**CS496 Senior Design**

**Lunar Image Processing and Feature Detection**

**Design and Progress**

Document Prepared by:

Natalie Gallegos

Tony Liang

Marvin Mendez

Raul Ornelas

Alberto Serrano

17 March 2015

CALIFORNIA STATE UNIVERSITY LOS ANGELES



Los Angeles, California

**Table of Contents**

1.0 INTRODUCTION

2.0 Hough Circle Transform..............................................................................

3.0 Ellipse Fitting……………………………………………………………...……..

4.0 Crater Depth and Diameter………………….…………………………………

5.0 CONCLUSION………………………………………………………………......

# INTRODUCTION

This document describes the progress achieved for each LIPFD module throughout the winter quarter, as well as individual summary for each member working under the each module.

# Hough Circle Transform

The Winter 2015 quarter continued refinements on the prototype crater detection algorithm from the Fall 2014 quarter. The algorithm is based of the methodology described in the paper “Linear and Non-Linear Feature Extraction Algorithms for Lunar Images” by Tamililakkiya V., Vani K., Lavanya A., and Anto Michael. The methodology consisted of three stages: a preprocessing stage, an edge detection stage, and a circular hough transform stage. By the end of the Fall 2014 quarter a rudimentary implementation of the aforementioned methodology was accomplished. The implementation involved a single scale retinex algorithm and two filters (median and gaussian) as a preprocessing stage. The canny edge detection was finished but involved empirical parameters. The hough transform was implemented to detect circles for one radii range and demonstrated difficulty detecting various circles within an image.

2.1 Marvin’s contribution

The first week of the Winter 2015 quarter updated the algorithm to union the results of three iterations of the circular hough transform at different radii ranges. Preliminary results demonstrated far more circles detected but, also, various false circles were detected. The algorithm was updated to view each intermediate step. The GUI was revamped to display images at various stages of the algorithm to determine where any information may be lost between steps. It was determined that more radii ranges would be beneficial to detecting craters that may have been missed. The parameters for each hough iteration would need to be updated to ensure less false positives were detected.

The previous quarter the hysteresis thresholds for the canny edge detector were provided by the user through a slider. These thresholds were empirical values and different for each image. After testing with a few crater images, it was found that the top 5% to 12% of edges returned by the sobel operator in an image would be ideal for most of the crater images that were tested on. The top 9% are being used for all testing purposes moving forward but, a slider is still provided when the user wants to manually change the threshold value.

Toward the middle of the Winter quarter the LIPFD team began comparing results amongst our different crater detection algorithms. We found strengths and weaknesses between our algorithms. The convolutional neural network through Saman’s work did well at detecting old craters and many small craters. Albert and Raul’s had great success with elliptical craters with proper light and dark patches.The circular hough transform did well at detecting craters with apparent rims. Unfortunately, the circular hough transform algorithm continued to detect many false positives, especially around shadows of a crater.

Around this time is when the team visited JPL for a tour and to present our current results. We returned with helpful advice on how to update our algorithms to obtain better results. The circular hough transform algorithm was updated to filter out false positives through gray level histogram analysis of the potential circles detected. Extreme cases of circles with almost pure dark pixels or light pixels were eliminated by the filtering algorithm.

The remainder of the quarter was spent finalizing the algorithm for data analysis to be done in the following quarter of Spring 2015. Our data set to test on is also in the process of being compiled into 30-40 images for the next quarter. The plan for next quarter will be to add multiple image analysis to the algorithm.

# Ellipse Fitting

At the end of the Fall quarter, we had implemented the ellipse algorithm from the "Detection of Craters and Its Orientation on Lunar" by Nur Diyana Kamarudin, Kamaruddin Abd. Ghani, Siti Noormiza Makhtar, Baizura Bohari and Noorlina Zainuddin. This crater search algorithm focuses on finding the light and dark patches and then linking them up following appropriate restraints. Our program needed a few manually inputted values, so for the Winter Quarter we looked to think to have the computer generate some of the values itself, further improve the algorithm and compare it with results of the other algorithms.

We designed a GUI that displays the image with detected craters with image preprocessing steps to the right of it. It allows the user to input a sun angle , dilation and erosion value. We implemented the otsu's method to determine an ideal threshold value that replaced our previous user inputted value. We spent time looking over how to fix the erosion and dilation. We also spend time comparing crater result images with other team mates algorithms to find the strength and weaknesses of each. We found our algorithm works best with well defined light and dark patches. We split up the crater search into three levels. Small, medium, and Large craters. Each using different values that would be better for each search. Also adjusted the GUI to fit the three level search. We added a results tab to the GUI to show the found craters and the information of each, also outputted to a txt file. We also made a way to eliminate duplicate craters that were being detected twice by the three split up algorithms.

3.1 Albert’s contribution

Working on creating a GUI for the ellipse algorithm. Spent time creating the GUI support the three level crater search. When through the algorithm to find all possible places that could be improved. Worked on implementing the otsu’s method to find the best threshold, got it working and removed the manually inputted threshold from the user. Close to the end of the Winter Quarter, Raul and I worked on implementing the three level crater search that would run the algorithm three times with different values and restraints. We then went on to find out what values would be best for each search. I also spend some time adjusting the program to be used by Natalie and Tony, to provide them with the center of the dark patches which they could use in their program. Spend time with Marvin comparing results from our algorithms with different images. I also spend time gathering interesting crater images, at least 4 each week to try and big out data set that we will use.

