

# IMAGE SIMULATION AND VISUALIZATION

## INTERNSHIP PROJECT REPORT

Submitted by

Harshalkumar SunilKumar Patel

210303124014

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY  
in

Computer Science and Engineering

Parul Institute of Engineering and Technology, Limbda



VADODARA

February - 2025



# Parul Institute of Engineering & Technology

## Internship Report



**Parul University, Limbda**

**February-2025**



Established & Incorporated Under Gujarat Private Universities  
(Second Amendment) Act, 2015 (Guj. Act No. 7 of 2015)

**Parul**<sup>®</sup>  
**University**



Date: 12/11/2024

To,  
**Indian Space Research Organisation**  
**Ahmedabad**

Subject: NOC of the selected student for the internship

Dear Sir / Madam,

This is to inform that **Enrollment No 210303124014, Harshalkumar Sunilkumar Patel** from division **8B3** from our institute is allowed to join the internship from date **09-12-2024** up to **12-04-2025**. This student can join your organisation on full time basis but at the same time, he/she will be required to appear for all Weekly Tests, Mid-Sem Exams, External Semester Exams, vivas, submission and practical exams and must perform satisfactorily in order to become eligible to get degree certificate.

We would request you to kindly consider the same and approve leaves accordingly as per the exam schedule as & when gets finalised.

**Yours Faithfully,**

**Dr.Sanjay Agal**  
Head-AI & AIDS Dept.,  
Parul Institute of Engineering & Technology,  
Parul University, Vadodara.



# PARUL UNIVERSITY

## Declaration

We here by declare that the Project report submitted along with the Project entitled **Image simulation and Visualization** submitted in partial fulfillment for the degree of Bachelor of Engineering in **Computer Science Engineering** to Parul University, Vadodara, is a bonafide record of original project work carried out by me at **Indian Space Research Organisation** under the supervision of **Krishna Raulji** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Harshalkumar Sunilkumar Patel

Name of the Student

Sign of Student



# PARUL UNIVERSITY

## CERTIFICATE

This is to certify that the project report submitted along with the project entitled Image simulation and Visualization has been carried out by **Harshalkumar Sunilkumar Patel** under my guidance in partial fulfillment for the degree of Bachelor of Engineering in Computer Science and Engineering, 8th Semester of Parul University, Vadodara during the AY 2024-25.

**Krishna Raulji,**

Project Guide

**Dr. Sanjay Agal,**

Head of Department,

AI and AIDS, PIET,

Parul University.

## **Acknowledgements**

I would like to express my sincere gratitude to SAC ISRO, Ahmedabad for providing me with the opportunity to work on this project, "Image Simulation and Visualization." This experience has been invaluable in enhancing my knowledge of satellite imaging and digital image processing.

I extend my heartfelt thanks to my industry mentor, Mr. K. Suresh, for his continuous guidance, encouragement, and technical insights that helped shape this project. His expertise in satellite imaging and simulation provided me with a strong foundation to develop this system.

I am also immensely grateful to my institute mentor, Ms. Krishna Rauji, for her unwavering support, valuable feedback, and constructive suggestions throughout the project. Her mentorship played a crucial role in refining my approach and ensuring the project's successful completion.

Lastly, I would like to thank my colleagues, peers, and everyone who supported and motivated me throughout this journey. This project would not have been possible without their constant encouragement and collaboration.

**Harshalkumar SunilKumar Patel**

**AI and AIDS, PIET**

**Parul University,**

**Vadodara**

## **Abstract**

The Image Simulation and Visualization project aims to assist in designing and evaluating satellite imaging systems by simulating how a camera would capture images in a lunar environment. By allowing users to define various camera parameters such as position, altitude, roll, yaw, pitch, and lens settings, this system generates simulated images that closely represent the expected results from actual lunar missions.

The project is developed using PyQt5 and Qt Designer for the graphical user interface, while NumPy, PyQtGraph, and GDAL handle image processing and spatial data operations. The iterative waterfall model is followed to ensure systematic development and refinement based on feedback at each stage.

This system plays a crucial role in optimizing camera settings before deployment on lunar satellites, reducing experimental costs and improving mission accuracy. The feasibility and effectiveness of the simulation ensure that future moon exploration missions can be planned with greater precision and confidence.

