rate; high frame rates allow smoother recording, while long exposures increase image brightness
 - Step 5e: On the Stellacam3 wired or wireless controller, select desired gain value; high gain values increase overall brightness of the image but can saturate the recorded images
 - Step 5f: Click "Record" button at the top of the preview window
 - Step 5g: Enter name of file to be recorded (.avi format)
 - Step 5h: Select "Save"
 - Step 5i: Allow recording to proceed until satisfied; end recording by clicking "Record" again
- Step 6: Disable optical feedback
 - Step 6a: Wait until optical feedback is no longer required, or object fails to be visible within WFOV
 - Step 6b: In the command line, type "end_track"
- Step 7: End open-loop control
 - Step 7a: In the control GUI, de-select "Track." Control will no longer be highlighted.

Bibliography

Adirondack Video Astronomy. "ASTROVID StellaCam3." *ASTROVID StellaCam3*. Hudson Falls, NY: Adirondack Video Astronomy, 2010.

Air Force Research Laboratory Directed Energy Directorate. *STARFIRE OPTICAL RANGE AT KIRTLAND AIR FORCE BASE, NEW MEXICO*. March 23, 2009. <http://www.kirtland.af.mil/library/factsheets/factsheet.asp?id=15868>.

American Honda Motor Co., Inc. *Honda Power Equipment - EU1000I*. American Honda Motor Co., Inc. 2009. <http://www.hondapowerequipment.com/products/modeldetail.aspx?page=modeldetail§ion=P2GG&modelname=EU1000I&modelid=EU1000IAN>.

AstroHaven Enterprises. *AstroHaven - Gallery*. 2008. http://siteground186.com/~astrohav/index.php?option=com_content&task=view&id=22&Itemid=20.

Belkin International, Inc. *Belkin: Network USB Hub*. Belkin International, Inc. 2010.

Boas, Mary L. *Mathematical Methods in the Physical Sciences*. Hoboken: John Wiley & Sons, Inc., 2006.

Candomes Observatories Ltd. *Dome Picture 2*. 2008. <http://www.candomes.com/domepic02.htm>.

Davidson, Joshua. *U.S. Military Seeks "Green" Energy Solutions*. September 2009. http://www.usaasc.info/alt_online/article.cfm?iId=0909&aid=01.

Digi International Inc. *AnywhereUSB - Network-Enabled USB Hub - USB over IP - Digi International*. Digi International Inc. 2010. <http://www.digi.com/products/usb/anywhereusb.jsp#overview>.

D-Link Corporation/D-Link Systems, Inc. *D-Link Ethernet Broadband Router*. D-Link Corporation/D-Link Systems, Inc. 2010. <http://www.dlink.com/products/?pid=478>.

Epiphan Systems Inc. "Epiphan Frame Grabber User Guide." Vol. Version 3.20.2 (Windows). Epiphan Systems Inc., 2009.

Hillier, Frederick S, and Gerald J Lieberman. *Introduction to Operations Research*. New York: McGraw-Hill, 2008.

ISS-Tracking. *ISS-Tracking*. February 22, 2007. <http://www.iss-tracking.de/images/stationpic.html>.

Lastar. *TV to PC Converter*. 2006.

http://www.impactacoustics.com/product.asp?cat_id=1007&sku=40971.

Meade Instruments Corporation. "Instruction Manual." *8", 10", 12", 14", 16" LX200GPS Schmidt-Cassegrain Telescopes, 7" LX200GPS Maksutvo-Cassegrain Telescope with Autostar II Hand Controller*. Irvine, CA: Meade Instruments Corporation, 2003.

Orofino, Don, and Siamak Faridani. *Target Tracking Using Matlab Image Processing Toolbox*. 2008. <http://www.ieor.berkeley.edu/~faridani/tracking.htm>.

Outfitter Satellite Phones. *Iridium Data Modems*. 2009.

http://www.outfittersatellite.com/iridium_9522.htm.

Pavlov, Igor. *7-Zip*. 2009. <http://www.7-zip.org/>.

Pelican Products, Inc. *Pelican Products 1520 Case*. 2010.

http://www.pelican.com/cases_detail.php?Case=1520.

Philips Consumer Electronics. "Instructions for Use." *Philips SPC 900NC PC Camera*. 2003.

