**CS496 Senior Design**

**Salient Feature Extraction from Planetary Images**

**Software Design Specification**

Document Prepared by:

Adam Berman

Andy Tong

Benjamin Evans

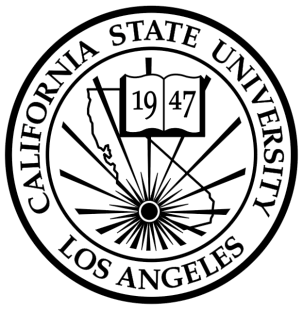
Bruno Recillas

Kevin Casado

Lineli Hairapetian

27 May 2016

CALIFORNIA STATE UNIVERSITY LOS ANGELES



Los Angeles, California

**Salient Feature Detection from Planetary Images**

**Section 1: Introduction**

This document describes the MIPFD (Mars Image Processing - Feature Detection) system and its components, along with the current plan for improvement. It includes a brief summary of the strategies used in implementing its features, along with a dissection of the software into components and an analysis of the structure of data shared between modules.

**Section 2: System Overview**

The MIPFD system is used to detect craters in images recorded by the Mars Reconnaissance Orbiter’s HiRISE (High Resolution Imaging Science Experiment) camera. Since crater detections is not perfect, general features are also detected and utilized to generate a hazard map of the scanned location.

**Section 3: Design Considerations**

***3.1 Goals***

* Accurately detect features in Mars images
* Use detected features  to generate a hazard map

**Section 4: Architectural Strategies**

***4.1 Use of a particular type of product (programming language, database, library, etc.)***

* Java
* Opencv
* Linux
* GDAL
* MySQL
* Octave
* USGS ISIS

***4.2 Reuse of existing software components to implement various parts/features of the system***

MIPFD is a project derived from LIPFD (Lunar Image Processing - Feature Detection). All software components and implemented system features of LIPFD will be reused for MIPFD.

***4.3 Future plans for extending or enhancing the software***

With MIPFD we aim to improve existing feature detection and recognition systems to provide better overall results for Mars images. We also plan to implement a hazard mapping system to identify which areas are safe for exploration and which areas aren’t.

***4.4 User interface paradigms (or system input and output models)***

(See LIPFD documentation for details)

***4.5 Error detection and recovery***

(See LIPFD documentation for details)

***4.6 External databases***

MIPFD does not use any external databases.

***4.7 Distributed data or control over a network***

MIPFD doesn’t deal with distributed data and is not controlled over a network

***4.8 Generalized approaches to control***

(See Section 4.7)

***4.9 Concurrency and synchronization***

There is no concurrency concerns with MIPFD

***4.10 Communication mechanisms***

There are no communication mechanism

**Section 5: System Architecture**

***5.1 component-1 Crater Detection***

Takes an image and detects what in the image is a possible crater.

***5.2 component-2 Duplicate Removal***

Takes a list of detected craters and checks to see if any of the craters have been detected more than once; if so, then the duplicate detections are removed.

***5.3 component-3 Crater Recognition***

Takes a list of detected craters and determines whether or not they are actually a craters

***5.4 component-4 Hazard Mapper***

Takes a list of detected craters that have been classified as not a crater and uses its features to identify parts of an image that are no safe for exploration

**Section 6: Detailed System Design**

***6.1 module-1 Crater Detection***

* ***Definition***

The purpose of this module is to detect potential craters within a given image

* ***Responsibilities***

This module is responsible for determining what areas of the image contains possible craters so that they can be sent to the Duplicate Removal module for further processing

* ***Composition***

This module utilizes three different algorithms for detecting craters: Ellipse fitting, Circular Hough, and template matching algorithm

1. ***Ellipse Fitting algorithm*** - Finds light and dark patched within an image and tries to matches and fit an ellipse around them to see if the size and orientation of the ellipse coincides with proper alignment and size of a possible crater within the image based on the sun azimuth.
2. ***Circular Hough algorithm*** - Finds circular features and determine whether or not they may be craters
3. ***Template Matching algorithm*** - Uses template images of actual craters and attempts to detect potential craters by comparing parts of the image with the crater templates

* ***Uses/Interactions***

After detecting craters, the result are sent to the Duplicate Removal module as input.

