

**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования**

**«Национальный исследовательский университет  
«Высшая школа экономики»**

*Факультет прикладной математики и кибернетики  
Кафедра «Механика и математическое моделирование»*

**БАКАЛАВРСКАЯ РАБОТА**

На тему: «Определение формы горизонта на Луне»

Студент группы № МТ-71  
Иванов Антон Михайлович

(подпись)

Руководитель ВКР  
К.ф.-м.н. доцент Малашкин А.В.

(должность, звание, Ф.И.О.)

(подпись)

Москва, 2013

## Оглавление

	№ страницы
Глава 1. <i>Аннотация</i>	3
Глава 2. <i>Введение</i>	4
Глава 3. <i>Физическая постановка</i>	6
Глава 4. <i>Формат IMG файла</i>	11
Глава 5. <i>Алгоритм построения горизонта</i>	34
Глава 6. <i>Программная реализация</i>	38
Заключение	49
Список использованной литературы	50
Приложение	52

## **Аннотация**

Целью данной работы является разработка программно-математического обеспечения (ПМО) для оценки пригодности места, задаваемого координатами точки на поверхности Луны, для посадки ПсМ(посадочный модуль) по двум критериям: визуальной оценки рельефа по строящейся компьютерной модели ландшафта местности и линии горизонта, оценки показателей освещённости места посадки Солнцем по строящемуся графику на заданный интервал времени и по карте продолжительности видимости Солнца и Земли в окрестностях места посадки ПсМ. Результаты работы разрабатываемого ПМО, в виде моделей, карт и графиков, могут служить средствами по обоснованию выбора возможных мест посадки ПсМ с учетом особенностей рельефа опасных для посадки и функционирования на поверхности Луны в приполярных районах Луны:  $70-85^{\circ}$  с.ш.,  $30^{\circ}$  з.д.- $60^{\circ}$  в.д. и  $70-85^{\circ}$  ю.ш.,  $0-60^{\circ}$  в.д.

## Введение

Данная научная работа проводилась в рамках проекта «Луна Глоб». Целью проекта «Луна-Глоб» являются контактные и дистанционные научные исследования поверхности и внутреннего строения Луны, кратеров на полюсе, разведки природных ресурсов и воздействий на поверхность Луны космических лучей и электромагнитных излучений. Запуск КК «Луна-Глоб» запланирован на осень 2014 года на ракетеносителе «Союз» с космодрома Байконур.

Так как для решения задачи ( оценки пригодности варианта посадки ) не было найдено готовое ПМО, соответствующее необходимым критериям, было принято решение реализовать свое ПМО, которое в любой заданной точке на произвольный момент времени вычисляет :

- Локальный горизонт ( топография с учетом ближайших окрестностей )
- Топоцентрические координаты Солнца и Земли
- Продолжительность видимости Солнца
- Продолжительность видимости Земли

Для реализации данного функционала требуются значительные вычислительные мощности , по этому в нашем ПМО применяются различные алгоритмы оптимизации. Применение данных алгоритмов позволило достичь значительные преимущества перед конкурентами. Задачи, решение которых обеспечивает достижение поставленной цели:

1. Создание программных библиотек по работе с данными, описывающими рельеф поверхности Луны (исходные данные: LRO, источник архив Planetary Data System (PDS)), геометрию и небесную механику Солнечной системы (исходные данные: библиотека SPICE, предоставляющая навигационное и вспомогательно-информационное

обеспечение) для их дальнейшего эффективного использования в разрабатываемом ПМО.

2. Построение ПМО для построения компьютерной модели локального ландшафта местности в окрестности точки задаваемой пользователем и учитывающей положение Солнца и Земли (с использованием библиотеки OpenGL, данных LRO, данных программы SPICE), на определённый момент времени для визуальной оценки рельефа и освещённости в окрестности предлагаемого варианта места посадки.

3. Разработка ПМО для построения графиков видимости Солнца и Земли из заданной точки на поверхности Луны, учитывающих рельеф местности и карт освещённости, и последующего использования данных алгоритмов в процедурах построения графика освещённости на заданный интервал времени для оценки пригодности предлагаемого места посадки по техническим критериям энергообеспечения.

4. Разработка ПМО для определения координат места на поверхности Луны по заданному пользователем панорамному фотоизображению горизонта (линия горизонта и освещённости деталей рельефа в известный момент времени) в месте посадки ПсМ на Луне.

## Физическая постановка

Исходные данные для построения линии горизонта хранятся в формате PDS и находятся в открытом доступе - [http://imbrium.mit.edu/DATA/LOLA\\_GDR/POLAR/IMG/](http://imbrium.mit.edu/DATA/LOLA_GDR/POLAR/IMG/).

Эти данные получаются при помощи LRO(Лунный орбитальный зонд).

Рассмотрим его структуру:

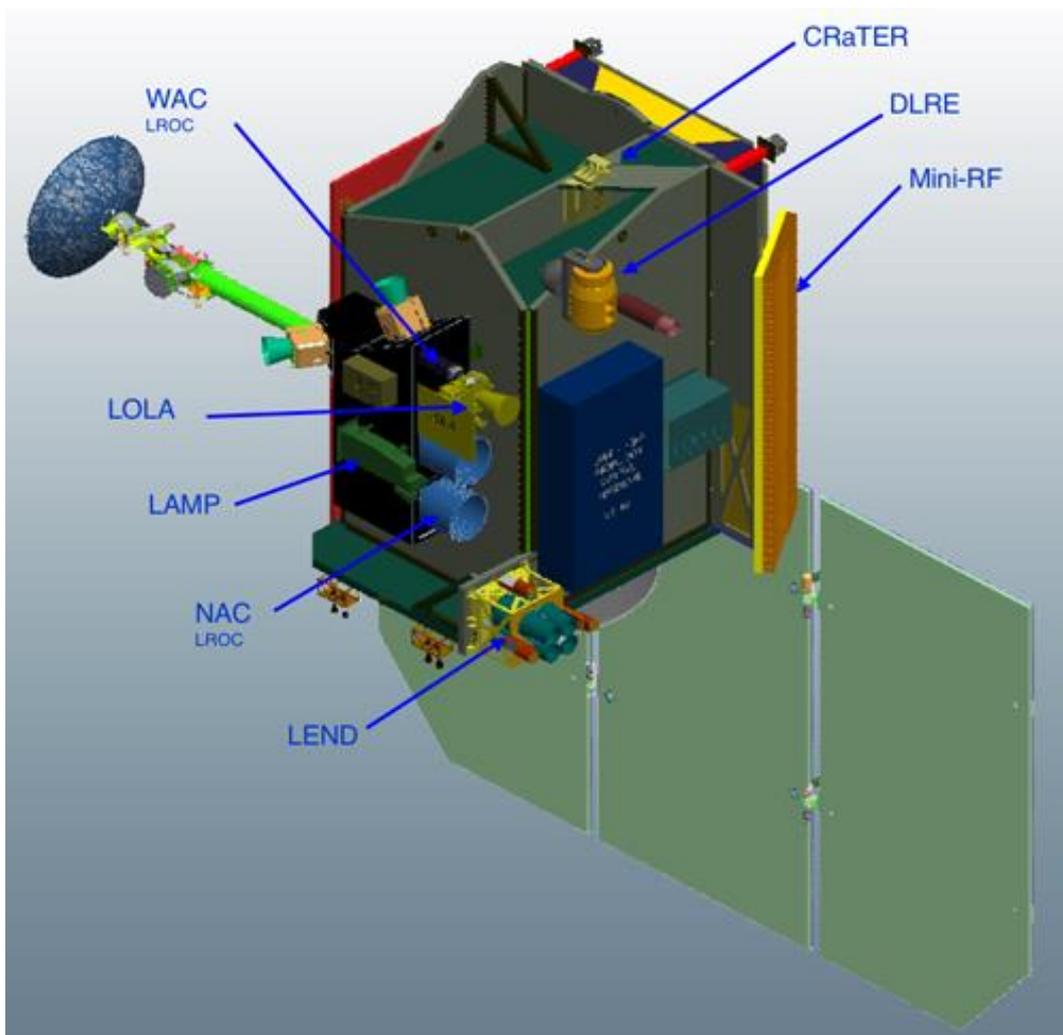


Рис. 1. Структура Лунного орбитального зонда

На данной схеме мы наблюдаем :

- **CRaTER** (**Cosmic Ray Telescope for the Effects of Radiation**) основным предназначением этого прибора является оценка вредного воздействия космических лучей и солнечной радиации на биологические объекты.

- **DLRE** (**The Diviner Lunar Radiometer Experiment**) — измерение теплового излучения лунной поверхности и его изменения в течение суток, информация нужна для будущих работ на поверхности Луны.

- **LAMP** (**The Lyman-Alpha Mapping Project**) — прибор для поиска льда в неосвещённых кратерах. Он наблюдает отражение ультрафиолетового излучения звёзд (линии серии Лаймана ) от лунной поверхности.

- **LEND** (**The Lunar Exploration Neutron Detector**) — прибор российского производства, для составления подробных карт содержания атомов водорода в лунной поверхности.

- **LOLA** (**The Lunar Orbiter Laser Altimeter**) лазерный альтиметр для составления точной карты высот.

- **LROC** (**The Lunar Reconnaissance Orbiter Camera**) главная оптическая камера, для получения фотографий поверхности Луны с разрешением до полуметра, с помощью которых будут искать подходящие места для посадки пилотируемых экспедиций.

LROC состоит из трех камер: камеры низкого разрешения (WAC) и двух камер высокого разрешения (NAC), первая из которых предназначена для получения общих планов местности, а две другие — для фотографий с большим разрешением. Планируется сделать фотографии примерно 8 % лунной поверхности, в том числе 1) всех мест, связанных с деятельностью человека: мест посадок пилотируемых кораблей «Аполлон», американских и советских автоматических станций, а также кратеров, образовавшихся при падении искусственных спутников Луны и других аппаратов; 2) 50 «зон повышенного интереса», отобранных учеными и 3) околополярных районов, которые сейчас считаются наиболее перспективными для организации обитаемой базы.

- **Mini-RF** (The Miniature Radio Frequency) — тестирование нового легковесного радара с синтезированной апертурой.

Аппарат будет производить следующие исследования:

- Изучение лунной глобальной топографии
- Измерение радиации на лунной орбите
- Изучение лунных полярных регионов, включающее в себя поиск залежей водяного льда и исследование параметров освещённости
- Составление сверхточных карт с нанесением объектов не менее 0,5 метра с целью найти лучшие посадочные площадки.

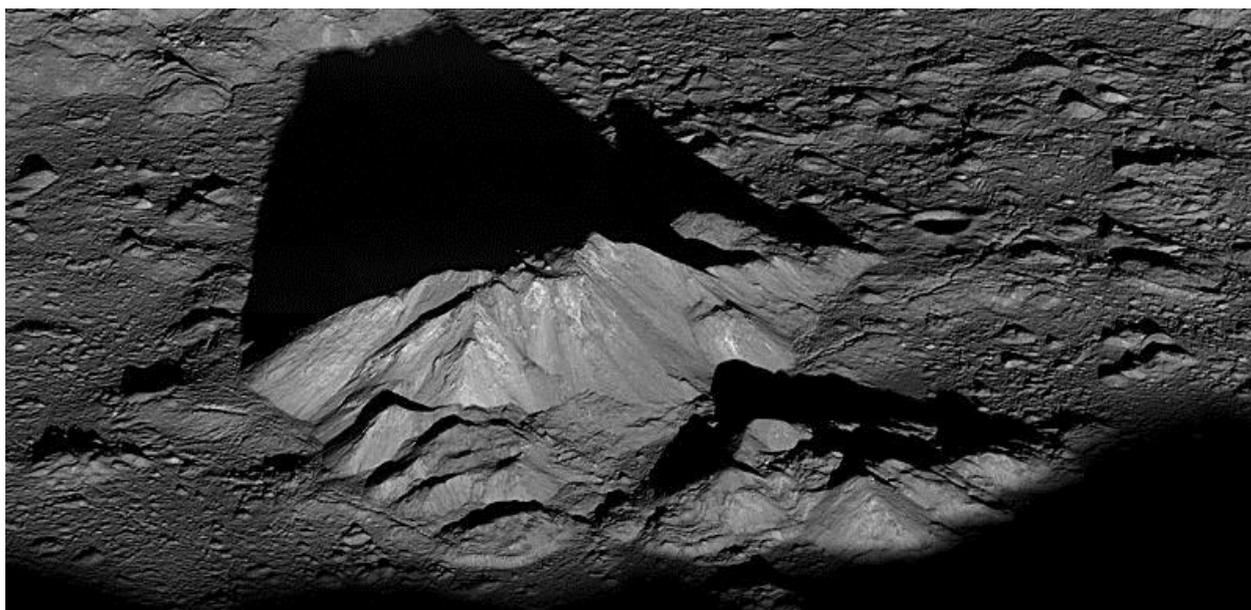
17 июля 2009 г., перед 40-летним юбилеем первой высадки на Луну, были опубликованы сделанные Orbiter'ом фотографии. На будущее запланированы съёмки других памятных мест Луны, например стоянок «Луноходов».

6 сентября 2011 г. NASA представила более детальные снимки мест пилотируемых экспедиций, сделанные LRO, для этого зонд был переведен на более низкую орбиту над поверхностью Луны.

16 августа 2012 г. NASA сообщила о нахождении в атмосфере Луны атомов гелия при помощи спектроскопа LAMP. Кроме того, в почве на поверхности Луны исследователи обнаружили атомы аргона.

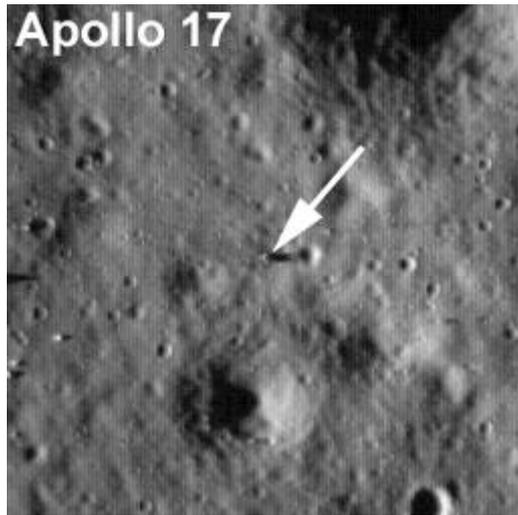
В начале сентября 2012 года с помощью легковесного радара с синтезированной апертурой (Mini-RF) были открыты залежи водяного льда, массовая доля которого составляет 5-10% вещества, слагающего стенки кратера Шеклтона. Эти цифры перекрыли предыдущие консервативные оценки количества воды в лунном грунте в 5-10 раз. Эти цифры, перекрывающие в несколько раз предыдущие консервативные оценки количества воды в лунном грунте, позволяют с еще большим оптимизмом смотреть на будущую колонизацию спутника Земли и строительство там стационарных населенных баз.

