INDUSTRIAL TRAINING REPORT

ON

DETECTION OF IMPACT CRATERS ON THE SURFACE OF VENUS IN SAR IMAGES

SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By:

Alok Tripathy

ROLL NO.:43514802716

Maharaja Agrasen Institute of technology, PSP area,

Sector – 22, Rohini, New Delhi – 110085

CERTIFICATE

This is to certify that the project report entitled " **DETECTION OF IMPACT CRATERS ON THE SURFACE OF VENUS IN SAR IMAGES**" submitted by **ALOK TRIPATHY** in fulfilment of the requirement of Industrial Training at **SPACE APPLICATIONS CENTRE, INDIAN SPACE RESEARCH ORGANIZATION (SAC/ISRO)** as part of the degree of Bachelor of Technology in **COMPUTER SCIENCE ENGINEERING** of **MAIT**, **ROHINI** session 2016-2020 is a record of bonafide work carried out under my supervision and has not been submitted anywhere else for any other purpose.

DECLARATION

I hereby declare that the training report entitled "Detection of Impact Craters on the Surface of Venus Using SAR Images" is an authentic record of my own work as requirement of 8 weeks training during the period from 10th June 2019 to 31st July 2019 for the award of degree of B.Tech. (Computer Science Engineering), GGSIPU, under the guidance of Shri Tathagata Chakraborty.

(Signature of Student)

Name : Alok Tripathy

Roll No. - 43514802716

Date :_____

Certified that above statement made by the student is correct to the best of our knowledge and belief.

Signatures

Examined by:

1) **2)**

(Guide/Trainer) **(Faculty Coordinator)**

ACKNOWLEDGEMENT

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behaviour, and acts during the course of study.

I express my sincere gratitude to Dr. Namita Gupta, Head of Department, Computer Science and Engineering for providing me an opportunity to undergo industrial training at Space Application Center, Indian Space Research Organisation (SAC/ISRO). I also thank Space Applications Center, Indian Space Research Organisation (SAC/ISRO) for giving me the opportunity to undergo industrial training at their premises.

I express my sincere gratitude to Sri Deepak Putrevu, Head (EPSA), for giving me the opportunity to work on this project. I am also thankful to Sri Tathagata Chakraborty, SCI/ENGG., EPSA, for his support, cooperation and motivation provided to me during the training for constant inspiration, presence and blessings.

Lastly, I would like to thank the almighty and my parents for their moral support and my friends with whom I shared my day-to-day experience and received lots of suggestions that improved my knowledge.

ABOUT THE ORGANIZATION

अंतरिक्ष उपयोग केंद्र

SPACE APPLICATIONS CENTRE

2.1 Overview

Space Applications Centre (SAC), is a major research and development centre of the Indian Space Research Organisation (ISRO). It plays a key role in realising vision and mission of ISRO. Located at Ahmedabad, SAC is spread across two campuses having multi-disciplinary activities.

The core competence of the centre lies in development of space borne and air borne instruments/payloads and their applications for national development and societal benefits. These applications are in diverse areas and primarily meet the communication, navigation and remote sensing needs of the country. Besides these, the centre also contributes significantly in scientific and planetary missions of ISRO like Chandrayan-1, Chandrayan-2, Mars Orbiter Mission etc.

The communication transponders developed at this centre for Indian National Satellite (INSAT) and Geo Synchronous Satellite (GSAT) series of satellites are used by government and private sector for VSAT, DTH, internet, broadcasting, telephony etc. These satellites are instrumental in reaching remote parts of the country. The payloads for major navigation systems of the country - Indian Regional Navigation Satellite System (IRNSS) and GPS Aided Geo Augmented Navigation (GAGAN) are being developed by this centre.

This centre designs and develops the optical and microware sensors for the satellites, signal and image processing software, GIS software and many applications for Earth Observation (EO) programme of ISRO. These applications are in diverse areas of Geosciences, Agriculture, Environment and Climate Change, Physical Oceanography, Biological Oceanography, Atmosphere, Cryosphere, Hydrosphere etc.

The facilities at SAC includes highly sophisticated payload integration laboratories, electronic and mechanical fabrication facilities, environmental test facilities, systems reliability/assurance group, image processing and analysis facilities, project management support group and a well-stocked library. SAC has active collaborations with industry, academia, national and international institutes for research and development. The centre also has state-of-art in-house and mobile exhibitions to propagate space technology and applications amongst students and public.

The Centre also conducts nine-month post graduate diploma courses for students from the Asia Pacific region under the aegis of the Centre for Space Science and Technology Education (CSSTEAP) in satellite meteorology and communication.

2.2 Genesis and History

The genesis of the centre dates back to 1966, with establishment of the Experimental Satellite Communication Earth Station (ESCES), by late Dr. Vikram Sarabhai in Ahmedabad. It was an experimental Earth Station and training centre where scientists and engineers of India and other developing countries could receive training and firsthand experience in the design, development and operations of an earth station for communications and broadcasting.

Later in 1972, the different units of ISRO in Ahmedabad pursuing research in applications of space technology were merged to form SAC. During 1975-76 a unique experiment called the Satellite Instructional Television Experiment (SITE) was conducted by SAC/ISRO utilizing the American ATS-6 satellite. This involved telecasting educational programmes aimed at socio-economic development of rural India, which covered 2400 villages - spread over six states - through experimental Direct Reception Sets. SITE was followed by communication techniques developmental project called Satellite Telecommunications Experiments Projects (STEP), carried out with the Frenco-German satellite, Symphony.

The payload for first experimental communication satellite of India, 'APPLE' was designed, fabricated and qualified at SAC. It was launched onboard the first experimental flight of the Arian. An exhaustive communication applications programme called the APPLE Utilization Programme (AUP) was also conceived and carried out.

The INSAT-1 series of satellite was custom designed and made as per the unique requirements of the country by a US company. The INSAT 2A, 2B, 2C, 2D and 2E, launched in the years 1992, 1993, 1995, 1997 and 1999 respectively, were designed, fabricated and qualified / indigenously at SAC.

The present remote sensing programme of ISRO started in early 1970s. Payload development at SAC was started with balloon experiments followed by aerial photography for remote sensing. At the same time activities were also carried in the field of meteorology with available data from foreign satellites and from indigenously developed airborne thermal scanner.

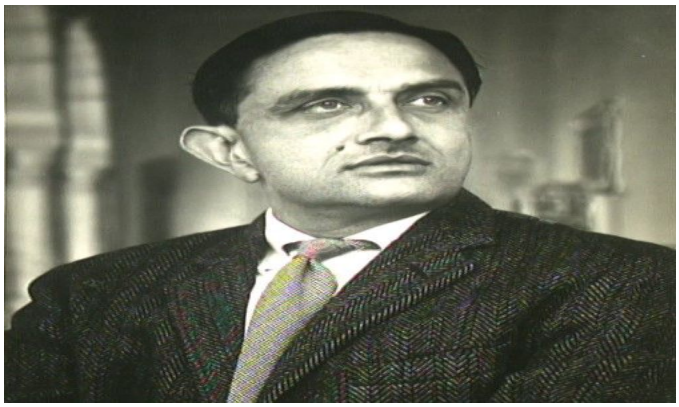
The first phase saw the development of airborne thermal sensors, multispectral scanner, linear Charge Coupled Device (CCD) camera, Side Looking Radar, Colour Infrared (CIR) based photographic systems and a number of photo interpretation and ground truth equipment. Based on above initial work, a strong applications programme was evolved around these instruments. Foundations for space borne sensors were laid during this period. Under the programme 'Satellite for Earth Observation (SEO)', two satellites were launched and called Bhaskara satellites after their Launch onboard Russian launch Vehicle. Bhaskara carried a 1 km resolution 2 बैंड TV camera systems and a three channel microwave radiometer. These were designed, developed and successfully qualified in house. Bhaskara I and II were the first Indian Meteorological satellites which carried microwave radiometer called SAMIR to provide information on sea state and atmospheric water vapour content for use in meteorological studies.