Pier-Tech, Inc. *Telescope Pier, Observatory, Roll Off Roof Observatory*. Pier-Tech, Inc. 2007. <http://www.pier-tech.com/>.

Pier-Tech, Inc. *Tele-Station 2 Roll Off Observatory*. 2007. http://www.pier-tech.com/telestation_2_roll_off_roof_observatory.htm.

Schmunk, Matthew M. "Initial Determination of Low Earth Orbits Using Commercial Techniques." *Master's Thesis*. Wright-Patterson AFB, OH: Air Force Institute of Technology, 2008.

Solar Stik. *Solar Stik*. Solar Stik Inc. 2009. www.solarstik.com.

Space-Track. *Space Track*. 2004. <http://www.space-track.org/>.

The MathWorks, Inc. *Image Acquisition Toolbox 3.4*. 2010.

http://www.mathworks.com/products/imaq/demos.html?file=/products/demos/shipping/imaq/demoimaq_Pendulum.html.

—. *Laser Tracking*. 2010.

http://www.mathworks.com/products/demos/shipping/imaq/demoimaq_LaserTracking.html?product=IA.

—. *Read Audio/Video Interleaved (AVI) file*. 2010.

<http://www.mathworks.com/access/helpdesk/help/techdoc/ref/aviread.html>.

Toyama Astronomical Observatory. March 17, 2005.
<http://www.tsm.toyama.toyama.jp/tao/intro-e.htm>.

Wal-Mart Stores, Inc. *Sylvania 7" Portable LCD TV, SRT702A: TVs*. Wal-Mart Stores, Inc. 2010. <http://www.walmart.com/ip/Sylvania-7-Portable-LCD-TV-SRT702A/12548058#ProductDetail>.

Vita

Captain Michael E. Graff graduated from Highland High School in Highland, NY. He entered undergraduate studies at the Georgia Institute of Technology in Atlanta, GA where he graduated with a Bachelor of Science degree in Mechanical Engineering May 2001. He was commissioned through Detachment 165 AFROTC at the Georgia Institute of Technology.

His first assignment was at Vandenberg AFB, CA where he was a Distinguished Graduate from Officer Space Prerequisite Training and a graduate of ICBM Initial Qualification Training December 2002. In January 2003 he was assigned to the 319th Missile Squadron as a missileer at F.E. Warren AFB, WY. While stationed at F.E. Warren, he served in capacities ranging from Deputy Missile Combat Crew Commander to ICBM Chief, Weapons and Tactics Training. In August 2008, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the Space Innovation & Development Center at Schriever AFB, CO.

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.				
1. REPORT DATE (DD-MM-YYYY) 15-03-2010		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From — To) 01-08-2008 – 15-03-2010
4. TITLE AND SUBTITLE Development Of a Remotely Operated Autonomous Satellite Tracking System			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) MICHAEL E. GRAFF, Capt, USAF			5d. PROJECT NUMBER	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENY) 2950 Hobson Way WPAFB OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GSS/ENY/10-M03	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Undisclosed sponsor			10. SPONSOR/MONITOR'S ACRONYM(S)	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT AFIT is currently developing a capability to remotely and autonomously track LEO satellites using commercial telescopes. Currently, the system is capable of open-loop tracking based on Two-Line Element sets (TLEs) downloaded from NORAD's space object catalog. The ability to actively track using a closed-loop control system would allow tracking of satellites which deviated from the published TLEs along with providing some information about the object's new orbital elements. To accomplish closed-loop tracking, the object is imaged by a digital camera connected to a wide field-of-view (WFOV) spotting scope. Software was developed to provide azimuth and elevation inputs in order to center the object within the WFOV. Pixel centroid location along with telescope azimuth and elevation commands can be recorded for use in estimating updated orbital elements. This thesis documents the current efforts towards achieving a remotely operated autonomous tracking optical system. Future application could include networking to other geographically-separated telescopes to allow simultaneous observation of the same space objects to accurately document orbital maneuvers.				
15. SUBJECT TERMS Closed-loop, Satellite, Tracking				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 136
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U		
			19b. TELEPHONE NUMBER (Include Area Code) (937)255-3636, ext 4559 richard.cobb@afit.edu	