Пример сделанного снимка (Кратер Тихо в лучах восходящего Солнца) :



**Рис. 2. Кратер Тихо в лучах восходящего Солнца**

Место посадки Аполлон 17 :



**Рис. 3. Место посадки Аполлон 17**

## Формат IMG файла.

Информация о поверхности предоставляется в следующих файлах:

- \*.LBL
- \*.IMG

LBL файл содержит в себе информацию о исследуемом регионе и параметры с которыми были произведены замеры.

Рассмотрим содержание произвольного LBL файла:

**Табл. 1. Описание LBL файла**

Название параметра	Значение
PDS_VERSION_ID	"PDS3"
PRODUCT_VERSION_ID	"V2.0"
DATA_SET_ID	"LRO-L-LOLA-4-GDR-V1.0"
PRODUCT_ID	"LDRM_85N_40M"
INSTRUMENT_HOST_NAME	"LUNAR RECONNAISSANCE ORBITER"
INSTRUMENT_NAME	"LUNAR ORBITER LASER ALTIMETER"
INSTRUMENT_ID	"LOLA"
MISSION_PHASE_NAME	{"COMMISSIONING","NOMINAL MISSION","SCIENCE MISSION"}
TARGET_NAME	MOON
START_TIME	2009-07-13T17:33:17.246
STOP_TIME	2011-12-11T13:33:12.807
PRODUCT_CREATION_TIME	2012-06-15T00:00:00
PRODUCER_ID	LRO_LOLA_TEAM

PRODUCER_FULL_NAME	"DAVID E. SMITH"
PRODUCER_INSTITUTION_NAME	"GODDARD SPACE FLIGHT CENTER"
DESCRIPTION	"This data product is a map of the surface roughness of the Moon at a resolution of 40m/pix by 40m/pix true at the pole in polar stereographic (spherical) projection, based on altimetry data acquired by the LOLA instrument [ZUBERETAL2012]. The LOLA Laser 1 and 2 data through mission phase LRO_SM_17 are the source for this data set. The residual standard deviation of altitudes from three successive laser shots with n-3 degrees of freedom after fitting a plane to n=5 to 15 profile returns, is averaged in each pixel. Depending on orbital velocity, probability of detection, and spacecraft altitude, the slope baseline may vary from 30 to 120 meters. This label describes an IMG file in the form of a binary resampling with pixel registration."
OBJECT	UNCOMPRESSED_FILE
FILE_NAME	"LDRM_85N_40M.IMG"
RECORD_TYPE	FIXED_LENGTH
FILE_RECORDS	7584

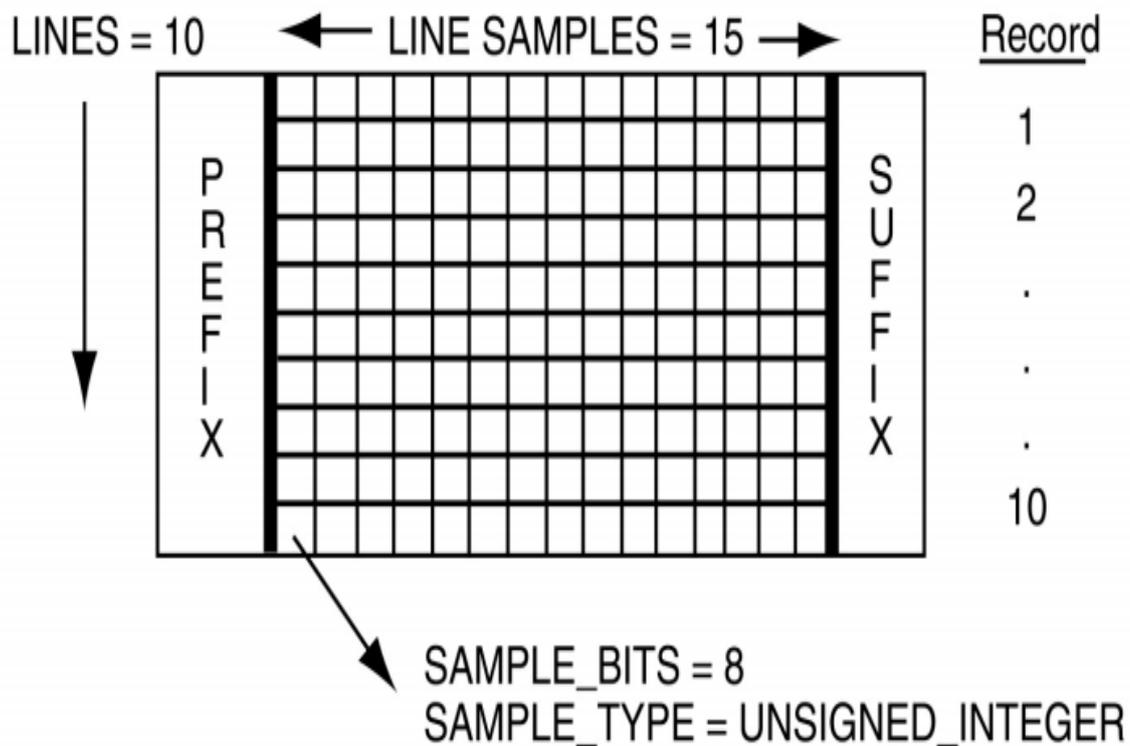
RECORD_BYTES	15168
^IMAGE	"LDRM_85N_40M.IMG"
OBJECT	IMAGE
NAME	ROUGHNESS
DESCRIPTION	"Each sample represents surface roughness in meters. A scaling was applied to make use of the signed 16-bit integer range: a flat surface corresponds to a numerical value of 0, and a surface with 32.767 meters standard deviation corresponds to a numerical value of 32767."
LINES	7584
LINE_SAMPLES	7584
DERIVED_MINIMUM	0
DERIVED_MAXIMUM	26732
SAMPLE_TYPE	LSB_INTEGER
SAMPLE_BITS	16
UNIT	METER
SCALING_FACTOR	0.001
OFFSET	0.
MISSING_CONSTANT	-32768
END_OBJECT	IMAGE
END_OBJECT	UNCOMPRESSED_FILE
OBJECT	IMAGE_MAP_PROJECTION
^DATA_SET_MAP_PROJECTION	"DSMAP_POLAR.CAT"
MAP_PROJECTION_TYPE	"POLAR STEREOGRAPHIC"
KEYWORD_LATITUDE_TYPE	"PLANETOCENTRIC"
MAP_RESOLUTION	758.084 <pix/deg>

A_AXIS_RADIUS	1737.4 <km>
B_AXIS_RADIUS	1737.4 <km>
C_AXIS_RADIUS	1737.4 <km>
FIRST_STANDARD_PARALLEL	'N/A'
SECOND_STANDARD_PARALLEL	'N/A'
POSITIVE_LONGITUDE_DIRECTION	"EAST"
CENTER_LATITUDE	90 <deg>
CENTER_LONGITUDE	0 <deg>
REFERENCE_LATITUDE	'N/A'
REFERENCE_LONGITUDE	'N/A'
LINE_FIRST_PIXEL	1
LINE_LAST_PIXEL	7584
SAMPLE_FIRST_PIXEL	1
SAMPLE_LAST_PIXEL	7584
MAP_PROJECTION_ROTATION	0.0
MAP_SCALE	40 <m/pix>
MAXIMUM_LATITUDE	90 <deg>
MINIMUM_LATITUDE	85 <deg>
WESTERNMOST_LONGITUDE	'N/A'
EASTERNMOST_LONGITUDE	'N/A'
LINE_PROJECTION_OFFSET	3791.5 <pix>
SAMPLE_PROJECTION_OFFSET	3791.5 <pix>
COORDINATE_SYSTEM_TYPE	"BODY-FIXED ROTATING"
COORDINATE_SYSTEM_NAME	"MEAN EARTH/POLAR AXIS OF DE421"
END_OBJECT	IMAGE_MAP_PROJECTION

Как видно из данных , этому LBL файлу соответствует IMG файл - LDRM\_85N\_40M.IMG

В IMG файле храниться точная карта высот , полученная с помощью LOLA.

Структура IMG файла следующая :

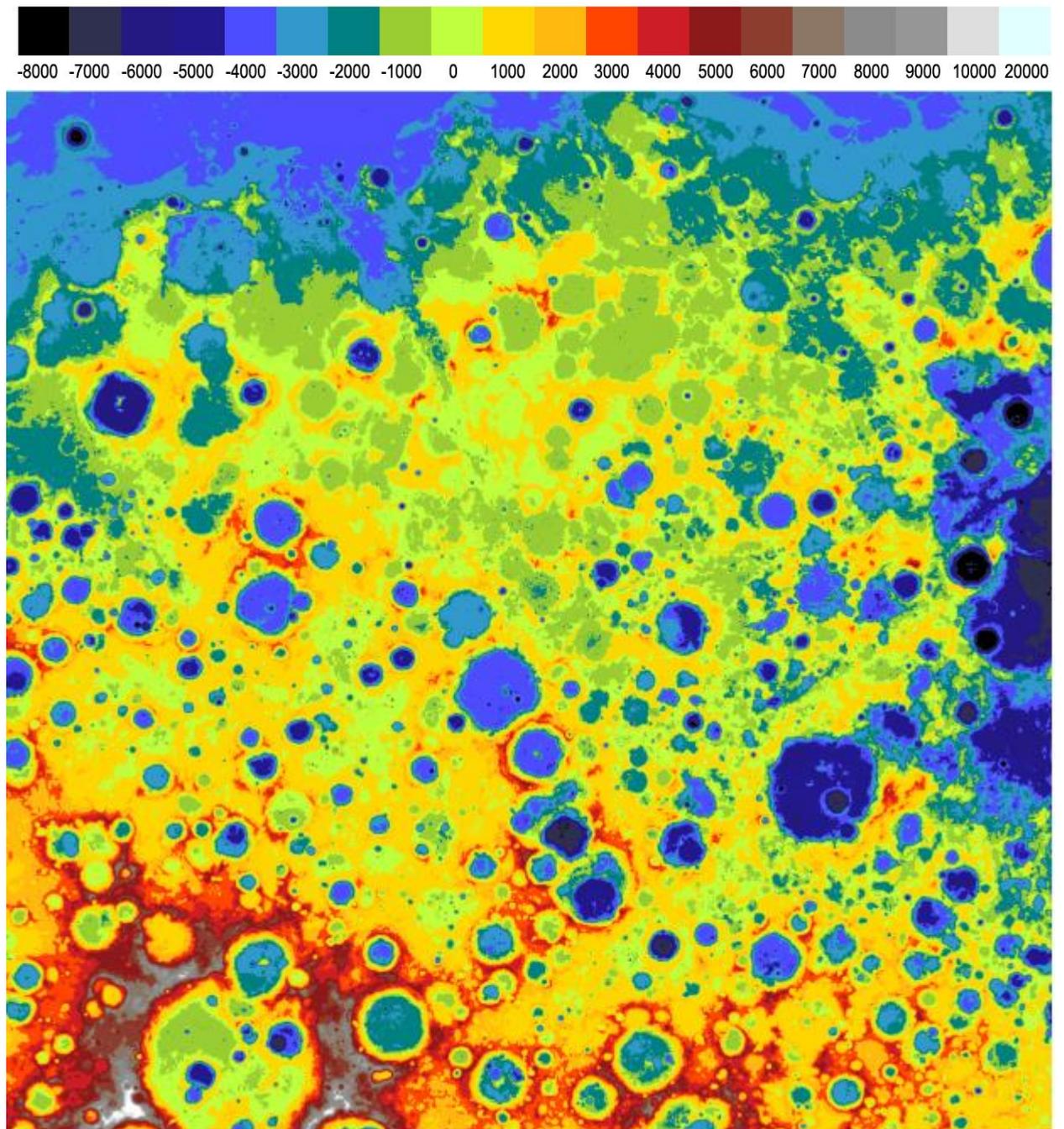


**Рис.4. Формат IMG файла**

- **LINES** - количество строк в файле
- **LINE SAMPLES** - количество элементов в строке.
- **SAMPLE\_BITS** - размера каждого элемента в битах.
- **SAMPLE\_TYPE** - тип данных элемента.

Взяв произвольный IMG файл мы можем построить карту высот.

Например :

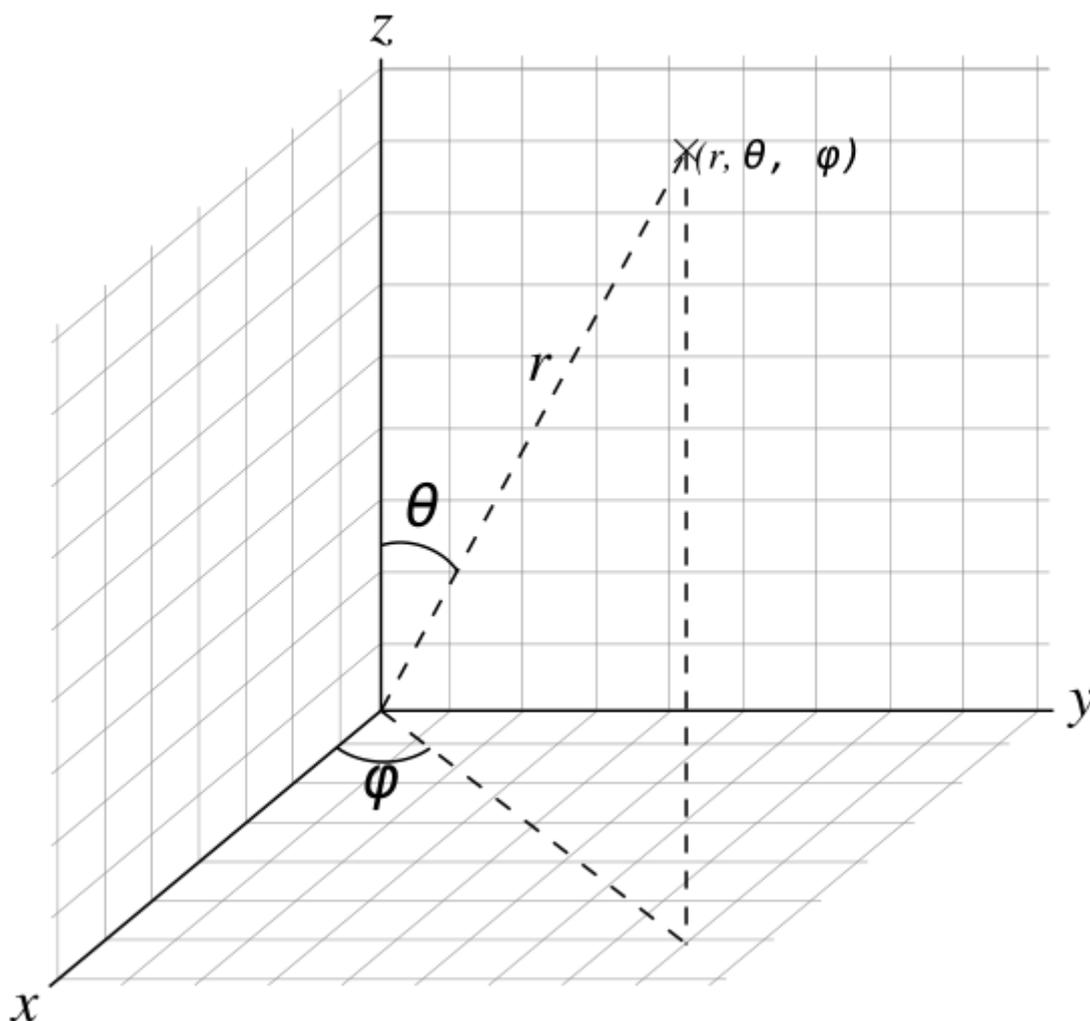


**Рис. 5. Карта высот**

По IMG файлу также можно построить трехмерную поверхность. Для этого надо сделать некоторые преобразования.