# Table of Contents

<b>Acknowledgements</b>	<b>vi</b>
<b>Abstract</b>	<b>vii</b>
<b>1 Overview of the Company</b>	<b>1</b>
1.1 History of SAC ISRO . . . . .	1
1.2 Different Products / Scope of Work . . . . .	1
1.3 Organization Chart . . . . .	2
1.4 Capacity of Plant . . . . .	2
<b>2 Operational Details</b>	<b>3</b>
2.1 Overview of Different Units and Departments . . . . .	3
2.1.1 Payload Design and Development . . . . .	3
2.1.2 Satellite Systems Integration and Testing . . . . .	3
2.1.3 Earth Observation and Remote Sensing . . . . .	3
2.1.4 Communication Systems . . . . .	4
2.1.5 Ground Systems and Control . . . . .	4
2.1.6 Applications Development . . . . .	4
2.2 Technical Specifications of Major Equipment Used in Each Department . . . . .	4
2.2.1 Payload Design and Development Department . . . . .	5
2.2.2 Satellite Integration and Testing Department . . . . .	5
2.2.3 Ground Systems Department . . . . .	5

2.3	2.3 Schematic Layout Showing Sequence of Operations for Manufacturing End Product . . . . .	5
2.4	2.4 Explanation of Each Stage of Production . . . . .	6
<b>3</b>	<b>Introduction to Project</b>	<b>7</b>
3.1	Project Summary . . . . .	7
3.2	Purpose . . . . .	7
3.3	Objective . . . . .	7
3.4	Scope (What it Can and Can't Do) . . . . .	8
3.5	Technology and Literature Review . . . . .	8
3.6	Project Planning . . . . .	9
3.6.1	Project Development Approach and Justification . . . . .	9
3.6.2	Project Effort and Time, Cost Estimation . . . . .	9
<b>4</b>	<b>System Analysis</b>	<b>10</b>
4.1	Study of Current System . . . . .	10
4.2	Requirements of New System . . . . .	10
4.3	System Feasibility . . . . .	11
4.3.1	Does the system contribute to the overall objectives of the organization? . . . . .	11
4.3.2	Can the system be implemented using the current technology and schedule constraints? . . . . .	11
4.3.3	Can the system be integrated with other systems already in place? . . . . .	11
4.4	Activity / Process in New System / Proposed System . . . . .	11
4.5	Features of New System / Proposed System . . . . .	12
4.6	List of Main Modules / Components / Processes / Techniques of New System / Proposed System . . . . .	12

<b>4.7 Selection of Hardware / Software / Algorithms / Methodology / Techniques / Approaches and Justification . . . . .</b>	<b>12</b>
<b>4.7.1 Hardware: . . . . .</b>	<b>12</b>
<b>4.7.2 Software: . . . . .</b>	<b>13</b>
<b>4.7.3 Algorithms: . . . . .</b>	<b>13</b>
<b>4.7.4 Methodology: . . . . .</b>	<b>13</b>
<b>4.7.5 Techniques: . . . . .</b>	<b>13</b>
<b>5 System Design . . . . .</b>	<b>14</b>
<b>5.1 System Design &amp; Methodology . . . . .</b>	<b>14</b>
<b>5.1.1 Graphical User Interface (GUI) . . . . .</b>	<b>14</b>
<b>5.1.2 Camera Parameters Handling . . . . .</b>	<b>14</b>
<b>5.1.3 Marker-Based Simulation . . . . .</b>	<b>14</b>
<b>5.1.4 Footprint Preview . . . . .</b>	<b>14</b>
<b>5.1.5 Image Processing . . . . .</b>	<b>14</b>
<b>5.2 Data Structure Design / Process Design / Structure Design . . . . .</b>	<b>15</b>
<b>5.2.1 Data Structure Design . . . . .</b>	<b>15</b>
<b>5.2.2 Process Design . . . . .</b>	<b>15</b>
<b>5.3 Input / Output and Interface Design . . . . .</b>	<b>15</b>
<b>5.3.1 State Transition Diagram (Optional) . . . . .</b>	<b>15</b>
<b>5.3.2 Samples of Forms, Reports, and Interface . . . . .</b>	<b>16</b>
<b>5.3.3 Access Control / Security (If Applicable) . . . . .</b>	<b>16</b>
<b>6 IMPLEMENTATION . . . . .</b>	<b>17</b>
<b>6.1 Implementation Platform / Environment . . . . .</b>	<b>17</b>
<b>6.2 Process / Program / Technology / Modules Specification(s) . . . . .</b>	<b>17</b>
<b>6.2.1 Process Flow . . . . .</b>	<b>17</b>