The second phase in 1980s witnessed the results of earlier efforts of experimental satellites. The IRS 1A programme was successfully launched and the users started receiving multispectral imagery with 36m resolution. Major applications in agriculture, hydrology, geology and other areas were defined in close interaction with user agencies and the IRS utilization programme was carried out successfully. These efforts led to semi-operational applications of IRS 1A data.

Strong foundation was also laid for airborne SAR system development, its data processing and applications at SAC. The advanced activities carried out at at the centre during the third phase in 1990s put India at par with many other advanced nations through the design of high resolution sensors in the optical and microwave regions including a successfully flown airborne SAR system and a very sophisticated application programme tuned to our country's needs. The 5.8 m resolution Pan Camera of IRC 1C & 1D revolutionized the applications concept in the country. Being the best resolution civilian sensor in the world at that time, it attracted the attention of foreign users.

SAC has state of the art General Circulation Models for experimentation with satellite data. Prediction of weather in the extended range and predon of Ocean state in the short range are the fields of active research.

2.3 Founder



Dr. Vikram Ambalal Sarabhai

Dr. Vikram Sarabhai was born on 12 August 1919 in Ahmedabad, Gujarat. Dr. Vikram A Sarabhai obtained his Tripos (1939) and Ph. D. (1947) in Cosmic Ray Physics from Cambridge University, U.K. He worked with Sir C.V.Raman, the Nobel Laureate, at the Indian Institute of Science, Bangalore. On his return from Cambridge, Dr. Sarabhai set up the Physical Research Laboratory (PRL), Ahmedabad. He also established the Ahmedabad Textile Industry’s Research Association (ATIRA) and the Indian Institute of Management (IIM), Ahmedabad.

With a belief that space technology could play a meaningful role in national development and solving the problems of common man, Dr. Vikram A. Sarabhai shaped the Indian Space Programme in 1960. He saw tremendous potential of using modern science to help in the process of development. He drew up plans to take education to the remotest villages by using satellite television. The Indian Space programme concentrated on achieving self reliance and developing capability to build and launch communication satellites for television broadcast, telecommunications and meteorological applications; remote sensing satellites for management of natural resources. Thus Indian Space Programme initiated space activities in the country and gradually ISRO came into existence on August 15, 1969.

Dr. Vikram Sarabhai further established the Community Science Centre under the aegis of the Nehru Foundation for Development, which was renamed as Vikram A. Sarabhai Community Science Centre. In 1966, Dr. Sarabhai was appointed as Chairman, Atomic Energy Commission, and Government of India. He was awarded in 1962 the Shanti Swaroop Bhatnagar Memorial Award for Physics. He was also honored with the award of Padma Bhusan in 1966 by the government of India. The Padma Vibhusan was awarded to him posthumously.

2.4 Vision

Harness space technology for national development, while pursuing space science research and planetary exploration.

2.5 Mission

- Design and development of launch vehicles and related technologies for providing access to space.
- Design and development of satellites and related technologies for earth observation, communication, navigation, meteorology and space science.
- Indian National Satellite (INSAT) programme for meeting telecommunication, television broadcasting and developmental applications.
- Indian Remote Sensing Satellite (IRS) programme for management of natural resources and monitoring of environment using space based imagery.
- Space based application for societal development.

2.6 Objectives

- Operational flights of Polar Satellite Launch Vehicle (PSLV).

- Developmental flight of Geo-synchronous Satellite Launch Vehicle (GSLV- Mk II).
- Development of heavy lift Geo-synchronous Satellite Launch Vehicle (GSLV-Mk III).
- Design, Development and Realization of Communication Satellites.
- Design, Development and Realization of Earth Observation Satellites.
- Development of Navigation Satellite Systems.
- Development of satellites for Space Science and Planetary Exploration.
- Earth Observation Applications.
- Space based systems for Societal Applications.
- Advanced Technologies and newer initiatives.
- Training, Capacity building and Education.
- Promotion of Space technology.
- Infrastructure / Facility Development for space research.
- International Cooperation.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER -1

INTRODUCTION

[1]Craters are topographic features on planetary surfaces that result from impacts with meteoroids. They are the most abundant landforms on the surface of planets. Their morphology and distribution enable studies of a number of outstanding issues, such as the measurement of the relative age of planetary surface, nature of degradational processes, and regional variations in geologic material. In addition, craters are ideal landmarks for autonomous spacecraft navigation. Optical landmark navigation utilizes crater information in remotely sensed image for determining spacecraft orbits around a celestial body for close flybys and low-altitude orbiting. Crater detection and identification have become key technologies in deep space exploration. All these scientific applications require crater detection and identification methods. However, craters are complex objects. Crater dimensions in an image might differ by orders of magnitude. Crater shapes may also vary depending on their interior morphologies (e.g., central peaks, peak rings, central pits, and wall terraces), level of degradation, and degree of overlap with other craters. The illumination conditions also vary their appearance in images. Thus, it is difficult to find discriminant features able to correctly detect and identify the crater object in a scene. Therefore, developing a highly efficient, robust crater detection and identification algorithm is significant in planetary research and deep space exploration.

Here I have used an unsupervised learning method to detect the impact craters on Venus. The images used are Synthetic Aperture Radar (SAR) images acquired by NASA Magellan Spacecraft. Here first we have attempted to detect the edges of the craters using Canny Edge Detection algorithm followed by circle detection equivalent to the crater rim using Circular Hough Transform and OpenCV library.

CHAPTER -2

TOOLS AND TECHNOLOGY USED

2.1 Environment Choice:

[2]In computer program and software product development, the development environment is the set of processes and programming tools used to create the program or software product. The term may sometimes also imply the physical environment. An integrated development environment is one in which the processes and tools are coordinated to provide developers an orderly interface to and convenient view of the development process (or at least the processes of writing code, testing it, and packaging it for use).

The environment used to create simulation and execute algorithms is **Jupyter Notebook**. Jupyter Notebook (Formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebooks documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the "**.ipynb**" extension.

Fig. 2.1 - Jupyter Notebook Logo

Advantages of Jupyter Notebook:

- To simplify visualization of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the file and display it to the user.
- Jupyter Notebook provides a browser-based REPL built upon a number of popular open-source libraries:
  - IPython
  - ØMQ
  - Tornado (web server)
  - jQuery
  - Bootstrap (front-end framework)
  - MathJax
- Jupyter Notebook has the ability to re-run individual snippets.
- In a notebook we can edit code snippets before re-running them.

2.2 Programming Language Choice:

Fig - 2.2 Python Logo

[3]**Python** is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features adynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution(late binding), which binds method and variable names during program execution.

The language's core philosophy is summarized in the document *The Zen of Python* , which includes aphorisms such as:

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications.

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is also sometimes termed the off-side rule.

2.3 NumPy :

Fig - 2.3 Python Logo

[4]**NumPy** is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. In comparison, MATLAB boasts a large number of additional toolboxes, notably Simulink, whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language. Moreover, complementary Python packages are available; SciPy is a library that adds more MATLAB-like functionality and Matplotlib is a plotting package that provides MATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient linear algebra computations.

2.4 MatPlotLib:

Fig - 2.4 Matplotib logo

[4]**Matplotlib** is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in an easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

2.5 GDAL :

Fig - 2.5 GDAL logo

[6]The Geospatial Data Abstraction Library (GDAL) is a translator library for raster and vector geospatial data formats that is released under an X/MIT style Open Source License by the Open Source Geospatial Foundation. As a library, it presents a single raster abstract data model and single vector abstract data model to the calling application for all supported formats. It also comes with a variety of useful command line utilities for data translation and processing.

As of version 2.2.3, GDAL/OGR provides at least partial support for 154 raster and 93 vector geospatial data formats. A subset of data formats is supported to ensure the ability to directly create files and georeferencing them with the default GDAL compiling options.

Core Features:

- Reading and writing of Raster and Vector Geospatial formats.
- Data format translation.
- Geospatial processing: subnetting, image warping, reprojection, mosaicing, tiling, DEM processing, SAR processing.