Дадим определения сферическим координатам:

Сферическими координатами называют систему координат для отображения геометрических свойств фигуры в трёх измерениях посредством задания трёх координат  $(r, \theta, \varphi)$ , где  $r$  — расстояние до начала координат, а  $\theta$  и  $\varphi$  — зенитный и азимутальный угол соответственно.



**Рис.6. Сферическая система координат**

Точка  $P$  имеет три декартовых и три сферических координаты.

Три координаты  $(r, \theta, \varphi)$  определены как:

- $r \geq 0$  — расстояние от начала координат до заданной точки  $P$ .

- $0 \leq \theta \leq 180^\circ$  — угол между осью  $Z$  и отрезком, соединяющим начало координат и точку  $P$ .
- $0 \leq \varphi < 360^\circ$  — угол между осью  $X$  и проекцией отрезка, соединяющего начало координат с точкой  $P$ , на плоскость  $XY$ .

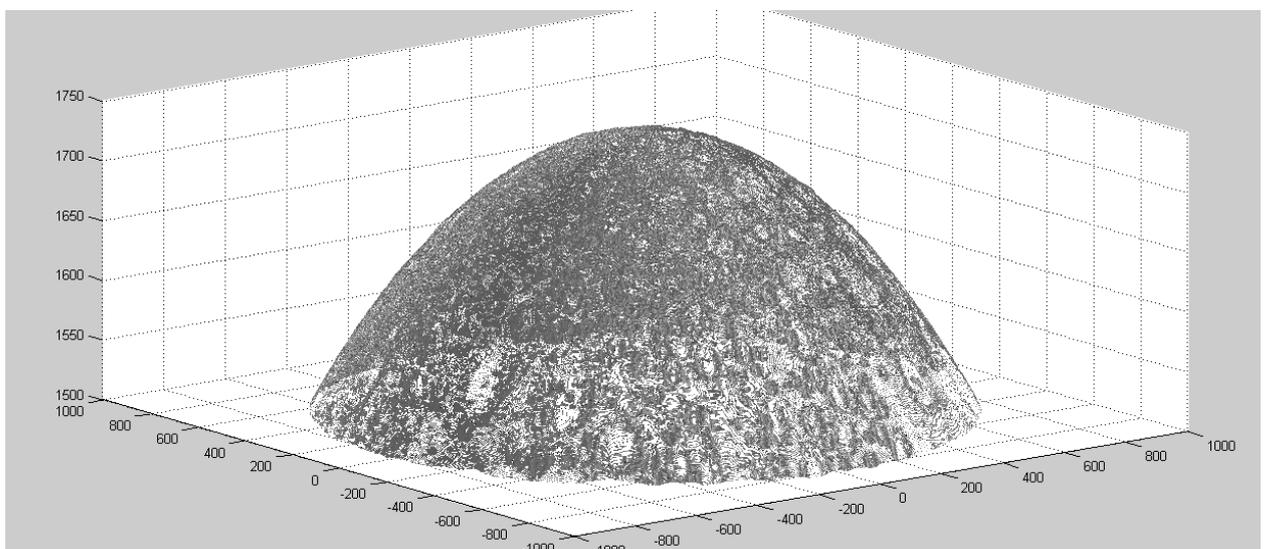
Для перехода к декартовой системе координат необходимо выполнить следующие преобразования :

$$\begin{cases} x = r \sin \theta \cos \varphi, \\ y = r \sin \theta \sin \varphi, \\ z = r \cos \theta. \end{cases}$$

Обратное преобразование происходит по следующим формулам:

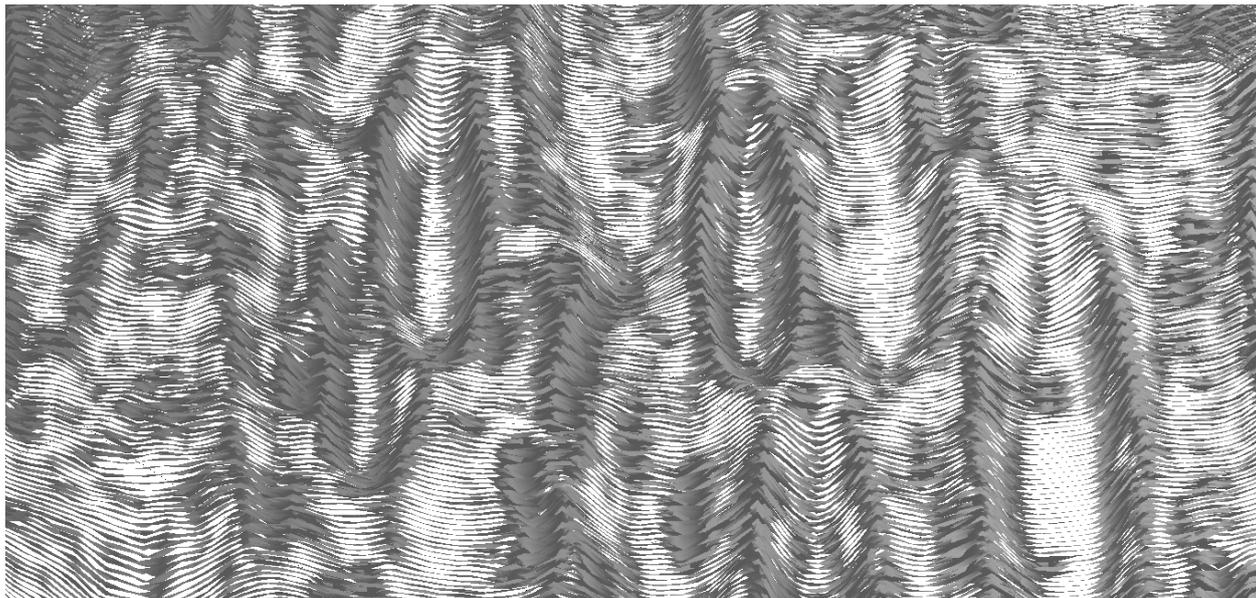
$$\begin{cases} r = \sqrt{x^2 + y^2 + z^2}, \\ \theta = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) = \operatorname{arctg}\left(\frac{\sqrt{x^2 + y^2}}{z}\right), \\ \varphi = \operatorname{arctg}\left(\frac{y}{x}\right). \end{cases}$$

Построение пробной поверхности в пакете MATLAB по 2 408 703 точкам



**Рис. 7. Общий вид построенной поверхности**

Приближение построенного участка поверхности:



**Рис. 8. Кусок построенной поверхности**

Для получения быстрого результата можно воспользоваться программным обеспечением Blender.

Blender — свободный пакет для создания трёхмерной компьютерной графики, включающий в себя средства моделирования, анимации, рендеринга, постобработки видео, а также создания интерактивных игр.

Характерной особенностью пакета Blender является его небольшой размер. Установленный пакет занимает от 30 до 45 МБ. В базовую поставку не входят развёрнутая документация и большое количество демонстрационных сцен.

Функции пакета:

- Поддержка разнообразных геометрических примитивов, включая полигональные модели, систему быстрого моделирования в режиме subdivision surface (SubSurf), кривые Безье, поверхности

NURBS, metaballs(метасферы), скульптурное моделирование и векторные шрифты.

- Универсальные встроенные механизмы рендеринга и интеграция с внешним рендерером YafRay, LuxRender и многими другими.

- Инструменты анимации, среди которых инверсная кинематика, скелетная анимация и сеточная деформация, анимация по ключевым кадрам, нелинейная анимация, редактирование весовых коэффициентов вершин, ограничители, динамика мягких тел (включая определение коллизий объектов при взаимодействии), динамика твёрдых тел на основе физического движка Bullet, система волос на основе частиц и система частиц с поддержкой коллизий.

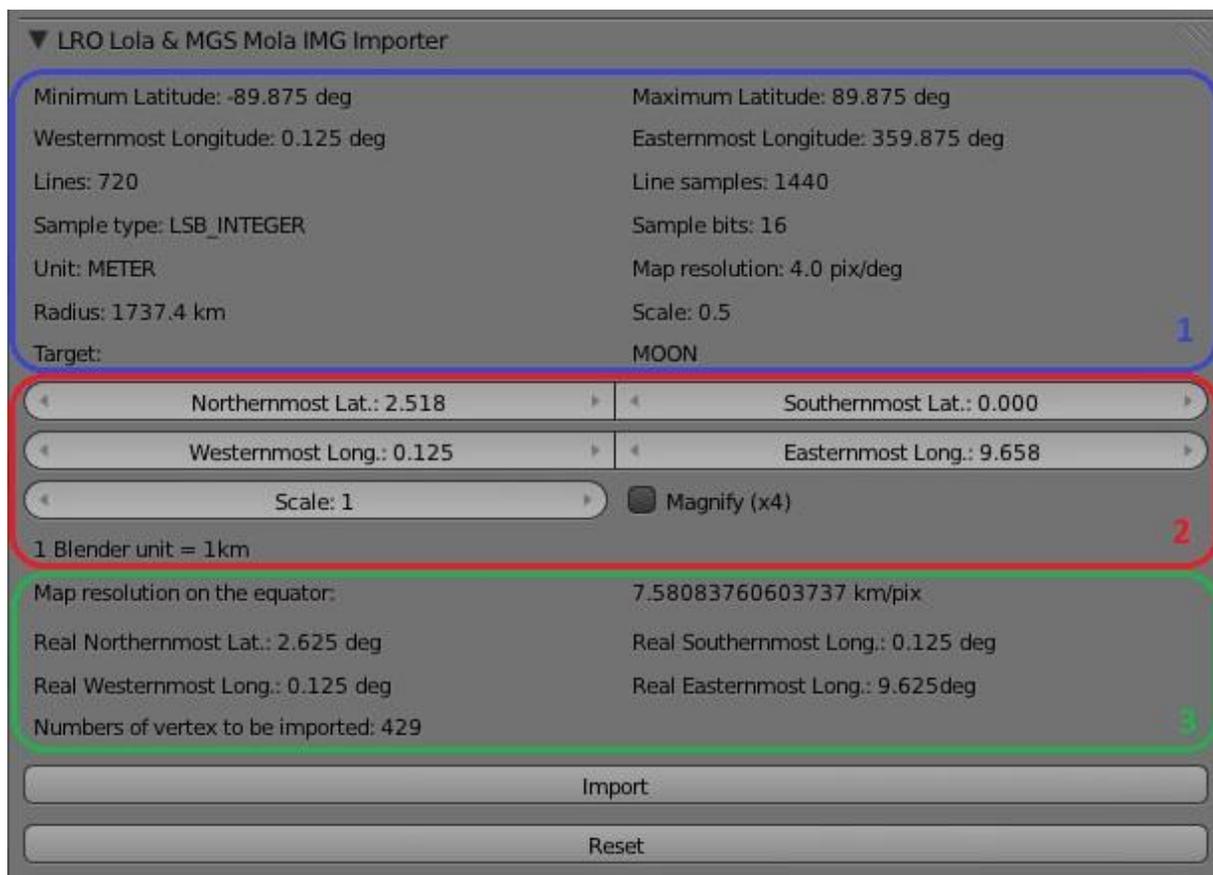
- Python используется как средство создания инструментов и прототипов, системы логики в играх, как средство импорта/экспорта файлов (например COLLADA), автоматизации задач.

- Базовые функции нелинейного редактирования и комбинирования видео.

- Game Blender — подпроект Blender, предоставляющий интерактивные функции, такие как определение коллизий, движок динамики и программируемая логика. Также он позволяет создавать отдельные real-time приложения начиная от архитектурной визуализации до видео игр.

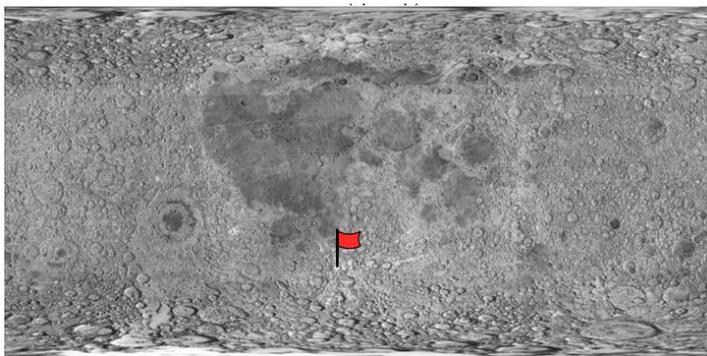
В данной программе присутствует возможность подключения модулей , которые существенно увеличивают функционал.

Для работы с форматом PDS IMG реализован специальный модуль , который позволяет обрабатывать подобные файлы(LRO Lola & MGS Mola img Importer).