6.2.2	Technologies & Modules Used . . . . .	18
6.2.3	Findings / Results / Outcomes . . . . .	18
6.2.4	Result Analysis / Comparison / Deliberations . . . . .	19
<b>7</b>	<b>Testing</b>	<b>20</b>
7.1	Testing Plan / Strategy . . . . .	20
7.1.1	Testing Types Used . . . . .	20
7.1.2	Testing Environment . . . . .	20
7.2	Test Results and Analysis . . . . .	20
7.2.1	Test Cases . . . . .	21
7.3	Analysis of Test Results . . . . .	21
<b>8</b>	<b>Conclusion and Summary</b>	<b>22</b>
8.1	Conclusion and Summary . . . . .	22
8.2	Limitations . . . . .	22

# **List of Tables**

7.1 Test Cases for Image Simulation System . . . . .	21
--	----

# **List of Figures**

# 1 OVERVIEW OF THE COMPANY

## 1.1 HISTORY OF SAC ISRO

The Space Applications Centre (SAC), located in Ahmedabad, is a premier unit of the Indian Space Research Organisation (ISRO), established in 1973. SAC plays a pivotal role in the development of space-based applications and technologies for communication, broadcasting, meteorology, Earth observation, and satellite payloads.

SAC is involved in designing and developing payloads and space systems for Indian satellites, contributing to various national and international missions.

SAC's primary focus has been on developing payloads that enable the delivery of services like telecommunication, television, remote sensing, meteorology, and satellite navigation.

SAC's efforts have been integral to the success of ISRO's commercial and scientific satellite programs.

## 1.2 DIFFERENT PRODUCTS / SCOPE OF WORK

SAC is responsible for the design and development of advanced payloads, satellite communication systems, and other critical components used in Earth observation and communication satellites. The major products and scope of work include:

- **Satellite Payloads** – Development of sensors, transponders, and communication equipment.
- **Spacecraft Bus Systems** – Design and development of the satellite structure, thermal management, and power systems.
- **Ground Systems** – Establishment of ground stations and communication infrastructure.
- **Earth Observation Systems** – Development of systems like remote sensing payloads for applications in agriculture, forestry, and disaster management.
- **Satellite Launch and Mission Support** – Assisting in mission planning and execution for satellite launches and operation of payloads.

### **1.3 ORGANIZATION CHART**

The organization structure of SAC ISRO consists of multiple divisions and departments focusing on different aspects of space research, design, and development. The key departments include:

- Payload Design and Development
- Communication Payload Systems
- Spacecraft Integration and Testing
- Ground Segment Operations
- Satellite Data Processing
- Applications Development

A detailed organization chart will display the hierarchical structure from the Director down to the specialized teams responsible for various tasks in the development, integration, and testing of space systems.

### **1.4 CAPACITY OF PLANT**

SAC is equipped with state-of-the-art facilities that cater to the design, integration, and testing of satellite payloads and communication systems. These facilities include:

- **Clean Room Facilities** – SAC's clean rooms ensure a contamination-free environment for sensitive satellite payload assembly.
- **Antenna Testing and Simulation Lab** – Used for the testing and simulation of communication systems and payloads.
- **Satellite Payload Integration Facility** – Where satellite payloads are integrated, tested, and validated.
- **Thermal and Vibration Testing** – Specialized testing facilities that simulate space conditions like temperature extremes and vibration during launch.
- **High-Performance Computing Systems** – Support the simulation, data analysis, and image processing for mission planning and satellite operation.

**Capacity:** SAC can handle the development and integration of multiple satellites simultaneously, supporting both Indian and international missions.

# **2 OPERATIONAL DETAILS**

## **2.1 OVERVIEW OF DIFFERENT UNITS AND DEPARTMENTS**

SAC ISRO is divided into various specialized departments that focus on the research, design, development, testing, and integration of space systems. The key departments involved in the development of satellite payloads and communication systems are:

### **2.1.1 Payload Design and Development**

**Scope:** This department is responsible for designing the payloads used in communication satellites, remote sensing satellites, and other space applications.

#### **Work Carried Out:**

- Design and development of payload systems like communication transponders, sensor payloads, and optical instruments for Earth observation.
- Development of payload integration strategies and testing procedures.

### **2.1.2 Satellite Systems Integration and Testing**

**Scope:** Focuses on integrating various subsystems of satellites (payloads, bus systems, etc.) and ensuring they function as a cohesive unit.

#### **Work Carried Out:**

- Integration of communication systems, electrical power systems, thermal management systems, and payload systems.
- Conducting rigorous testing (thermal, vibration, EMI, etc.) to ensure that the satellite meets the performance standards required for space missions.

