

THERMAL EVOLUTION OF MOON

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

by

Arshdeep Singh Gill

March 2017

© 2017
Arshdeep Singh Gill
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Thermal Evolution of Moon

AUTHOR: Arshdeep Singh Gill

DATE SUBMITTED: March 2017

COMMITTEE CHAIR: John A. Saghri, Ph.D.
Professor of Electrical Engineering

COMMITTEE MEMBER: Gary Hughes, Ph.D.
Professor of Statistics

COMMITTEE MEMBER: William L. Ahlgren, Ph.D.
Associate Professor of Electrical Engineering

ABSTRACT

Thermal Evolution of Moon

Arshdeep Singh Gill

In August, 2014 three experiments were conducted using infrared systems deployed at White Mountain Research center, CA. The data was acquired for the whole month of August. Teams of 3-4 students from Cal Poly San Luis Obispo and UC Santa Barbara were stationed at the research center for 2-3 days to operate the equipment. The three experiments were:(1) creating spatial-temporal time series of lunar surface temperatures;(2) identifying atmospheric meteor trails;(3) search for meteor impacts on the Moon surface. Out of the three this thesis focusses on experiment 1 and the results from this experiment, could also help with the other experiments.

We propose to use a thermal infrared camera mounted on a telescope to acquire time-series observations of lunar surface temperatures to get a novel insight into the thermal evolution of the Moon over a complete lunar cycle. Half a lunar cycle would account from morning to night and lasts for approximately 14.75 days. Seeing how the pixel value changes from morning to night the pixel and temperature trends can be observed throughout the day. Apart from that one can get the two temperature peaks that could maybe help to get an estimate for the Thermal inertia of the surface in the presence of Moon regolith. The temperature trends and the thermal inertia could potentially provide some insight for methods that seek to determine the properties of asteroids from ground based observations.

ACKNOWLEDGMENTS

This project would not have been successful without the help and support from wonderful people and their valuable advice and feedback. First, I would like to thank Dr. Gary Hughes for all his help and patience throughout this Thesis. Dr. Hughes has been extremely helpful; from making me part of his project, giving me opportunity to get involve in the data gathering process, getting me started with the data processing, helping me troubleshoot all the little issues, providing me vision for this project. Without Dr. Hughes, this project would not be finished; he always took time out of his busy schedule to help me, even on holidays.

Dr. John Saghri, my department thesis advisor has always been there for me, providing me with valuable feedback every time I asked for one. He kept pushing me to get this project done in time. Knowledge gained from his Image Processing and Remote Sensing classes have helped me better understand the image basics and made me familiar with the Matlab tools, that I have used throughout my thesis. I would like to thank him for being there during my Undergraduate and Graduate degree. Lastly I would like to thank everyone at the Barcroft station, for making my stay there fun, healthy and safe. Also, my team members who helped me with installing the heavy equipment during freezing nights and keeping us all warm with heaters outside.

I would like to thank my Parents and rest of the family for their continuous support and encouragement throughout my college career.

Finally, I want to thank my Wife by,

Dedicating this study to: Dr. Prabhjot Kaur Gill

TABLE OF CONTENTS

	Page
LIST OF TABLES	viii
LIST OF FIGURES.....	ix
CHAPTER	
1. Introduction	1
2. Background.....	2
2.1 Thermal Inertia.....	2
2.2 Previous Works.....	3
3. Infrared over visible	5
4. Data Collection	6
5. System Architecture	8
5.1 Software Design	9
5.1.1 Block 1: Lunar Data.....	11
5.1.2 Block 2: Lunar Convert	11
5.1.3 Block 3: Intensity Based Image Registration.....	13
5.1.4 Block 4: Control Point Image Registration.....	14
5.1.5 Block 5: Background Normalization	17
5.1.6 Block 6: Pixel trends.....	18
5.1.7 Block 7: Temperature trends.....	19
5.2 Hardware Design.....	21
6. Image Registration	26
6.1 Intensity-Based Automatic Image-Registration	26
6.2 Control Point Image Registration.....	28
6.3 Feature Based Image Registration.....	30
7. Processing	31

7.1 Pre-processing	31
7.2 Short Time Series	33
7.3 Hour Time Series	35
7.4 Final Time Series	38
7.5 Error Analysis	39
7.6 Background Normalization	42
8. Results and Analysis.....	44
8.1 Short Time Series	46
8.1.1 Pixel Plots.....	46
8.2 Analysis.....	48
8.3 Hour Time Series	49
8.3.1 Pixel Plots	49
8.4 Analysis.....	52
8.5 Final Time Series	56
8.5.1 Pixel Plots	56
8.5.2 Temperature Plots	59
8.6 Analysis.....	61
8.6.1 Pixel Plots	61
8.6.2 Temperature Plots	67
9. Final Analysis.....	70
10. Conclusion.....	73
10.1 Future Work	75
BIBLIOGRAPHY	76
Appendix A: Source Code	77

LIST OF TABLES

Table	Page
Table 1: The Intensity Based Image Registration Process Step	28
Table 2: Control Point Registration Process Steps	29
Table 3: Hour Time Series Dataset for Mare Serenitatis.....	53
Table 4: Hour Time Series Dataset for Mare Tranquillitatis	55
Table 5: Final Time Series Dataset for Mare Serenitatis.....	62
Table 6: Final Time Series Dataset for Mare Tranquillitatis	66

LIST OF FIGURES

Figure	Page
Figure 1: The Moon	4
Figure 2: Visible vs. Infrared camera	5
Figure 3: Zero Level Hardware Diagram	8
Figure 4: Level 1 Hardware Diagram	9
Figure 5: Overall Software Flowchart	9
Figure 6: Lunar Convert Flowchart	11
Figure 7: Intensity based Image	13
Figure 8: Control Point Image Registration Flowchart	14
Figure 9: Background Normalization Flowchart	17
Figure 10: Pixel trends Flowchart	18
Figure 11: Temperature trend flowchart	20
Figure 12: Hardware equipment	21
Figure 13: Scene Pixel Resolution	23
Figure 14: Final Pixel Resolution of 2.6 Km/pixel	23
Figure 15: Camera Response	24
Figure 16: Intensity Based Image Registration Block diagram	27
Figure 17: Control Point Selection Tool	30
Figure 18: Glider caught during data collection	31
Figure 19: Frame out of focus	32
Figure 20: Registration error in one Video file	34
Figure 21: Frame1 and Frame 450 (Registration Bad)	36
Figure 22: Frame 1 and Frame 900 (Registration good)	36
Figure 23: Frame1 and Frame 451 (Registration Bad)	37
Figure 24: White Rectangle on the same crater in all Frames	40
Figure 25: Frame 225 and 3150 showing maximum horizontal error	41
Figure 26: Frame 225 and 3175 showing maximum vertical error	41
Figure 27: Hour time series plot before normalization	42
Figure 28: Hour time series plot after normalization	42
Figure 29: Final time series plot before normalization	43
Figure 30: Final time series plot after normalization	43
Figure 31: Mare Tranquillitatis region on Moon	44
Figure 32: Mare Serenitatis region on Moon	45
Figure 33: Mare Tranquillitatis 3x2 kernel plot	46
Figure 34: Mare Tranquillitatis average of 3x2 kernel plot	47
Figure 35: Mare Serenitatis 3x2 kernelplot	47
Figure 36: Mare Serenitatis average of 3x2 kernel plot	48
Figure 37: Mare Tranquillitatis 3x2 kernel plot	49
Figure 38: Mare Tranquillitatis average over 3x2 kernel plot	50
Figure 39: Mare Tranquillitatis Average pixel plot	50
Figure 40: Mare Serenitatis 3x2 kernel plot	51
Figure 41: Mare Serenitatis average over 3x2 kernel plot	51
Figure 42: Mare Serenitatis Average pixel plot	52

Figure 43: Detailed Mare Serenitatis average 3x2 kernel plot	52
Figure 44: August 14 Mare Serenitatis region with white test pixel	54
Figure 45: Detailed Mare Tranquillitatis average 3x2 kernel plot.....	55
Figure 46: Mare Tranquillitatis plot for 3x2 kernel.....	56
Figure 47: Mare Tranquillitatis plot for average of 3x2 kernel.....	57
Figure 48 : Mare Tranquillitatis plot for 7 Average values	57
Figure 49: Mare Serenitatis plot for 3x2 kernel.....	58
Figure 50: Mare Serenitatis plot for average of 3x2 kernel.....	58
Figure 51: Mare Serenitatis plot for 7 Average values.....	59
Figure 52: Infrared temperature plot for the Mare Tranquillitatis	60
Figure 53: Infrared temperature plot for the Mare Serenitatis.....	60
Figure 54: Detailed Mare Serenitatis average plot	61
Figure 55: August 5(Left) and August 8(Right)	63
Figure 56: August 11(Left) and August 14(Right).....	63
Figure 57: August 15(Left) and August 16(Right)	63
Figure 58: August 17	64
Figure 59: Actual Moon Phases in August, 2014	65
Figure 60: Detailed Mare Tranquillitatis average plot.....	65
Figure 61: Mare Serenitatis Average Infrared Temp plot.....	67
Figure 62: Mare Tranquillitatis Average Infrared Temp plot.....	68

1. Introduction

In the quest to study the thermal evolution of the Moon surface, it is proposed to acquire digital video data of the Moon using a long-range infrared camera mounted on a telescope. It was planned to observe the Moon for approximately 4 hours per day, when the Moon was within $\pm 30^\circ$ of its peak position in the sky. For this experiment, the team travelled to the Barcroft research center located on White Mountains in Death Valley at an elevation of 12,470 ft. This location is ideal for Astronomical observations as the light and other forms of pollutions are minimized. The experiment was conducted over the period of 01-31 August, 2014. From the infrared data, an attempt is made to create the spatial time-series observations of Lunar surface temperatures over half a Lunar cycle. The temperature information is related to the pixel value in the images. Mapping how the Moon surface heats and cools throughout the day could provide some good insight into the Thermal inertia of the surface in the presence of Regolith (dust and debris that cover the Moon surface). This could further provide some great insight for methods to determine asteroid properties from passive observations.

At the end of the data gathering experiment 2 Terabytes of digital data was acquired. After some preprocessing and putting the image data in a video format, the clearest days are chosen for further experimentation. Image registration is then performed on the data to obtain pixel/temperature trends for few chosen locations on the surface. Temperature slopes are found for the Final data over half the Lunar Cycle. The infrared camera with the telescope has a surface resolution of 2.6Km/pixel and operates at a peak response around 10 Microns, wavelength.

2. Background

2.1 Thermal Inertia

In scientific terms, Thermal inertia is the measure of the thermal wave that controls the surface temperature. In other words, it is the measure of the resistance of the material to temperature change. Mathematically, it is defined as the square root of the product of the material's Volumetric Heat capacity and the bulk thermal conductivity. The Volumetric heat capacity is the product of density and the specific heat capacity.

Hence, $I = \sqrt{(\kappa * \rho * c)}$, where

- κ is thermal conductivity, with unit [$\text{W m}^{-1} \text{K}^{-1}$]
- ρ is density, with unit [kg m^{-3}]
- c is specific heat capacity, with unit [$\text{J kg}^{-1} \text{K}^{-1}$]
- I has SI units of thermal inertia of [$\text{J m}^{-2} \text{K}^{-1} \text{s}^{-1/2}$].

In Planetary systems, Thermal inertia is an important quantity in determining the diurnal and seasonal temperature variations. It depends largely on the physical properties of the geologic surface and its surroundings. It is a volumetric quantity which implies that it depends on the surface composition approximately 20 cm beneath, to which a typical diurnal heat wave can reach. A low Thermal inertia of a surface typically means that the temperature of the surface changes rapidly during the day. Whereas the surface with high Thermal inertia value, have a small swing in the temperature during the day. With the help of additional data various surface features can be derived along with the surface composition [Wikipedia Contributors, 2017].

It is possible to get a rough estimate of the surface thermal inertia from the temperature slopes, which can be found from known maximum and minimum temperature values. Price (1977), derived an approximation of the Thermal Inertia, P, using the temperature difference and albedo of the surface. The *apparent thermal inertia (ATI)* is calculated as:

$$ATI = (1-A) / (T_{Day} - T_{Night})$$

where A is the albedo, defined as the reflection coefficient of the surface implying the portion of the light being reflected by the surface. The known value of the Moon's average visual albedo is 0.12.

2.2 Previous Works

The Moon is thought to have been formed approximately 4.5 billion years ago, due to a collision between the pre-Earth and a Mars size body named 'Theia' [Benz et al, 1986]. On an average the Moon is 2,38,855 miles away from Earth, with virtually no atmosphere to capture the heat during day for long. Due to which there is a large variation in the daily temperatures for the surfaces under darkness and illumination [Thomas, 2009].

In general, the Lunar surface can be separated into two categories, the maria (15%) and the highlands. The maria that makes about 15 % of the surface, is composed of basaltic lava flowing in the craters and basins formed from large impacts. These rocks are younger than the surrounding highlands and have a lower albedo value that makes it appear darker. Most of the big maria is on the near side of Moon as seen in Figure 1. The highlands make up to 75% of the surface and are mainly formed of older rocks that have a higher albedo value and therefore appears lighter when viewed from Earth [Thomas, 2009].

According to NASA, observing the Moon at wavelengths higher than 5 μ m one can determine the apparent surface temperature during Lunar day. Also, during the day the temperatures are very insensitive to Thermal inertia but at night the ground based measurements can yield reliable Thermal inertia values. The depth and temperature change are big factors for varying thermal properties. There are many hotspots present on the surface such as the small craters that cool off very slowly. This NASA study in 1971 also states that the Thermal inertia values of the loose surface is 50 times higher than that of the rocks [Allen, D.A. 1971].

One of the other papers published in 2003, explore the thermal properties of the eclipsed Moon and finds the brightness temperatures with a dynamic range of 160K to 250 K. The results were derived from a sensor operating at 11 μ m [Price et al. 2003].



Distance: 238,900 miles from Earth

Lunar Day: 29.5 Earth Days

Category: Earth Satellite

Temperature: -233 to 123 °C

Diameter: 3,475 Km

Figure 1: The Moon

3. Infrared over visible

For this experiment, an Infrared camera is preferred over the visible camera for some obvious reasons such as the infrared camera can be used equally well during the day or night. These infrared sensors create an image of the heat signature in the scene as oppose to the visible camera that rely on the reflected or emitted visible light.



Figure 2: Visible vs. Infrared camera

As an example, the visible camera video recordings of the meteorite impact near Chelyabinsk on 15 February 2013 didn't provide any insight to the impact events. Comparatively, an infrared video could have also captured the meteor trails created by small debris in the upper atmosphere with the complete heat signature of the meteor tail without any interference from the light reflected off the smoke. Also, if the event had taken place during the night, the dash cam videos would have been completely ineffective. Additionally, the infrared camera zoom optics along with the reflective telescope can capture high-resolution videos of distant objects such as the Moon.

4. Data Collection

In the summer of 2014, a Long-wave Infrared camera mounted on a Telescope installed in a wooden frame, was taken to White mountains research center in Death Valley, CA. The experiment lasted for about twenty days. The camera captured the Moon data for approximately 4 hours per day when the Moon was within $\pm 30^\circ$ of its peak position above. The process of capturing data was manual for two reasons: First due to the small *Field of view* of the camera which could not capture the entire surface of the Moon, thus requiring an active person on duty to move the telescope every minute to capture the entire surface. Secondly, as the night progresses the Moon moves in the sky and the telescope didn't have any automatic tracking system.

The process of moving the telescope was broken down into four steps. At each step a quarter of the Moon would be captured in the frame. Starting with the upper right quarter, the telescope was pointed at the outer sky near the edge of the Moon surface. As the Moon moves to the right, the camera capture the frames until the center of the surface is reached. Moving to next step, the personnel moves the camera to point at the top edge of the surface and the camera again capture the frames until the entire Moon goes out of view. Once all the four quarters of the Moon surface were captured the process was repeated.

Note: The experiment was intended for the whole month of August but due to atmospheric disturbances, clouds and storms on some days, the team was forced to pack the equipment and move it indoors to prevent it from damage. At the end, only eight days of data proved to be of any use.

Method: Each section/quarter of the Moon surface is analyzed separately. Since, this thesis is focused on determining the effectiveness of the ground based technique the focus is concentrated on few spots on the Moon surface rather than every single pixel on the entire surface, which is not required.

Due to the lack of an automatic tracking system and smaller FOV of the infrared camera, the frames captured separately for the four sections of the Moon require to be first aligned to get any useful pixel value trends over time. Looking closely, for each section two different image registration problems arise. First, the image registration for each .Avi file (450 frames) is required which is affected by the *movement* of the Moon in the sky and thus in the frames. Secondly, the image registration in between different .Avi files (corresponding to different times), which is now affected by the *Rotation* of the Moon surface.

5. System Architecture

Figure 3 shows the Level zero block diagram of the system; where the IR camera provides the input to the system, which then processes the input raw 16-bit data to output the Pixel trends and optionally the temperature trends, related to the Thermal Inertia.

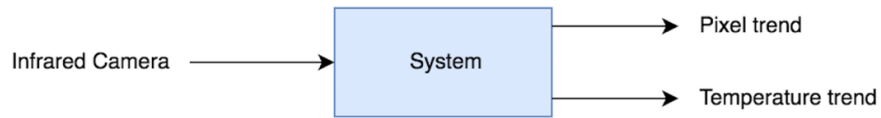


Figure 3: Zero Level Hardware Diagram

Figure 4 below shows the level 1 diagram of the system. The long wave infrared (IR) camera is mounted on a telescope using a connector specifically designed for this experiment. The IR camera connects to a network adapter which connects to a laptop through a data grabber program, written in C++. This data is then processed using an algorithm in Matlab which performs Image registration before getting pixel trends. The whole system is powered through long cables attached to an outlet. The laptop runs on its own battery.

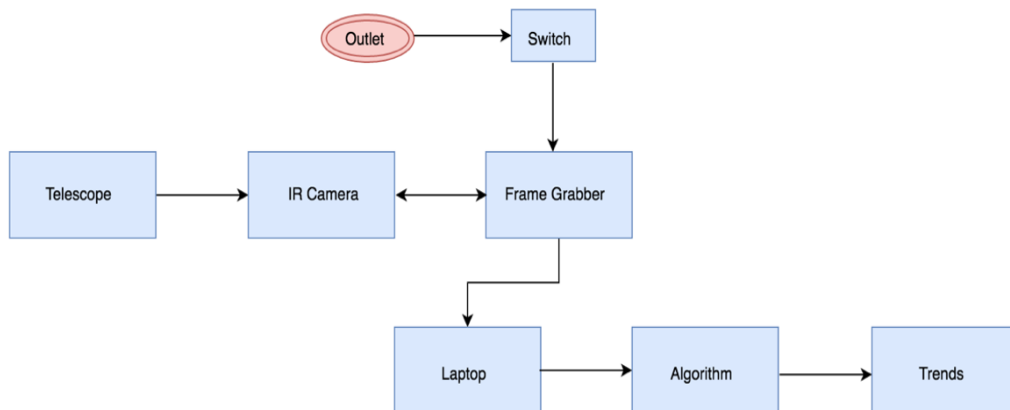


Figure 4: Level 1 Hardware Diagram

5.1 Software Design

Figure 5 shows the simplified flowchart of the overall Software workflow. The working of the different blocks is as explained below:

Lunar Data: This first block acts as a Data grabber after the camera is connected onto the telescope and to the laptop. With a single click this script is responsible for talking to the infrared camera and saving the raw 16-bit data onto an external hard disk connected to the laptop. The infrared images are then put in Tiff files, each containing 450 images. This script is re-initiated during each new phase of data capture, which begins every time the telescope is moved and focused on the new area on the Moon.

Lunar Convert: Once the data is saved on the hard disk, the next block takes the Tiff files and puts them in a video format to visually make sure the frames are in a sequential order.

Intensity Based Image Registration: This block is responsible for creating the *Short Time Series*. Once the Tiff files (*450 frames each*) are obtained, the intensity based image registration is performed on the selected files for selected days.

Control Point Image Registration: This block is responsible for creating the *Longer Time Series*. After obtaining the individual registered files,

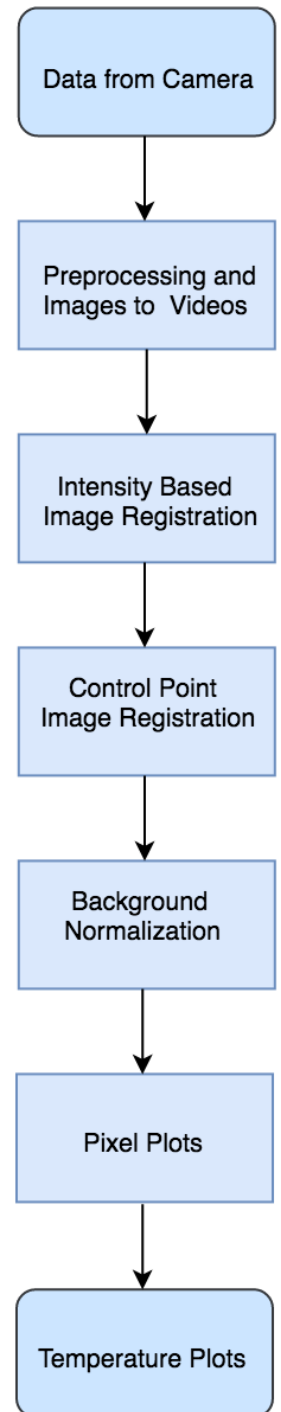


Figure 5: Overall Software Flowchart

the Control Point Registration routine now registers in between Tiff files (individually registered). Using one file as a reference the rest of the Tiff files in the folder are registered again and saved. It is to be noted that this step is done manually. There are two parts in this block as explained in the next section.