**Рис. 9. Интерфейс модуля LRO Lola & MGS Mola img Importer**

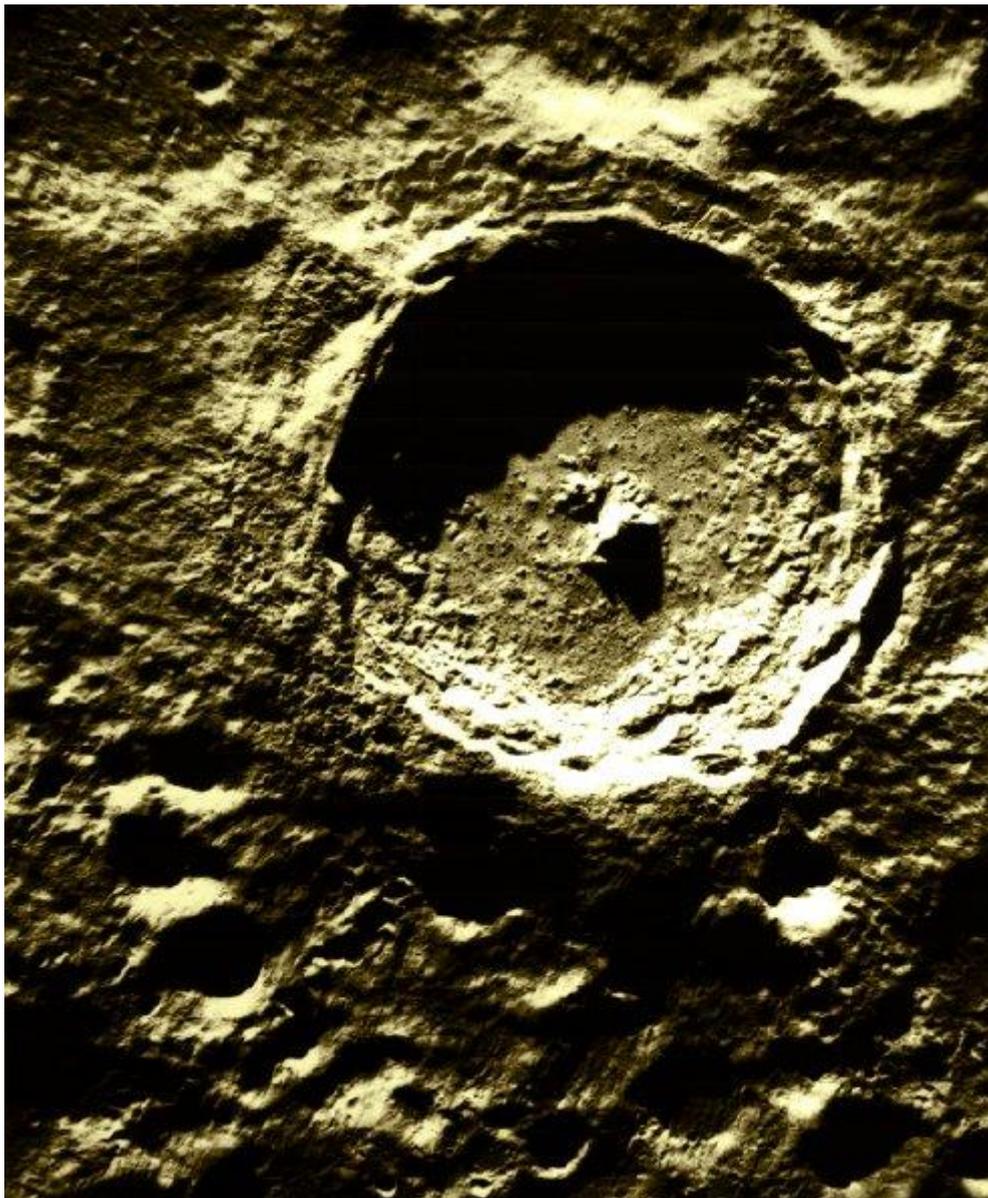
Попробуем построить трехмерную поверхность описывающую кратер Тихо( координаты на Луне -  $43^{\circ} 18' 0''$  S,  $11^{\circ} 12' 0''$  W ( $-43.3^{\circ}$ ,  $-11.2^{\circ}$ )).



**Рис. 10. Координаты кратера Тихо**

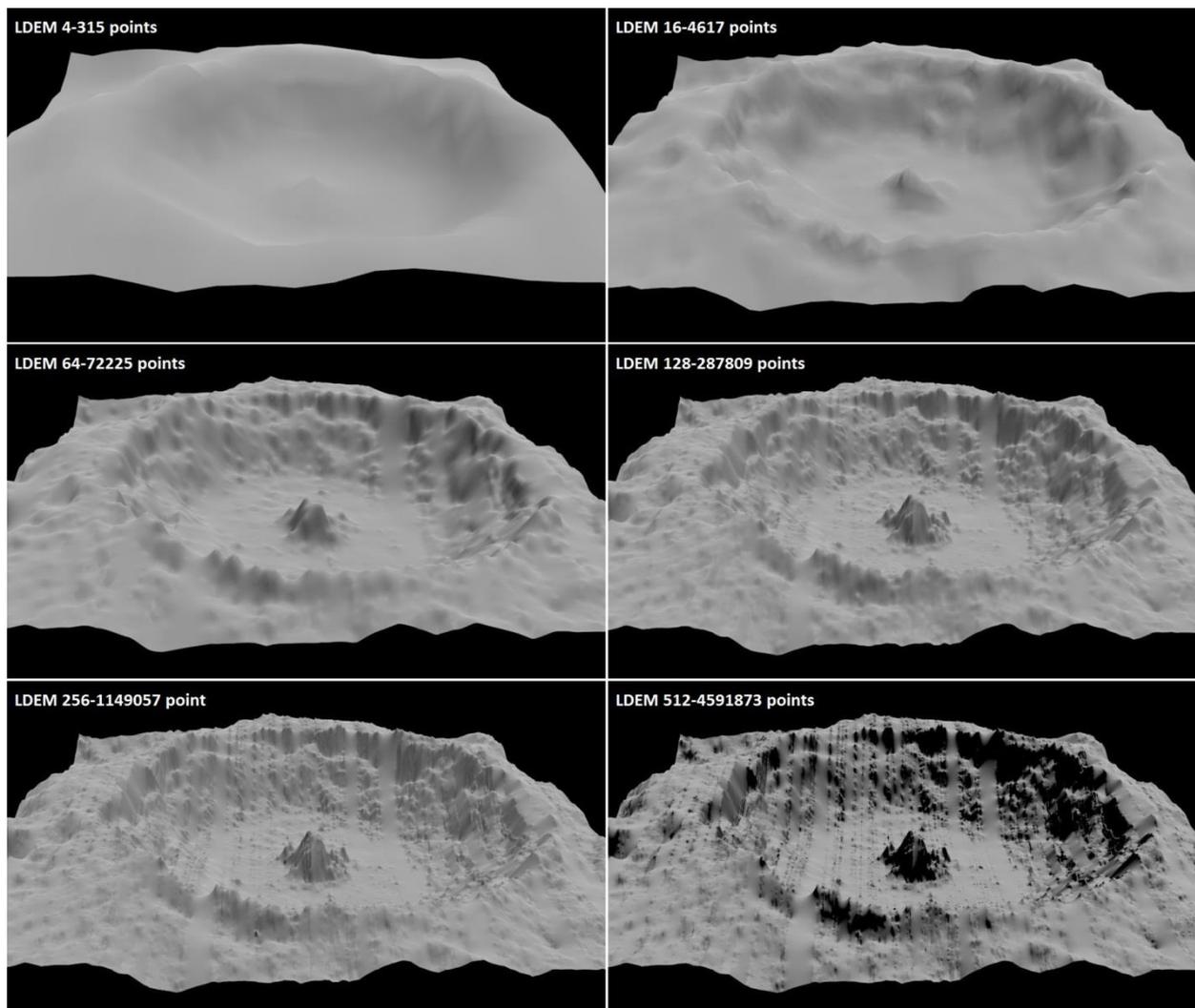
Тихо — ударный кратер на Луне, один из наиболее интересных лунных кратеров, назван в честь датского астронома и алхимика середины XVI века Тихо Браге.

Фотография NASA:



**Рис. 11. Фотография кратера Тихо**

Построенная трехмерная модель с помощью модуля NASA\_IMG\_Importer :



**Рис. 12. Построение 3-D модели кратера Тихо**

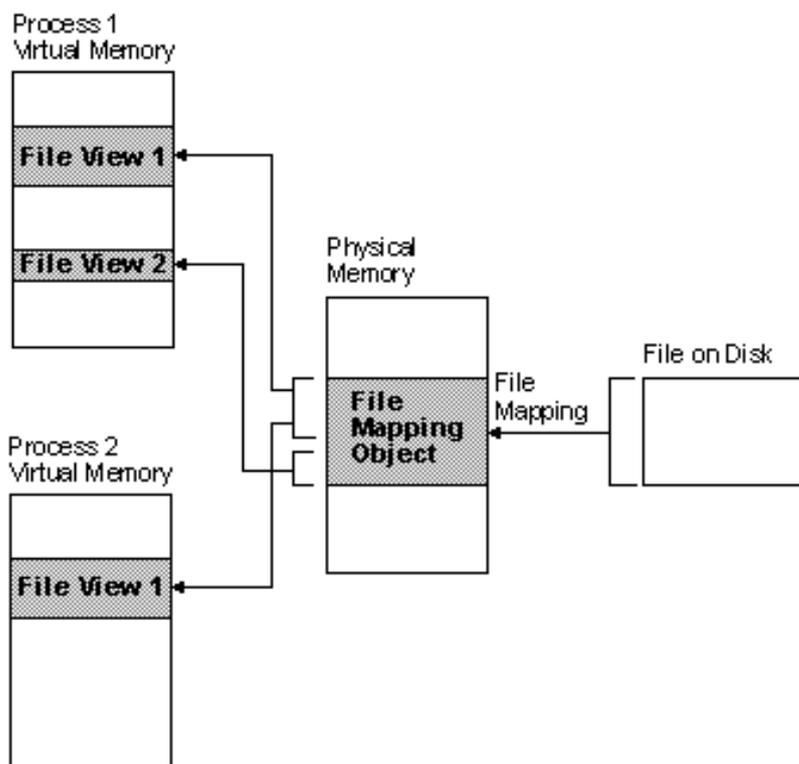
Как видно из предоставленных моделей , точность построения поверхности очень сильно зависит от количества точек, самый лучший результат можно наблюдать при 4 591 873 точках.

Размеры поверхностей , которые интересуют нас по расположению и по качеству , достаточно больших размеров в среднем от 500 МВ.

Даже для современного персонального компьютера - это достаточно большой размер для обработки. Так как для построения линии горизонта , нам нужно обрабатывать такие большие файлы и

следовательно считывать их с диска это приведет к значительному снижению производительности.

Для того , чтобы снизить нагрузку было решено использовать механизм отображения файла в память.



**Рис. 13. Механизм работы отображения файла в память**

Отображение файла в память (на память) — это такой способ работы с файлами в некоторых операционных системах, при котором всему файлу или некоторой непрерывной части этого файла ставится в соответствие определённый участок памяти (диапазон адресов оперативной памяти). При этом чтение данных из этих адресов фактически приводит к чтению данных из отображенного файла, а запись данных по этим адресам приводит к записи этих данных в файл. Примечательно то, что отображать на память часто можно не только обычные файлы, но и файлы устройств.

Альтернативой отображению может служить прямое чтение файла или запись в файл.

Такой способ работы менее удобен по следующим причинам:

1. Необходимо постоянно помнить текущую позицию файла и вовремя её передвигать на нужное место, в которое необходимо записать или из которого необходимо прочитать.

2. Каждый вызов смены/чтения текущей позиции, записи/чтения — это системный вызов, который приводит к потере времени.

3. Для работы через чтение/запись всё равно приходится выделять буферы определённого размера, таким образом, в общем виде работа состоит из трёх этапов: чтение в буфер -> модификация данных в буфере -> запись в файл. При отображении же работа состоит только из одного этапа: модификация данных в определённой области памяти.

Дополнительным преимуществом использования отображения является меньшая по сравнению с чтением/записью нагрузка на операционную систему — дело в том, что при использовании отображений операционная система не загружает в память сразу весь файл, а делает это по мере необходимости, блоками размером со страницу памяти (как правило 4 килобайта). Таким образом, даже имея небольшое количество физической памяти (например 32 мегабайта), можно легко отобразить файл размером 100 мегабайт или больше, и при этом для системы это не приведет к большим накладным расходам. Также выигрыш происходит и при записи из памяти на диск: если вы обновили большое количество данных в памяти, они могут быть одновременно (за один проход головки над диском) сброшены на диск.

Файл, отображенный на память, удобен также тем, что можно достаточно легко менять его размер и при этом (после переотображения) получать в своё распоряжение непрерывный кусок памяти нужного

размера. С динамической памятью такой трюк не всегда возможен из-за явления фрагментации. Когда же мы работаем с отображенным на память файлом — менеджер памяти автоматически настраивает процессор так, что странички ОЗУ, хранящие соседние фрагменты файла, образуют непрерывный диапазон адресов.

Основная причина, по которой следует пользоваться отображением — выигрыш в производительности. Однако, необходимо помнить о компромиссах, на которые придется пойти. Обычный ввод-вывод чреват накладными расходами на дополнительные системные вызовы и лишнее копирование данных, использование отображений чревато замедлениями из-за страничных ошибок доступа.

Допустим, страница, относящаяся к нужному файлу, уже лежит в кэше, но не ассоциирована с данным отображением. Если она была изменена другим процессом, то попытка ассоциировать её с отображением может закончиться неудачей и привести к необходимости повторно зачитывать данные с диска, либо сохранять данные на диск. Таким образом, хотя программа и делает меньше операций для доступа через отображение, в реальности операция записи данных в какое-то место файла может занять больше времени, чем с использованием операций файлового ввода-вывода (при том, что в среднем использование отображений даёт выигрыш).

Другой недостаток в том, что размер отображения зависит от используемой архитектуры. Допустим, 32-битные архитектуры (Intel 386, ARM 9) не могут создавать отображения длиной более 4 Гб.

Для использования данного механизма, необходимо воспользоваться API операционной системы.

Практически все операционные системы (UNIX, Windows, Mac OS, и т. д.) имеют API, с помощью которого программисты могут создавать приложения для этой операционной системы. Главный API операционных систем — это множество системных вызовов.

В индустрии программного обеспечения общие стандартные API для стандартной функциональности имеют важную роль, так как они гарантируют, что все программы, использующие общий API, будут работать одинаково хорошо или, по крайней мере, типичным привычным образом. В случае API графических интерфейсов это означает, что программы будут иметь похожий пользовательский интерфейс, что облегчает процесс освоения новых программных продуктов.