* ***Processing***

(See composition section of this module)

* ***Interface/Exports***

This module outputs a list of coordinates (with respect to the image being processed) representing the bounding box around potential craters within the image.

***6.2 module-2 Duplicate Removal***

* ***Definition***

The purpose of this module is to remove duplicate craters from the list of possible craters produced by the Crater Detection module.

* ***Responsibilities***

This module is responsible for ensuring that any craters that are detected more than once only appear once in the crater candiddate list.

* ***Composition***

There are no subcomponents for this module.

* ***Uses/Interactions***

The output of the crater detection algorithm is used as input for this module. The output of this module is used as input for the crater recognition module.

* ***Processing***

This module removes duplicate craters by finding detected craters that have an intersecting bounding box. If the intersecting boxes overlap by more than 50% of their respective areas then the detected craters are considered duplicate detections. To resolve this duplication this module creates a new box which is the average of the two overlapping boxes to newly represent the detected crater.

* ***Interface/Exports***

This module outputs a set of coordinates that represent the bounding boxes (with respect to the input image) around a set of unique craters.

***6.3 module-3 Crater Recognition***

* ***Definition***

The purpose of this module is to identify which detected craters are actually craters.

* ***Responsibilities***

This module is responsible for taking a list of possible craters and filtering out which of these craters are actually craters.

* ***Composition***

There are no subcomponents for this module

* ***Uses/Interactions***

This module takes the output of the duplicate removal module as its input. After it is complete, any detected features that are not recognized as craters are sent to the Hazard Mapping Module.

* ***Processing***

This module utilizes a neural network to identify what is a crater and what isn’t a crater.

* ***Interface/Exports***

This Module outputs a filtered set of what has been recognized as a crater.

***6.4 module4 Hazard Mapping***

* ***Definition***

The purpose of this module is to identify which areas of the image are not safe for exploration.

* ***Responsibilities***

This module is responsible for calculating and color coding hazardous areas of the image according to calculated hazard levels.

* ***Composition***

There are no subcomponents for this module

* ***Uses/Interactions***

This module takes a list of non-crater features determined by the Crater Recognition Module.

* ***Processing***

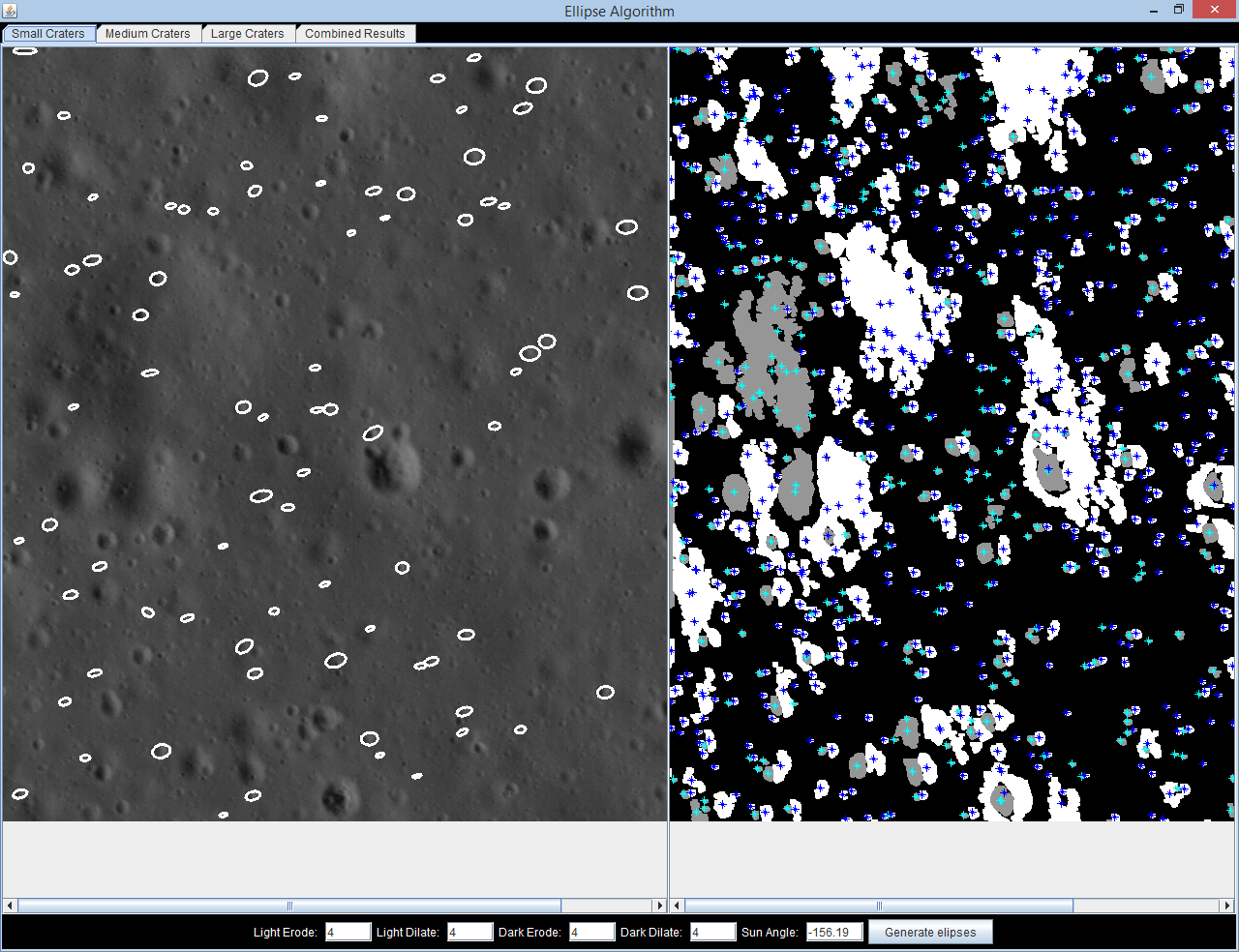
This module splits the input image into a per hectare grid and calculates the crater density for each hectare in the image. The crater density value indicates the level of danger; value above a certain value will be considered hazardous and there will be three levels of danger ranging from unsafe to extremely dangerous based on certain threshold values.