- **2.6 OpenCV :**

Fig - 2.6 OpenCV Logo

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

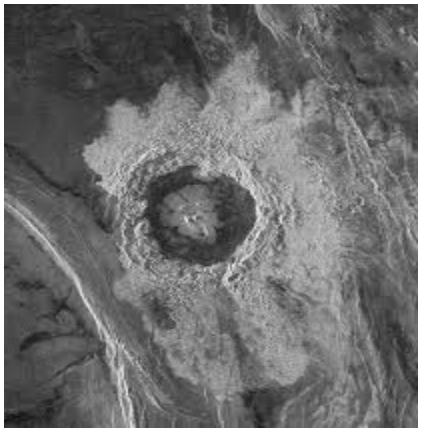
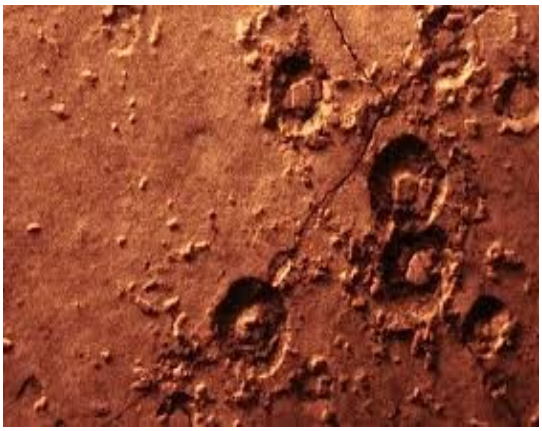
It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are

being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms.

CHAPTER -3

TECHNICAL CONTENT

3.1 Impact Craters on Venus :

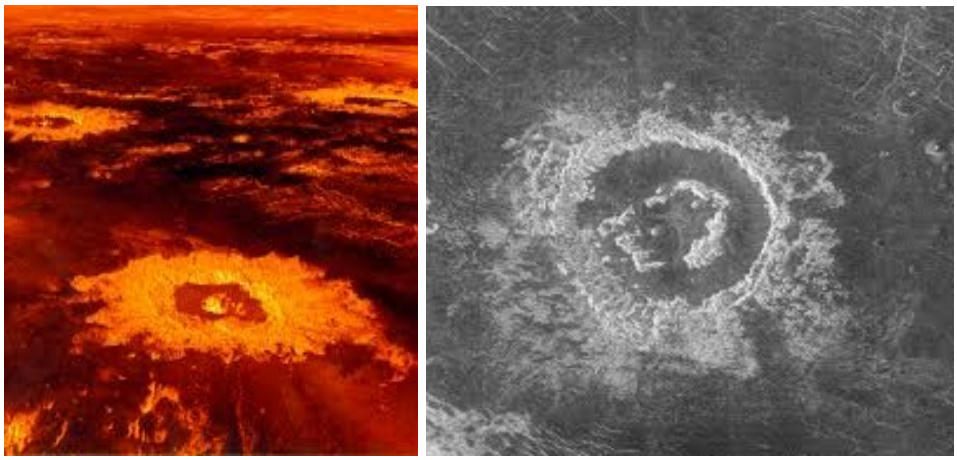


Fig. 3.1 - Impact craters on Venus

Craters on Venus are different from craters on other planets. Venus is having pausity of impact craters compared to Moon and Mars. The thick atmosphere of Venus stops any smaller objects reaching its surface by burn up in the atmosphere. There are about 1000 craters identified on the surface of Venus. Crater Mead is the largest known crater on Venus, named after the American anthropologist, Margaret Mead. It measures 280 km in diameter, and contains several concentric rings. In general the smallest impact craters are 2 - 5 km in radius.

Fig. 3.2 – Impact Crater Mead

3.1.1 Clusters of Impact Crater:

Because Venus's atmosphere is so dense, small meteorites break up as they fall, creating clusters

of overlapping craters.

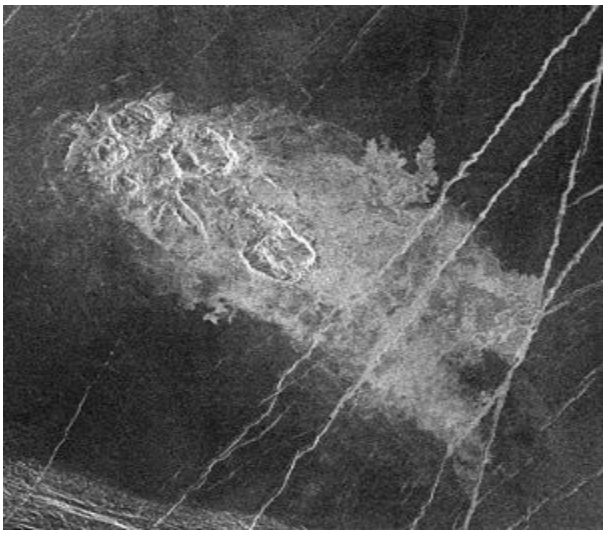


Fig. 3.3 - Width of image area = 95 km

3.1.2 Flows around crater:

Due to high surface temperature, meteorite impacts cause large amounts of rock to melt and flow like lava. If this molten material spills from the crater, it can form flows like those shown here around the crater Markham.

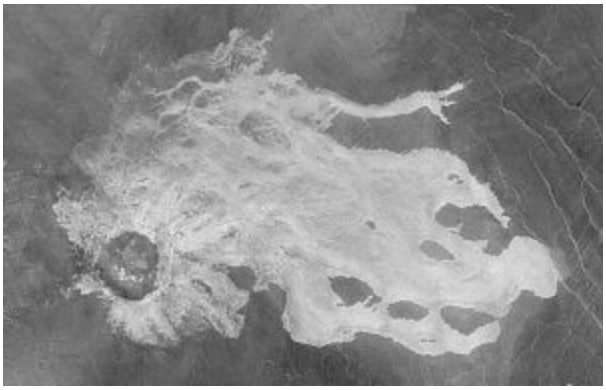


Fig 3.4 - Crater Markham

3.1.3 Impact Deposits:

The thick atmosphere keeps material thrown aloft by meteorite impacts from traveling far. Because of this, craters on Venus lack the long "rays" of material and chains of secondary craters found around many lunar craters. But young craters, such as the one shown in Fig. 3.5, are surrounded by parabola-shaped deposits of fine soil that was ejected and carried to the west (left) by winds high in the atmosphere. Over time the gentle surface winds erase these features.

Fig. 3.5

3.2 Data used for Crater detection on Venus :

The data used for our analysis are the Synthetic Aperture Radar (SAR) images provided by the NASA Magellan SpaceCraft. NASA's Magellan mission to Venus was one of the most successful deep space missions. It was the first spacecraft to image the entire surface of Venus and made several discoveries about the planet.

Fig. 3.6 NASA's Magellan SpaceCraft

3.2.1 Synthetic Aperture Radar Images

Synthetic Aperture Radar Images (SAR Images) are obtained from the Synthetic Aperture Radar (SAR). **Synthetic-aperture radar (SAR)** is a form of radar that is used to create two-dimensional images or 3 dimensional objects of objects, such as landscapes. SAR uses the motion of the radar antenna over a target region to provide finer spatial resolution than conventional beam-scanning radars.

3.2.2 The Data to be analysed:

The images of impact craters which were used for the analysis as under:

Fig. 3.7 Impact Crater 1

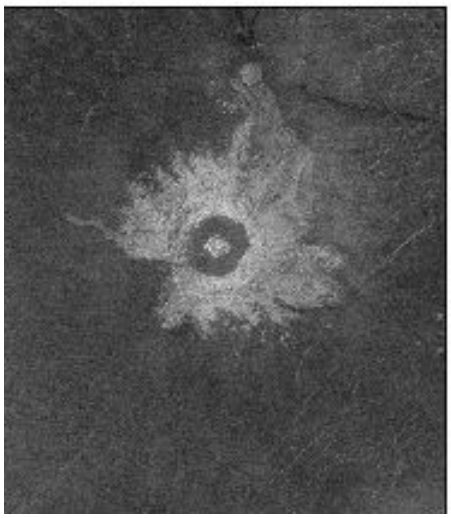


Fig 3.8 Impact Crater 2

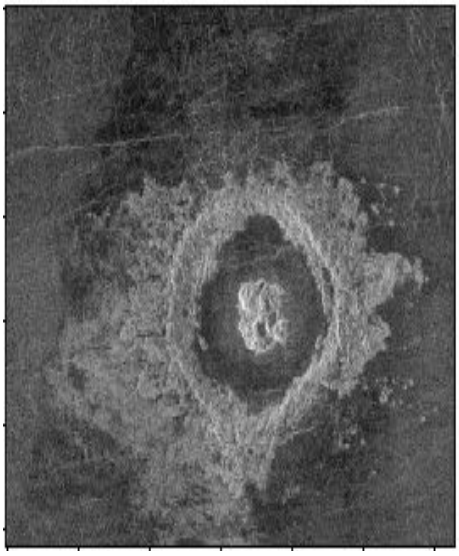


Fig . 3.9 Impact Crater 3

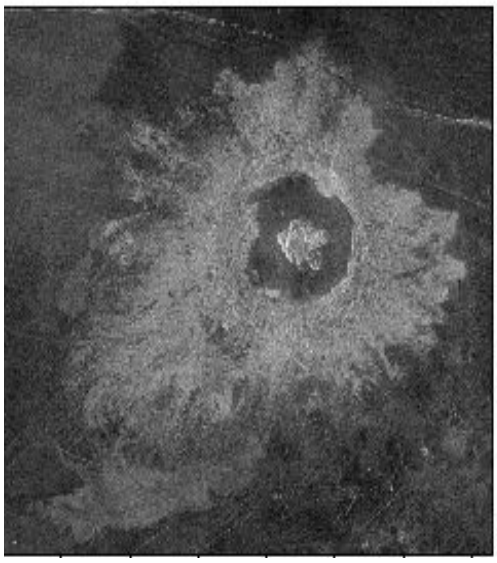


Fig. 3.10 Impact Crater 4

3.3 Algorithm used to detect edges of the crater:

Canny Edge Detection:

The **Canny edge detector** is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations. The general criteria for edge detection include:

1. Detection of edge with low error rate, which means that the detection should accurately catch as many edges shown in the image as possible
2. The edge point detected from the operator should accurately localize on the center of the edge.
3. A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

To satisfy these requirements Canny used the calculus of variations – a technique which finds the function which optimizes a given functional. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

Among the edge detection methods developed so far, Canny edge detection algorithm is one of the most strictly defined methods that provides good and reliable detection. Owing to its optimality to meet with the three criteria for edge detection and the simplicity of process for implementation, it became one of the most popular algorithms for edge detection.

3.3.1 Process of Canny Edge Detection algorithm

The Process of Canny edge detection algorithm can be broken down to 5 different steps:

1. Apply Gaussian Filter to smooth the image in order to remove the noise
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

The detailed flow chart of Canny Edge Detection Algorithm is shown below in Fig. 3.6.

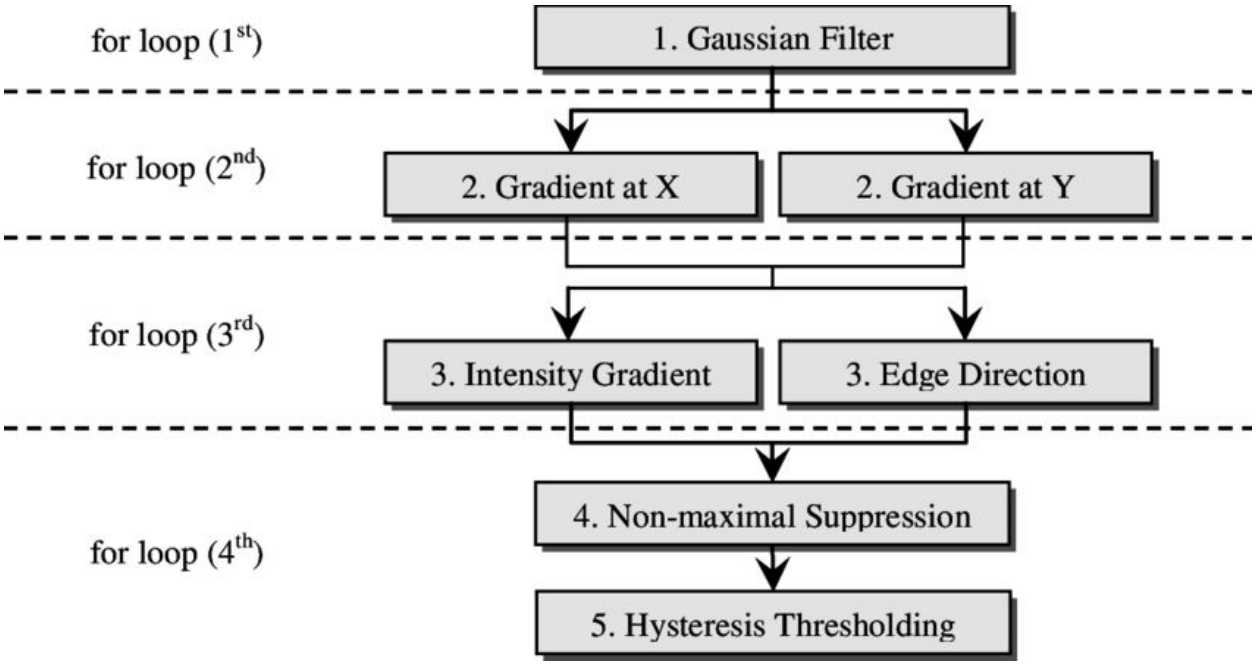


Fig. 3.11 - Flow chart for Canny Edge Detection Algorithm

1) Gaussian Blur:

Since all edge detection results are easily affected by image noise, it is essential to filter out the noise to prevent false detection caused by noise. To smooth the image, a Gaussian filter is applied to convolve with the image. This step will slightly smooth the image to reduce the effects of obvious noise on the edge detector. The equation for a Gaussian filter kernel of size $(2k+1)\times(2k+1)$ is given by:

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2} \right); 1 \leq i, j \leq (2k+1)$$

Here is an example of a 5×5 Gaussian filter, used to create the adjacent image, with sigma = 1.4

$$\mathbf{B} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbf{A}.$$

It is important to understand that the selection of the size of the Gaussian kernel will affect the performance of the detector. The larger the size is, the lower the detector's sensitivity to noise. Additionally, the localization error to detect the edge will slightly increase with the increase of the Gaussian filter kernel size. A 5×5 is a good size for most cases, but this will also vary depending on specific situations. In this present study 5x5 kernel size is used.

2) Finding the intensity gradient of the image

An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (such as Roberts, Prewitt, or Sobel) returns a value for the first derivative in the horizontal direction (G_x) and the vertical direction (G_y). From this the edge gradient and direction can be determined:

$$\mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$$

$$\mathbf{\Theta} = \operatorname{atan2}(\mathbf{G}_y, \mathbf{G}_x)$$

where G can be computed using the hypot function and atan2 is the arctangent function with two arguments. The edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals (0°, 45°, 90° and 135°). An edge direction falling in each color region will be set to a specific angle values, for instance θ in [0°, 22.5°] or [157.5°, 180°] maps to 0°. In this present study Sobel filter is used for gradient computation.

3) Non Maximum Suppression:

Non-maximum suppression is an edge thinning technique. It is applied to find "the largest" edge. After applying gradient calculation, the edge extracted from the gradient value is still quite blurred. With respect to criterion 3, there should only be one accurate response to the edge. Thus non-maximum suppression can help to suppress all the gradient values (by setting them to 0) except the local maxima, which indicate locations with the sharpest change of intensity value. The algorithm for each pixel in the gradient image is:

1. Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions.
2. If the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction (e.g., a pixel that is pointing in the y-direction will be compared to the pixel above and below it in the vertical axis), the value will be preserved. Otherwise, the value will be suppressed.

In some implementations, the algorithm categorizes the continuous gradient directions into a small set of discrete directions, and then moves a 3x3 filter over the output of the previous step (that is, the edge strength and gradient directions). At every pixel, it suppresses the edge strength of the center pixel (by setting its value to 0) if its magnitude is not greater than the magnitude of the two neighbors in the gradient direction. For example,

- if the rounded gradient angle is 0° (i.e. the edge is in the north-south direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the **east and west** directions,
- if the rounded gradient angle is 90° (i.e. the edge is in the east-west direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the **north and south** directions,
- if the rounded gradient angle is 135° (i.e. the edge is in the northeast-southwest direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the **north west and south-east** directions,
- if the rounded gradient angle is 45° (i.e. the edge is in the north west–south east direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels in the **north east and south west** directions.