Background Normalization: Background normalization is required before getting the pixel curves to minimize the background noise. In the dataset, ideally in every frame, the pixel value for the background sky should be a constant but carefully observed it's not. This is explained in more detail later. This block takes the average value of a point in the background sky using a 5x5 kernel for each Tiff file and then takes another average over 450 frames to get one value for each Tiff file in the dataset. For example, if the dataset contains 7 files from 7 different days then 7 average values for the background sky are obtained. The middle value is then kept as a reference and the other six values are subtracted to get the difference. Then add or subtract that difference from the corresponding Tiff file (from every frame) and save the new data. This could be little tricky as the point in dark sky also moves in different Tiff files, so at this point it's best to manually select that point for each of the 7 files.

Pixel Trends: Once the newly saved registered files are concatenated together to make a final Tiff file ($(450 * \# \text{ of Tiff files}) \text{ frames}$), its ready to be analyzed. For selected locations, this script then logs the pixel data to get the pixel trends in time.

Temperature Trends: This is the final step in the process to convert the pixel values to some infrared temperature values. It should be noted that the temperatures obtained from this script are the infrared temperatures and not the actual surface temperature values. The temperature difference can still perhaps be approximated using these values.

5.1.1 Block 1: Lunar Data

The Lunar data was collected and saved on the hard drives using the frame grabber.

5.1.2 Block 2: Lunar Convert

The purpose of the Lunar Convert script is to put the sequential 450 frames in each Tiff file, obtained from Block 1, in a video format (.Avi format). If the files are not in a sequential order the video would not be smooth, instead would move back and forth. Figure 6 shows the flowchart for this block and individual processes are explained below.

Create Data Path: The first step is to create path to the data directory on the hard disk to be able to access data.

Constructing File names: Next construct file names to easily read all the data files ending with Tif in the data folder and construct corresponding file names ending with .Avi to be used to name the newly created video files in the next steps.

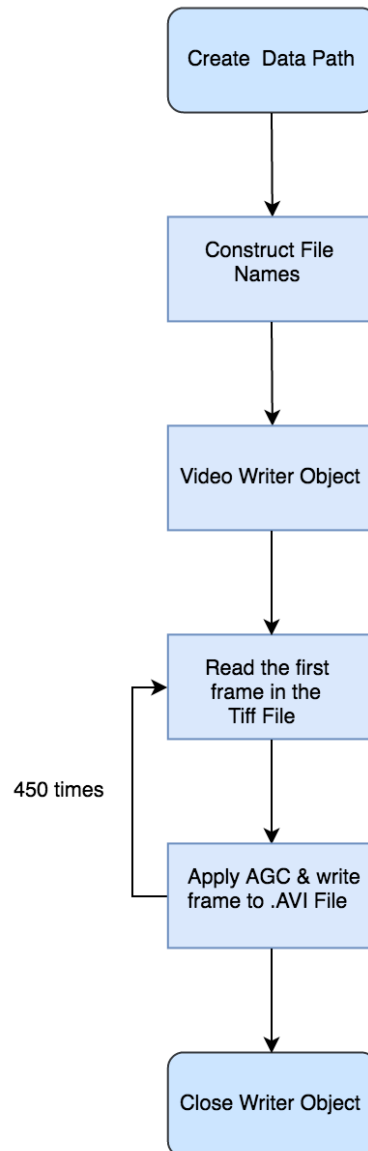


Figure 6: Lunar Convert Flowchart

Video Writer object: Using the VideoWriter (), Matlab function, a VideoWriter object is created with the same name constructed in the previous step, ending in .Avi, to write a motion Tiff Avi.

AGC and Final Avi: The first Tiff file is read from the data directory and starting with the first frame, the script applies the Automatic Gain Control algorithm before writing the frame content to the VideoWriter object created previously. The loop runs for 450 frames to finish creating one Avi file for the related Tiff file. This process is repeated for all the Tiff files in the data directory.

5.1.3 Block 3: Intensity Based Image Registration

Once the Tiff files, each containing 450 frames, are obtained the Intensity based Image registration algorithm is applied to the selected Tiff files to register them individually. Figure 7 on the side shows the process flowchart and each process is explained in detail below.

Input Transformation Type: As explained in the Intensity based registration technique under the Image registration section, an input transformation type is specified by the user. In this case, the transformation type specified is “monomodal”, which means images taken with the same device. If different cameras are used to take the images the type would be “multimodal”.

Create Data Path: This step creates path to the data directory to be able to access the data files.

Reference Frame: After reading the first Tiff file in the workspace, the middle frame (*Frame 225*) is specified as the reference frame, to be used by rest of the frames to align with the reference frame.

Reading Tiff File: Next each frame of the Tiff file is read in a loop, and using the *imregtform*, Matlab function, an optimizer and metric are internally determined for each of the 450 frames.

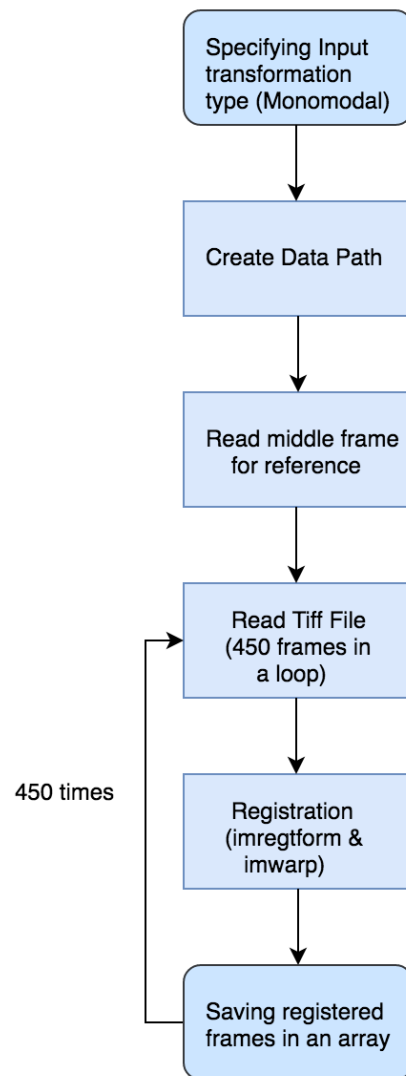


Figure 7: Intensity based Image Registration Flowchart

Registration: The input transformation type along with the internally determined transformation type using the optimizer and metric determined in the previous step, the Matlab function, `imwarp` is used to register and thus align the frames with the reference frame.

Registered images in Array: Once the registration is complete the registered frames are saved in an array from which new Tiff files with all the frames aligned can be easily created.

5.1.4 Block 4: Control Point Image Registration

To create Long Time series, registration in between Tiff files is attempted using the Intensity based technique but eventually, the Control based algorithm is applied. This would be explained more in a later section. At this point eight individually registered Tiff files, at an average thirty minutes apart, for the Upper right quadrant of the Moon makes the data set.

Figure 8 on the side shows the process flowchart for the Control Point Registration algorithm and each process is explained in detail below.

Specifying Fixed Frame: Out of the eight registered Tiff files, the first file is chosen arbitrarily as the Fixed/reference file. Further the middle frame is read in

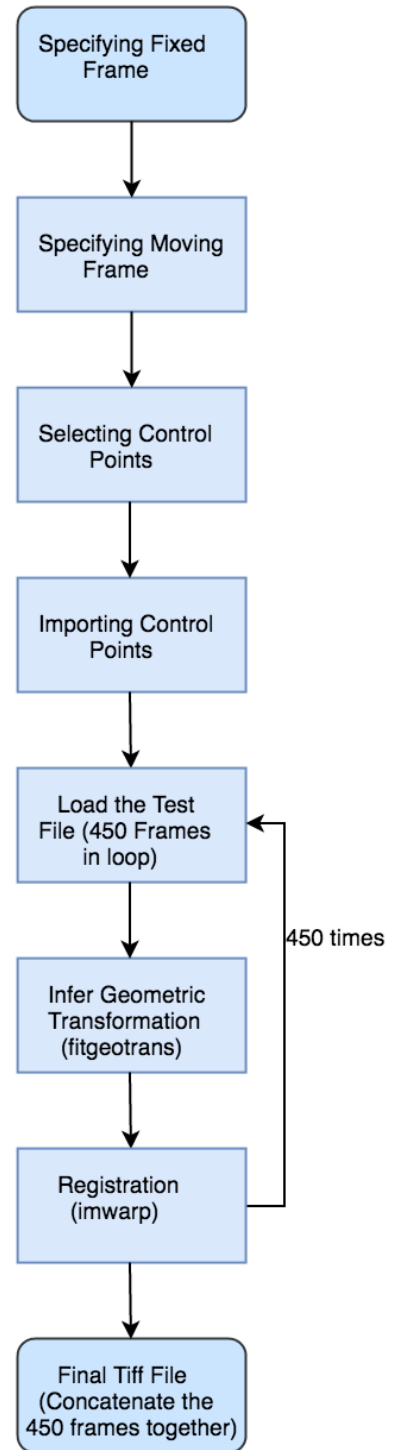


Figure 8: Control Point Image Registration Flowchart

the workspace and chosen to be the reference or fixed frame to be used to register the whole data set.

Specifying Moving Frame: Under this process, rest of the seven Tiff files are read in the Matlab one at a time and the middle frame of the active file is set to be the moving frame to figure out the transformation type to be applied to all the 450 frames in the active Tiff file.

Selecting Control Points: Through a Matlab function, *cpselect*, a window showing both the Moving and Fixed frames adjacent to each other pops up. The user selects the four reference points in both the images and export them to the workspace. This process is explained in detail under the Image registration section.

Importing Control Points: Once the reference points are created they are exported to the Matlab workspace.

The first four processes make up the Control Point Part 1 script.

Load the test File: Once the control points are obtained, in the following script the Tiff file corresponding to the moving frame is loaded in the workspace. All the 450 frames are loaded in a loop to be registered w.r.t. the fixed frame.

Infer Geometric Transformation: Next a Matlab function, *fitgeotrans*, is used and a transformation type is specified, which in this case is 'NonreflectiveSimilarity'.

Nonreflective similarity transformation is desirable over the other types since it can include a rotation, a scaling, and a translation. The *fitgeotrans* uses the control points and the transformation type information to infer the geometric transformation to be used for alignment of the frames.

Registration: As the name suggests this transformation uses, *imwarp* command again to align the moving frame with the fixed frame. The same transformation can be applied to all the 450 frames, since they are already aligned to each other.

Final Tiff: All the 450 frames after being aligned are concatenated together to make a Final Tiff file, in which all the frames are aligned with the fixed frame from the reference File.

The last four processes make up the Control Point Part 2 script.

The whole Control Point registration process is applied to all the seven remaining Tiff files creating corresponding registered Tiff files that are aligned with each other.

5.1.5 Block 5: Background Normalization

Figure 9 shows the process flowchart for Background normalization with individual processes explained below.

Select file in Dataset: After Block 4 get the different Tiff files registered individually and in between. Select the first file in the dataset.

Select a point in Sky: After carefully observing the background sky pixels using *imtool* in Matlab, choose a location that would be visible in all other files too and not out of frame in any of them.

Average with 5x5 kernel: Next using a 5x5 kernel an average value over 25 pixels at that location is taken for each of the 450 frames in the selected Tiff file from the dataset.

Average over 450 frames: Once an array of 450 average values for each frame is obtained,

another average value over 450 frames is then

calculated to get one value for one Tiff file. All the above steps are repeated X # of times, where X is the # of Tiff files in the dataset. At the end, obtain X average values each corresponding to a different Tiff file.

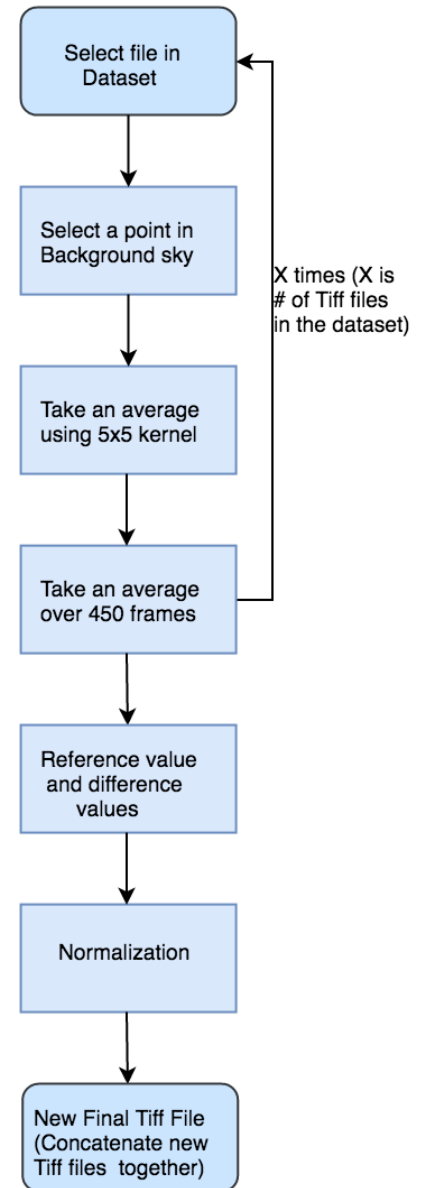


Figure 9: Background Normalization Flowchart

Reference value and Difference values: Arbitrarily choose one of the average values as a reference and subtract all the other values from it to find X difference value, where one of the values would be 0.

Normalization: At this step the corresponding difference value to the right Tiff file is added or subtracted depending on if the value is negative or positive. Each Tiff file is saved under a new name.

New Final Tiff: At the end, all the new Tiff files are concatenated together to form a bigger Tiff file that should contain $(X * 450)$ frames, all aligned and with background normalized for a smoother trend in the next block.

5.1.6 Block 6: Pixel trends

Figure 10 shows the process flowchart to get the pixel trends with individual processes as explained below.

Long Time Series: The output of Block 5 is the New Tiff file that represents the Long-Time series. It is loaded in the workspace to obtain the Pixel trends for selected locations on the Moon.

In a later section, the error in the registration will be determined.

Select Location: The pixel location is found using a simple script which is then used in the pixel plot script to obtain the plots. This repeated for the number of locations to be observed.

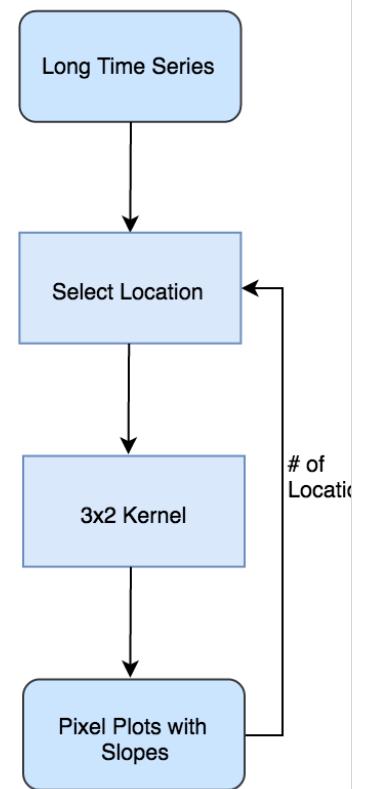


Figure 10: Pixel trends Flowchart

3x2 Kernel: To get the pixel trends is now as simple as selecting any point in the frames and saving the pixel values corresponding to different frames in a separate array and plotting against, frame number. Due to some error in the registration, instead of logging the value of a single pixel, an average pixel value using a simple 3x2 kernel could be useful to compensate for the error.

Note: Using a 3x2 kernel decreases the resolution from 1 pixel (2.6 Km) to 6 pixels (5.2 x 7.8 Km²). For a smooth surface this doesn't make a significant difference but its predicted to make some difference for a crater dense location.

Pixel Plots and Slopes: Using Matlab plot window, different functions could be selected under basic fitting to fit the curve with a corresponding equation. This would visually show the pixel rise and fall. This is shown under results and analysis section. Further by choosing the average pixel value for every 450 frames, instead of plotting 450 points for each set only one point can be plotted to visually observe the pixel trend better and possibly obtain the slope of the graph using a regression line.

5.1.7 Block 7: Temperature trends

Figure 11 shows the process flowchart to covert pixels to infrared temperatures and get the Temperature trends. All the individual processes as explained below.

Read Data File: After the script starts running it prompts the user to select the test file.

The data Tiff file is loaded in the workspace for which the temperature conversion needs to be done.

Sequence max. and min. temperature: In this step the maximum and minimum pixel values for the test file is found and after the pixels are converted to temperatures the corresponding maximum and minimum temperatures are also saved.

Temperature conversion: Each frame is read in a loop, running to the length of the test Tiff file, and a camera calibration file (*calcurves.mat*) is read for temperature conversion. From this file the slope and y-intercept for the temperatures is obtained and using the linear equation the pixels are converted to infrared temperature values. They are then converted from Celsius to Fahrenheit.

Save and write frames to Avi: Through a Matlab Writerobject, after the frames are converted to temperatures and are saved in an array, they are written in a Avi file to create a video stream.

Build color map for sequence: After converting the pixels to temperatures and writing to a video Avi file, a color map is built to make the video visually intriguing and see the temperatures for every pixel in the image through a heat map.

Create New Tiff File: As all the pixels are getting converted to temperatures, each Tiff file is being saved in a *struct* which towards the end is read in a loop to concatenate all the resultant temperature converted Tiff files to make one big final Tiff.

Plot Temperature curve: Once the new Tiff file is created, the

same test pixel location, used in pixel plots is used again to obtain the temperature curves. The plotting script remains the same.

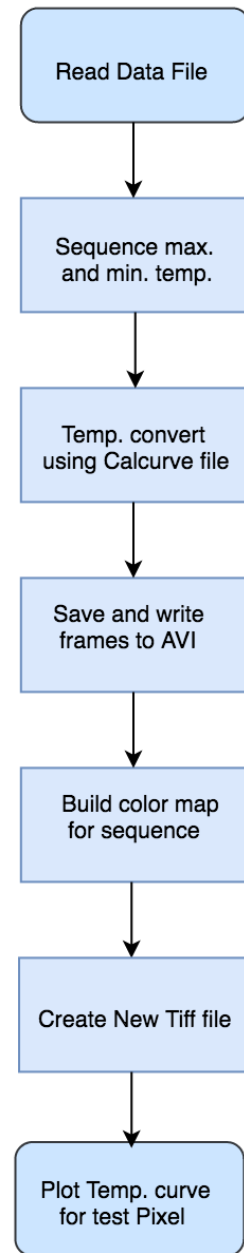


Figure 11:
Temperature trend
flowchart

After obtaining the pixel/temperature plots, the Matlab plot window is used to find the best fitting curve. The curve or the low and high data values could be then used to obtain the slopes.

5.2 Hardware Design

The complete hardware for lunar observations is shown below in Figure 12. Additionally, there is a wooden frame to protect and hold the telescope still and to make the process of maneuvering it easy. The infrared camera mounted on the telescope using a customized adaptor captures the raw 16-bit data, which is saved on the laptop through a data grabber software and is immediately backed up to external hard drives. Specifications for key blocks are described below.



Figure 12: Hardware equipment

Dobsonian telescope: The Dobsonian telescope is the key element in the hardware design. The telescope with 2.5 m focal length helps in achieving a resolution of 2.6 km

from ground. Before being deployed, an interposer (Tau barrel adaptors) to connect the telescope eyepiece with the infrared camera was built by the UCSB team. Secondly, the original mirror of the telescope is made from Silicon Dioxide that doesn't reflect the infrared energy well. So, the mirror was coated with a thin layer of Gold to capture most of the infrared energy.

Long-wave Infrared Camera: The Infrared camera was supplied by FLIR Systems, Inc. It was proposed to use FLIR's Tau-640 camera, with a 640 by 512 uncooled micro bolometer array sensor, 30 Hz frame rate, and 14-bit digital resolution. For lunar observations, the Tau camera was mounted onto a Dobsonian telescope with 2.5 m focal length, providing full view of the lunar disk with approximately 2.6 km pixel resolution on the Moon's surface. Figure 13 and 14 below shows the resolution calculations. The infrared camera is connected to the laptop via Frame grabber. There are no other wires connected to the camera.

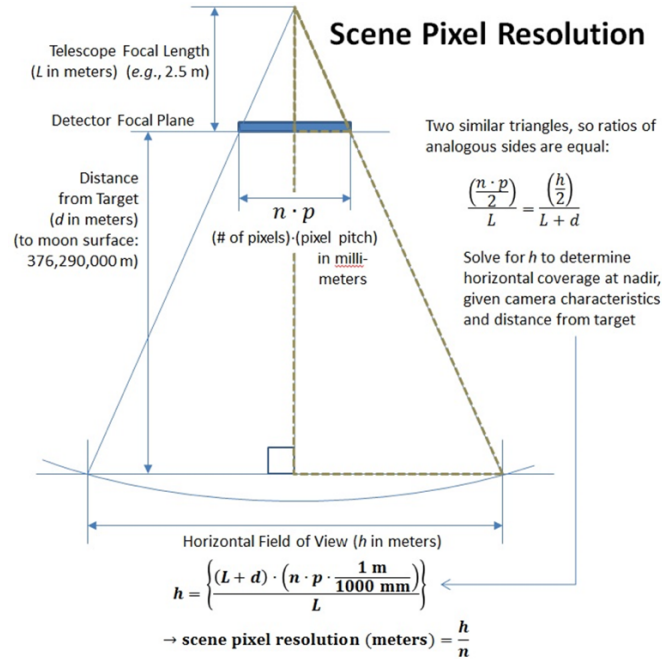


Figure 13: Scene Pixel Resolution

Tau-640, 2.5 m Telescope, Lunar Surface Resolution

focal length (m)	2.5
hpitch (mm)	0.017
hres (pixels)	640
vpitch (mm)	0.017
vres (pixels)	512
hsize (mm)	10.880
vsize (mm)	8.704
hfov (degrees)	0.249
vfov (degrees)	0.199
Coverage (meters):	1,637,614
Pixel Resolution (m/p):	2,559

Figure 14: Final Pixel Resolution of 2.6 Km/pixel

Figure 15 below shows the response of the camera. The camera can operate from 7 Microns to 20 Microns with a peak response at around 10 Micron.