Для проецирования файлов в память в операционной системе Windows используются следующие API функции:

- CreateFile
- CreateFileMapping
- MapViewOfFile
- UnmapViewOfFile

Для Unix систем:

- open
- mmap

Наше ПМО реализовано для операционной системы MS Windows , поэтому рассмотрим сигнатуры данных функций(создание / открытие файла , создание / открытие объекта проецирования , проецирование файла в адресное пространство , удаление спроецированного объекта ) :

HANDLE WINAPI CreateFile(

    \_In\_    LPCTSTR lpFileName,  
    \_In\_    DWORD dwDesiredAccess,  
    \_In\_    DWORD dwShareMode,  
    \_In\_opt\_ LPSECURITY\_ATTRIBUTES lpSecurityAttributes,  
    \_In\_    DWORD dwCreationDisposition,  
    \_In\_    DWORD dwFlagsAndAttributes,  
    \_In\_opt\_ HANDLE hTemplateFile

);

HANDLE WINAPI CreateFileMapping(

    \_In\_    HANDLE hFile,  
    \_In\_opt\_ LPSECURITY\_ATTRIBUTES lpAttributes,  
    \_In\_    DWORD flProtect,  
    \_In\_    DWORD dwMaximumSizeHigh,  
    \_In\_    DWORD dwMaximumSizeLow,  
    \_In\_opt\_ LPCTSTR lpName

);

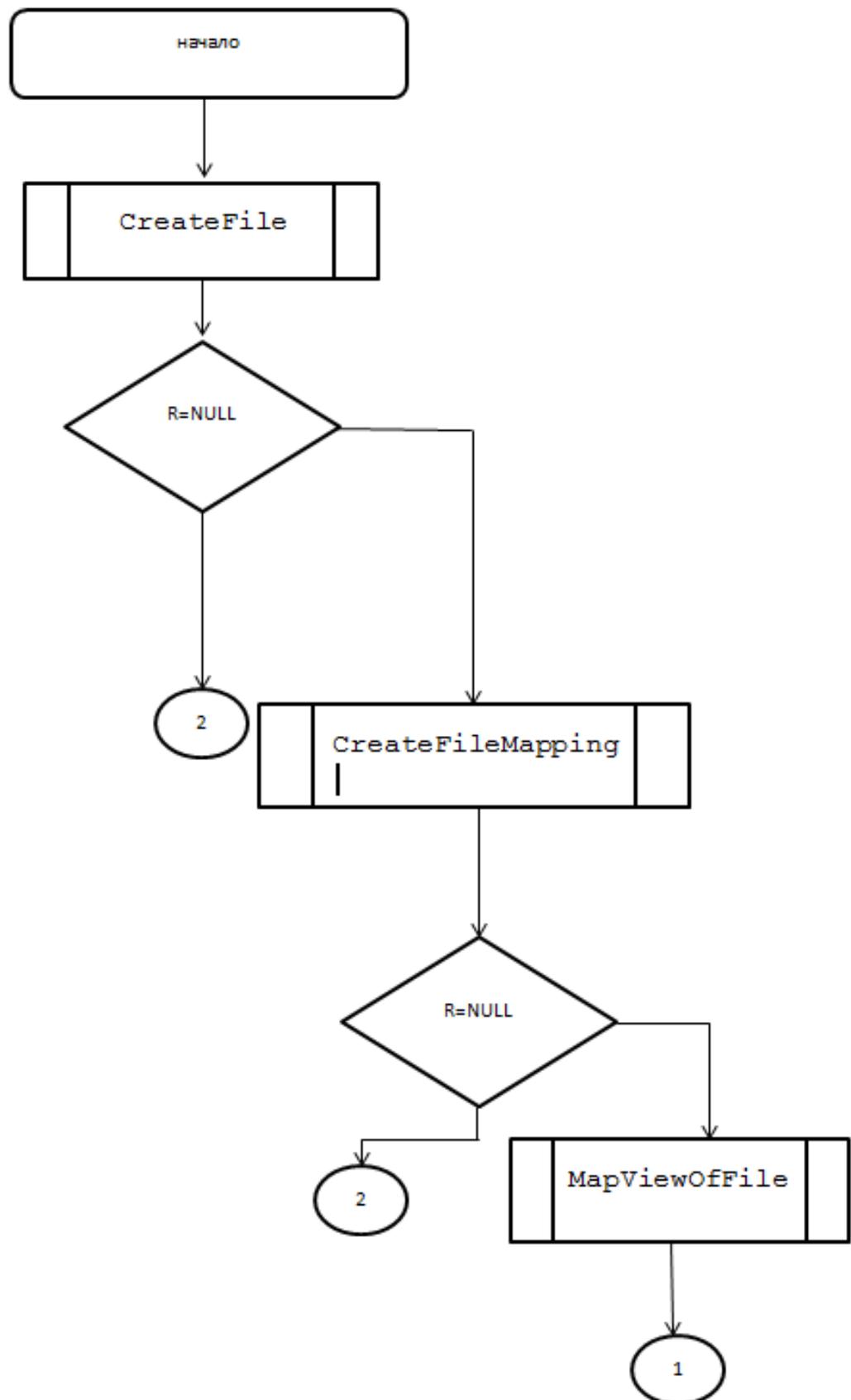
LPVOID WINAPI MapViewOfFile(

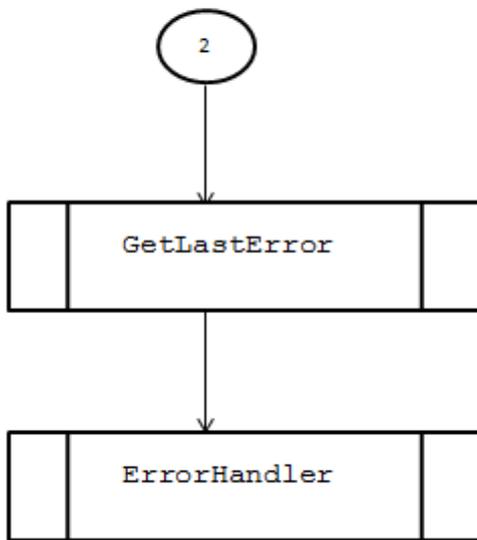
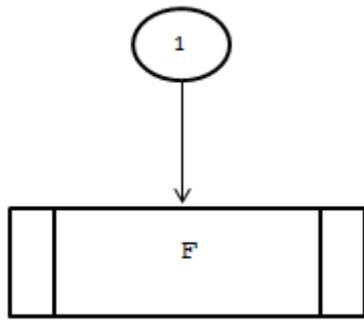
    \_In\_    HANDLE hFileMappingObject,

```
_In_ DWORD dwDesiredAccess,  
_In_ DWORD dwFileOffsetHigh,  
_In_ DWORD dwFileOffsetLow,  
_In_ SIZE_T dwNumberOfBytesToMap  
);
```

```
BOOL WINAPI UnmapViewOfFile(  
_In_ LPCVOID lpBaseAddress  
);
```

Блок Схема алгоритма:





На данной блок схеме введены следующие обозначения:

R – результат выполнения WinApi функции

F – функция реализуемая программистом в которую передается спроецированный объект

ErrorHandler - функция реализуемая программистом для обработки исключительных ситуации.

Функция на языке Си которая реализует данную технологию в нашем ПМО :

```
void *PDS_Read( WCHAR*filename )
{
    void *Mapped = NULL;
    HANDLE hFile;
    HANDLE hMap;
    DWORD FileSize = 0;
    hFile = CreateFile( filename , GENERIC_WRITE |
GENERIC_READ , 0 , 0 , OPEN_EXISTING , FILE_ATTRIBUTE_NORMAL
, 0 );
    if( hFile == INVALID_HANDLE_VALUE )
    {
        MessageBoxA( NULL , "CreateFile failed" , "Error:" ,
MB_ICONERROR );
        return NULL;
    }

    FileSize = GetFileSize( hFile , 0 );

    hMap = CreateFileMapping( hFile , 0 , PAGE_READWRITE , 0 ,
FileSize , 0 );
    if(hMap == INVALID_HANDLE_VALUE)
    {
        MessageBoxA( NULL , "CreateFileMapping failed" ,
"Error:" , MB_ICONERROR );
        CloseHandle(hFile);
        return NULL;
    }
}
```

```

    }
    Mapped = (void*)MapViewOfFile(hMap , FILE_MAP_READ |
FILE_MAP_WRITE , 0 , 0 , FileSize);
    if(Mapped == NULL)
    {
        MessageBoxA( NULL , "MapViewOfFile failed" , "Error:" ,
MB_ICONERROR );
        CloseHandle(hFile);
        CloseHandle(hMap);
        UnmapViewOfFile(Mapped);
    }
    CloseHandle(hFile);
    CloseHandle(hMap);
    return Mapped;
}

```

Данной функции на вход подается путь до файла , который надо спроецировать в адресное пространство нашего процесса.

## Алгоритм построения горизонта

Построение горизонта для базовой точки на поверхности Луны начинается с задания количества секторов (HORIZONT) для разбиения  $360^\circ$  горизонта на равные углы. Обычно это 1080, соответствует  $20'$ , 2160, соответствует  $10'$  или 4320, соответствует  $5'$  угла по горизонту в одном секторе. Далее считываются данные высот из исходного файла, который имеет размеры  $n_{\max} * n_{\max}$  (в файле MoonFloat.bin  $n_{\max} = 15520$ ) и содержит высоты, измеренные для точек на поверхности Луны с равными угловыми расстояниями по широте (dlat) и с равными угловыми расстояниями по долготе (dlon) между точками. Данные в файле MoonFloat.bin хранятся в формате чисел float (4 байта). Изучаемая область лежит севернее  $60^\circ$  северной широты. Равномерное разбиение по широте и долготе приводит к различным линейным расстояниям между точками, которые сгущаются к Северному полюсу.

Задаем координаты точки, для которой будет построена линия горизонта. Базовая точка должна совпадать с одной из точек, хранящихся в файле MoonFloat.bin с измеренными высотами.

Расстояние между двумя точками на поверхности Луны находим по формуле  $R \cdot \arccos((\cos(\varphi_1 - \varphi_2) \cdot \cos \theta_1 \cdot \cos \theta_2 + \sin \theta_1 \cdot \sin \theta_2))$ , где  $\theta_i$  – широта точки  $i$ ,  $\varphi_i$  – долгота точки  $i$ ,  $R$  – средний радиус Луны. Поскольку радиус Луны постоянный и арккосинус монотонно убывающая функция, то в программе для сравнения расстояний используется косинус дуги  $(\cos(\varphi_1 - \varphi_2) \cdot \cos \theta_1 \cdot \cos \theta_2 + \sin \theta_1 \cdot \sin \theta_2)$  между точками на поверхности, это позволяет избегать вычисления арккосинусов.

Для каждой точки поверхности, находящейся внутри окружности радиуса  $R1$  км с центром в базовой точке, находим пересечение отрезка с

концами в данной точке и точке наблюдения и цилиндра радиуса  $1$  с осью, проходящей через центр Луны и базовую точку. Максимумы углов наблюдения полученных точек на поверхности цилиндра образуют линию горизонта.

Алгоритм, полученную величину высоты, сравнивает с хранящейся максимальной высотой для данного сектора горизонта, и выбирает максимальную из них. Для ускорения счета, чтобы избежать вычисления арктангенсов, хранятся именно высоты точек на поверхности цилиндра, а не углы горизонта. Это также позволяет ускорить интерполяцию, если она необходима.

Полученный горизонт может оказаться сильно осциллирующей функцией, так как точки с известными высотами, находящиеся недалеко от базовой и образующие часть горизонта, могут находиться на больших угловых расстояниях и создавать разрывы горизонта. Несмотря на это, полученный горизонт самый точный из всех возможных для данного множества точек с известными высотами и его точность определяется только густотой покрытия поверхности Луны точками измерений. Причем критически важной областью, где необходима максимальная частота измерений высот, является область вблизи базовой точки, радиус которой зависит от количества секторов горизонта (HORIZONT). Попытка получить более гладкую линию горизонта при помощи интерполяции неизбежно приводит к ухудшению его качества.

Чтобы сгладить линию горизонта необходимы предположения о том, какие высоты могут быть в точках поверхности Луны, находящихся между измеренными. Сделав наиболее естественное и простое предположение о линейности поверхности Луны между точками с измеренными высотами в

направлениях Север-Юг и Восток-Запад, получим интерполированную сглаженную линию горизонта.

Для этого найдем проекцию высоты точки, находящейся слева (западнее) от очередной обрабатываемой, ее характеристики были рассчитаны на предыдущем шаге по  $j$  (по долготе). Поскольку через 3 точки: базовую, очередную обрабатываемую и точку к западу от нее можно провести плоскость, используем это. Пересечение плоскости с цилиндром – эллипс. На развертке цилиндра получим синусоиду с периодом  $2\pi$ . Амплитуда и фаза синусоиды определяются по двум вычисленным высотам очередной обработанной и слева от нее точек.

Если проекции обрабатываемой точки и левой находятся в одном и том же секторе или смежных секторах горизонта, то интерполяция не производится и промежуточные высоты не вычисляются, если же есть промежуточные сектора, то для них вычисляются высоты. Эти высоты соответствуют прямой линии соединяющей две точки с измеренными высотами на поверхности Луны.

Аналогично обрабатывается отрезок, соединяющий обрабатываемую точку и смежную к ней точку находящуюся севернее (сверху), ее характеристики были вычислены на предыдущем шаге по  $i$  (по широте), с тем же индексом  $j$ , что и у обрабатываемой точки.

Поскольку каждый отрезок внутри области интерполяции  $R3 < r < R2$  имеет или северный, или западный конец, то тем самым интерполяция проводится для всех отрезков внутри этой области.

Все горизонты, рассчитанные в областях с близко расположенным сильно пересеченным высоким рельефом образующим часть горизонта, будут малодостоверны, до тех пор, пока на расстояниях до 1 км от точки

наблюдения частота точек с измеренными высотами не будет, минимум в 5 раз выше, чем сейчас. Сейчас высоты определены для точек, находящихся друг от друга на расстоянии 60 метров у полюса и на расстоянии 300 метров у  $60^{\circ}$  северной широты. (Расстояние между измеренными точками зависит от их широты.)