In more accurate implementations, linear interpolation is used between the two neighbouring pixels that straddle the gradient direction. For example, if the gradient angle is between 89° and 180°, interpolation between gradients at the **north** and **north east** pixels will give one interpolated value, and interpolation between the **south** and **south west**pixels will give the other (using the conventions of the last paragraph). The gradient magnitude at the central pixel must be greater than both of these for it to be marked as an edge. Note that the sign of the direction is irrelevant, i.e. north–south is the same as south–north and so on.

4) Double Threshold

After application of non-maximum suppression, remaining edge pixels provide a more accurate representation of real edges in an image. However, some edge pixels remain that are caused by noise and color variation. In order to account for these spurious responses, it is essential to filter out edge pixels with a weak gradient value and preserve edge pixels with a high gradient value. This is accomplished by selecting high and low threshold values. If an edge pixel's gradient value is higher than the high threshold value, it is marked as a strong edge pixel. If an edge pixel's gradient value is smaller than the high threshold value and larger than the low threshold value, it is marked as a weak edge pixel. If an edge pixel's value is smaller than the low threshold value, it will be suppressed. The two threshold values are empirically determined and their definition will depend on the content of a given input image.

5) Edge Tracking by Hysteresis

So far, the strong edge pixels should certainly be involved in the final edge image, as they are extracted from the true edges in the image. However, there will be some debate on the weak edge pixels, as these pixels can either be extracted from the true edge, or the noise/color variations. To achieve an accurate result, the weak edges caused by the latter reasons should be removed. Usually a weak edge pixel caused from true edges will be connected to a strong edge pixel while noise responses are unconnected. To track the edge connection, blob analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels. As long as there is one strong edge pixel that is involved in the blob, that weak edge point can be identified as one that should be preserved.

3.3.2 Results of Edge Detection:

The result of our edge detection are as shown below:

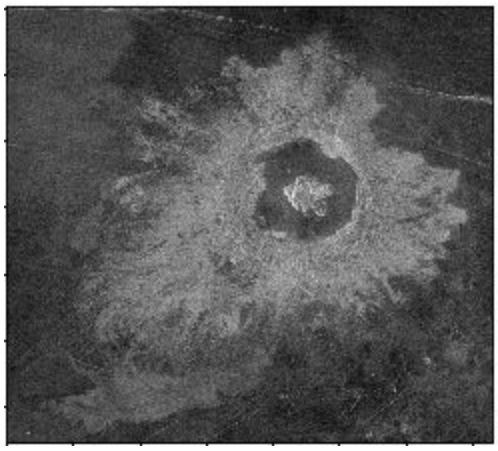


Fig. 3.12 - Original Image of Impact crater 3 3

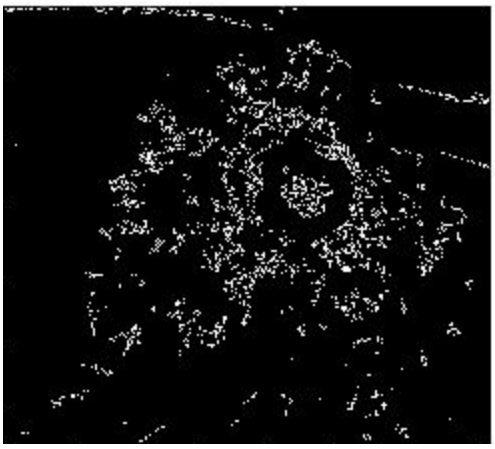


Fig. 3.13 - Image after edge detection of Impact Crater

3.4 Algorithm used to detect circular rim of craters:

Circular Hough Transform:

The circle Hough Transform (CHT) is a basic technique used in Digital Image Processing, for detecting circular objects in a digital image. The circle Hough Transform (CHT) is a feature extraction technique for detecting circles. It is a specialization of Hough Transform. The purpose of the technique is to find circles in imperfect image inputs. The circle candidates are produced by "voting" in the Hough parameter space and then select the local maxima in a so-called accumulator matrix.

3.4.1 Theory:

In a two-dimensional space, a circle can be described by:

$$(x-a)^2 + (y-b)^2 = r^2 \quad (1)$$

where (a,b) is the center of the circle, and r is the radius. If a 2D point (x,y) is fixed, then the parameters can be found according to (1). The parameter space would be three dimensional, (a, b, r). And all the parameters that satisfy (x, y) would lie on the surface of an inverted right-angled cone whose apex is at (x, y, 0). In the 3D space, the circle parameters can be identified by the intersection of many conic surfaces that are defined by points on the 2D circle. This process can be divided into two stages. The first stage is fixing radius then find the optimal center of circles in a 2D parameter space. The second stage is to find the optimal radius in a one

dimensional parameter space.

3.4.2 Find parameter with known radius:

If the radius is fixed, then the parameter space would be reduced to 2D (the position of the circle center). For each point (x, y) on the original circle, it can define a circle centered at (x, y) with radius R according to (1). The intersection point of all such circles in the parameter space would be corresponding to the center point of the original circle.

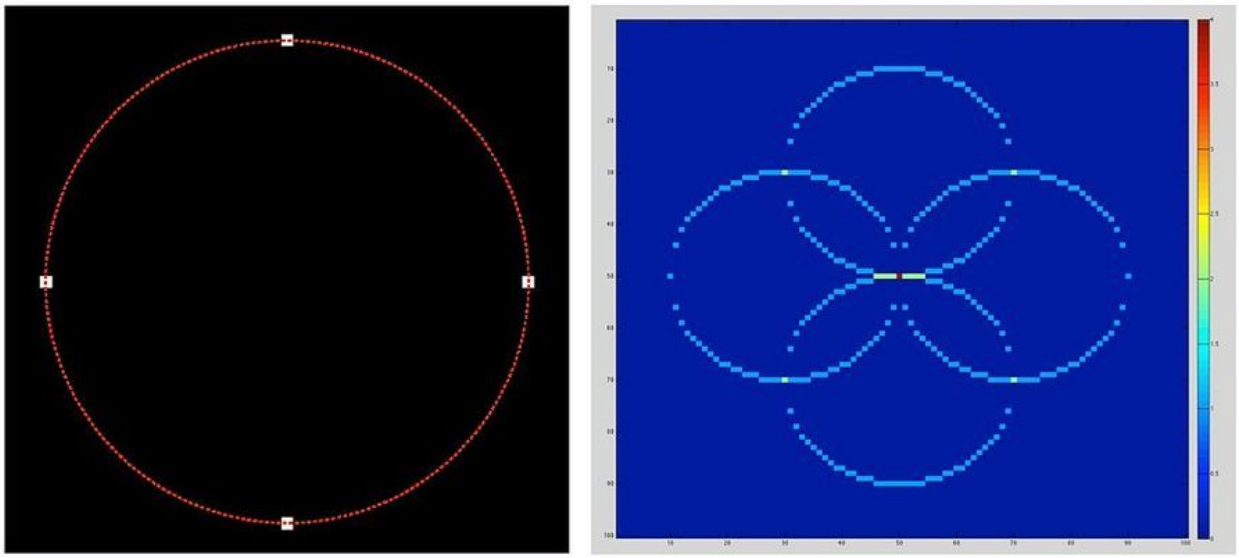


Fig. 3.14

Consider 4 points on a circle in the original image (left). The circle Hough transform is shown in the right. Note that the radius is assumed to be known. For each (x,y) of the four points (white points) in the original image, it can define a circle in the Hough parameter space centered at (x, y) with radius r. An accumulator matrix is used for tracking the intersection point. In the parameter space, the voting number of points through which the circle passing would be increased by one. Then the local maxima point (the red point in the center in the right figure) can be found. The position (a, b) of the maxima would be the center of the original circle.

3.4.3 Multiple Circle with known Radius R

Multiple circles with same radius can be found with the same technique.