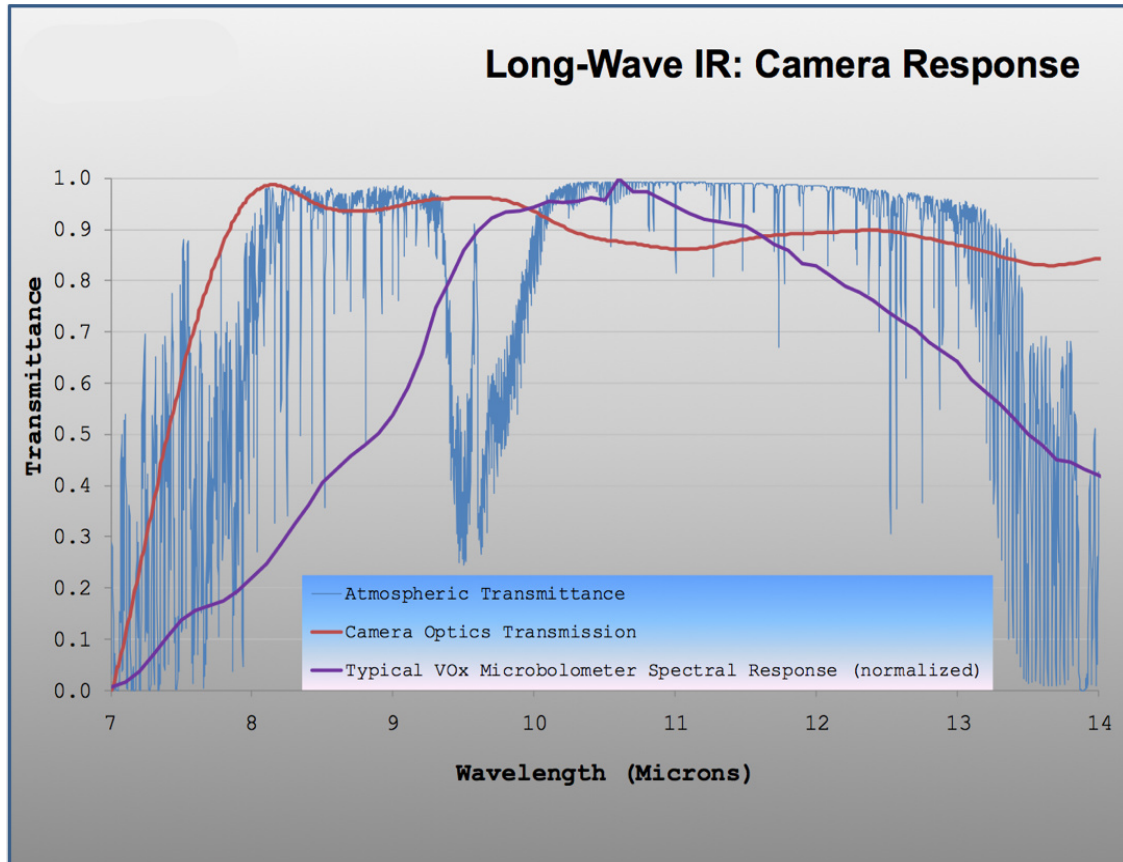


Figure 15: Camera Response

Frame Grabber: The frame grabber connects between the infrared camera and the computer. The digital data from the camera is sent to the computer through the grabber, connected to the camera with a cable and to the laptop through one of the side slots. The Imperx camera link Express card helps the user to store real time video images on computers, where the images are stamped with date and time. It comes with very user friendly and easy to use application program and driver software.

Laptop: Through the Laptop all the data is saved on external hard drives to be processed later. The real time infrared image stream in the field of view of the camera is seen on the laptop. This real-time video helps the user to see where the telescope is pointed and when

the Moon moves out of the frame. Without this live feed one cannot move the telescope to capture different quadrants of the Moon. Laptop connects to camera through the Frame grabber that plugs directly into one of the available slots on the side. Laptop has Windows 7 64-bit, photon SDK, Matlab, and Microsoft Visual Studio to run the system. Photon SDK is needed to be able to communicate with the camera. Matlab automates the process of grabbing and saving the data. Microsoft Visual Studio is only needed if the laptop needs to talk to the camera through some other technique.

6. Image Registration

Image registration is a technique to align two different images of the same scene which might have been taken while the camera was moving/rotating or the object in the scene was not still. In both cases the angle could be different for the two images or one image could contain only a part of the scene compared to the entire scene in the other image. In this process one image is designated as the *reference image* or the *fixed image*. The other image is called the *moving image*, which is aligned with the fixed image through various geometric transformations.

Geometric transformations perform the task of mapping locations on the fixed image to locations on the moving image. Hence, the process performance metric is the correct Geometric transformation parameters.

Note: Since the image registration area is new to me, I decided to stick with the Matlab toolboxes.

The Matlab offers three different Image registration techniques. Focus will be put on the two being used in this thesis. The three techniques are described below:

6.1 Intensity-Based Automatic Image-Registration

The intensity based Image registration technique works in a closed loop system as shown in Figure 16. The process requires a set of images, a transformation type, a metric, and an optimizer as explained below.

Input images: The two input images are, the fixed image and the moving image, that needs to be transformed for alignment with the reference image.

Transformation type: A 2-D transformation determined from a user specified transform type and an internally determined transform type.

Metric: Similarity metric is a scalar value to describe how similar the two images are, hence evaluating the registration accuracy.

Optimizer: This defines the methodology to minimize or maximize the similarity metric.

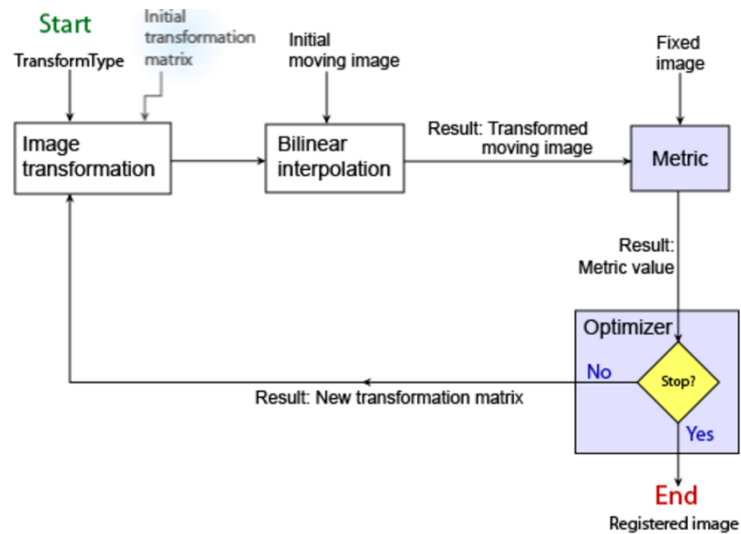


Figure 16: Intensity Based Image Registration Block diagram

The figure shows a closed loop system with the feedback term being the new transformation matrix between the fixed and the transformed moving image. After the system determines the specific transformation, it is applied to the moving image with bilinear interpolation. The resulting transformed image and the fixed image are the inputs for the Similarity Metric block that follows. The output scalar value is the input to the optimizer that checks for the stop condition, which could be a specified number of iterations. Such a condition determines to either stop the registration, producing the final Registered image or output a new transformation matrix which is applied again through a feedback to repeat the process [MathWorks].

To use this process in Matlab one can follow the simple steps in Table 1:

Table 1: The intensity based Image registration process steps

<i>Step</i>	<i>Function</i>	<i>Matlab command</i>
1	Reading images in Matlab	imread()
2	Creating optimizer and metric	imregconfig()
3	Registering the images	imregister()

6.2 Control Point Image Registration

Control Point registration requires a user to manually select four points, corresponding to four distinct features in one image, which are then mapped to the same pixel locations on the other image. The first image is called the Fixed image (*Reference Frame*) and the other is referred to as the Moving image (*Test Frame*). After reading the images in Matlab, using the Control Point Selection tool select the four points on both images. After selecting the points, they are exported in the Matlab workspace. An example showing the process of selecting the control points is shown below. Using the *fitgeotrans* command, the Matlab then specifies the type of transformation needed for alignment. Finally use *imwarp*, to align the images [Math Works]. The summary is shown in table 2 below:

Table 2: Control Point registration process steps

<i>Step</i>	<i>Function</i>	<i>Matlab command</i>
1	Reading images in Matlab	imread()
2	Select control points	cpselect()
3	Estimate transformation type	fitgeotrans()
4	Registering the images	imwarp()

Once the *Moving* image and the *Fixed* image is loaded in the workspace, using *cpselect* command the Matlab control point selection tool is used to select the four points as shown in Figure 17 below.

Four separate windows appear, where the Moving image is displayed on the left and the Fixed image is displayed on the right. The rectangles in the images are there to zoom in on a specific area on the surface and the zoomed portion is shown in the above window. The zoom makes it easier for the user to select the points corresponding to same locations on both the images. As shown below, the point pairs are numbered from 1 to 4 on both the images. The coordinates of these points can be exported to Matlab workspace using this tool which are then used to figure out the transformation required to register the two frames.

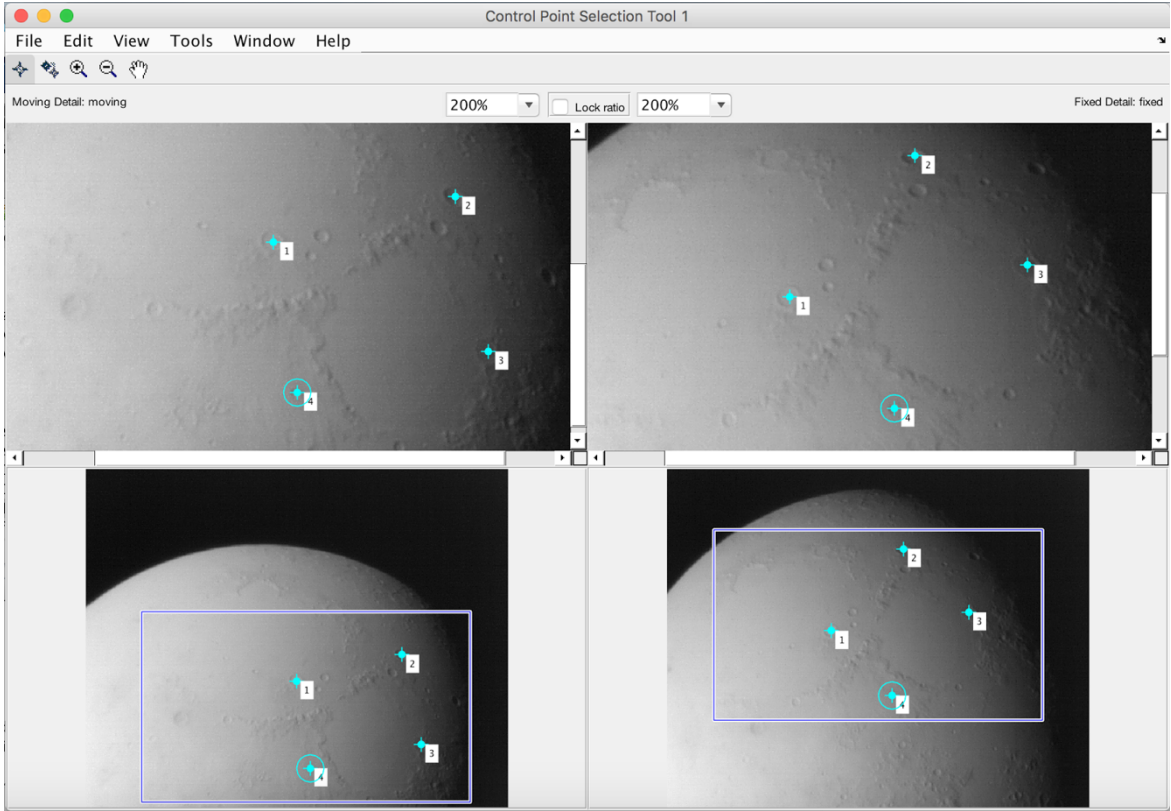


Figure 17: Control Point Selection Tool

6.3 Feature Based Image Registration

This Image registration technique is based on feature detection, feature extraction and matching. This technique would work better if the pictures have constant pixel values as opposed to the case here where pixel values change constantly with time. The feature detection would be highly dependent on looking for the same features represented by same pixel values in the two images. With the change in pixel values the error would be higher using this technique.

7. Processing

7.1 Pre-processing

Step1: The first step after collecting the data is to sort out and get rid of all the anomalies to work with clear frames. This is an important step since anything blocking the view of the Moon surface would disrupt the whole process of reading the surface pixel values correctly. This step was accomplished by the UCSB team by analyzing each frame. Anomalies could be due to some bad pixels in the camera, some airplane in the view blocking the Moon surface or frames with meteors and even the international space station. Some of the frames with anomalies are shown in figure 18 and 19 below.

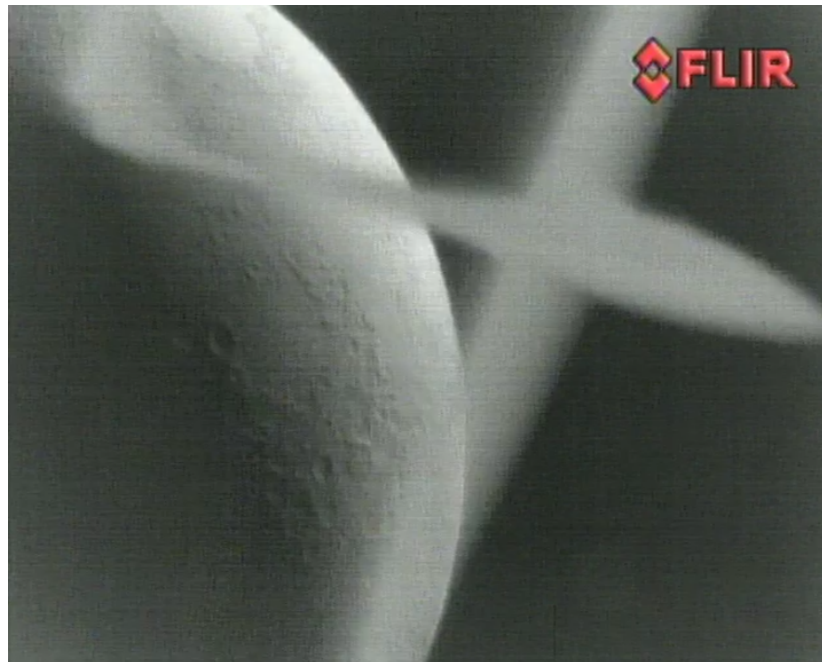


Figure 18: Glider caught during data collection

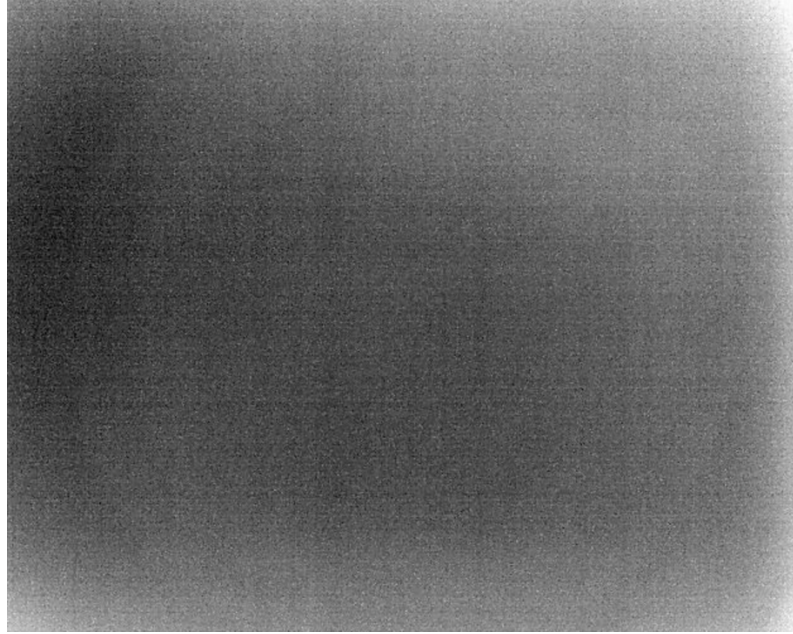


Figure 19: Frame out of focus

All such frames were discarded from the data set.

Step2: In the second step, the frames were put in a video format, creating video files (Avi format), made up of 450 frames each. The video files make it visually easy to see the Moon moving through time as oppose to just raw images which wouldn't tell which way the darkness on the surface is moving. The 450 number was chosen using hit and trial method. This was accomplished through an automated Matlab script that reads in 450 frames taken back to back and concatenate them together to make a video file. The script created is in Appendix at the end. After this step, all the data is sorted and labeled as Filename.Avi and corresponding Filename.Tiff for each day. There are equal number of video files and Tiff files for all the four quadrants pertaining each day.

Creating Time series: An attempt to create the special-temporal time series of the Lunar Surface is made by experimenting on few selected locations. Once the methodology is created it can be extended to the whole Moon surface.

7.2 Short Time Series

To get started a clear day is picked to minimize the atmospheric disturbance that would influence the pixel values. August 14, 2014 was selected as the test day. To create a small-time series, a Matlab script is created to perform image registration on each video file containing 450 frames. All the frames must be aligned, to analyze the change in the pixel value for any specific point in the given frame over time.

From the above described three Image registration techniques the *Intensity Based Image Registration* is chosen. Since, there is a lot of data to process one needs a technique that's automated. Following experiments were performed applying this technique:

Experiment 1: In this first experiment, the first Matlab script for Intensity based Image registration is created. To check the registration accuracy, it is first performed on one video file to register all the 450 frames. In the newly created registered file, all the frames should be aligned. Figure 20 shows the registration error in one video file. The registration is done by using the middle frame (Frame 225), as the reference frame. In the Figure below, Frame 1 and Frame 225 are shown adjacent to each other. Using four scales and carefully placing the one end on the center of the same small crater in both frames, one can see how much error is introduced. Vertically the error is negligible since it is less than 1 pixel and horizontally the error is about 3 pixels.



Figure 20: Registration error in one Video file

Once the registration is done one can get a change in the pixel value over the time span of 450 frames. But since the total time for each video file is approximately 15 seconds, any change in pixel value would be very small and not much useful.

Experiment 2: To get any useful pixel value trend, the pixel should be observed over longer periods of time. One way to do this is by first registering individual video files and then eyeballing the same pixel location on all the files and saving the pixel values. So, in this experiment it was attempted on two video files. They were first registered individually and for each file the pixel value for the same test location on the surface was read and saved manually. Total number of pixels obtained for the test location were 900 (2*450).

Limitations: The limitations of the Small-time series are obvious. From the individual Tiff files, not much information can be extracted as the change in pixel value over less than a minute in time, doesn't change that much. By manually selecting a point on the surface of the Moon for each Tiff file, and saving the pixel values in an array before plotting, gives us the result but this process is not practical. It would take an unknown amount of time to select every surface point manually in each video file to plot it. Also, due to eyeballing for the same point in each file, the error would be unknown. These limitations lead us to create Longer time series to get the result with a practical approach.

7.3 Hour Time Series

Keeping in mind the limitations of the shorter time series, a longer time series is created. To accomplish this, various small experiments were performed to check the validity and accuracy of the current Image registration routine.

Experiment 1: Intensity Based Image Registration

For this experiment a short script was written to combine two Tiff files, with 450 frames each, to make a new Tiff file with 900 frames. Then all the 900 frames are registered using the same intensity based image registration routine. This was attempted a few times, combining different sets of Tiff files or in other words experimenting with combining Tiff files in different instances of time.