## Программная реализация

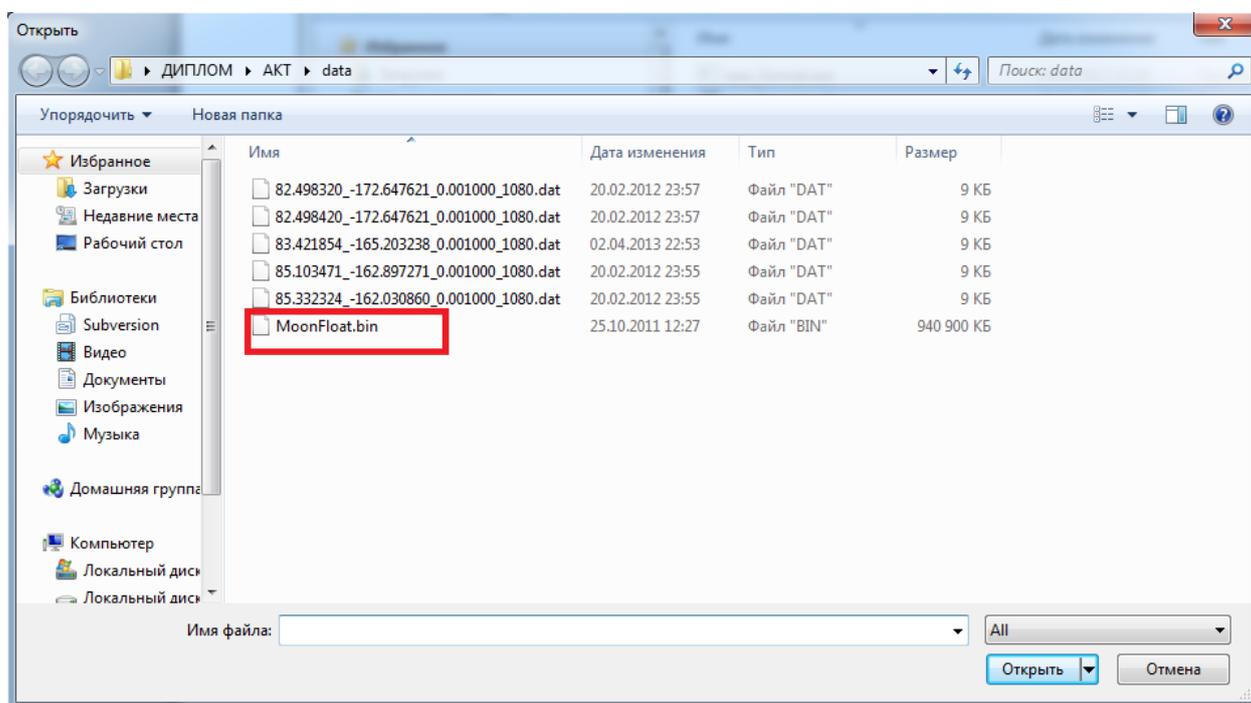
Системные требования для запуска данного ПО:

- Видеокарта — объём памяти от 64 Мб, совместимость с DirectX 9.0b и новейшие драйвера.
- Операционная система — Windows 2000/XP/7.
- Процессор — Pentium 4 с частотой 1,5 ГГц либо Athlon XP 1500+ и выше.
- 1 Гб Мб ОЗУ.
- CD-ROM 8x.
- 2,2 Гб свободного места на жёстком диске.
- Звуковая карта — 16-разрядная, совместимая с DirectX 9.0b

Входные данные :

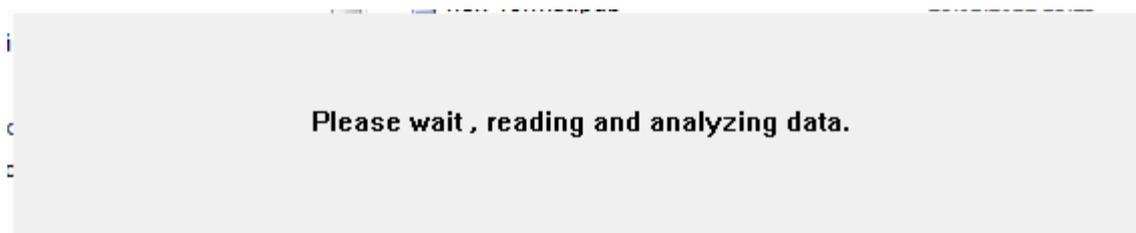
- IMG файл описывающий поверхность
- Широта
- Долгота
- Расстояние наблюдающего элемента от земли
- Число разбиение секторов.

При запуске нашего ПМО пользователю предлагается выбрать файл исходных данных , где находится информация о поверхности:



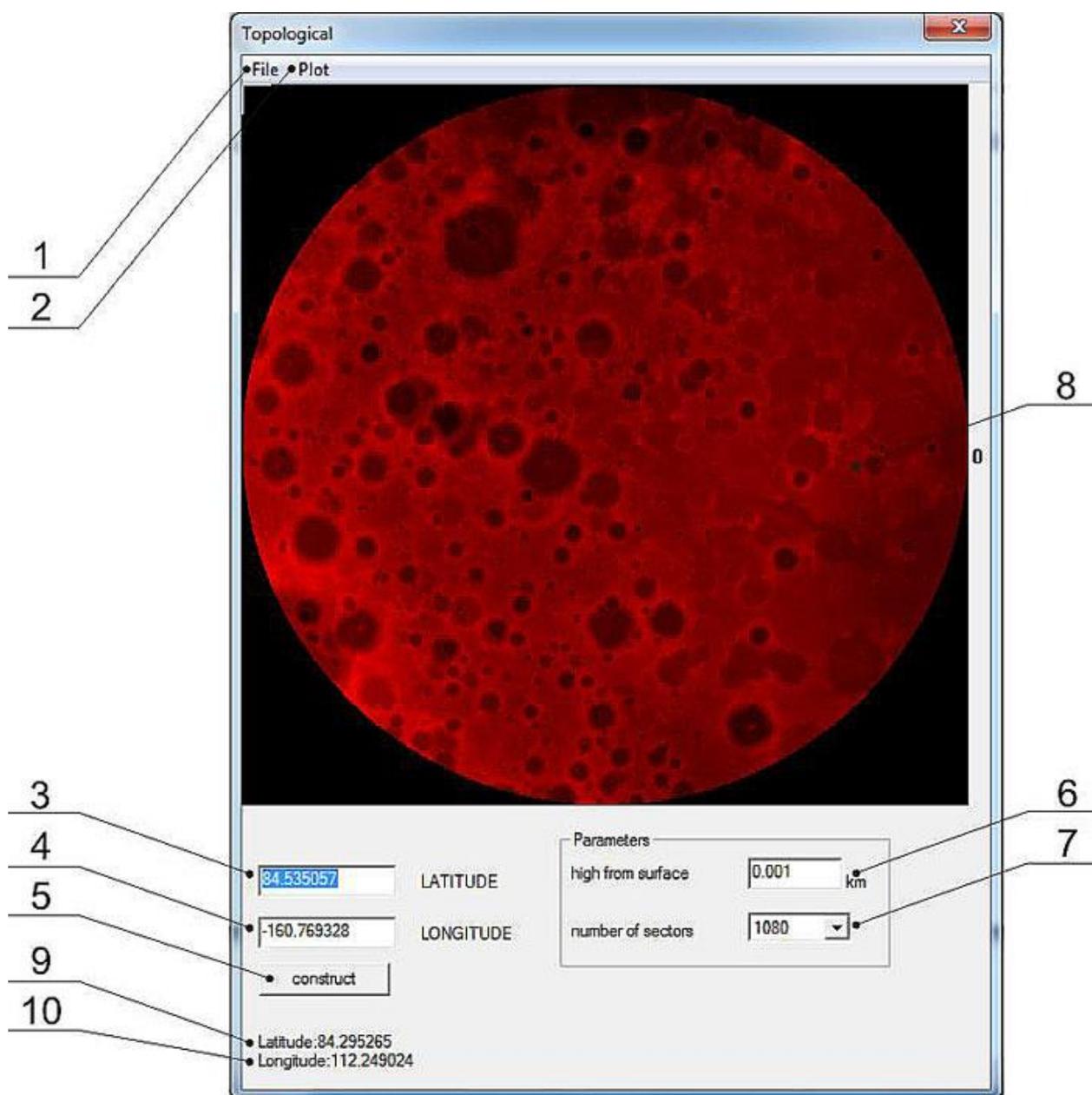
**Рис. 14. Выбор исходных данных**

Во время загрузки и обработки информации пользователю показывается заставка :



**Рис. 15. Заставка программы**

После выполнения всех предварительных расчетов и построения карты высот , пользователю показывается основной интерфейс программы:

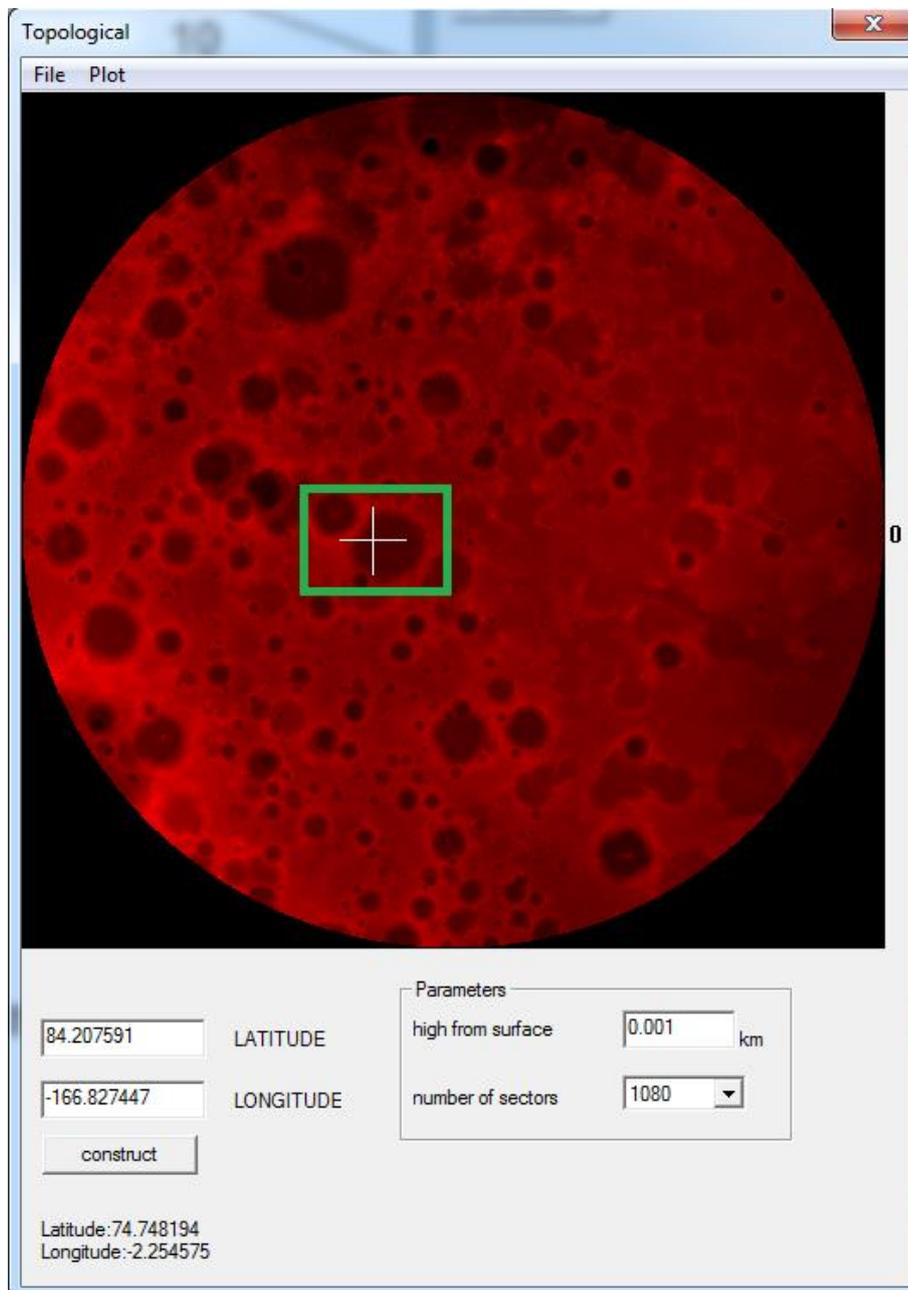


**Рис. 16. Интерфейс главного окна**

- 1) Системное меню
- 2) Меню работы с графиками горизонта
- 3) Ручной ввод широты
- 4) Ручной ввод долготы
- 5) Начать построение линии горизонта

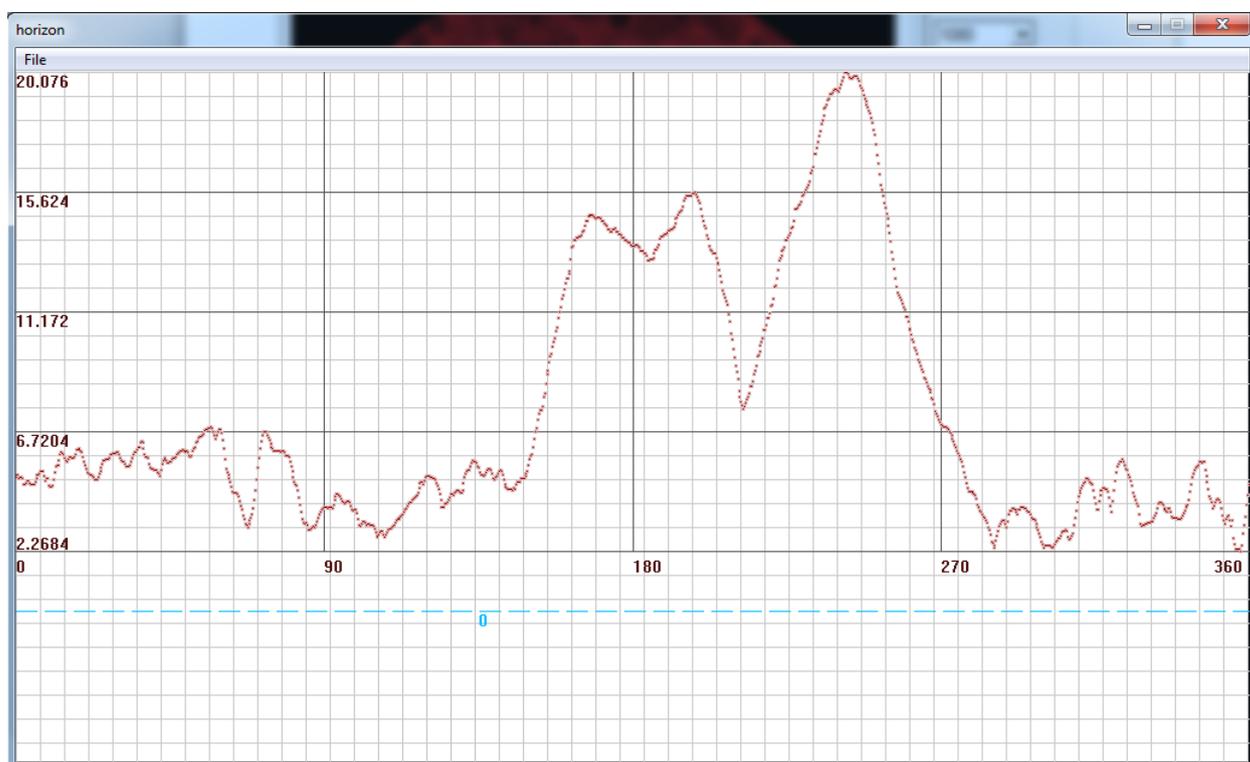
- 6) Ручное задание расположение наблюдающего устройства от поверхности( наблюдающее устройство расположено на луноходе).
- 7) Число разбиение секторов
- 8) Топологическая карта выбранного участка Луны
- 9) Текущее значение широты
- 10) Текущее значение долготы