3.2 Raul’s contribution

Winter Quarter, Albert and I continued refining the algorithm. Through testing and suggestions from advisor, we found that otsu’s method for thresholding provided better results than manually inputed values. We added a sun vector input and later changed from a vector to an angle in degrees, measuring azimuth of the sun. This angle, along with other parameters, are used to qualify the the candidates for each matching light and dark patches in crater images. We explored using the sub solar azimuth provided in the PDS file as a more precise angle, however, after consulting with JPL experts, found that the PDS angle were not consistent and not completely reliable. Therefore, it was determined that the manually input angle was sufficient. Worked with Albert to divide detection into three crater diameter ranges, which allowed for varying levels of erosion and dilation. This variation in preprocessing allowed for reduction in background noise in smaller crater detection and clearer definition in larger craters. Began evaluating crater images from JPL results for a test data set, to be used in the performance analysis of algorithm.

# Crater Depth and Diameter

By the end of the 2014 fall quarter, the crater depth and diameter module was in its initial stages and could only extract the diameter of the “Alan” crater. Many updates have been added to the program since then, which now allow for extraction of both crater depth and diameter for all detected craters on a given image.

This quarter the first update added simple GUI to the program which allowed the user to input metadata for the image in question in order to be able to accurately extract diameter information. Additional functions were added to our program GUI, namely a drawing tool where the user can draw both the length of the shadow of a crater, as well as the crater’s diameter and have the program scale these values and provide diameter and crater depth in meters. By the mid quarter, we found that our crater depth extraction varied when using different illumination angles, thus another update was needed, and the upgrade needed to be ready by the 13th of February.

February 13th was the day of our JPL tour. Tony and I wanted to update our program to use two images and compute the average depth and diameter of a crater given two different lamination angles. We had to restructure the GUI and make sure the drawing tools worked for multiple images. After the presentation our next program updates focused on extracting crater depth and diameter automatically instead of using the drawing tools. We reverted back to single images, and tested automatic crater depth and diameter extraction on the Yoshi crater. Currently our program can automatically extract diameter and depth for most detected craters. In the final week we are debugging further, and testing our algorithm on earth craters to measure our accuracy, given the limited ground truth available for moon craters.

## Natalie’s Contribution:

 At the start of the quarter I began to add GUI to the program to make it more interactive and modular for diameter extraction in order to test on more craters. Our next step was depth extraction. I researched the different types of possible sun angles to better understand how we can get the sun direction from the images provided on the LROC website. I came to understand that we would have to use three different angles to properly calculate the depth of a crater, we would need the elevation angle, which is not provided but can be calculated using the incidence angle, and we would need the solar azimuth. To understand the idea behind light direction we updated our program to calculate the depth of a crater. While Tony worked on drawing tools which would allow the user to manually draw the shadow length and diameter crater, I implemented the methods to calculate the depth of a crater as well as helped Tony make general updates to the program to take in more input for metadata.

 By mid quarter we were assigned to update our program to take in two images to find the average of diameter and depth of the same crater in order to provide a demo on the JPL tour. Tony and I both worked on updated the GUI for our program as well as implementing the minor changes that would calculate the average length and diameter. I also prepared the slides for our presentation for JPL the same week. For the latter part of the quarter automatic calculation of crater depth was our focus. This required a little more research on the sub solar azimuth angles provided by the website and how we could use them. I implemented a methods for automatic measuring of shadow lengths as well as a method that would mimic the orientation of the azimuth angle given by the LROC website. The last week I spent debugging these implementations to try and obtain better results, as well look for information on Earth craters that we could use to further test our algorithm’s accuracy. Of course, throughout the entire quarter, along with the rest of the team, I collected images from JPL’s test results for use during analysis next quarter.

## Tony’s Contribution:

In the beginning of the quarter, I started working on finding the light direction of crater images that contain only one crater. Natalie and I started off working with a single image at a time for our program. We were focused on testing our program on the Yoshi crater. I began with implementing a manual approach to retrieving the light direction. I allowed the user of our program to use the mouse to click on two spots that would serve as two endpoints that connect and form a line that represent the shadow length of the crater in the image. The shadow length is needed to calculate the depth of a crater. Then, I added the same option for the user to click on two spots that represent the diameter of the crater in the image. Later on, Natalie and I moved on to work with two crater images at a time. I modified the program to allow the user to use the two options previously discussed with both images. After achieving successful results with the Yoshi crater’s diameter and depth calculations, we began implementing an automatic approach to calculate diameter and depth of craters. With this approach, we went back to working with one crater image at a time. While Natalie implemented the automatic approach to calculating the diameter and depth of a crater image, I focused on making GUI (Graphical User Interface) changes such as convenient browsing and automatic diameter and shadow length drawings.

**5.0 Conclusion**

 Each of our modules have gone through many updates this quarter, but there is still plenty left to do next quarter. Next quarter the entire team will focus using multiple images and different light direction in hopes of extracting more craters that may have been missed with previous light conditions. We will also combine all of our programs in order to deliver a single product to JPL, and finally we will perform a deeper analysis and compare our results with those of JPL for crater detection.