The image registration routine is expected to register the two Tiff files with the same accuracy, but fails to do the job. It is observed that none of the frames are registered properly. Registration between Frames 1 and 900 looks good but it happens just by

chance. Various iterations give the same type of results. The result can be observed from Figure 21, 22 and 23 below. (Reason for bad registration: Rotation of Moon)

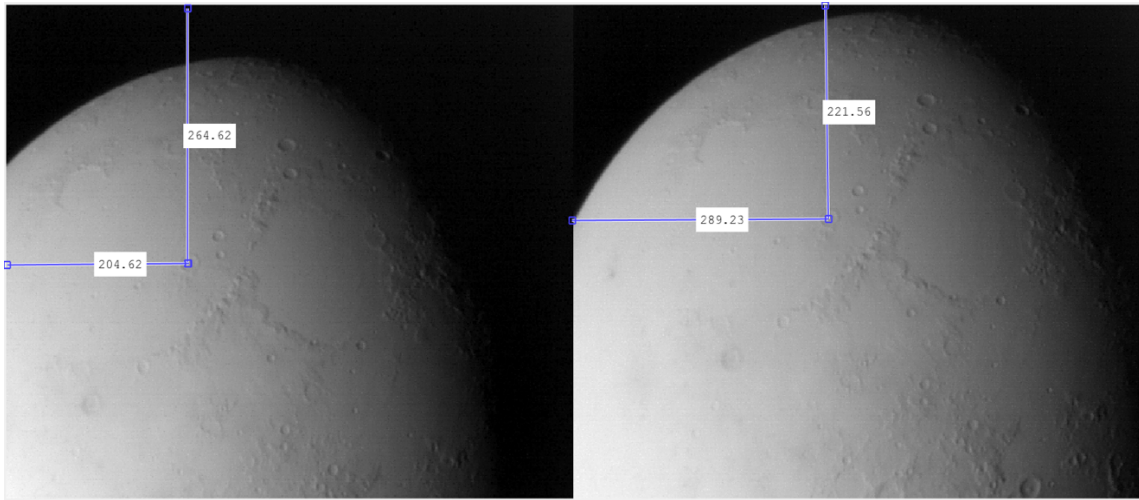


Figure 21: Frame1 and Frame 450 (Registration Bad)

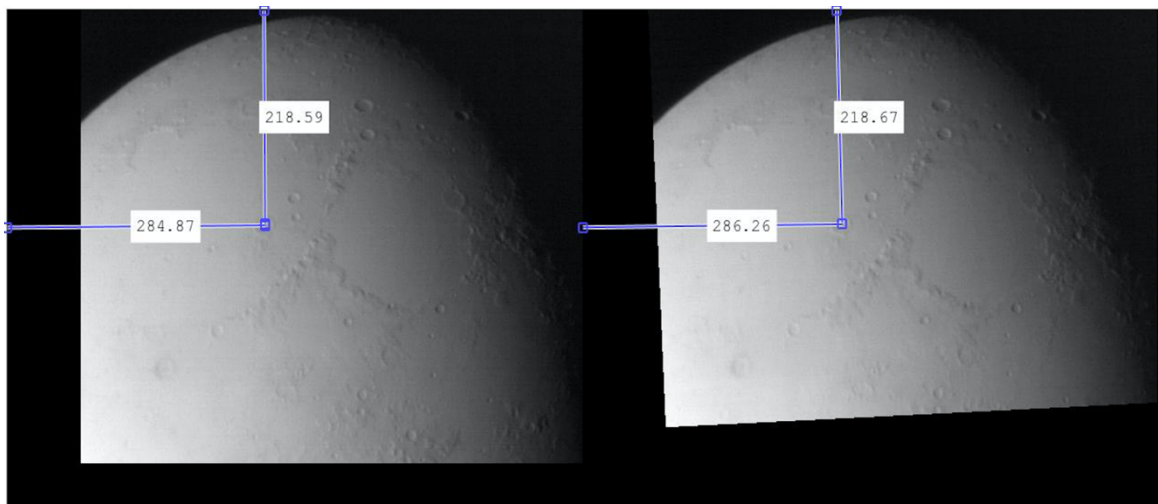


Figure 22: Frame 1 and Frame 900 (Registration good)

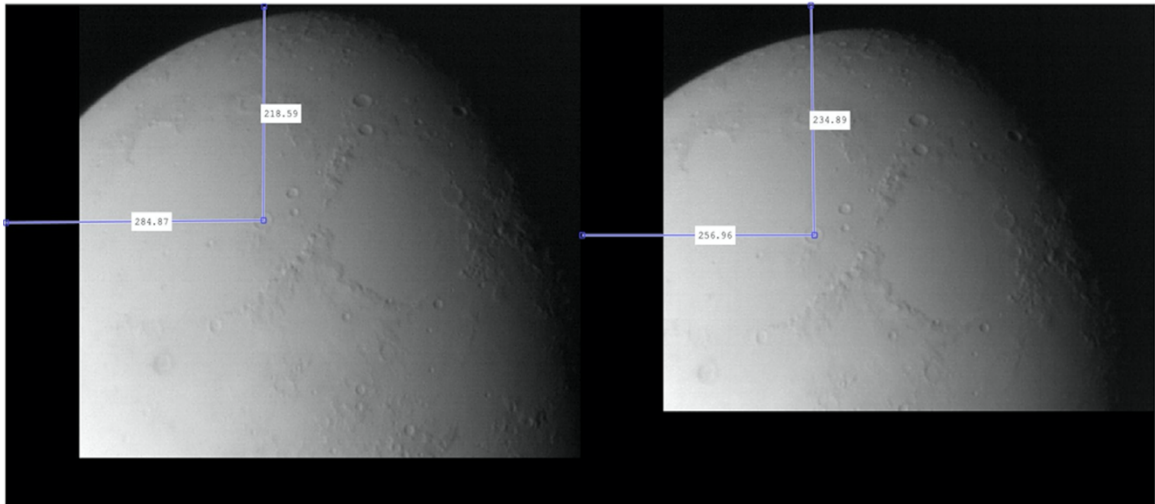


Figure 23: Frame1 and Frame 451 (Registration Bad)

Figure 21 shows frames 1 and 450 are registered with a vertical error of 33 pixels and a horizontal error of 15 pixels. Figure 20 shows frames 451 and 900 again registered with a no vertical error and a horizontal error of just 2 pixels. But Figure 22 shows frames 1 and 451 registered with a vertical error of 16 pixels and a horizontal error of 28 pixels. This is a significant error since the resolution per pixel is 2.6 Km and 28 pixels would mean an error of 72.8 Km.

Hence, it is determined that the Intensity based registration script alone is not good enough for longer time series.

Experiment 2: Control Point Image Registration

To encounter the problem of large registration error, a separate routine, this time based on Control point registration is implemented to register in between Tiff files. The downside of this technique is that it makes the process manual but the registration error is significantly low. The new script loads in two video files (from the Upper Right Moon quadrant data set) in Tiff format and reads the middle frame for both the files. Using the

cpselect, Matlab function, a window allows us to select 4 control points corresponding to the same locations in both frames, as explained earlier under Image registration section. Through this first step the two middle frames are aligned, which eventually help in aligning all the 900 frames in both Tiff files. One of the files act as fixed/reference file while the other file is the moving file that is to be registered. The output of the script is the moving file, with all the frames registered with respect to the fixed Tiff file.

For this experiment eight different files are registered with respect to the same fixed file (used as reference), using the same procedure. Once all the eight registered files are obtained, a separate script concatenates all the nine files (including the fixed file), to make one large Tiff file with 4050 frames registered.

This is the final Tiff file for one quadrant (Upper Right quadrant) of the Moon for August 14th from about 2:30 A.M. to 6 A.M., which accounts for 3 and half hours.

Limitations: The limitation of the Hour Time series is that the registration process becomes manual with the introduction of the Control Point Image registration, although its way faster than the Intensity based Registration. It was realized later in time that the Hour Time series is still very short when seen from the time frame on the Moon, since one Moon day is 29.5 days on Earth. With the interest in data over half the Lunar cycle one would have to create a longer Time series by taking data from different days and account for 14 days on Earth to get pixel values from the morning to night on Moon.

7.4 Final Time Series

The previous experiments are not a waste since they helped us develop the registration technique which can now be used for any type of dataset. For the Final Time Series,

select days that would constitute half a Lunar cycle and further select the best files for one quadrant on the Moon. After carefully looking at the data 7 different files for seven days starting Aug. 5th and ending Aug.17th were selected This is explained in detail under the Results and Analysis section.

Limitations: The limitation of the Final Time series is that, again the control point registration makes the process manual. There is no additional limitation added for the Final time series other than being limited to the data at hand. At this point it was realized that the data at hand is not enough to cover, for example the full Lunar cycle. The reason being some days were too cloudy to gather data.

7.5 Error Analysis

Data from Experiment 2 is used for the Error analysis. In the nine registered files in Experiment 2 above, the fixed file is loaded in the workspace. It is also known that while registering the individual files, the middle frame is used as a reference. This implies that the Frame 225 in the fixed Tiff file is the reference frame for rest of the 4049 frames. This frame is useful in determining the registration error and is called the reference frame.

A simple Matlab script is written to display any frame and placing a red rectangle at any point on the image. A small crater is picked in the reference Frame and a 2x2 pixel rectangle is superimposed on it. Now the script can be used to see how the rectangle moves in any other frame. Two test cases were formed as shown below:

Test Case I: Rectangle is superimposed on all the middle frames in each of the 9 files. It is determined that the registration error is less than 1 pixel for all the middle frames.

Figure 24 shows three out of the nine test frames with rectangles displayed on them. It can be observed that the rectangle doesn't move for Frames 225, 1525 and 3825 (from left to right).



Figure 24: White Rectangle on the same crater in all Frames

Test Case II: Various frames are analyzed visually and the ones, with the most deviation are analyzed further to find the maximum error. In this case, it is found that Frame 3150 has the maximum horizontal deviation w.r.t. the reference frame and Frame 3175 has the maximum vertical deviation w.r.t. the reference frame. Both are shown in Figure 24 and 25 respectively.

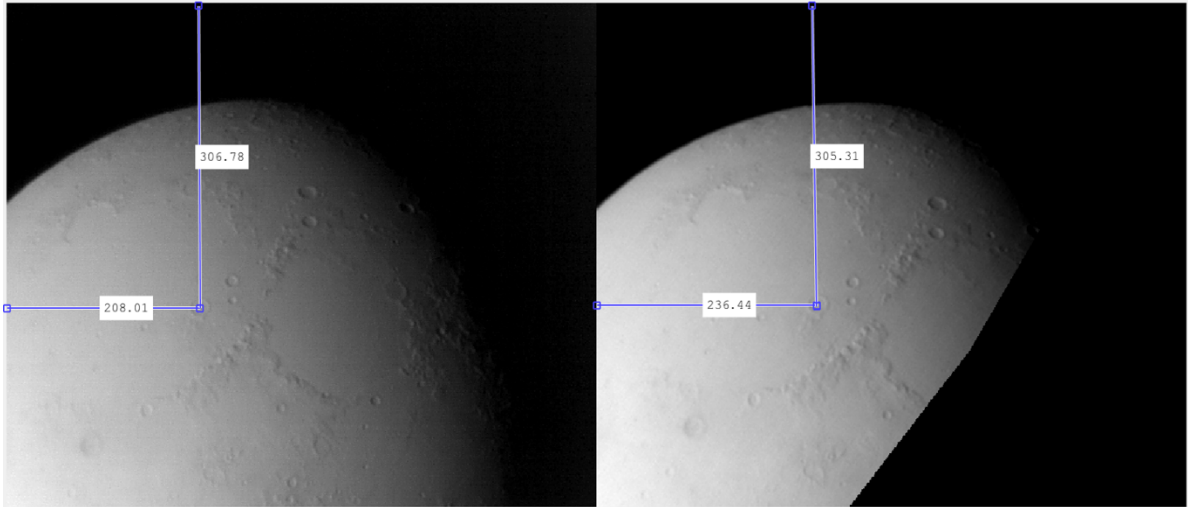


Figure 25: Frame 225 and 3150 showing maximum horizontal error

In Figure 25: Horizontal error: 28 Pixels

Vertical error: 1 Pixel

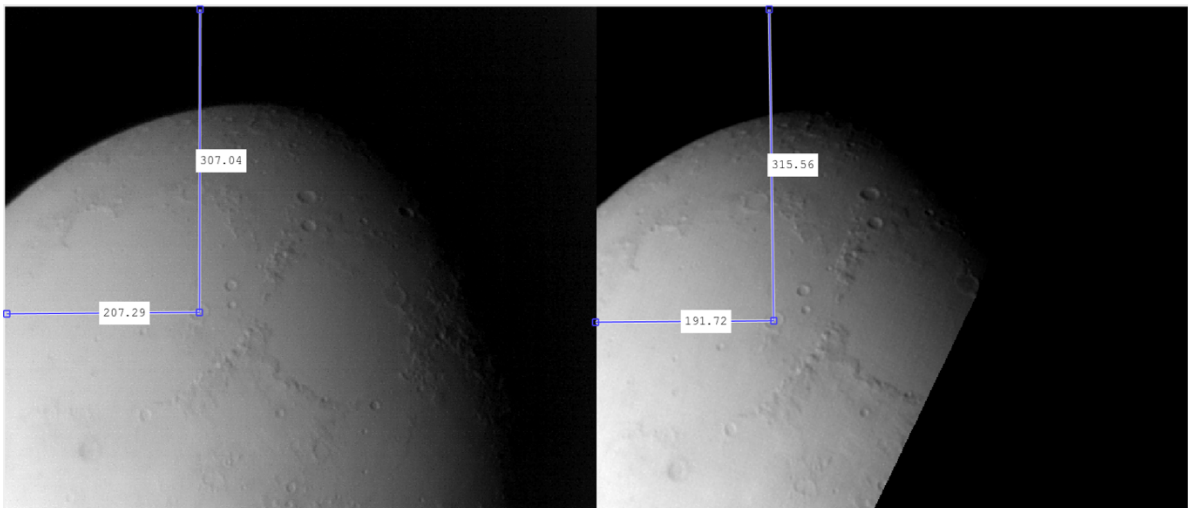


Figure 26: Frame 225 and 3175 showing maximum vertical error

In Figure 26: Horizontal error: 16 Pixels

Vertical error: 8 Pixels

It is noted that the vertical and horizontal errors in over 97% frames is less than 5 Pixels. Only few frames have higher errors with maximum errors in frames shown above.

7.6 Background Normalization

Analyzing the results, it was realized that the jumps between the different Tiff files are partly because of the Background noise. The background sky pixels should be constant in all the frames as the sky should have a constant temperature. It is observed that the sky pixel value is not constant which is due to the camera response going up with fluctuating temperatures. With temperatures going down the response increases and in return the pixel values also go up. So, to reduce those gaps and make the curves smoother, background normalization is performed on each file and new final Tiff file is saved. The process is explained in detail under the Software design section. Over here let's analyze some before and after images to see the difference. Figure 27 below shows the pixel plot before the normalization and the Figure 28 shows the plot after the normalization, for the hour Time series. It can be noted that after the normalization it is much easier to see the

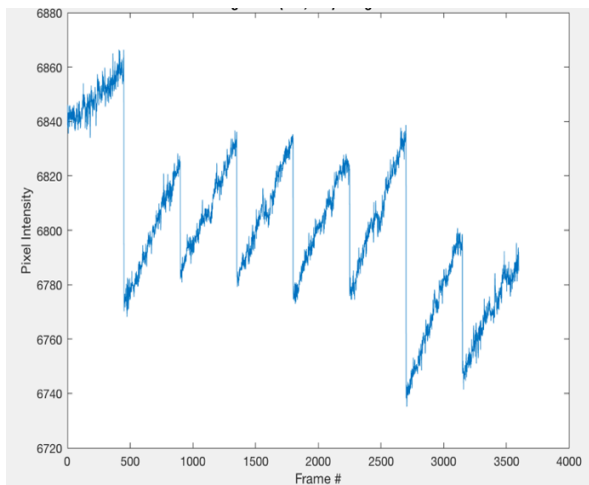


Figure 27: Hour time series plot before normalization

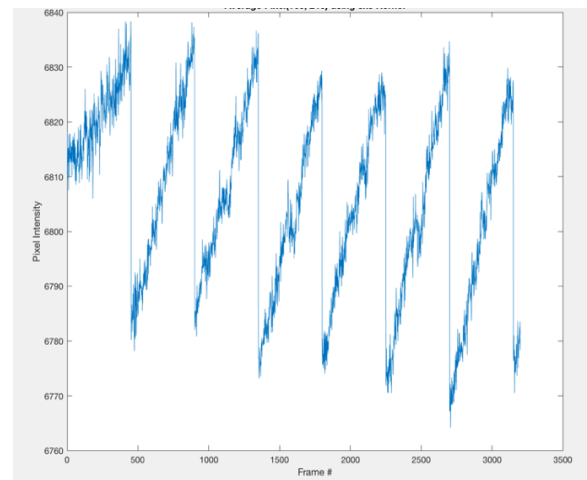


Figure 28: Hour time series plot after normalization

pixel trend.

The Background normalization was also done for the Final Time series. The data from different days made this tricky as some of the frames didn't have the background sky at all, which made obtaining the average sky pixel value hard. Figure 29 and 30 below shows the results. The pixel trend is lot more clear in Figure 30.

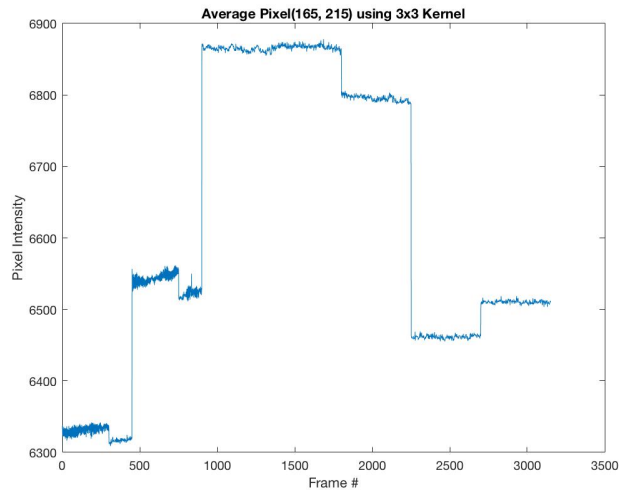


Figure 29: Final time series plot before normalization

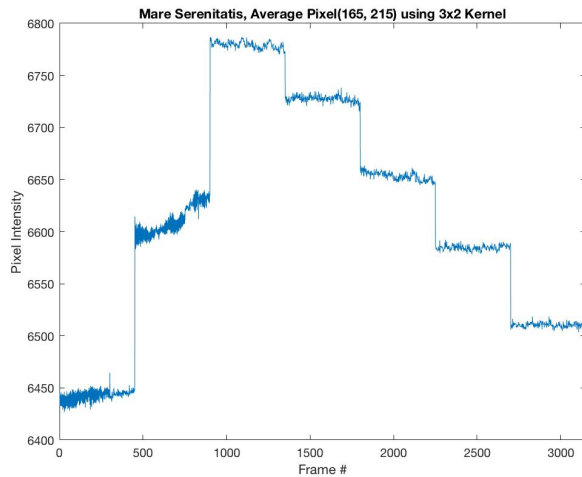


Figure 30: Final time series plot after normalization

8. Results and Analysis

The three different time series were analyzed throughout the experiment. Although the Short time series don't contribute much to the results, it's the first building block of the image registration process which helped throughout the process of building the code.

Two locations on the Moon surface were chosen to be analyzed. Figure 31 and 32 below shows the selected locations.



Figure 31: Mare Tranquillitatis region on Moon

Figure 31 shows the region of the Moon called the *Mare Tranquillitatis*. The historical significance of this region lies in the fact that this is where the first manned Moon landing took place. This region is believed to have been formed during the early history of the Moon due to a large impact after which the crater was flooded with mare basalts making it smooth and flat [Wiki Contributors]. That is the reason why it also appears dark from the surface of Earth. Its long known that the Thermal inertia of the Moon, depends on the

dusty surface rather than rocks. Per NASA data Lunar surface has 50 times higher Inertia values than Lunar Rocks. This is the reason for choosing this location.



Figure 32: Mare Serenitatis region on Moon

Figure 32 shows the second chosen location on the Moon, called the *Mare Serenitatis*. Just to the West of *Mare Tranquillitatis*, this region is similarly smooth and flat. The other reason for choosing this location is the data set at hand. The infrared data obtained for this experiment contains images from August 5th to August 17th, which constitutes approximately half the Moon cycle. The infrared images from August 5th shows half of this region in darkness making the other half see the day light. As the month progresses the darkness moves to the West lightening this region more and starts creeping back from the East. The images from August 17th show the *Mare Tranquillitatis* region to the right in darkness. This could show some interesting results in the pixel trends, as to how the pixel values change as the day progresses.

For all the three chosen locations, the three different Time series were applied. A pixel is chosen somewhere in the center of the region arbitrarily to analyze. In figures above the pixel location is shown with a White dot. For all three time-series, first one pixel is plotted through time against the number of frame. Secondly, to get rid of some registration error an average pixel value over six pixels using a 3x2 kernel is calculated and plotted.

The different results are as discussed below.

8.1 Short Time Series

8.1.1 Pixel Plots

Location 1: Figure 33 and 34 shows the pixel plots for the *Mare Tranquillitatis* region.

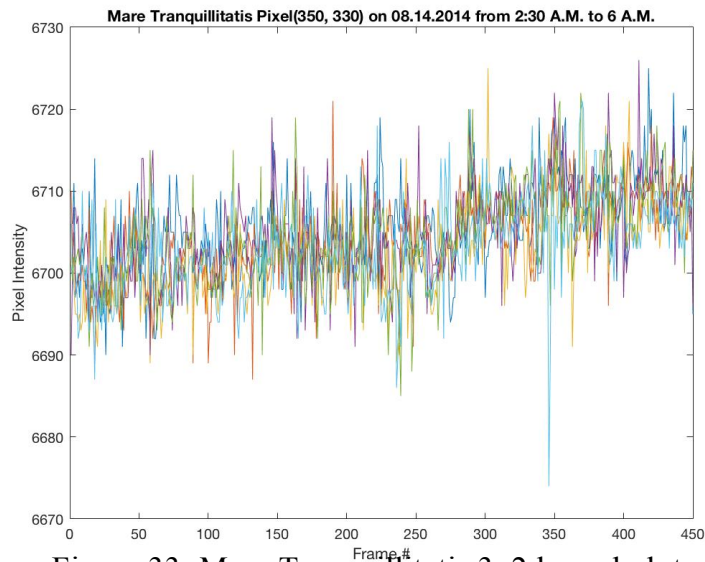


Figure 33: Mare Tranquillitatis 3x2 kernel plot

Figure 33 shows the six pixels from the 3x2 kernel all plotted against frame number.

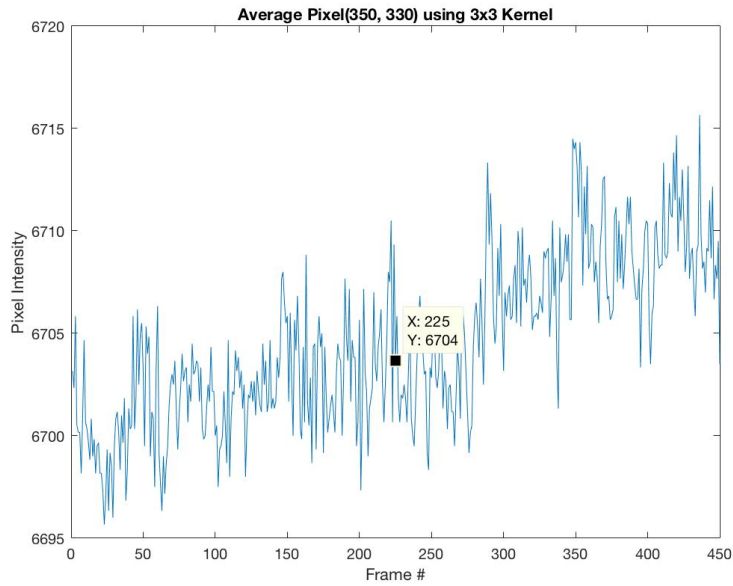


Figure 34: Mare Tranquillitatis average of 3x2 kernel plot

Figure 34 above shows the average of six pixels. The cursor on the plot further represents the average pixel value for the 450 frames all in sequence that last for fifteen seconds, in time.

Location 2: Figure 35 and 36 shows the pixel plots for the *Mare Serenitatis* region.