При выборе нужной точки на участке появится перекрестие :

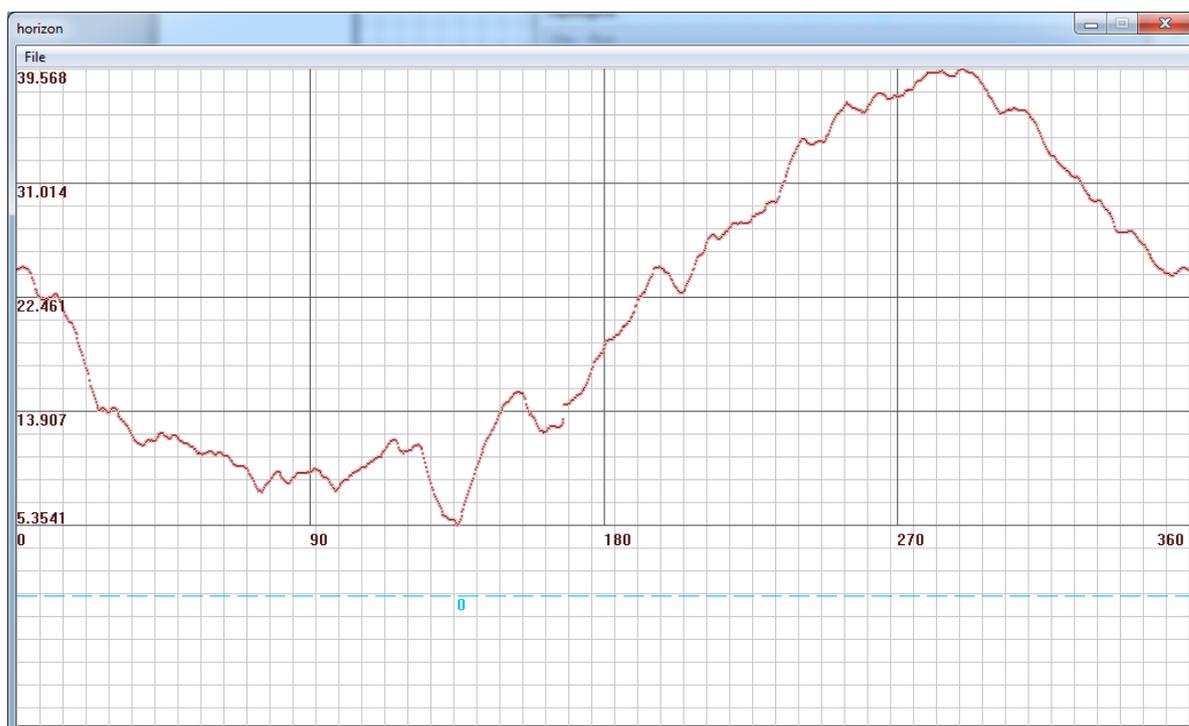


**Рис. 17. Выбор координаты точки**

После нажатия на управляющий элемент “construct” выполнится построение линии горизонта:



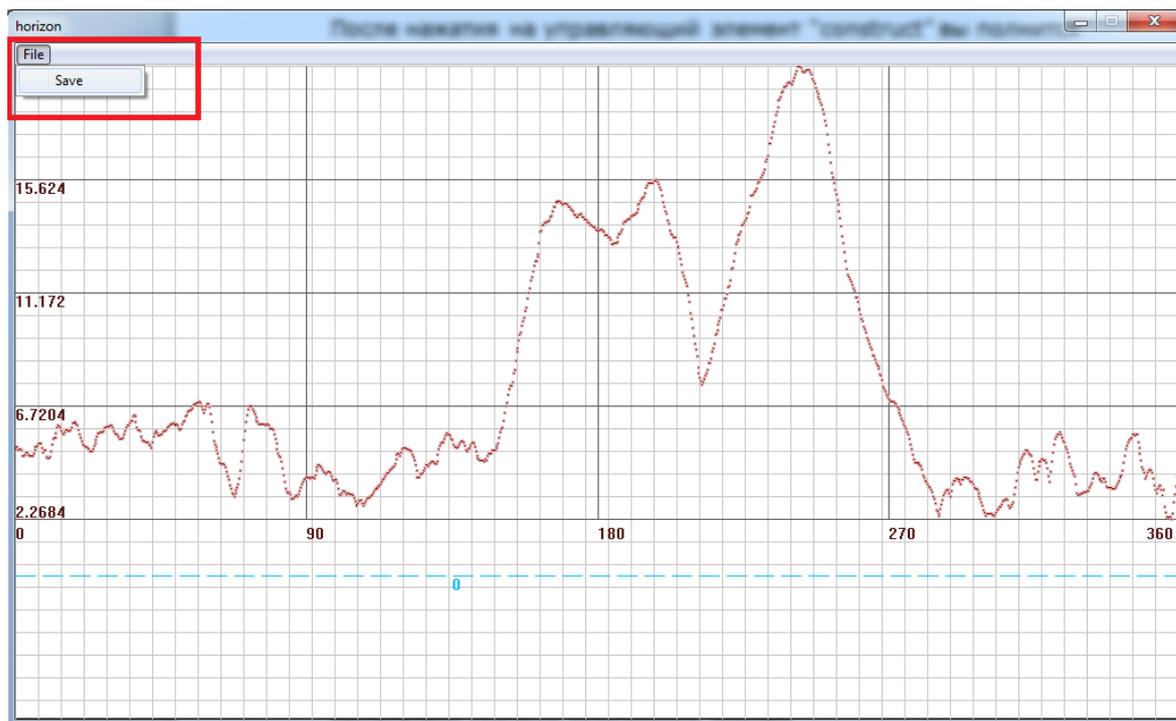
**Рис. 18. Построение линии горизонта**



**Рис. 19. Построение линии горизонта**

На данном графике расположение наблюдающего элемента на луноходе обозначено голубой пунктирной линией.

Для ведения базы наблюдений в ПМО предусмотрена возможность сохранять результаты. Для этого в окне результат надо нажать на меню File-> Save.



**Рис. 20. Сохранение построенной линии горизонта**

Результаты будут сохранены в файле :

latitude\_longitude\_high\_from\_surface.dat:

	82.498320_-172.647621_0.001000_1080.dat	20.02.2012 23:57	Файл "DAT"	9 КБ
	82.498420_-172.647621_0.001000_1080.dat	20.02.2012 23:57	Файл "DAT"	9 КБ
	83.421854_-165.203238_0.001000_1080.dat	02.04.2013 22:53	Файл "DAT"	9 КБ
	85.103471_-162.897271_0.001000_1080.dat	20.02.2012 23:55	Файл "DAT"	9 КБ
	85.332324_-162.030860_0.001000_1080.dat	20.02.2012 23:55	Файл "DAT"	9 КБ
	85.747189_-163.610460_0.001000_1080.dat	07.04.2013 19:11	Файл "DAT"	9 КБ
	MoonFloat.bin	25.10.2011 12:27	Файл "BIN"	940 900 КБ

**Рис. 21. Сохранённый файл линии горизонта**

После получения нужных данных у пользователя есть возможность произвести сравнение полученных участков.

Для этого в главном окне программы нужно выбрать меню Plot->Compare:

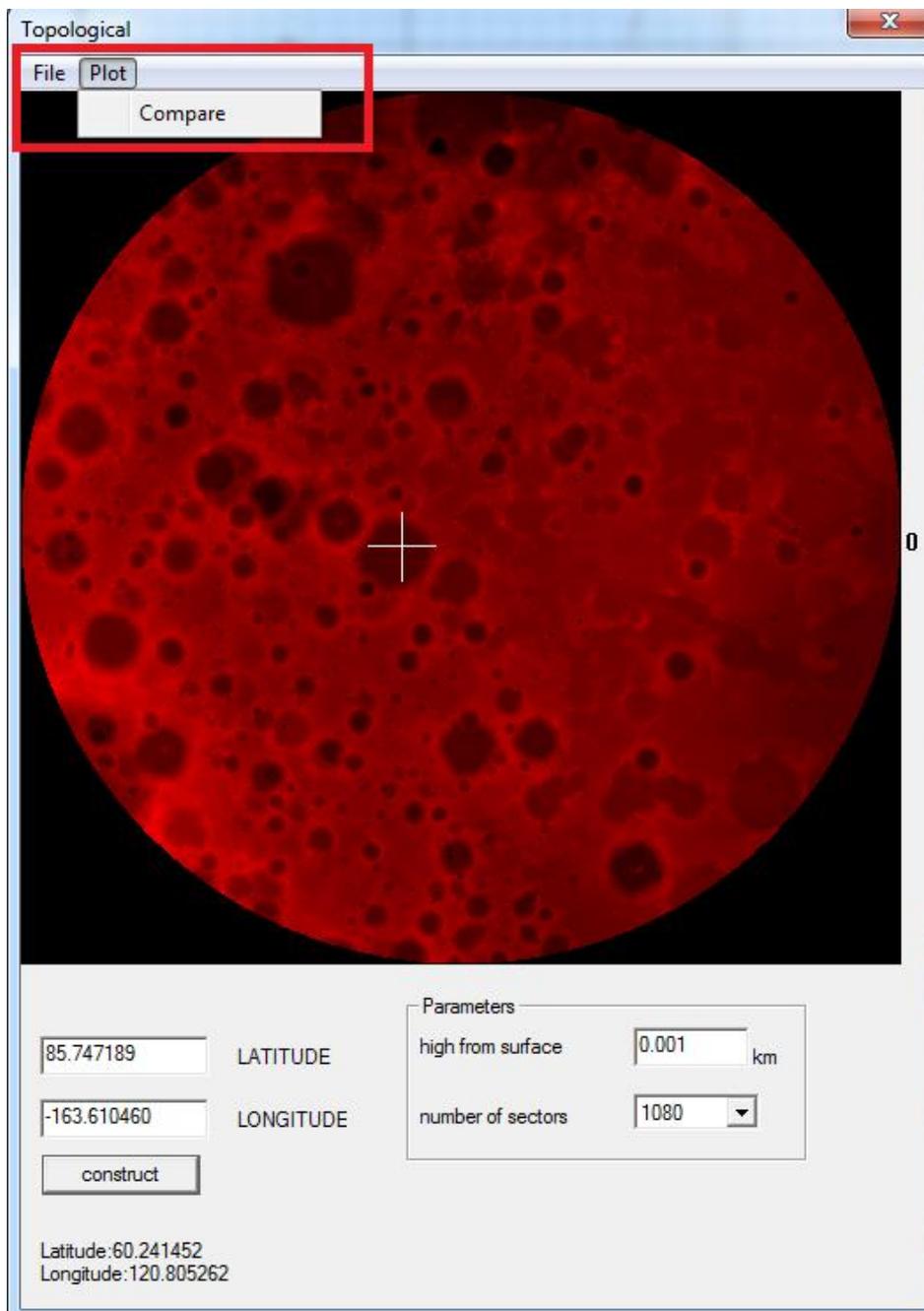
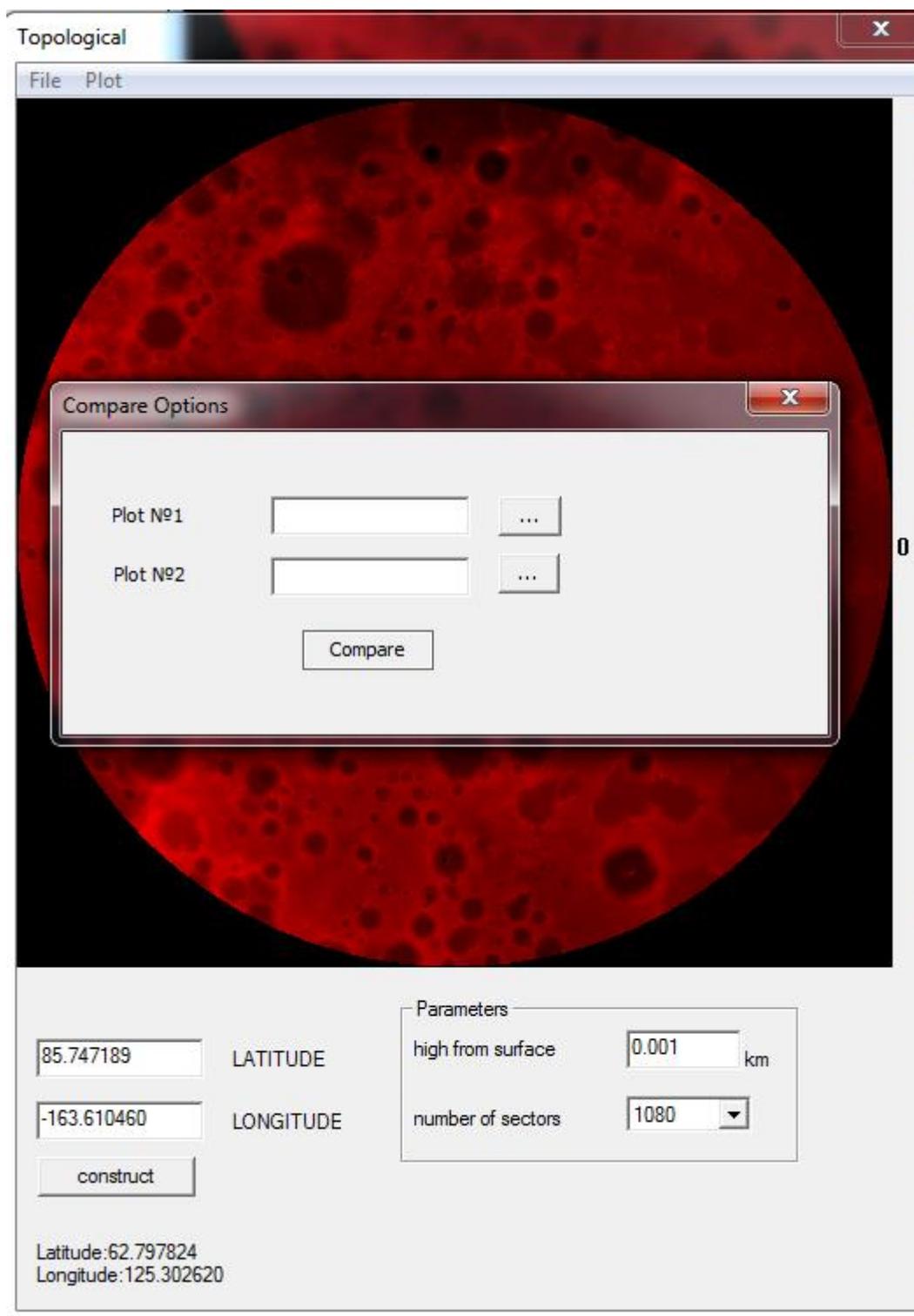