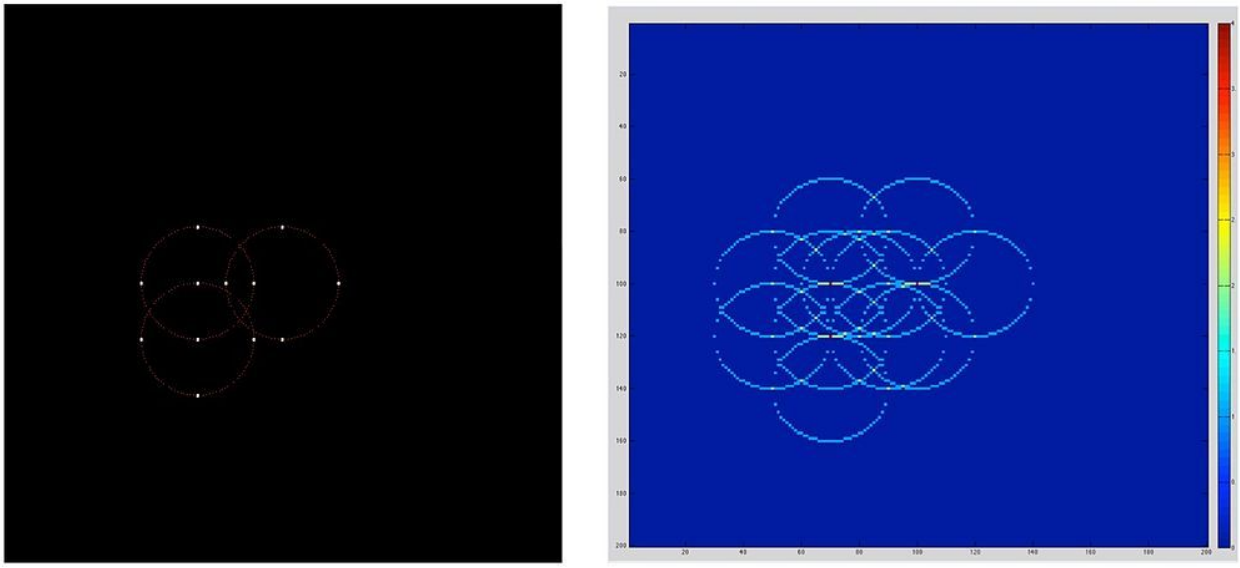


Fig 3.15

Note that, in the accumulator matrix (right fig), there would be at least 3 local maxima points.

3.4.4 Accumulator Matrix and Voting

In practice, an accumulator matrix is introduced to find the intersection point in the parameter space. First, we need to divide the parameter space into "buckets" using a grid and produce an accumulator matrix according to the grid. The element in the accumulator matrix denotes the number of "circles" in the parameter space that passing through the corresponding grid cell in the parameter space. The number is also called "voting number". Initially, every element in the matrix is zeros. Then for each "edge" point in the original space, we can formulate a circle in the parameter space and increase the voting number of the grid cell which the circle passing through. This process is called "voting".

After voting, we can find local maxima in the accumulator matrix. The positions of the local maxima are corresponding to the circle centers in the original space.

3.4.5 Circle Parameter with unknown radius:

Since the parameter space is 3D, the accumulator matrix would be 3D, too. We can iterate through possible radii; for each radius, we use the previous technique. Finally, find the local maxima in the 3D accumulator matrix. Accumulator array should be A[x,y,r] in the 3D space. Voting should be for each pixels, radius and theta A[x,y,r] += 1

3.4.6 The algorithm :

1. For each A[a,b,r] = 0;
2. Process the filtering algorithm on image Gaussian Blurring, convert the image to grayscale (grayScaling), make Canny operator, The Canny operator gives the edges on image.
3. Vote the all possible circles in accumulator.
4. The local maximum voted circles of Accumulator A gives the circle Hough space.
5. The maximum voted circle of Accumulator gives the circle.

3.4.7 Results of Crater Detection:

The detected circular craters are shown below.

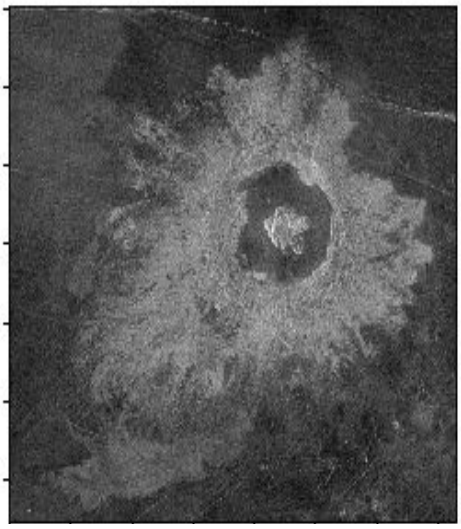


Fig. 3.16 - Original Image of Impact Crater 4

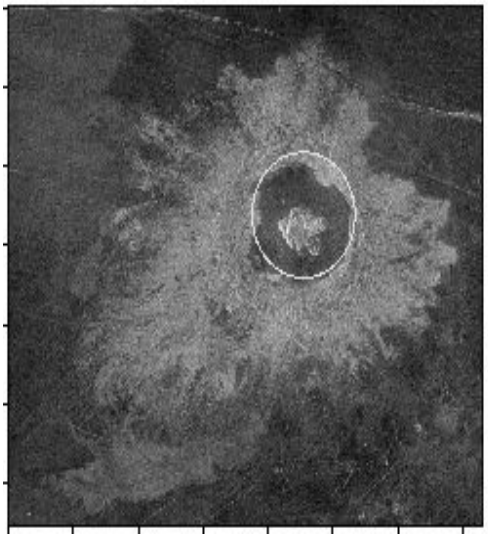


Fig. 3.17 - Detection of Crater on Impact crater 4

CHAPTER -4

SNAPSHOTS

1. Result of Analysed Impact Crater 1:

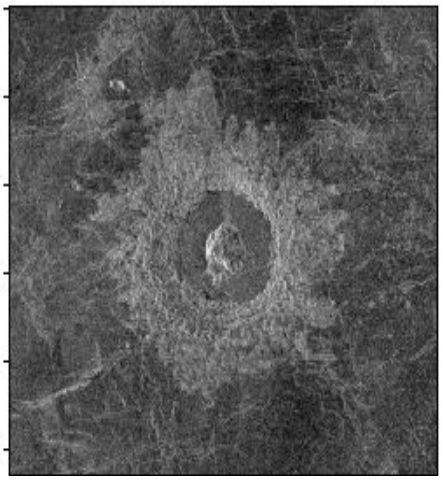


Fig 4.1 - Impact Crater 1

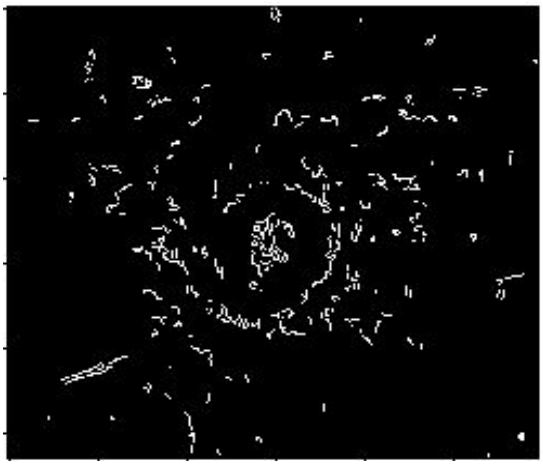


Fig 4.2 - Edge Detection on Impact Crater 1

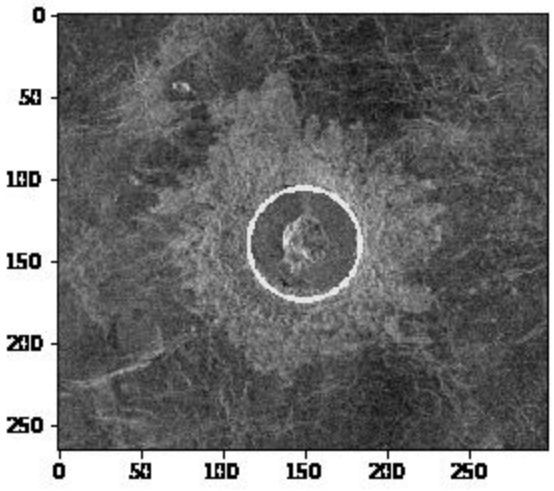


Fig 4.3 - Crater Detection on Impact Crater 1

2) Result of Analysed Impact Crater 2:

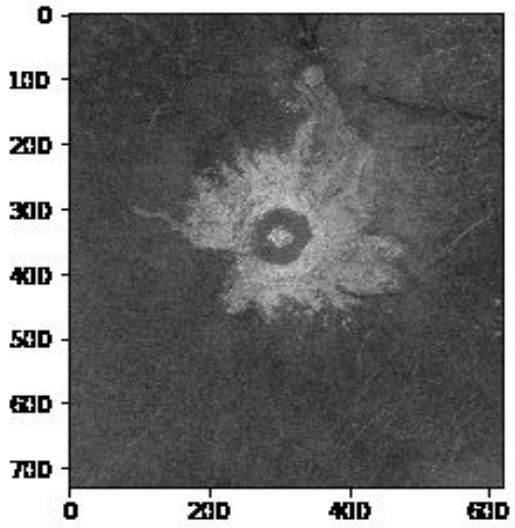


Fig 4.4 Impact Crater 2

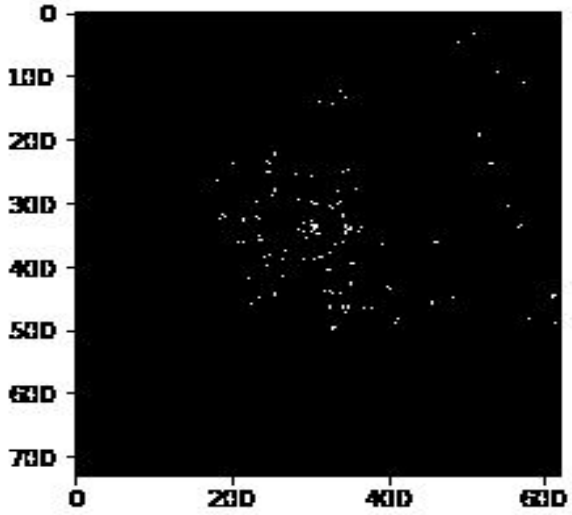


Fig 4.5 Edge Detection On Impact crater 2

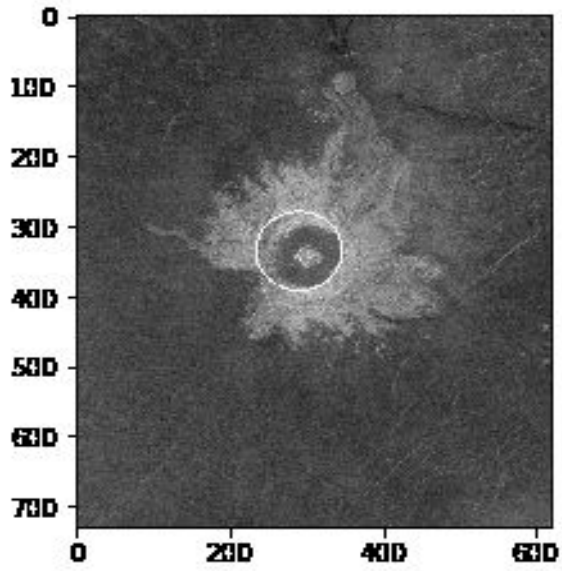


Fig 4.6 Crater Detection on Impact Crater 2

3) Result of Analysed impact crater 3:

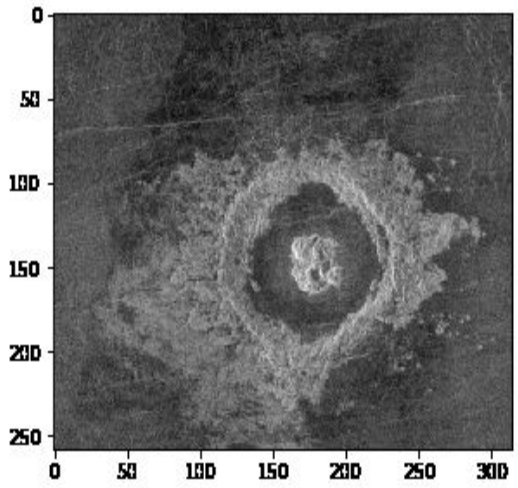


Fig 4.7 - Impact Crater 3

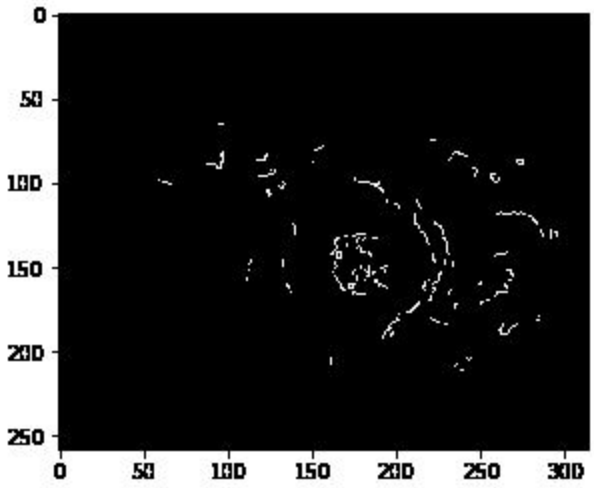


Fig 4.8 - Edge Detection on Impact Crater 3

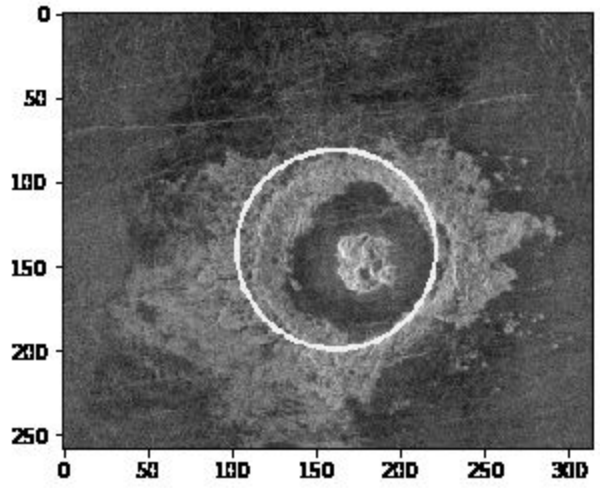
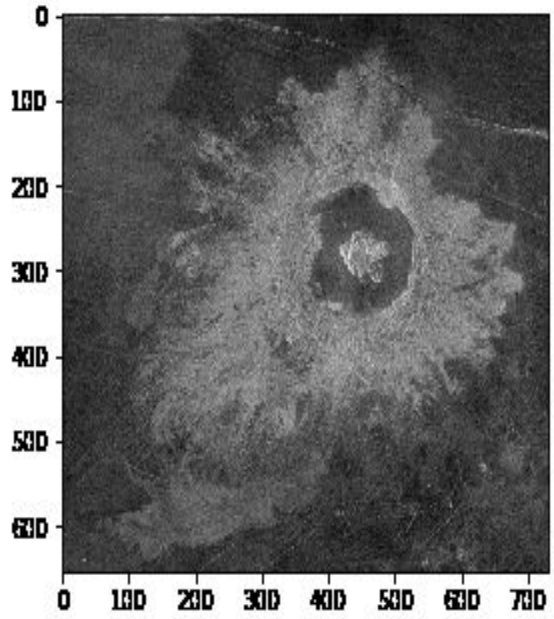
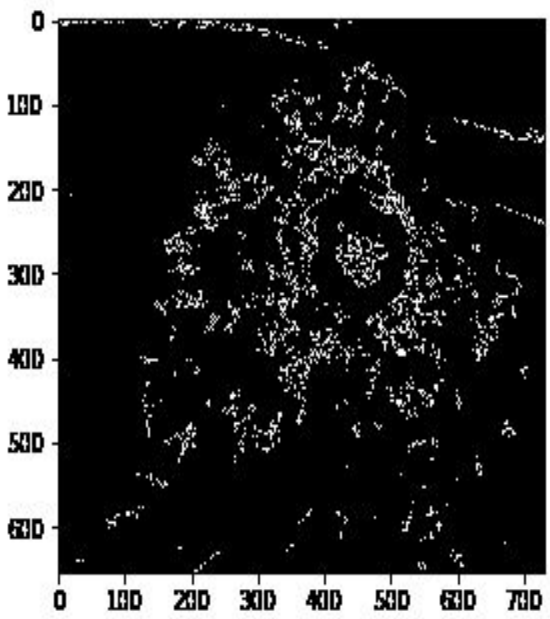
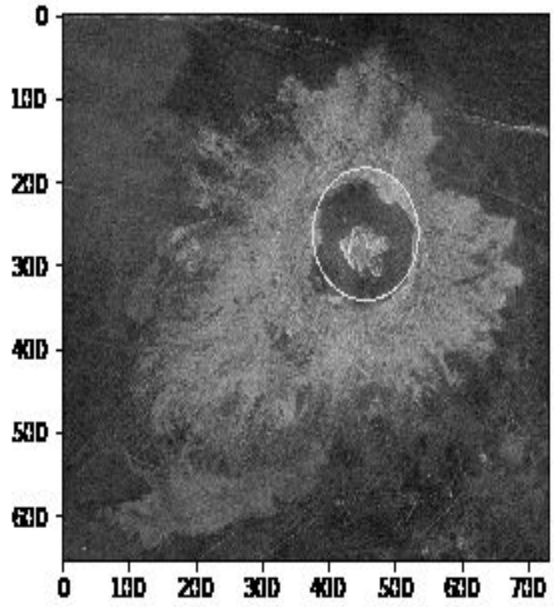