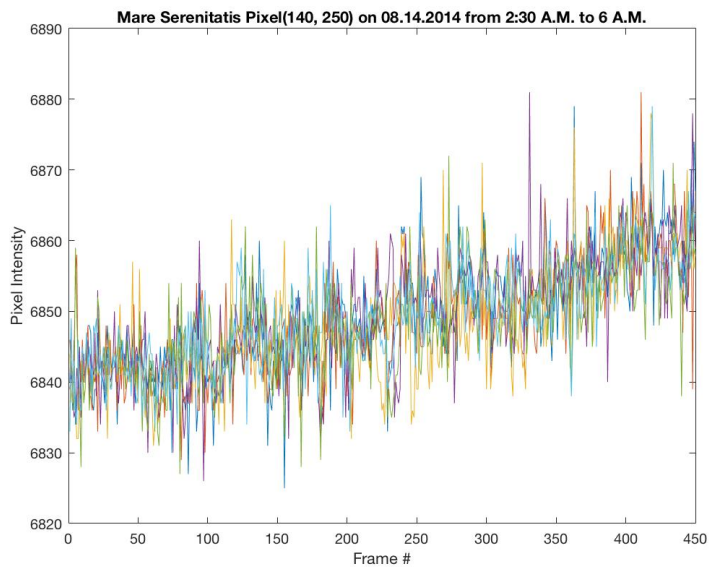


Figure 35: Mare Serenitatis 3x2 kernel plot

Figure 35 shows the six pixels from the 3x2 kernel all plotted against frame number.

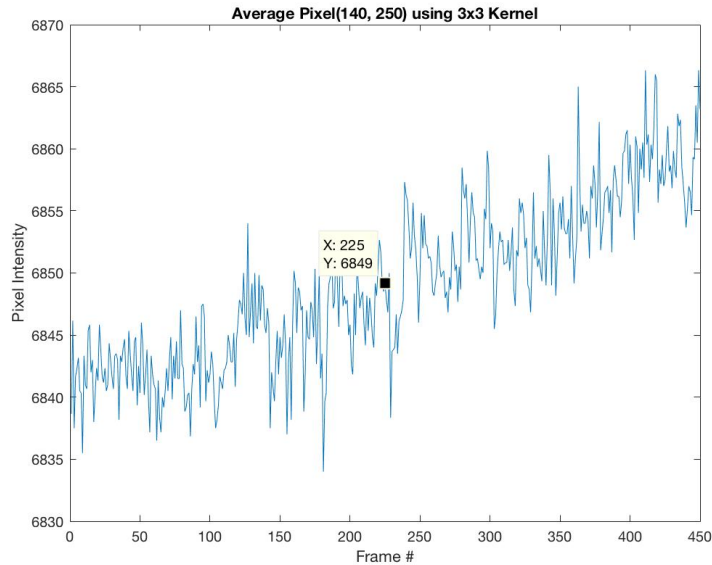


Figure 36: Mare Serenitatis average of 3x2 kernel plot

Figure 36 shows the average of six pixels. The cursor on the plot further represents the average pixel value for the 450 frames all in sequence that last for fifteen seconds in time.

8.2 Analysis

The Short Time Series is the first step towards building the Longer Time series. Not much information about the pixel change can be derived from the graphs above. The pixel change of about 20 is because of the Earth's atmospheric disturbances and the camera response going up as it gets cold. The infrared camera is made from resistors embedded on a substrate and the resistivity of the resistors go up with temperatures going down, hence higher pixel values. There shouldn't be any pixel value change since the 450 frames last only for about 15 seconds which further means much less time interval on the Moon surface.

8.3 Hour Time Series

8.3.1 Pixel Plots

Location 1: Figure 37, 38 and 39 shows the test pixel plots for the *Mare Tranquillitatis* region over a period of 4 hours on August 14th.

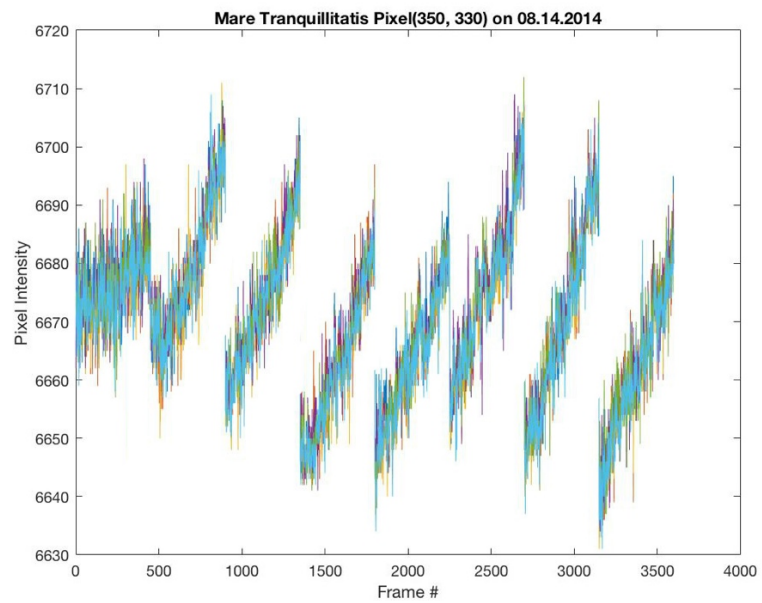


Figure 37: Mare Tranquillitatis 3x2 kernel plot

Figure 37 shows the six pixels that make up the 3x2 kernel, plotted against the frame number on the same graph.

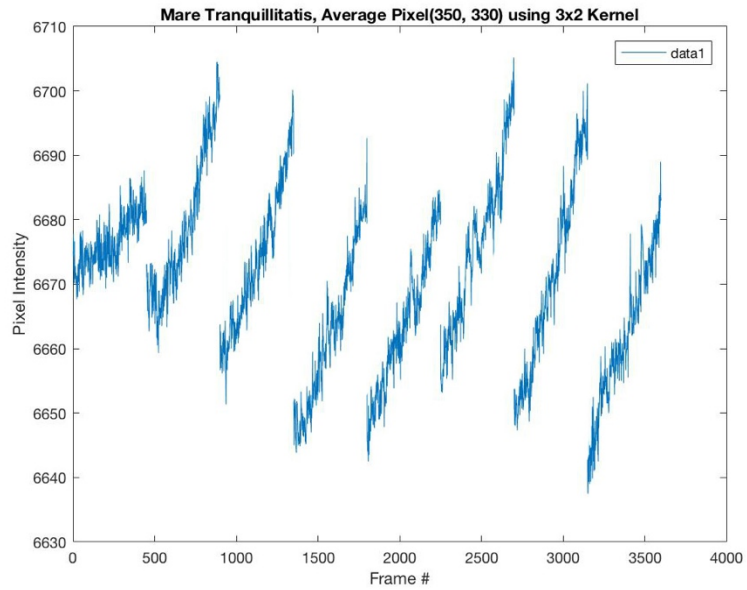


Figure 38: Mare Tranquillitatis average over 3x2 kernel plot

Figure 38 shows the average of six plots above, plotted against frame number.

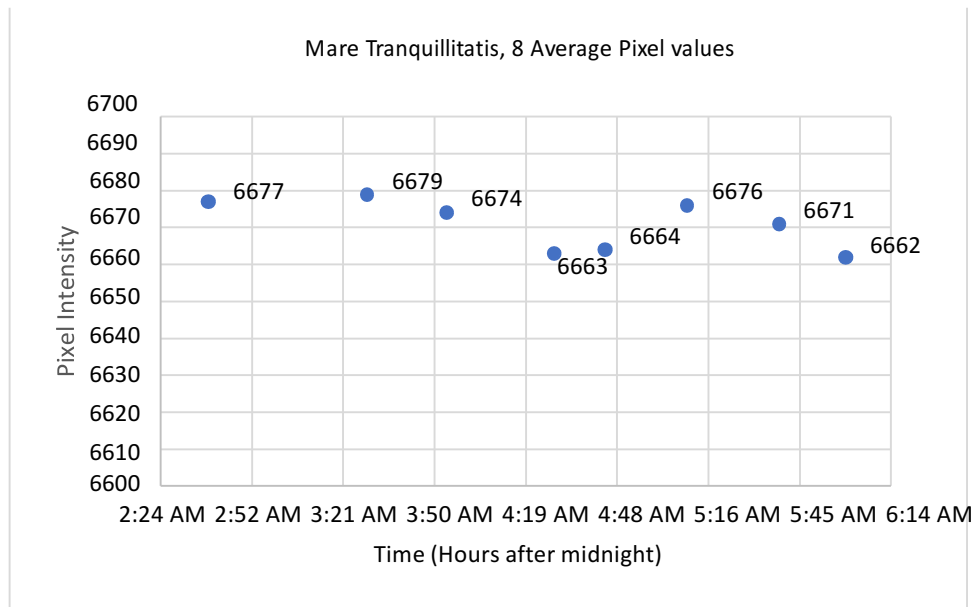


Figure 39: Mare Tranquillitatis Average pixel plot

Figure 39 shows the eight average values plotted. Each average value represents individual Tiff file containing 450 frames and is calculated by taking the average of 450 pixels for each file.

Location 2: Figure 40, 41 and 42 shows the test pixel plots for the *Mare Serenitatis* region over a period of 4 hours on August 14th.

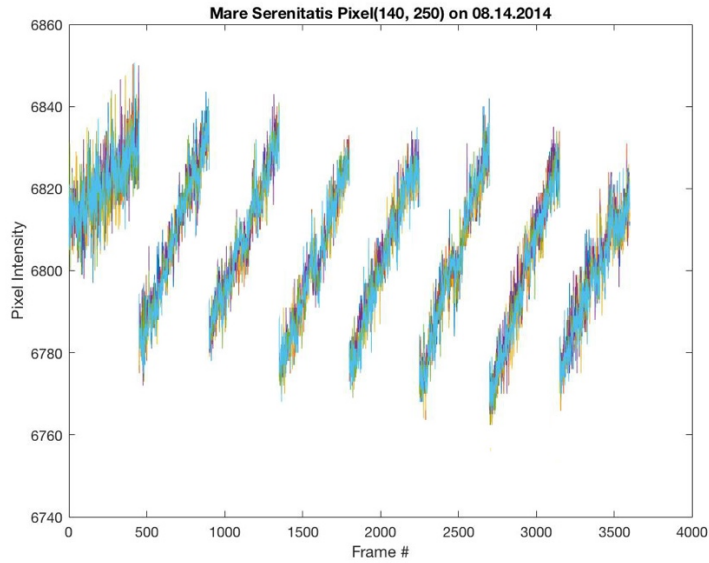


Figure 40: Mare Serenitatis 3x2 kernel plot

Figure 40 shows the six pixels that make up the 3x2 kernel, plotted against the frame number on the same graph.

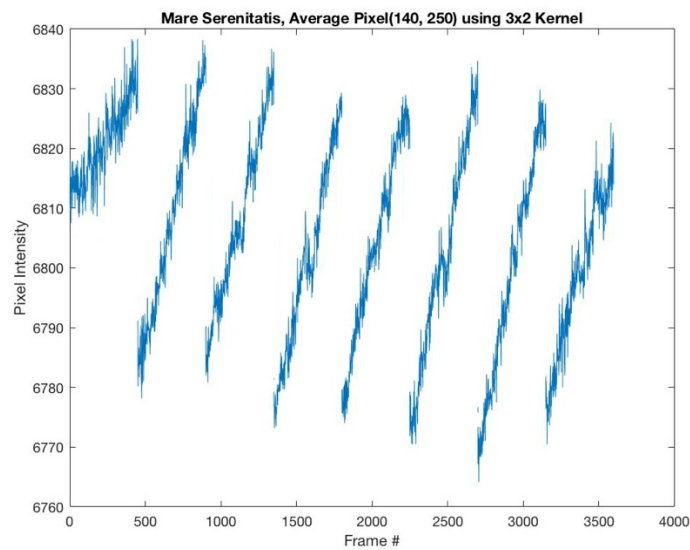


Figure 41: Mare Serenitatis average over 3x2 kernel plot

Figure 41 shows the average of six plots above, plotted against frame number.

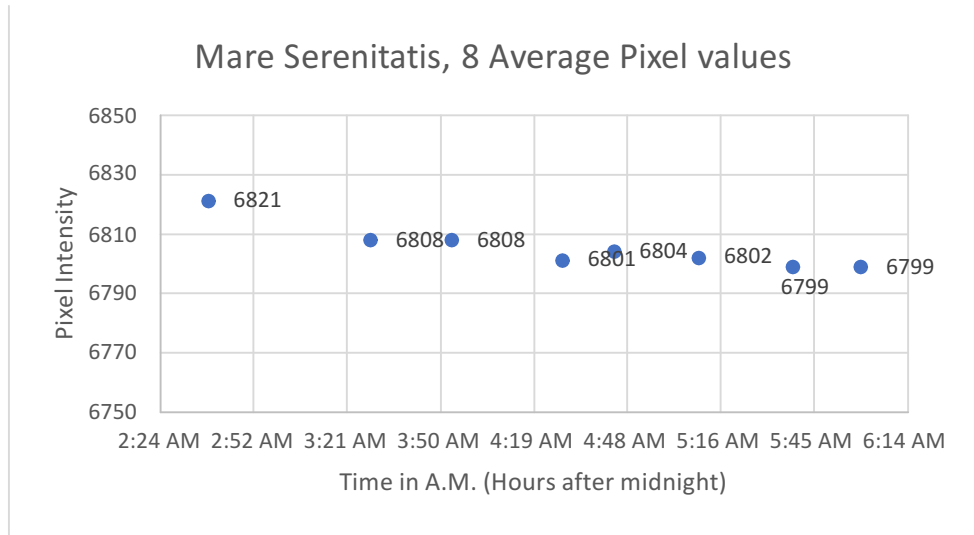


Figure 42: Mare Serenitatis Average pixel plot

Figure 42 shows the eight average values plotted. Each average value represents individual Tiff file containing 450 frames and is calculated by taking the average of 450 pixels for each file.

8.4 Analysis

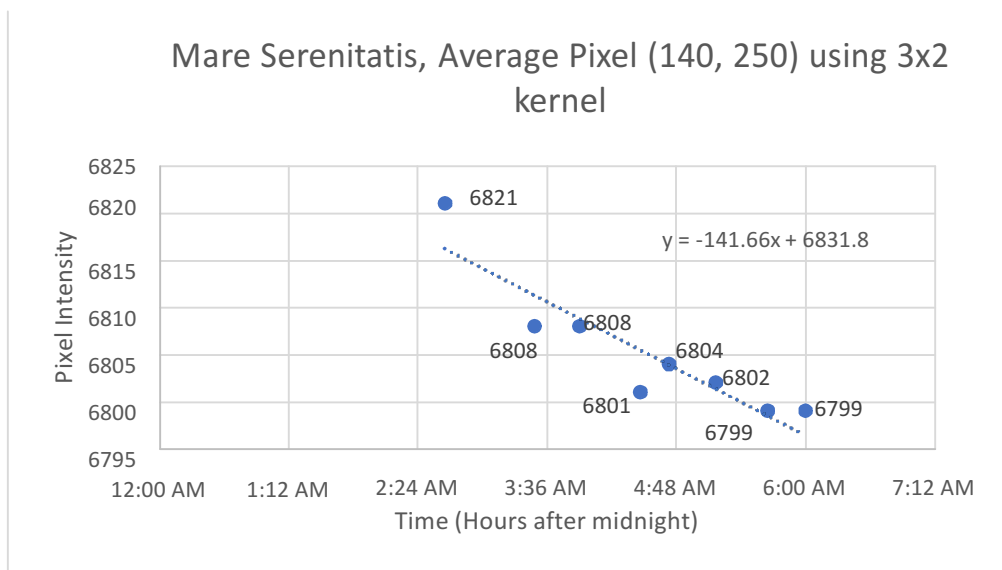


Figure 43: Detailed Mare Serenitatis average 3x2 kernel plot

Mare Serenitatis: Figure 43 above shows the detailed test pixel plot for the *Mare Serenitatis* region. All the individual data points are labeled with the best fitting curve. The y-axis shows the Pixel intensity and the x-axis represent the time on Earth after midnight when the data was collected. The data points are not at exact equal time intervals and the selected data is the best selected data in terms of having even intervals in between.

Table 3 shows the different files that make up the dataset. Data time interval column shows the time gap with the previous data point. These are chosen such as the time gap between the files could be as even as possible. But the infrared data available poses the limitation; the files selected make the best possible dataset.

Table 3: Hour Time Series dataset for Mare Serenitatis

Day [Year 2014]	Time	Data time interval	Pixel location	Avg. of 450 pixels
August 14	2:39 A.M.	0	[350, 330]	6821.0
August 14	3:29 A.M.	50 mins.	[350, 330]	6808.1
August 14	3:54 A.M.	25 mins.	[350, 330]	6808.3
August 14	4:28 A.M.	34 mins.	[350, 330]	6801.4
August 14	4:44 A.M.	16 mins.	[350, 330]	6803.6
August 14	5:10 A.M.	26 mins.	[350, 330]	6801.8
August 14	5:39 A.M.	29 mins.	[350, 330]	6798.7
August 14	6:00 A.M.	21 mins.	[350, 330]	6798.6

The test pixel for which the data is taken is under Sun's illumination on the Moon for the chosen day. Figure 44 below shows the infrared image on Aug. 14th.



Figure 44: August 14 Mare Serenitatis region with white test pixel

The pixel value starts at a highest value of 6821 at about 2:39 A.M. as shown in the table and ends in a lowest value of 6798 at 6 A.M. From the infrared image one can see the darkness on the East side of the Moon, slowly moving to the West implying it's getting cold and hence the pixel value goes down. The darkness moving from right to left in the image can be seen in the Analysis section under Final Time Series. There is only a difference of 23 in the highest Pixel value and the lowest one. The pixel value doesn't change much because the data is over approximately 4 hours and 30 minutes, Earth time. This converted to Moon time would account for much less time as one Moon day is 29.5 Earth days.

Mare Tranquillitatis: Figure 45 shows the detailed plot for the *Mare Tranquillitatis* region. The y-axis shows the Pixel intensity and the x-axis represent the time on Earth after midnight when the data was collected.

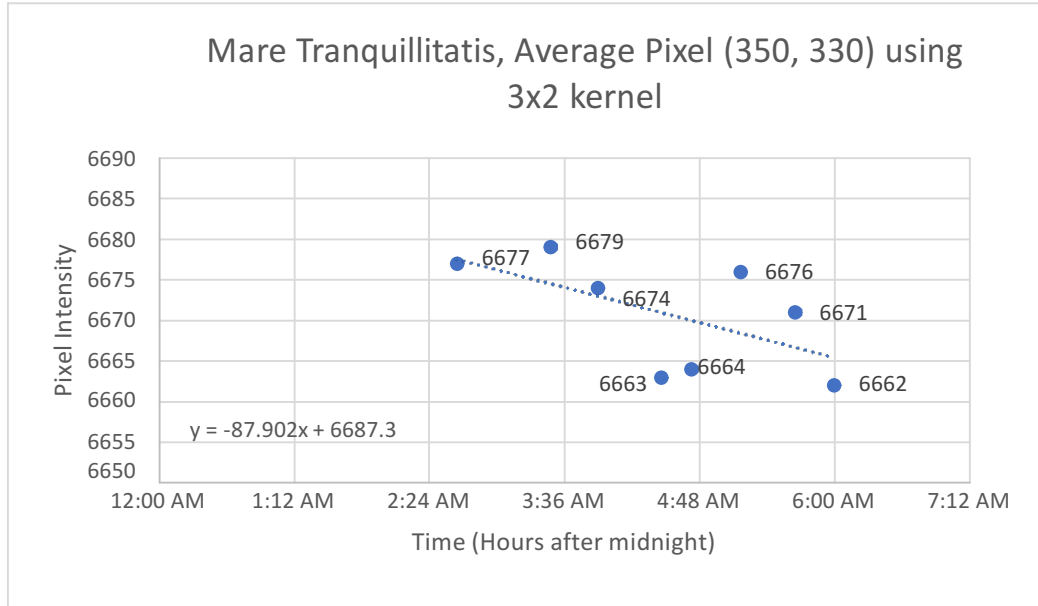


Figure 45: Detailed Mare Tranquillitatis average 3x2 kernel plot

Table 4 below shows the different files that make up the dataset for Aug. 14th from 2:39 A.M. to 6 A.M. at an arbitrarily chosen pixel location in the *Mare Tranquillitatis* region. Data time interval column shows the time gap with the previous data point.

Table 4: Hour Time Series dataset for Mare Tranquillitatis

Day [Year 2014]	Time	Data time interval	Pixel location	Avg. of 450 pixels
August 14	2:39 A.M.	0	[140, 250]	6676.5
August 14	3:29 A.M.	50 mins.	[140, 250]	6679.1
August 14	3:54 A.M.	25 mins.	[140, 250]	6673.8
August 14	4:28 A.M.	34 mins.	[140, 250]	6662.6
August 14	4:44 A.M.	16 mins.	[140, 250]	6663.8
August 14	5:10 A.M.	26 mins.	[140, 250]	6676.1
August 14	5:39 A.M.	29 mins.	[140, 250]	6670.7
August 14	6:00 A.M.	21 mins.	[140, 250]	6661.8

Similar type of result is obtained as before since both the locations are similar in formation and in terms of smoothness. Using the same reasoning, the data starts with the highest pixel value of 6679 as shown in the table and ends in a lowest value of 6662 at 6 A.M. This is again because of illumination going down and temperatures dropping. Comparing the two locations one can see that the pixel values are lower for the *Mare Tranquillitatis* region when compared with the *Mare Serenitatis*. This is simply because the *Mare Tranquillitatis* is closer to darkness, which means it colder than the other location.

8.5 Final Time Series

8.5.1 Pixel Plots

Location 1: Figure 46, 47 and 48 shows the pixel plots for the *Mare Tranquillitatis* region over a period of half a Lunar cycle using data from 7 days, starting August 5th and ending August 17th.