Рис. 22. Меню сравнения линий горизонта

После выбора данного пункта появится окно, в котором нужно задать пути до сохранённых горизонтов:



**Рис. 23. Окно выбора сохраненных линий горизонта**

После выбора файлов и нажатия на управляющий элемент “Compare” , пользователь увидит графическое сравнение горизонтов:

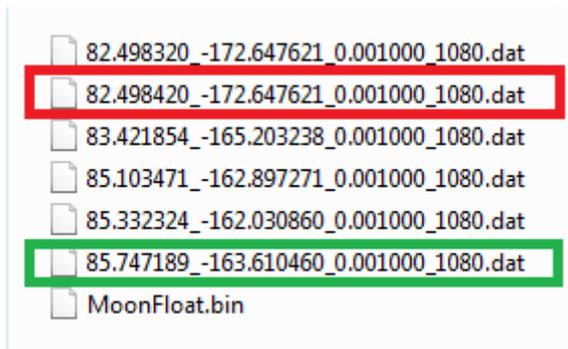


Рис. 24. Сохраненные линии горизонта

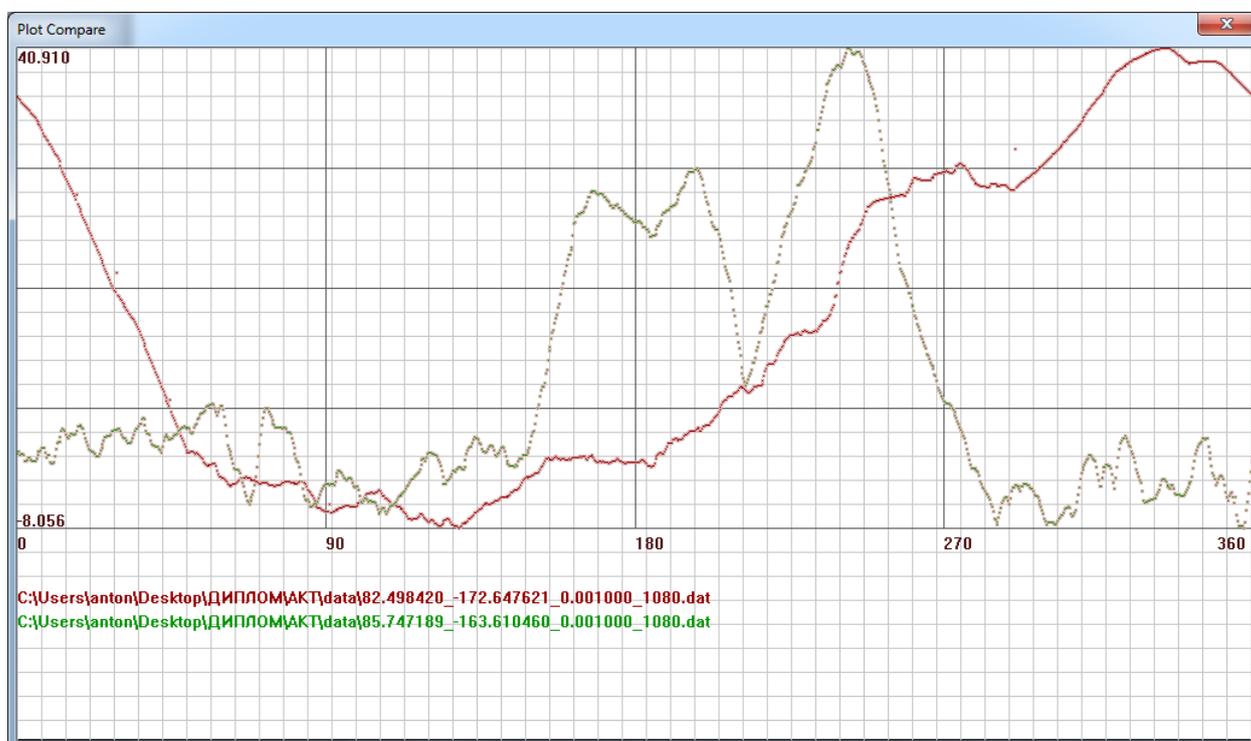
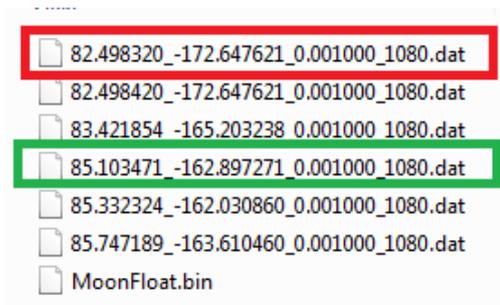
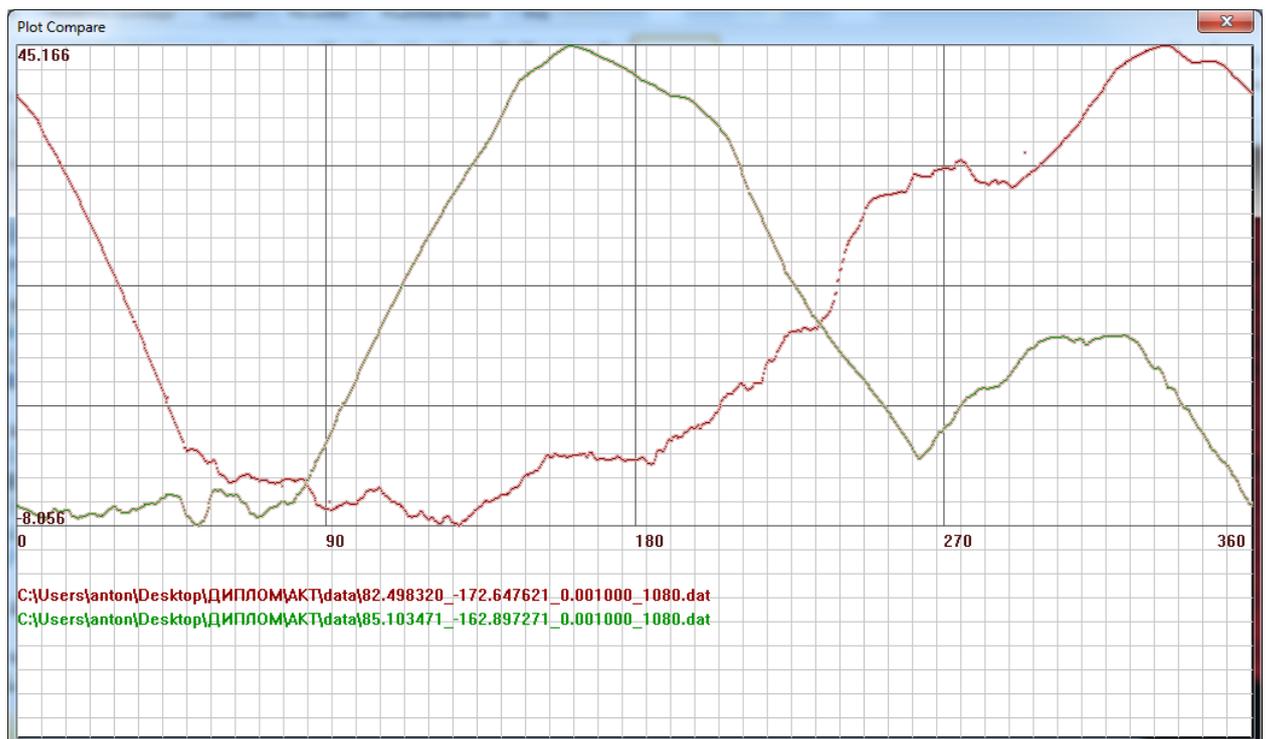


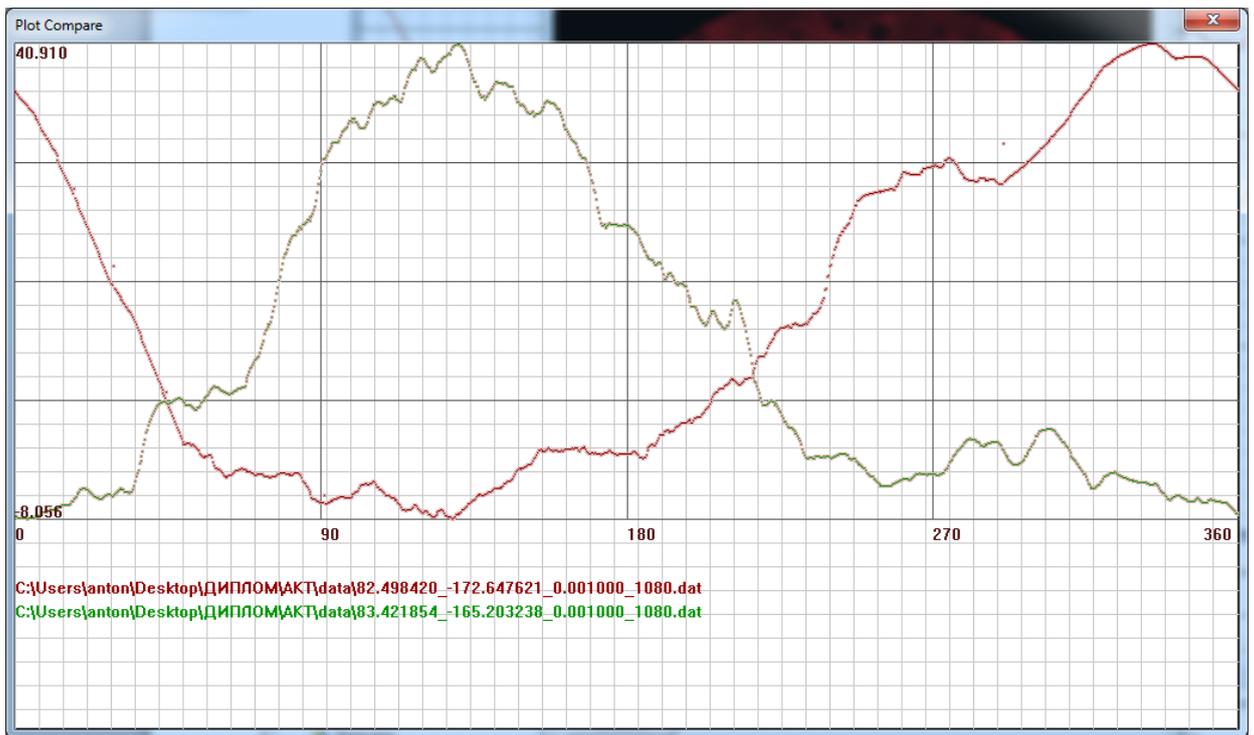
Рис. 25. Сравнение линий горизонта



**Рис. 26. Сохраненные линии горизонта**



**Рис. 27. Сравнение линий горизонта**



**Рис. 28. Сравнение линий горизонта**

## **Заключение**

В результате данной научной работы , были разработаны быстродействующие алгоритмы для построения горизонта и топологической карты. Так же было разработано программное обеспечение реализующее данные алгоритмы. Данный расчетный модуль можно использовать для выбора оптимальных мест посадки лунных модулей.

## Список использованной литературы

1. Брайан Керниган, Деннис Ритчи. Язык программирования Си. (ISBN 978-5-8459-0891-9, 0-13-110362-8; 2012 г.)
2. Джеффри Рихтер. Windows для профессионалов. Создание эффективных WIN-32 приложений с учетом спецификации 64-разрядной версии Windows(Питер, Русская Редакция , 752 стр.).
3. Бьерн Страуструп. Язык программирования C++(Бином,1136 стр.).
4. М. Руссинович, Д.Соломон. Внутреннее устройство Microsoft Windows( Питер , 800 стр. ).
5. wikipedia.org
6. <http://moonmaplab.blogspot.ru/>
7. <http://wms.lroc.asu.edu>
8. <http://pds-geosciences.wustl.edu/>
9. [http://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Import-Export/NASA\\_IMG\\_Importer](http://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Import-Export/NASA_IMG_Importer)
10. Dave Shreiner. OpenGL Programming Guide: 4th Edition(Addison-Wesley Professional , 816 pages ).
11. Thomas H.Corman, Charles E. Leiserson, Ronald L. Rivest , Clifford Stein. Introduction to Algorithms(The MIT Press; third edition edition , 1312 pages ).
12. Harold Abelson , Gerald Jay Sussman , Julie Sussman. Structure and Interpretation of Computer Programs(The MIT Press; second edition edition , 683 pages ).
13. <http://msdn.microsoft.com>
14. Herbert Schildt. C++ from the Ground Up(McGraw-Hill Osborne Media; 3 edition , 624 pages).
15. Yan Backgrounder: Introduction to LRO's instruments.

16. Николас Вирт. Алгоритмы и структуры данных(ДМК Пресс , 272 стр.).
17. Стивен Скиена. Алгоритмы Руководство по разработке(БХВ-Петербург , 720 стр. ).
18. Е.А. Никулин. Компьютерная геометрия и алгоритмы машинной графики(Книга по Требованию , 554 стр.).
19. Herbert Schildt. C++: The Complete reference(McGraw-Hill Osborne Media; 4 edition , 1056 pages).
20. Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra.Head First Design Patterns(O'Reilly Media; 1 edition , 678 pages).

## Награды

Данная работа получила диплом первой степени на следующих конференциях:

- АКТ-2011
- Научно-техническая конференция студентов, аспирантов и молодых специалистов МИЭМ