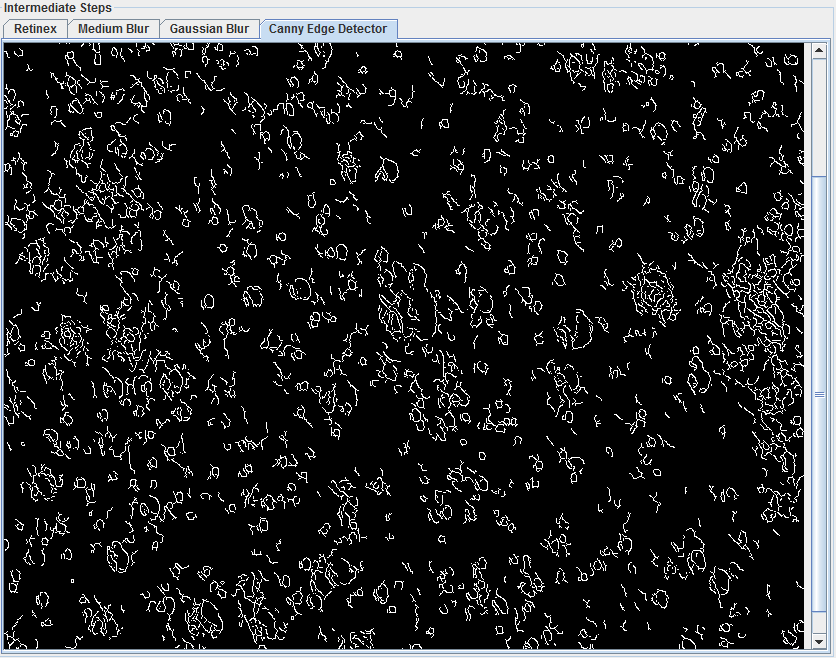
* ***Interface/Exports***

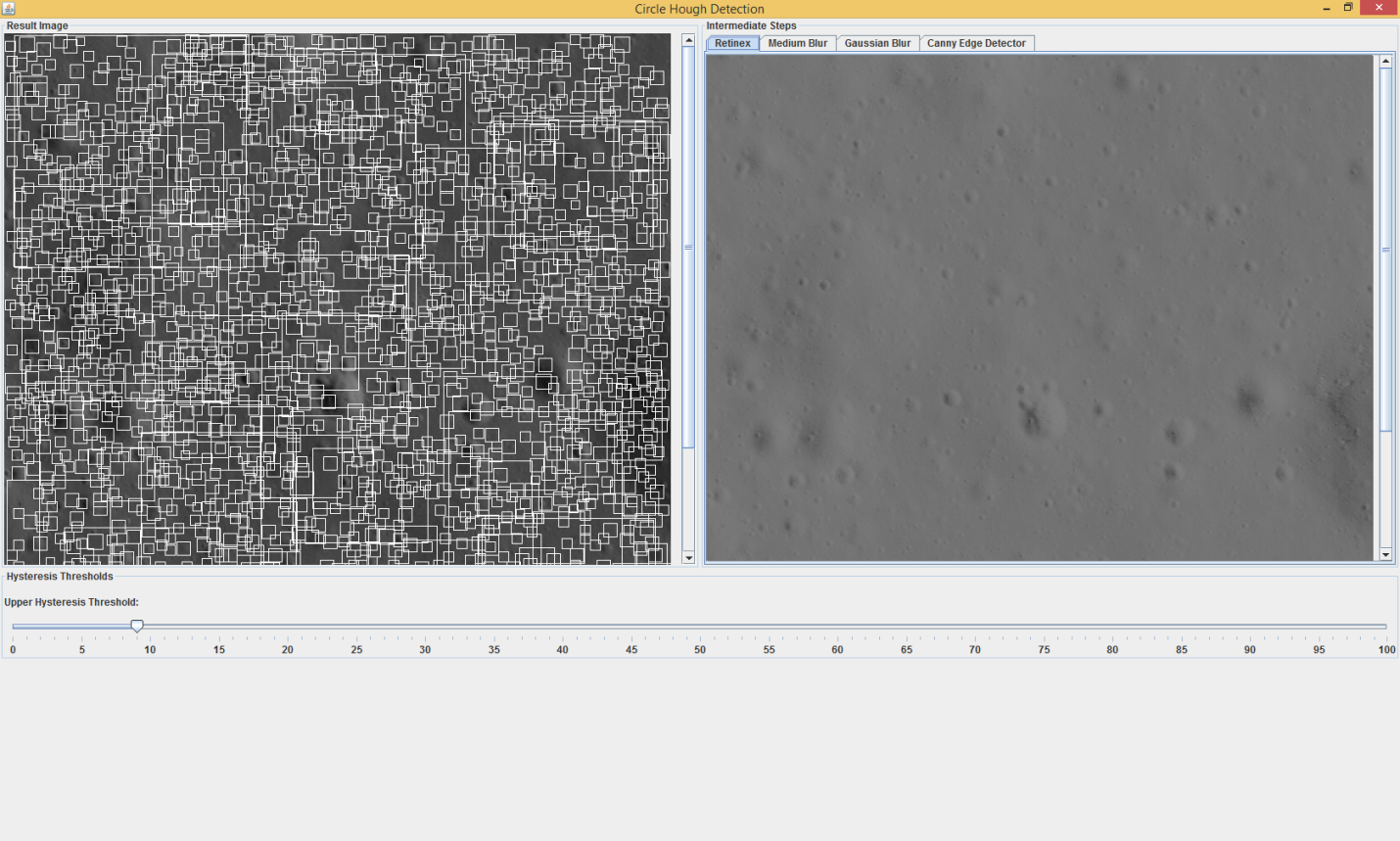
This modules outputs an color coded image where there is a gradient between green (corresponding to safe locations) and red (corresponding to extremely hazardous locations)

**Section 7: Graphical User Interface Design**

***GUI Design.***







**Section 8: Glossary**

|  |  |
| --- | --- |
| Term | Definition |
| LIPFD | Lunar Image Processing – Feature Detection |
| MIPFD | Mars Image Processing – Feature Detection |
| Neural network | Models inspired by biological neural networks used to estimate or approximate functions that can depend on a large number of inputs |
| Crater candidates | A list of detected craters that have not yet been classified/confirmed to be actual craters |
| Hectare | A metric unit of square measure, equal to 100 ares |
| Crater density | The number of craters or crater like features detected in a given area |