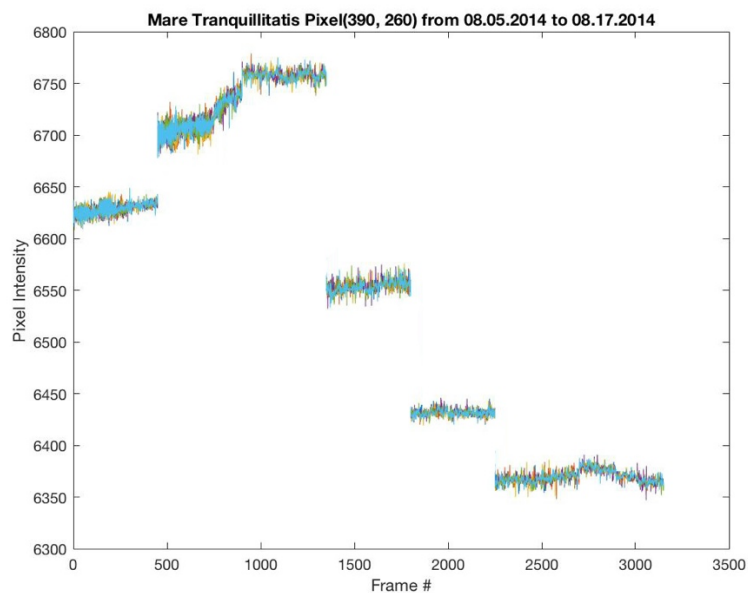


Figure 46: Mare Tranquillitatis plot for 3x2 kernel

Figure 46 shows the six pixels that make up the 3x2 kernel all plotted against frame number.

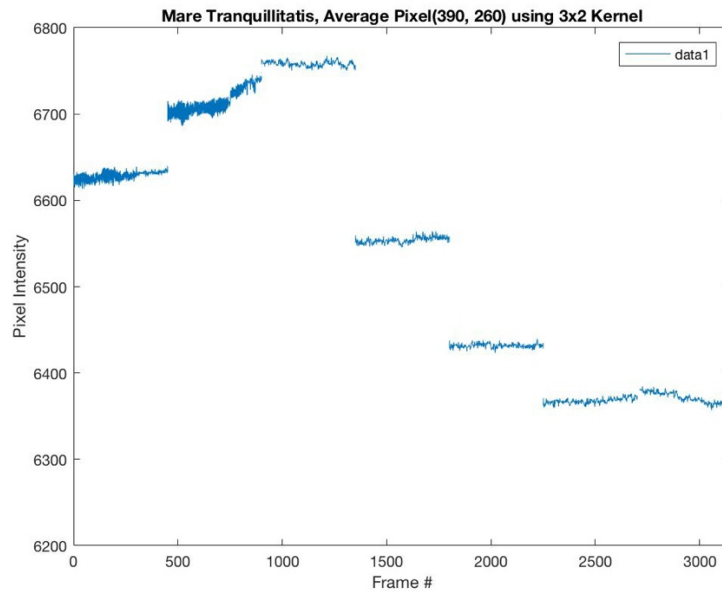


Figure 47: Mare Tranquillitatis plot for average of 3x2 kernel

Figure 47 shows the average of six plots, plotted against frame number. There are 3150 Frames in the bigger Tiff File created for the Final Time series, with sets of 450 frames representing each of the 7 different days. Again, each of the 450 frames are taken consecutively over a period of just 15 seconds in time.

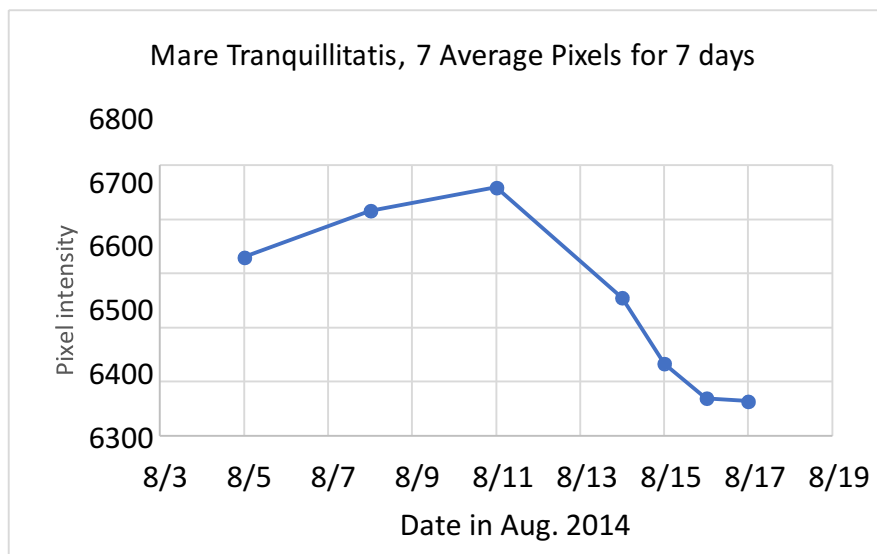


Figure 48 : Mare Tranquillitatis plot for 7 Average values

Figure 48 shows the seven average values plotted. Each average value represents individual Tiff file containing 450 frames and is calculated by taking the average of 450 pixels for each file.

Location 2: Figure 49, 50 and 51 shows the pixel plots for the *Mare Serenitatis* region.

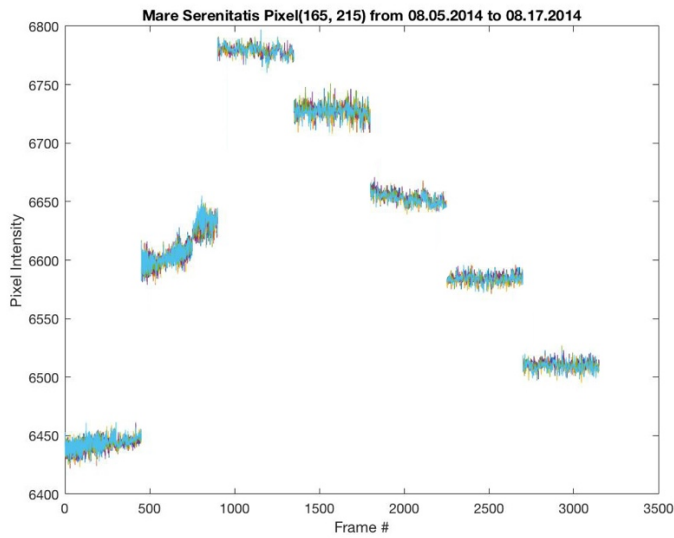


Figure 49: Mare Serenitatis plot for 3x2 kernel

Figure 49 shows the six pixels that make up the 3x2 kernel all plotted against frame number.

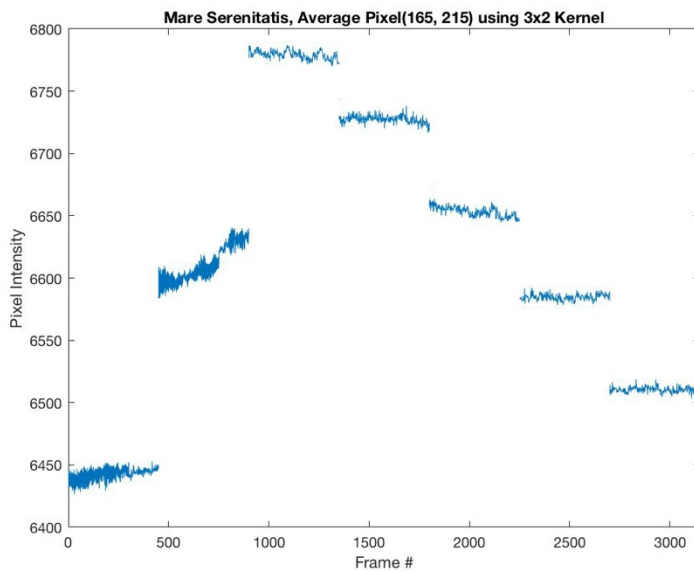


Figure 50: Mare Serenitatis plot for average of 3x2 kernel

Figure 50 shows the average of six plots, plotted against frame number. This plot simply shows the 3150 pixel values, one for each frame in the large Tiff file.

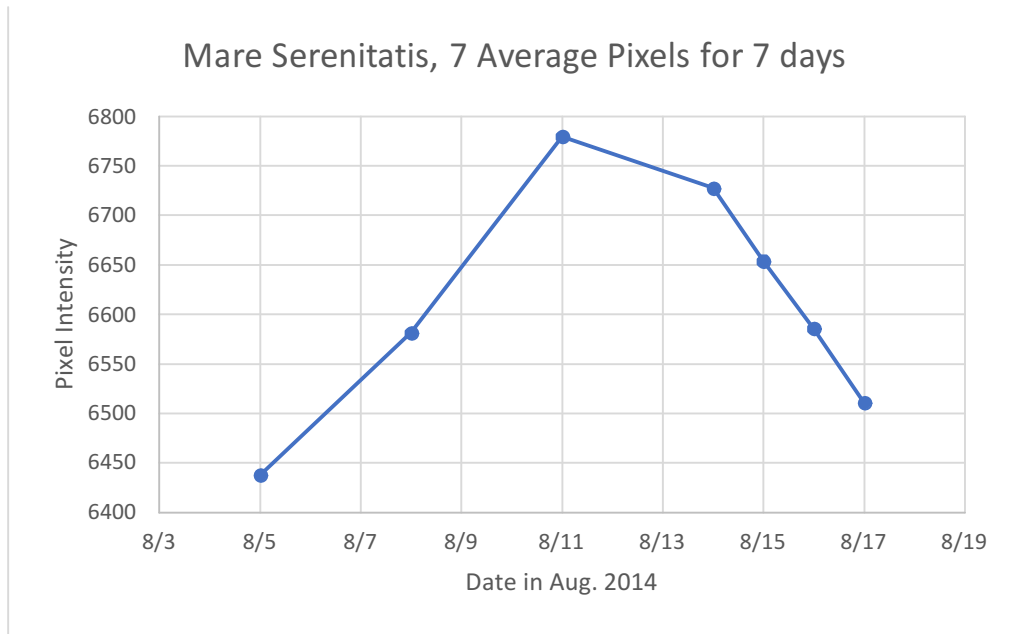


Figure 51: Mare Serenitatis plot for 7 Average values

Figure 51 shows the seven average values plotted. Each average value represents individual Tiff file containing 450 frames and is calculated by taking the average of 450 pixels for each file.

8.5.2 Temperature Plots

The temperatures obtained are not the absolute true temperatures of the surface of Moon. Instead these are the infrared temperatures obtained from a calibration file that comes with the infrared camera and is based on an emissivity of 0.95.

Location 1: Figure 52 shows the infrared temperature plot for the *Mare Tranquillitatis* region over half a Lunar cycle lasting from Aug. 5th to Aug. 17th. The same test pixel location is chosen for Temperature plots as well.

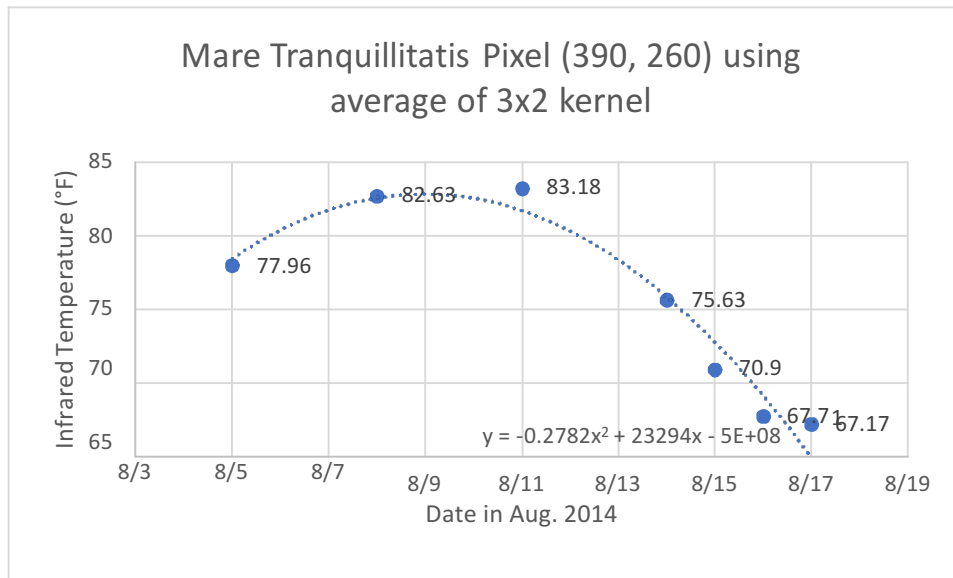


Figure 52: Infrared temperature plot for the Mare Tranquillitatis

Location 2: Figure 53 shows the infrared temperature plot for the *Mare Serenitatis* region.

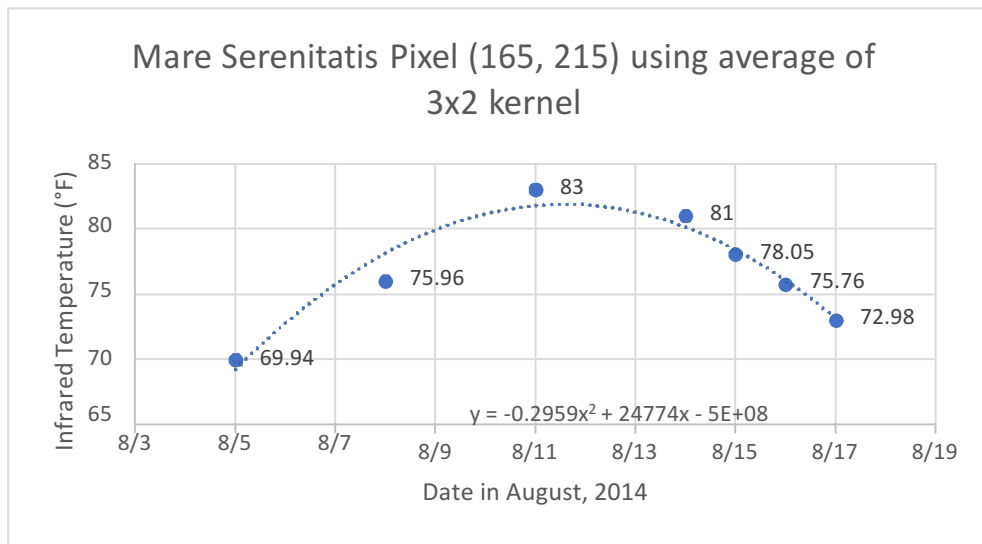


Figure 53: Infrared temperature plot for the Mare Serenitatis

Both the above plots show the best curve fits to simply see the temperature trend.

8.6 Analysis

8.6.1 Pixel Plots

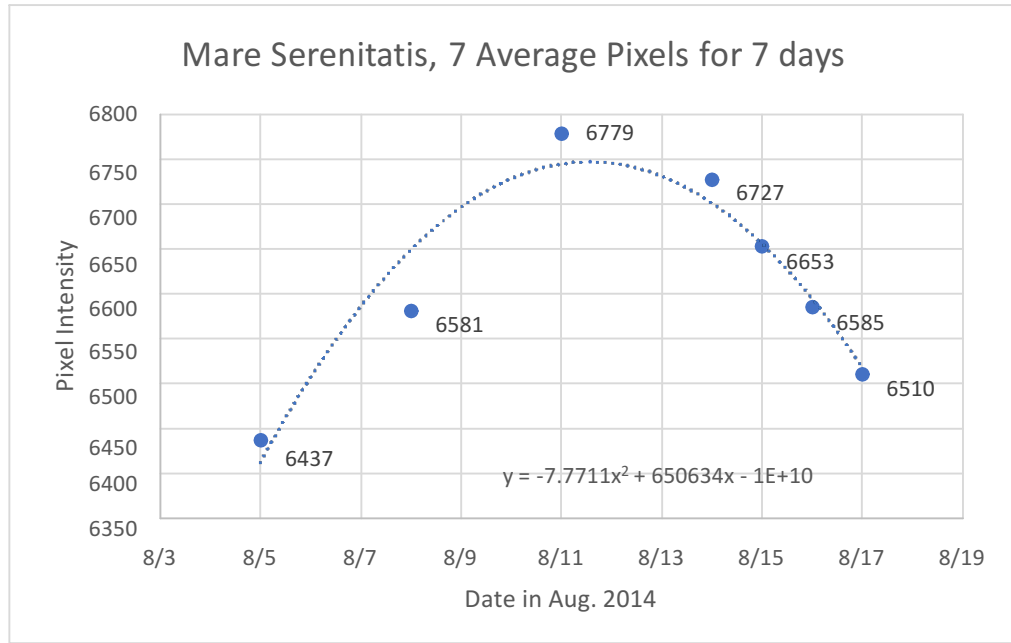


Figure 54: Detailed Mare Serenitatis average plot

Mare Serenitatis: Figure 54 shows the more detailed plot for the Mare Serenitatis location. The x-axis represents the day the data was taken in Aug., 2014. After trying different types of functions, it was determined that the 4th degree Polynomial gives the best fitting curve. The corresponding equation is also shown on the plot. From this curve the rising and the falling slopes can be calculated which would tell us how the pixels and eventually temperature changes as the day progresses on the Moon. Table 5 below shows the details for the dataset used in above plot.

Table 5: Final Time Series dataset for Mare Serenitatis

#	Day [Year 2014]	Time	Time interval	Pixel location	Avg. of 450 pixels
1	August 5	7:55 P.M.	0	[165, 215]	6437
2	August 8	10:14 P.M.	75 hrs.	[165, 215]	6581
3	August 11	00:01 A.M.	73 hrs. 47 mins.	[165, 215]	6779.2
4	August 14	2:29 A.M.	74 hrs. 28 mins.	[165, 215]	6727.4
5	August 15	3:32 A.M.	25 hrs. 3 mins.	[165, 215]	6653.2
6	August 16	4:07 A.M.	24 hrs. 35 mins.	[165, 215]	6584.6
7	August 17	4:46 A.M.	24 hrs. 39 mins.	[165, 215]	6510.4

The table shows the dates for which the data was selected. The dates selected depends on number of factors such as: the data should be clear to get best results, the selected locations on the Moon should be visible in all the individual frames in the dataset to get pixel value, the data should cover half the Lunar Cycle (day to night), and data with as close and even time interval in between as possible. Other than the above factors the experiment is limited by the availability of the data on different days. For instance, there were some days that were too cloudy to capture data, so data for those days is not available to us. The data time interval shows the exact hours in between each data point. The hours are accounted for in the plots but the minutes are all approximated. Figure 55 to 58 below shows the darkness moving on the surface of the Moon w.r.t. the Sun starting with Aug. 5th.

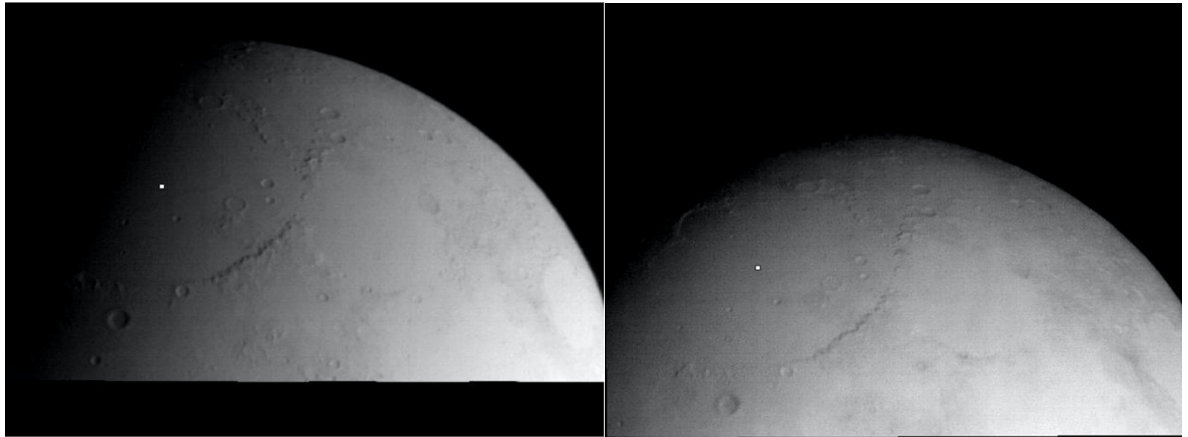


Figure 55: August 5(Left) and August 8(Right)

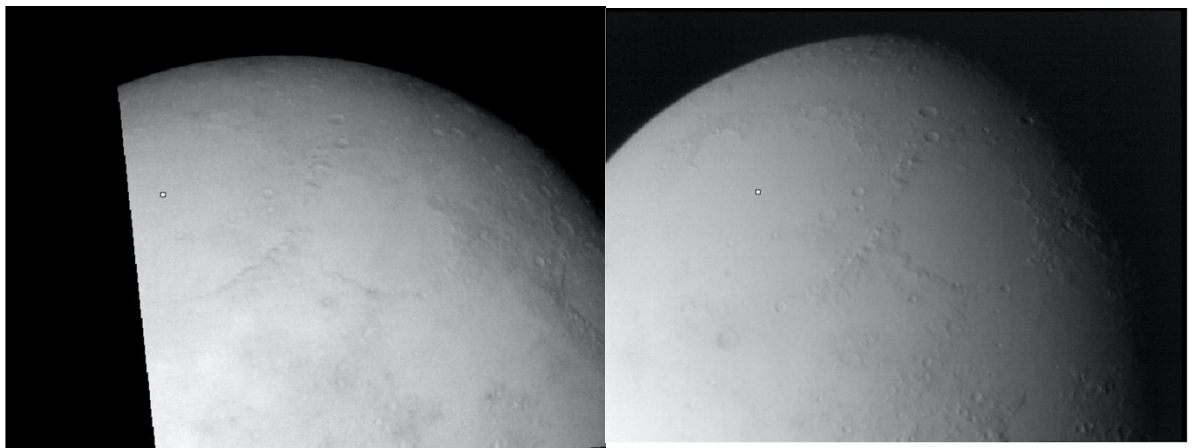


Figure 56: August 11(Left) and August 14(Right)

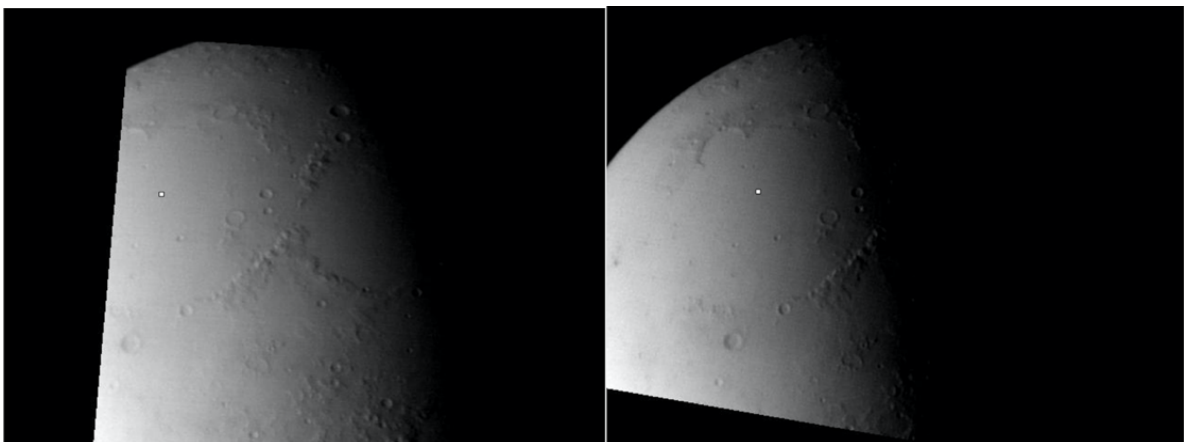


Figure 57: August 15(Left) and August 16(Right)

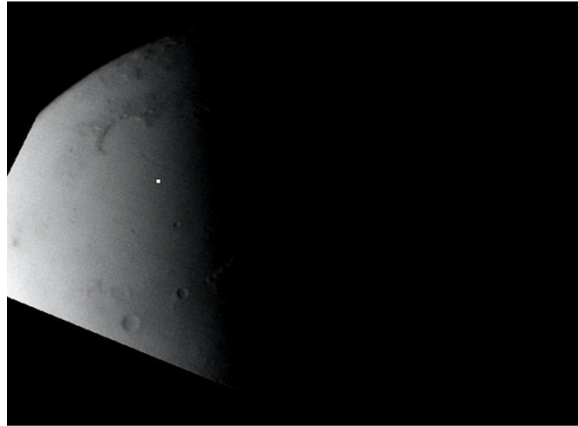


Figure 58: August 17

The white dot on each figure represents the test pixel location, shown in table above.