Fig 4.9 - Crater Detection on Impact Crater 3

4) Result of Analysed Impact Crater 4 :







CHAPTER -5
RESULTS AND DISCUSSIONS

5.1 Results:

In our project we found out that Circular Hough Transform works decently well on the problem of crater detection. The main advantages of circular Hough Transform technique is that it is tolerant of gaps in feature boundary description and is relatively unaffected by image noise. Also we see that edge detection plays a major role in detection of circular features. The results obtained by the canny edge detector has some noise but the detection of circle is unaffected by the noise.

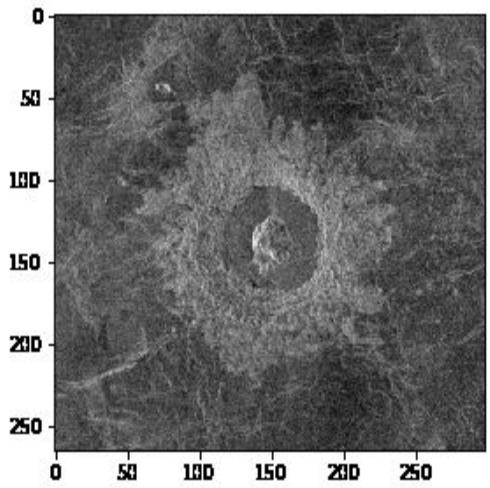


Fig. 5.1 - Original Image

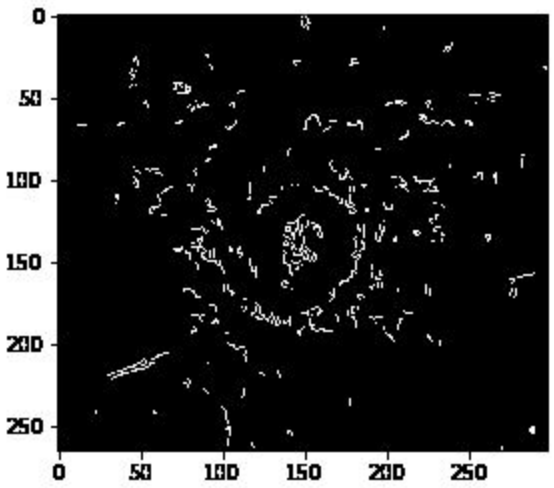


Fig. 5.2 - Edge Detection

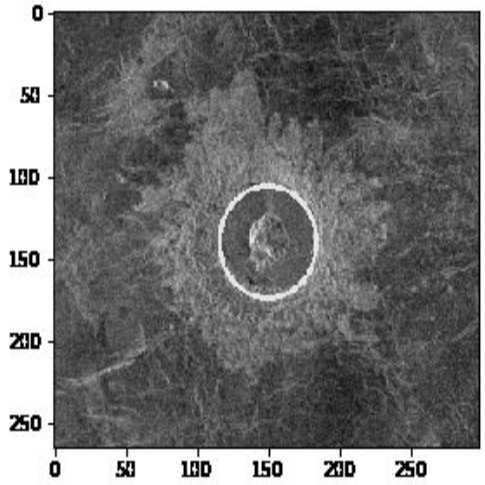


Fig. 5.3 - Crater Detected Image

5.2 Discussions :

Since Hough Transform is a brute force method, it is very complex in computation. It has 2 main drawbacks: large memory requirement and slowness. In order to find the plane parameters accurately, parameter space must be divided finely in all three directions, and an accumulator assigned to each block. Also it takes a long time to fill the accumulators when there are so many.

So, in order to improve its performance we can use Fast Circular Hough Transform. The Fast Hough Transform may be able to give considerable speed up and reduces memory requirement. Instead of dividing parameter space uniformly into blocks, the FHT homes in on the solution, ignoring areas in parameter space relatively devoid of votes. The relative speed advantage of the FHT increases for higher dimensional parameter spaces.

CHAPTER -6
CONCLUSIONS AND FUTURE SCOPE

6.1 Conclusions:

After applying circular Hough Transform on the SAR images of the impact crater on the surface of venus, we can conclude that the circular Hough Transform works decently well in identification of craters. The images of our project are shown below.

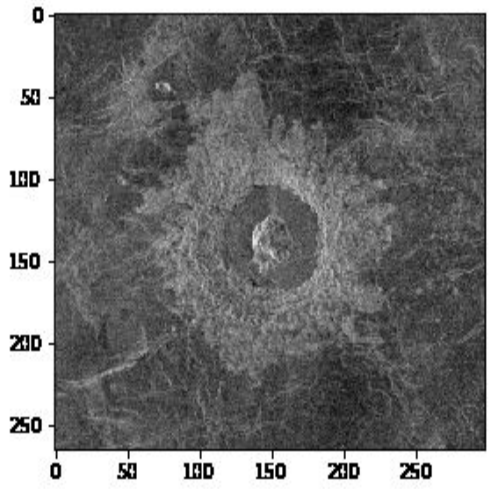


Fig. 6.1 - Original Image

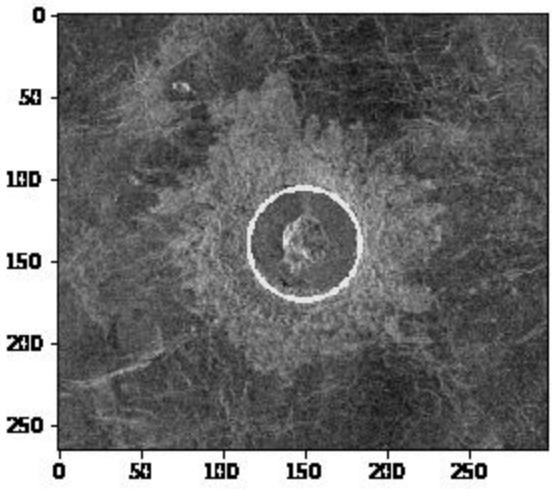


Fig. 6.2 - Crater Detection

5.2 Future Scope:

In this project I have considered the images with craters only. To make this work more efficient we can apply the circular hough transform on the images where there are object other than and including craters and try to detect only the craters in the image. For this we can follow the procedure discussed in this report and then apply classification to classify and detect the craters.

REFERENCES

[1]https://www.prl.res.in/~rajiv/planexnews/oldarticles/Volume%20-3,%20Issue-3.24-27.pdf

[2]https://jupyter.org/

[3]https://www.python.org/doc/

[4]https://numpy.org/

[5]https://matplotlib.org/tutorials/index.html

[6]