Since all the frames are aligned the location of the test pixel is same in each figure.

Figure 56 showing the Moon on Aug. 14th was kept as the reference frame and all the other days were aligned w.r.t. it. That's the reason why the Moon is tilted in all other figures.

This is a good dataset since the Lunar Cycle is 29 days long and from Aug. 5th to Aug.17th its 13 days which is almost half the Lunar Cycle or in other words from morning to night on Moon. One can see how the surface pixel value changes from morning to night by observing the pixel plot shown above. By carefully comparing the plot, the Average value in the table, and Figure 55, one can see that the average pixel value is the lowest (6437) on Aug. 5th since the point is very close to darkness and hence its cold. As the darkness moves away to the left the pixel value goes up as its getting hot, with the peak average pixel value of 6779 (from table) on Aug. 11th. This is also evident from the fact that Aug.11th, 2014 was a full moon day with maximum illumination from the Sun.

Figure 59 below is a good reference for seeing the Sun illumination for the month of August in 2014.

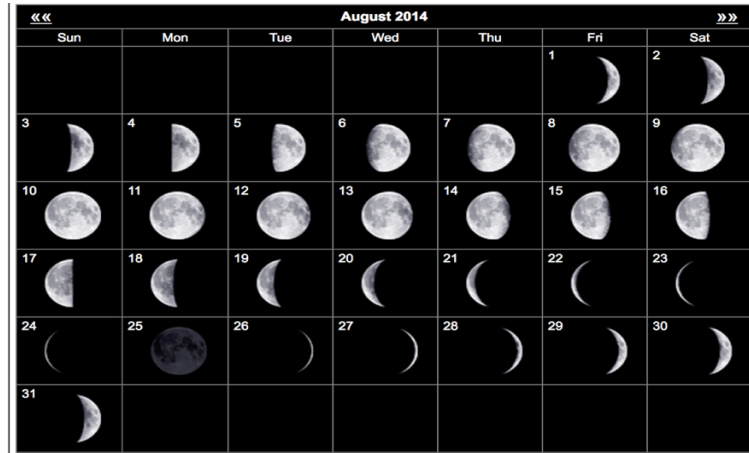


Figure 59: Actual Moon Phases in August, 2014

From the figure, it can also be seen that on Aug. 5th the left hemisphere is in darkness and on Aug. 17th the right hemisphere is in darkness which completes half the lunar cycle. Finally, on Aug. 17th the pixel value is at its minimum (6510) as it gets cold close to night.

Mare Tranquillitatis : Figure 60 above shows the detailed version of Figure 47, representing

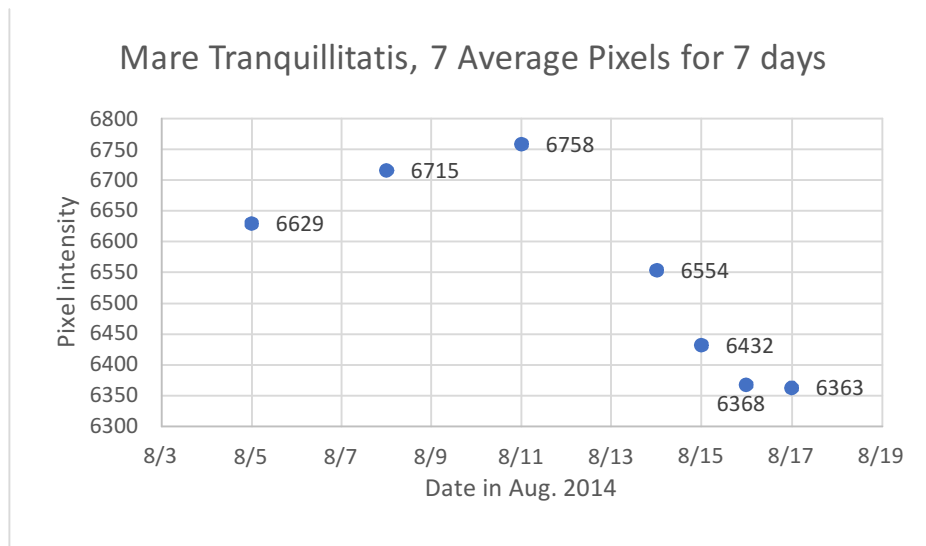


Figure 60: Detailed Mare Tranquillitatis average plot

the *Mare Tranquillitatis*. The x-axis shows the day in the month of Aug., 2014 when the data was taken. Each pixel value is an average of the 450 pixel values. The dates were selected as explained earlier. Again, from the data the rising and the falling slopes can be calculated to find how the pixels and eventually the temperature changes as the day progresses on the Moon.

Table 6 below shows the details for the dataset used for the *Mare Tranquillitatis* region.

Table 6: Final Time Series dataset for Mare Tranquillitatis

#	Day [Year 2014]	Time	Time interval	Pixel location	Avg. of 450 pixels
1	August 5	7:55 P.M.	0	[390, 260]	6628.5
2	August 8	10:14 P.M.	75 hrs.	[390, 260]	6714.8
3	August 11	00:01 A.M.	73 hrs. 47 mins.	[390, 260]	6758.1
4	August 14	2:29 A.M.	74 hrs. 28 mins.	[390, 260]	6553.9
5	August 15	3:32 A.M.	25 hrs. 3 mins.	[390, 260]	6431.7
6	August 16	4:07 A.M.	24 hrs. 35 mins.	[390, 260]	6367.7
7	August 17	4:46 A.M.	24 hrs. 39 mins.	[390, 260]	6362.7

The pixel trend for this location is slightly different than the Mare Serenitatis. Once again, comparing the plot in Figure 60, the Average values in the table, and Figures 55 to 58, it is observed that this time the lowest average pixel value of 6362.7 occurs on Aug. 17th. This makes complete sense as on that day the test point is under maximum darkness, which moves from right to left in the frames. The average pixel value of 6367.7 on Aug. 16th is also close, as the test point is again covered in darkness but slightly earlier in night. Aug. 5th value is not the lowest this time since the test point on that day is away

from the darkness. Again, the peak value of 6758 occurs on Aug. 11th, when there is a maximum illumination.

NOTE: The timings listed in Tables above corresponds to Earth time which could be confusing since one Moon day is approximately 29.5 Earth days, hence the day and night time is totally different.

8.6.2 Temperature Plots

Mare Serenitatis: Figure 61 below shows the Average Temperature plot for the pixel located in the Mare Serenitatis region.

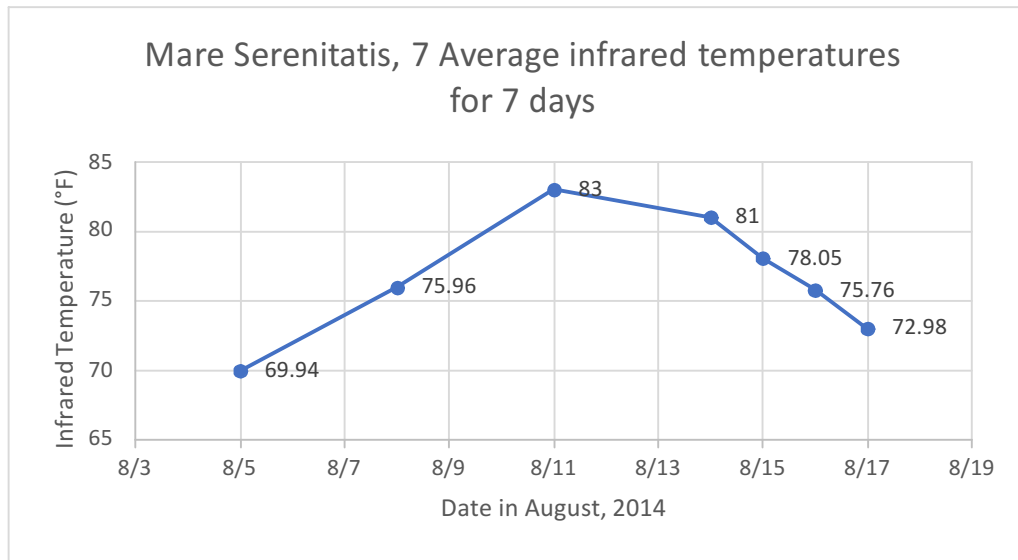


Figure 61: Mare Serenitatis Average Infrared Temp plot

The plot shows all the seven infrared temperature values for 7 different days. The results are consistent with the results derived from the pixel plots and the plots follow the very same trend. The infrared temperature value on Aug. 5th (about morning on Moon at this pixel location) is 69.94 °F and goes up to 83 °F on Aug. 11th during Full Moon, in three

Earth days. The temperature then goes down after the peak with a minimum temperature of 72.98 °F on Aug. 17th (Early night on Moon at this location).

From this plot, the slope of the temperature going up and going down can be found after the mid-day on Moon.

Rising Slope:

of Earth days = Aug. 5 – Aug. 11 = 6 days

Temperature change = (83 – 69.94) °F = 13.06 °F

Rising Slope = Temp. change / # of Earth days = 13.06 °F / 6 days = **2.18 °F/day.**

Falling Slope:

of Earth days = Aug. 11 – Aug. 17 = 6 days

Temperature change = (72.98 – 83) °F = -10.02 °F

Falling Slope = Temp. change / # of Earth days = -10.02 °F / 6 days = **-1.67 °F/day.**

Mare Tranquillitatis: Figure 62 below shows the Average Temperature plot for the pixel located in the *Mare Tranquillitatis* region.

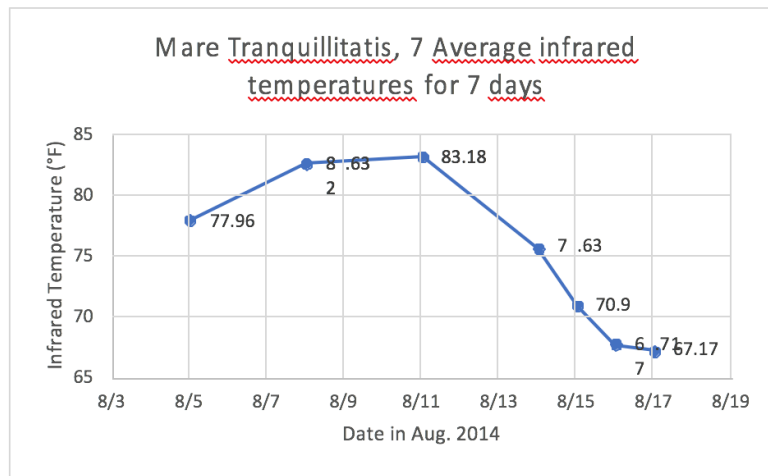


Figure 62: Mare Tranquillitatis Average Infrared Temp plot

The plot shows all the seven infrared temperature values for 7 different days. The results are again consistent with the previously derived results from the pixel plots and the curves follow the very same trend. The infrared temperature value on Aug. 5th (about mid-day on Moon at this pixel location) is 77.96 °F and goes up to 83.18 °F on Aug. 11th during Full Moon, in three Earth days. The temperature then goes down after the peak with a minimum temperature of 67.17 °F on Aug. 17th late in night (Later in night on Moon at this location).

From this plot, one can find the slope of the temperature going up and then going down after the mid-day on Moon.

Rising Slope:

of Earth days = Aug. 5 – Aug. 11 = 6 days

Temperature change = (83.18 – 77.96) °F = 5.22 °F

Rising Slope = Temp. change / # of Earth days = 5.22 °F / 6 days = **0.87 °F/Earth day.**

Falling Slope:

of Earth days = Aug. 11 – Aug. 17 = 6 days

Temperature change = (67.17 – 83.18) °F = -15.17 °F

Falling Slope = Temp. change / # of Earth days = -15.17 °F / 6 days = **-2.66 °F/Earth day.**

9. Final Analysis

From the Short Time-Series it can be learned that the camera response starts at a low and then slowly goes up while capturing different images. This can be seen from the Figures under the Short Time series section. Also, the ripple in the pixel data is somewhat due to the camera response fluctuating. But other factors such as Atmospheric disturbance, Earth temperatures changing and just the distance in general play a role. The 15-second long, stream of images makes it visually easy to see how the Moon is moving in the video sequence. Other than that, not much significant information regarding the Pixel/Temperature trends can be derived from the Short Time Series.

The hour time series gives us some good talking points as to how the pixel values for the two selected locations change w.r.t. the Sun. The first pixel location is in the *Mare Serenitatis* region that is to the West in the image shown and hence away from the darkness. The second pixel location is in the *Mare Tranquillitatis* region that is East of the first location and is closer to the darkness. Comparing the pixel values for both the locations it is observed that the Mare Serenitatis starts at a higher pixel value than the Mare Tranquillitatis, with the former going from 6821 to 6798 and the later going from 6679 to 6662. The reason being the Mare Serenitatis region is colder than the other location. However, the pixel value is going down for both the locations since both the points are moving closer to the darkness and away from sunshine as the day on Moon

progresses. The Hour Time series is still short relative to time-period on Moon, hence there is not a big change in the pixel value.

The Final Time Series gives us better results in terms of pixel/temperature change over 13 days that constitutes approximately half the lunar cycle. Luckily the dataset is available that goes over that amount of time-period, since half a Lunar cycle in other words is morning to night w.r.t. the day on Moon. It can be observed that the pixel plots for the Final Time Series is somewhat like a parabola which implies that the test pixels start in the morning with a lower value when it's cold and then goes through a peak temperature during the mid-day and eventually goes down again as it gets cold later in the afternoon. Again, comparing the pixel values for the two locations: The Mare Serenitatis pixels change from 6442 to 6779 from morning to mid-day and then 6779 to 6510 from mid-day to night. This makes complete sense with the darkness moving as shown in Figures 53 to 56. For the Mare Tranquillitatis, the pixels change from 6629 to 6758 from later in the morning to mid-day and 6758 to 6363 from mid-day to later in night. When both the locations are carefully compared with the pixel values, it is noted that the starting value for the Mare Tranquillitatis is higher than the Mare Serenitatis by about 186 pixels. Then the ending value for last data point is lower than the Serenitatis by about 148 pixels. But the Peak value is somewhat similar with only a difference of 21 pixels. The reason for this is that both the locations don't start at the same time w.r.t. to the Moon day/night cycle. For instance, the Mare Tranquillitatis region is not as close to the darkness as the Mare Serenitatis which means its little late in the morning. Since the data for both locations is the same, the Mare Tranquillitatis also approaches the night before Mare Serenitatis. Hence, the pixel values are lower for last data point. During the

full moon on Aug. 11th, both the locations are almost equally illuminated hence the pixel values are also similar. Additionally, for the Final time series the Temperature trends were also obtained by converting the pixel values to temperature values using a camera calibration file. The plots show a very similar behavior in terms of trends seen in the pixel plots. Under this section the Rising and the Falling slopes were also calculated. For the Mare Sernitatis, the infrared temperature is seen going from 70 °F to 83 °F to 73 °F at night. From the rising and the falling slopes, it is inferred that the infrared temperature rises at a rate of 2.18 °F/Earth day and falls at a rate of -1.67 °F/ Earth day. For the Mare Tranquillitatis, the infrared temperature is seen going from 77.96 °F to 83.18 °F to 67.17 °F at night. Again, from the rising and the falling slopes, it is inferred that the infrared temperature rises at a rate of 0.87°F/Earth day and falls at a rate of -2.66 °F/ Earth day.

10. Conclusion

In this study an algorithm is developed to obtain Pixel/temperature plots for any selected location on the Moon surface over half a Lunar cycle in this study (could be over any amount of time), using an infrared camera mounted on a simple telescope. The infrared camera used here operates with a peak response at 10 microns. To test the algorithm, the experiment starts by choosing pixel locations that are smooth rather than rocky or covered with craters. Such locations would be better because small registration error could be compensated, as the pixel intensity won't change much for a smoother surface as oppose to surfaces with too many craters that would show a larger variation in pixel intensity from point to point. The graphs obtained make complete sense with the darkness moving on the Moon. For instance, the parabolic shape of the curves show how a pixel value starts at a lower intensity at the edge of the darkness in morning and then hits the peak value on full moon or mid-day and then again goes low when the darkness approaches from the other side during night.

The infrared temperature is seen to rise at a rate of $2.18\text{ }^{\circ}\text{F/Earth day}$ and fall at a rate of $-1.67\text{ }^{\circ}\text{F/Earth day}$ for the Mare Serenitatis region. That means the infrared temperature value rises by $13.06\text{ }^{\circ}\text{F}$ from morning to mid-day and falls by $10\text{ }^{\circ}\text{F}$ from mid-day to night. For the Mare Tranquillitatis, the temperature is seen to rise at a rate of $0.87\text{ }^{\circ}\text{F/Earth day}$ and fall at a rate of $-2.66\text{ }^{\circ}\text{F/Earth day}$. In this case the infrared temperature value rises by $5.22\text{ }^{\circ}\text{F}$ from mid-morning to after mid-day and falls by $15.17\text{ }^{\circ}\text{F}$ from after mid-day to late in night.

For determining the accurate values of Thermal inertia one needs other surface constant values and perhaps the absolute temperatures, which is not covered in this study. But one

can infer from the small temperature change that for the smooth surfaces the Thermal inertia is a high value due to observed resistance to large temperature change. In the next step, the temperature slope can be determined for the rocky surfaces to compare with the smooth surfaces. Due to lack of time, not enough research is done to determine the temperature slope for the rocky surfaces to be published here.

The registration algorithm could be used to register any set of Moon images to build the Time series over any amount of time depending on the data available at hand. Although, if the camera with a large field of view that can cover the whole surface in one frame is used the need for image registration can be eliminated to some extent which would reduce the processing time by a lot. Also, if the infrared camera used is temperature controlled, the background noise could further go down to reduce the error. The error due to atmospheric disturbances is always going to be there.

10.1 Future Work

Since such an experiment is being documented for the first time, there is a lot of room for improvements. Many things were learned from this project and many are yet to be learned in the future experiments. Firstly, the data gathering process needs to get much more efficient by getting imaging equipment with a wider field of view, that can image the whole Moon in one frame rather than in quadrants. That alone can get rid a lot of the unnecessary image registration, which in return also adds error. Secondly, the imaging equipment should have something to track the Moon, so the user won't have to manually move the telescope. This could make the data gathering process automated. Thirdly, the data should be collected smartly by making sure that the data is taken evenly throughout the month to account for full Lunar cycle.

In terms of the software, if the frames contain the full Moon with no movement with the help of the tracker, perhaps the image registration could be eliminated for data pertaining same day. A better image registration technique could be developed to register frames from different days, more accurately and automatically with no manual step. The background noise cancellation techniques should also be improved to further smooth out the curves. One way to do this is by using a cooled infrared camera to reduce temperature sensitivity.

BIBLIOGRAPHY

1. Price, J.C., 1977, Thermal inertia mapping: A new view of the earth. *Journal of Geophysical Research* 82, 2582-2590.
2. Benz, W., Slattery, W. L. and Cameron, A. G. W., 1986, "The origin of the moon and the single-impact hypothesis I". *Icarus*. Vol. 66 (3), pp. 515-535
3. "Volumetric heat capacity." Wikipedia. Wikimedia Foundation, 11 Feb. 2017. Web. 09 Mar. 2017.
4. Thomas, Ian "Measurement of properties of the lunar surface using the Diviner Lunar Radiometer experiment on the NASA Lunar Reconnaissance Orbiter." ORA -Oxford University Research Archive. University of oxford, Aug. 2009.
5. Allen, D. A. "1971Moon....2..320A 320." 1971Moon....2..320A 320. NASA, Feb. 1971.
6. Price, Stephan D., Don Mizono, and Thomas L. Murdock. "Thermal profiles of the eclipsed moon." *Thermal profiles of the eclipsed moon - ScienceDirect*. Elsevier Ltd., 2003. Web. 09 Mar. 2017.
7. "Mare Tranquillitatis." Wikipedia. Wikimedia Foundation, 08 Mar. 2017. Web. 09 Mar. 2017.
8. "Documentation." *Intensity-Based Automatic Image Registration - MATLAB & Simulink*. MathWorks, n.d. Web. 09 Mar. 2017.

Appendix A: Source Code

Block 2: Lunar Images to videos

```
% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----%
Make short videos from Lunar Image data producing Avi
% files containing 450 frames each.
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----

% EDIT THE DATE, THEN SAVE AND RUN THE FILE
yyyy = 2014;
mm = 08;
dd = 05;

% create date directory paths for both USB storage
locations

path_F = sprintf ('/Users/AGill/Desktop/My_Hood/Moon
thesis/LunarDataMatlab/lunar.%04d.%02d.%02d', yyyy, mm,
dd);
path_H = sprintf ('/Users/AGill/Desktop/My_Hood/Moon
thesis/LunarDataMatlab/lunar.%04d.%02d.%02d', yyyy, mm,
dd);

% loop through lunar image files in a DATE directory
% make a video clip of the digital file

% image dimensions
nrows = 512;
ncols = 640;

% get a list of all the image (.tif) files in the selected
date directory
tif_files = sprintf ('%s.tif', path_F);
list_files = dir (tif_files);
n_files = length (list_files);

for i_file = 1:n_files
    % construct filename of the .tif file that is to be
    read
    imgfilepath_F = sprintf ('%s.tif', path_F);
```

```

imgfilepath_H = sprintf ('%s.tif', path_H);

% construct filename of the .avi file that is to be
written
avifilepath_F = sprintf ('%s.avi', path_F);
avifilepath_H = sprintf ('%s.avi', path_H);

% set up the .avi file for writing, and write the
first frame
writerObj = VideoWriter (avifilepath_F); %#ok<TNMLP>
open (writerObj);

% up to 450 frames in each video file
for iFrame = 1:450
    try
        irframe = imread (imgfilepath_F, iFrame);
    catch err
% exit the read loop if there are less than 450 frames
        break;
    end

    % ignore 4 blind rows
    irframe = irframe(:, :);

    % apply agc and write frame to .avi file
    irframeagc = agc (irframe, nrows, ncols-4);
    try
        writeVideo (writerObj, irframeagc);
    catch err
        continue;
    end
end

% close the .avi file, since the handle will be used
again for next
% file
close (writerObj);
try
    copyfile (avifilepath_F, avifilepath_H);
catch err
    continue;
end
end
end

```


Block 3: Intensity Based Image Registration

```
% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----%
Intensity Based Image Registration on Tiff files
% containing 450 frames each
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----
clear
[optimizer, metric] = imregconfig ('monomodal');

% EDIT THE DATE, THEN SAVE AND RUN THE FILE
yyyy = 2014;
mm = 08;
dd = 14;

% create date directory paths for both USB storage
locations
dataPath = sprintf ('/Users/AGill/Desktop/My_Hood/Moon
thesis/Test/lunar.%04d.%02d.%02d', yyyy, mm, dd);

% Consider one lunar image file in DATE directory

% image dimensions
nrows = 512;
ncols = 640;
nActiveCols = ncols - 4;

nframes = 450;
frameseq = zeros (nrows, nActiveCols, nframes);

%Get the .tif file in the Date directory
imgfilepath = sprintf ('%s.tif', dataPath);

% get middle frame and use as reference for all other
frames in the
% video sequence of nframes

try
    irframe = imread (imgfilepath, nframes/2);
    %nframes/2
catch err
    % error if cannot read middle frame
    disp 'Error: cannot read middle frame';
    return;
```

```

end

% ignore 4 blind rows
irrefframe = double (irframe(:,5:640));

% up to 450 frames in each video file
for iFrame = 1:nframes
    try
        irframe = imread (imgfilepath,
iFrame);
    catch err
        % exit the read loop if there are less than 450
frames
        nframes = iFrame - 1;
        break;
    end

    % ignore 4 blind rows
    irframe = double (irframe(:,5:640));

    % register images
    tform = imregtform (irframe, irrefframe, 'rigid',
optimizer, metric);
    Rmoving = imref2d (size (irframe));
    Rfixed = imref2d (size (irrefframe));
    registered = imwarp (irframe, Rmoving, tform,
'OutputView', ...
    Rfixed, 'Interp', 'linear', 'FillValues', NaN);
    frameseq(:, :, iFrame) = registered;

end
avgframe = mean (frameseq(:, :, 1:nframes), 3);
avgframe(avgframe < 0) = 0;

outfilepath = sprintf ('%s.avg450.tif', dataPath);

imwrite (uint16 (avgframe), outfilepath);

imagesc (double (avgframe))
set (gcf, 'position', [70 253 847 715])
axis square
colormap (gray)

outfilepath = sprintf ('%s.avg450.flat.jpg', dataPath);

print ('-djpeg', outfilepath)

```

```

surf (double (avgframe), 'edgecolor', 'none')
view (135, 45)
axis square
colormap (gray)

outfilepath = sprintf ('%s.avg450.surf.jpg', dataPath);

print ('-djpeg', outfilepath)

```

Block 4: Control Point Image Registration Part 1

```

% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----%
Control Point Image Registration code Part 1
% This script aligns to frames using cpselect, to manually
select
% 4 control points, corresponding to same locations in both
frames. This
% could be used as a first step to align two frames from
two different
% registered tif files, to eventually help in aligning all
900 frames in
% both tif files.
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----

% Allocating reference frame
moon_fixed = imread('lunar.2014.08.14.02.29.04.tif', 225);
% Allocating moving/test frame
moon_moving = imread('lunar.2014.08.14.03.54.49.tif', 225);

desiredMin = 0;
desiredMax = 255;
actualMin = double(min(min(moon_moving)));
actualMax = double(max(max(moon_moving)));
slope = (desiredMax - desiredMin) / (double(actualMax) -
double(actualMin));
scaledImage = slope * (double(moon_moving) - actualMin) +
desiredMin;

% Rescaling moving frame
moving = uint8(scaledImage);

```

```

actualMin = double(min(min(moon_fixed)));
actualMax = double(max(max(moon_fixed)));
slope = (desiredMax - desiredMin) / (double(actualMax) -
double(actualMin));
scaledImage = slope * (double(moon_fixed) - actualMin) +
desiredMin;

% Rescaling reference frame
    fixed = uint8(scaledImage);

    % Select Control points to import in the workspace
cpselect(moving, fixed);

```

Block 4: Control Point Image Registration Part 2

```

% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----%
Control Point Image Registration code Part 2
% After the 4 control points are imported, running a loop
% all the 900 frames in both files are aligned
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----

% Reading the 450 frames in the moving file
counter = 0;
for frame = 1:450

    counter = counter + 1;

    moving_frame =imread('Registered_450Frames.tif',frame);

    % Registering the moving frame with the reference
    frame
    mytform = fitgeotrans(movingPoints, fixedPoints,
'NonreflectiveSimilarity');
    registered =
    imwarp(moving_frame,mytform,'OutputView',imref2d(size(
fixed)));

    % Saving the newly registered moving frame
        reg_Cell{frame} = registered;

    % Saving the registered moving frames in a cell

```

```

        imwrite(uint16(reg_Cell{frame}), 'Registered.tif' ,
'WriteMode', 'append');
end

```

Block 5: Background Normalization

```

% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----
% Step 1: Select background sky pixel location on each test
file
% Step 2: Find the average background pixel value over a
5x5 kernel
% Step 3: Selecting one as a reference, calculate
Testfile_difference
% (average pixel value subtracted from refrence)
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----%
Each Step is done separately.
%-----%
Step 1
%-----

tstframe = imread('test_filename.tif', 255);
tstframe = double(tstframe);
tstframe(tstframe < 5500) = NaN;
tstframe(tstframe == 0) = NaN;
minValue = min(min(tstframe))
maxValue = max(max(tstframe))
imshow(tstframe, [])

hold on
% Show a rectangle on the image to visually see the
location selected
% and croess relate to other test files
rectangle('position',[Sel_Pixl_x_coord Sel_Pixl_y_coord 5
5], 'FaceColor', 'w')
hold off

%-----%
Step 2
%-----

Average_Matrix = zeros(450,1);
loop = 0;

```

```

% Manually get Pxl_x and Pxl_y values for inertia
Pxl_x = Sel_Pixl_x_coord;
Pxl_y = Sel_Pixl_y_coord;
% Get average with 5x5 kernel for each frame in the test
file and save in
% an array
for loo = 1:450
    loop = loop+1;
    Image = imread('test_filename.tif',loo);

    Inertia_Matrix(loop,1) = Image(Pxl_y,Pxl_x);
    Inertia_Matrix(loop,2) = Image(Pxl_y,Pxl_x+1);
    Inertia_Matrix(loop,3) = Image(Pxl_y,Pxl_x+2);
    Inertia_Matrix(loop,4) = Image(Pxl_y,Pxl_x+3);
    Inertia_Matrix(loop,5) = Image(Pxl_y,Pxl_x+4);

    Inertia_Matrix(loop,6) = Image(Pxl_y+1,Pxl_x);
    Inertia_Matrix(loop,7) = Image(Pxl_y+1,Pxl_x+1);
    Inertia_Matrix(loop,8) = Image(Pxl_y+1,Pxl_x+2);
    Inertia_Matrix(loop,9) = Image(Pxl_y+1,Pxl_x+3);
    Inertia_Matrix(loop,10) = Image(Pxl_y+1,Pxl_x+4);

    Inertia_Matrix(loop,11) = Image(Pxl_y+2,Pxl_x);
    Inertia_Matrix(loop,12) = Image(Pxl_y+2,Pxl_x+1);
    Inertia_Matrix(loop,13) = Image(Pxl_y+2,Pxl_x+2);
    Inertia_Matrix(loop,14) = Image(Pxl_y+2,Pxl_x+3);
    Inertia_Matrix(loop,15) = Image(Pxl_y+2,Pxl_x+4);

    Inertia_Matrix(loop,16) = Image(Pxl_y+3,Pxl_x);
    Inertia_Matrix(loop,17) = Image(Pxl_y+3,Pxl_x+1);
    Inertia_Matrix(loop,18) = Image(Pxl_y+3,Pxl_x+2);
    Inertia_Matrix(loop,19) = Image(Pxl_y+3,Pxl_x+3);
    Inertia_Matrix(loop,20) = Image(Pxl_y+3,Pxl_x+4);

    Inertia_Matrix(loop,21) = Image(Pxl_y+4,Pxl_x);
    Inertia_Matrix(loop,22) = Image(Pxl_y+4,Pxl_x+1);
    Inertia_Matrix(loop,23) = Image(Pxl_y+4,Pxl_x+2);
    Inertia_Matrix(loop,24) = Image(Pxl_y+4,Pxl_x+3);
    Inertia_Matrix(loop,25) = Image(Pxl_y+4,Pxl_x+4);

    Average_Matrix(loop,1) = (Inertia_Matrix(loop,1)+
    Inertia_Matrix(loop,2)+ Inertia_Matrix(loop,3) ...
        + Inertia_Matrix(loop,4)+ Inertia_Matrix(loop,5)+
    Inertia_Matrix(loop,6)+Inertia_Matrix(loop,7) ...
        + Inertia_Matrix(loop,8)+
    Inertia_Matrix(loop,9)+Inertia_Matrix(loop,10)+
    Inertia_Matrix(loop,11) ...

```

```

        + Inertia_Matrix(loop,12)+Inertia_Matrix(loop,13)+
Inertia_Matrix(loop,14)+ Inertia_Matrix(loop,15)...
        + Inertia_Matrix(loop,16)+ Inertia_Matrix(loop,17)+
Inertia_Matrix(loop,18)+Inertia_Matrix(loop,19)...
        + Inertia_Matrix(loop,20)+
Inertia_Matrix(loop,21)+Inertia_Matrix(loop,22)+
Inertia_Matrix(loop,23)...
        + Inertia_Matrix(loop,24)+
Inertia_Matrix(loop,25))/25;
end
minValue = min(min(Average_Matrix))

% Average the 450 Average pixel values obtained using the
5x5 kernel
Average = 0;
for l = 1:450
    Average=Average_Matrix(l)+Average;
end
Avg_value=Average/450;

%-----%
Step 3
%-----

% For each test file subtract the corresponding difference
and concatenate
% new files to form a New normalized Tiff
for q=1:450
    Image1 = imread('Test_filename.tif',q);
    Image = Image1 - (Testfile_difference) ;
    reg_Cell{q} = Image;
    imwrite(uint16(reg_Cell{q}), 'Regist.tif', 'WriteMode',
'append');
end

```

Block 5: Pixel Trends & Block 6: Temperature Trends

```
% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----%
This script simply uses the 3x2 kernel to plot average %
Pixel/temp. plots in 3 different formats
% containing 450 frames each
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----%

Inertia_Matrix = zeros(3150,6);
Average_Inertia_Matrix = zeros(3150,1);
loop = 0;

% Manually get Pxl_x and Pxl_y values for inertia
Pxl_x = 356;
Pxl_y = 159;

% Reads the test file and saves the 6 pixel values around
the specified
% pixel in each frame in 450x6 Inertia_Matrix array
for loo = 1:3150
    loop = loop+1;
    Image = imread('test_filename.tif',loo);
    Inertia_Matrix(loop,1) = Image(Pxl_y,Pxl_x);
    Inertia_Matrix(loop,2) = Image(Pxl_y,Pxl_x+1);
    Inertia_Matrix(loop,3) = Image(Pxl_y,Pxl_x+2);

    Inertia_Matrix(loop,4) = Image(Pxl_y+1,Pxl_x);
    Inertia_Matrix(loop,5) = Image(Pxl_y+1,Pxl_x+1);
    Inertia_Matrix(loop,6) = Image(Pxl_y+1,Pxl_x+2);

    % Takes an average for each row in the Inertia_Matrix
array
    Average_Inertia_Matrix(loop,1) =
(Inertia_Matrix(loop,1)+ Inertia_Matrix(loop,2)+
Inertia_Matrix(loop,3) ...
    +Inertia_Matrix(loop,4)+ Inertia_Matrix(loop,5)+
Inertia_Matrix(loop,6))/6;
end

% Displays the minimum pixel value in the newly created
array
minValue = min(min(Average_Inertia_Matrix))
```



```

% Plots the 6 different pixels/temp. values saved in the
Inertia array
figure;
plot(Inertia_Matrix)
hold on
title('Mare Serenitatis Pixel(165, 215) from 08.05.2014 to
08.17.2014')
xlabel('Frame #')
ylabel('Infrared Temperature (°F)')
hold off

% Plots the average pixels/temp. value saved in the
% Average_Inertia_Matrix array
figure;
plot(Average_Inertia_Matrix)
hold on
title('Mare Serenitatis Pixel(165, 215) using average of
3x2 Kernel')
xlabel('Frame #')
ylabel('Infrared Temperature (°F)')
hold off

% Takes the average of the individual 450 frame segments
sum = 0;
New_Intertia=zeros(5,1);
flag =0;
for G=1:3150
    sum = sum+Average_Inertia_Matrix(G);
    if (mod(G,450)==0)
        flag=flag+1;
        New_Intertia(flag,1) = sum/450;
        sum = 0;
    end
end

% Plots the average pixels/temp. value saved in the
% New_Intertia array
figure;
plot(New_Intertia)
hold on
title('Mare Tranquillitatis, 8 Average pixels for 8 days')
xlabel('Day #')
ylabel('Pixel Intensity')
hold off

```

Block 6: Temperature convert (Additional block)

```
% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----%
This script converts the pixel values to temperature
% values using the camera calibration file.
% containing 450 frames each
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----%

% make a video clip of a digital thermal file
[imgfilename, imgdirectory, ~] = uigetfile('*.tif');
imgfilepath = strcat (imgdirectory, imgfilename);
fileinfo = imfileinfo (imgfilepath);
% image dimensions
nrows = fileinfo(1).Height;
ncols = fileinfo(1).Width;
nFrames = length (fileinfo);

load ('CalCurves');

maxFrame = zeros (nFrames, 1);
minFrame = zeros (nFrames, 1);

% construct .avi file path from image filepath
avifilepath = sprintf ('%s.avi', imgfilepath);

% up to nFrames in each thermal video file
for iFrame = 1:nFrames
    try
        irframe = imread (imgfilepath, iFrame);
    catch err
        % exit the read loop if there are less than nFrames
        break;
    end

    % ignore 4 blind rows
    irframe = double (irframe(:,1:636));
    % p(1) is the slope and p(2) is the intercept
    p1 = squeeze (CalCurves(1, :, :));
    p2 = squeeze (CalCurves(2, :, :));
    calFrame = p1.*irframe + p2;
    calFrame = (9/5)*calFrame + 32;

    maxFrame(iFrame) = max (calFrame(:));
end
```

```

        minFrame(iFrame) = min (calFrame(:));
end
nFrames = iFrame;
seqMax = max (maxFrame(1:nFrames));
seqMin = min (minFrame(1:nFrames));

% set up the .avi file for writing, and write the first
frame
writerObj = VideoWriter (avifilepath);
open (writerObj);

% display the first frame
irframe = imread (imgfilepath, 1);
% ignore 4 blind rows
irframe = double (irframe(:,1:636));
% p(1) is the slope and p(2) is the intercept
p1 = squeeze (CalCurves(1, :, :));
p2 = squeeze (CalCurves(2, :, :));
calFrame = p1.*irframe + p2;
calFrame = (9/5)*calFrame + 32;
%Saving the first temperature converted frame
step = 1;
reg_Cell{step} = calFrame;

% build a color map based on the sequence temperature range
imagesc (calFrame)
colormap(jet)
caxis([seqMin seqMax])
colorbar
cb = colorbar;
degreelabel = sprintf('%cF', char(176));
xlabel (cb, degreelabel)
axis off;
axis image;
temptext = sprintf ('Max Temp: %0.2f %cF', maxFrame(1),
char(176));
text(10, 500, temptext, 'Color', 'white');

set (gca, 'nextplot', 'replacechildren');
set (gcf, 'Renderer', 'zbuffer');
frame = getframe;

% write frame to .avi file
try
    writeVideo (writerObj, frame);
catch err
    disp (err);
    close (writerObj);

```

```

    return;
end

for iFrame = 2:nFrames
    try
        irframe = imread (imgfilepath, iFrame);
    catch err
        disp (err);
        close (writerObj);
        return;
    end

    % ignore 4 blind rows
    irframe = double (irframe(:,1:636));
    % p(1) is the slope and p(2) is the intercept
    p1 = squeeze (CalCurves(1, :, :));
    p2 = squeeze (CalCurves(2, :, :));
    calFrame = p1.*irframe + p2;
    calFrame = (9/5)*calFrame + 32;
    %Saving rest of the temperature converted frames
    step = step + 1;
    reg_Cell{step} = calFrame;

    % build a color map based on the sequence temperature
    range
    imagesc (calFrame)
    colormap ('jet')
    caxis ([seqMin seqMax])
    cb = colorbar;
    xlabel (cb, degreeLabel)
    axis off;
    axis image;
    temptext = sprintf ('Max Temp: %0.2f %cF',
maxFrame(iFrame), char(176));
    text(10, 500, temptext, 'Color', 'white');

    frame = getframe;
    % write frame to .avi file
    try
        writeVideo (writerObj, frame);
    catch err
        disp (err);
        close (writerObj);
        return;
    end
end

% close the .avi file

```

```

close (writerObj);

% Making Final Temperature converted Tiff File
for step=1:3150
imwrite(uint16(reg_Cell{step}), 'Backnorm_TempFrames.tif' , '
WriteMode', 'append');
end

```

Miscellaneous scripts

Script 1:

```

% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----
% This script is to display the images in right format.
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----

function img_agc = agc (img, n_rows, n_cols)

% perform Plateau AGC on unsigned 16-bit image
max_bit = 2^16;
hist_bins = zeros (max_bit, 1);
img_agc = uint8 (zeros (n_rows, n_cols));

% set up the histogram bins, one for each 16-bit grayscale
level
for i_bin = 1:max_bit
hist_bins(i_bin) = i_bin - 1;
end

% matlab calculates the image histogram
img_hist = histc (img(:), hist_bins);

% for Plateau Equalization, clip the bins to some plateau
level
img_hist(img_hist > 150) = 150;

% calculate the cumulative frequency histogram from the
clipped
% histogram
cum_hist = cumsum (img_hist);

% scale the cumulative frequency histogram to 8-bit
grayscale

```

```

cum_hist = uint8 (255 * cum_hist / cum_hist(max_bit));

% use the scaled cumulative frequency histogram as an
intensity
% transform table to map the original 16-bit data to 8-bit
level
for i_row = 1:n_rows
for j_col = 1:n_cols
img_agc(i_row,j_col) = cum_hist(img(i_row,j_col) + 1);
end
end

return

```

Script 2:

```

% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----
% This simple script is to save registered files from Block
3.
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----

step = 0;

for q=1:450

    step = step + 1;
    reg_frame = frameseq(:, :, q);
    reg_Cell{step} = reg_frame;
    imwrite(uint16(reg_Cell{step}), 'lunar.2014.08.08
_Registered_450Frames.tif' , 'WriteMode', 'append');
end

```

Script 3:

```
% Electrical Engineering Department
% California Polytechnic State University
% Thesis: "Thermal Evolution of the Moon"
%-----%
This script simply concatenates different Tiff files
% in a folder to maek one big Tiff file.
% Author: Arsh Gill, Dr. Gary Hughes
% Revised: 03/14/2017
%-----
tif_files=dir('*.tif');
files={tif_files.name};
for k=1:numel(files)

    for l=1:450
        frame1{l}=imread(files{k}, l);
        imwrite(frame1{l}, 'Concatenate_Tiffs.tif' , 'WriteMo
de', 'append');
    end
end
```