### **2.1.3 Earth Observation and Remote Sensing**

**Scope:** Development of remote sensing payloads for Earth observation and monitoring.

#### **Work Carried Out:**

- Design and testing of imaging sensors, radar systems, and hyperspectral sensors for applications like agriculture, disaster management, weather monitoring, and environmental monitoring.
- Development of data processing and analysis systems to handle large volumes of satellite imagery.

#### **2.1.4 Communication Systems**

**Scope:** Responsible for the development of space-based communication systems including telecommunication transponders, broadcasting systems, and satellite networks.

**Work Carried Out:**

- Designing transponders, antennas, and communication protocols to enable satellite communication services.
- Implementation of ground communication infrastructure to support satellite data transmission.

#### **2.1.5 Ground Systems and Control**

**Scope:** Design and implementation of ground stations to track, communicate with, and control satellites during their mission life cycle.

**Work Carried Out:**

- Setting up ground stations equipped with tracking and data receiving equipment.
- Managing satellite health and performing routine control functions for satellites.

#### **2.1.6 Applications Development**

**Scope:** Develop innovative space-based applications using the data provided by the satellites, such as satellite navigation, telemedicine, and e-learning.

**Work Carried Out:**

- Implementation of space technology for practical solutions in areas like disaster management, telemedicine, and education.
- Develop user interfaces and systems for end-users to access satellite-based services.

## **2.2 TECHNICAL SPECIFICATIONS OF MAJOR EQUIPMENT USED IN EACH DEPARTMENT**

Each department at SAC ISRO employs specialized equipment and technology to carry out their tasks effectively. Below are the major technical specifications of some of the equipment used:

**2.2.1 Payload Design and Development Department****Imaging Sensors:**

- Resolution: Up to 1-meter resolution for Earth observation.
- Spectral Range: Multi-spectral (visible, infrared, thermal).

**Communication Transponders:**

- Frequency Range: C-band, Ku-band, S-band.
- Power Output: Up to 100 watts for high-efficiency communication.

**2.2.2 Satellite Integration and Testing Department****Thermal Vacuum Chamber:**

- Temperature Range: From -150°C to 150°C.
- Size: Can accommodate satellites of up to 6 meters in diameter.

**Vibration Testing Equipment:**

- Frequency Range: Can simulate vibration from 1 Hz to 10,000 Hz for satellite testing.

**2.2.3 Ground Systems Department****Tracking and Data Reception Antennas:**

- Size: 10m to 30m parabolic dish antennas.
- Frequency Range: Covers L-band, S-band, C-band, and Ku-band.

**2.3 2.3 SCHEMATIC LAYOUT SHOWING SEQUENCE OF OPERATIONS FOR MANUFACTURING END PRODUCT**

The manufacturing process at SAC ISRO follows a structured sequence for designing, developing, integrating, and testing space systems. The schematic layout of the workflow includes the following steps:

**1. Design Phase:**

- Conceptual design of payloads, subsystems, and satellite bus systems.
- Detailed system architecture design and simulations.

**2. Development Phase:**

- Procurement of components.
- Fabrication and assembly of satellite payloads and systems.

**3. Integration Phase:**

- Assembly of satellite components.
- Integration of payloads with satellite bus systems.

**4. Testing Phase:**

- Thermal, vibration, and electromagnetic interference (EMI) tests.
- Performance verification through ground-based simulations.

**5. Launch and Deployment Phase:**

- Satellite launch via ISRO's launch vehicles.
- Monitoring and control through ground stations post-launch.

**2.4 2.4 EXPLANATION OF EACH STAGE OF PRODUCTION**

- 1. Design:** In the design stage, concepts for payloads, spacecraft systems, and ground infrastructure are developed. The design is based on mission objectives and space conditions.
- 2. Development:** Components such as payloads, sensors, and communication equipment are designed, fabricated, and tested for functionality.
- 3. Integration:** In this phase, the various subsystems such as payloads, power systems, and thermal systems are integrated into the spacecraft.
- 4. Testing:** The integrated system undergoes rigorous performance testing, which includes thermal vacuum tests, vibration tests, and power system validation.
- 5. Launch and Deployment:** The final satellite is launched, and real-time data transmission and satellite health checks are conducted from the ground.

# **3 INTRODUCTION TO PROJECT**

## **3.1 PROJECT SUMMARY**

This project focuses on Moon Surface Image Simulation using satellite imaging technologies to simulate and visualize images of the moon's surface. The goal is to understand how a camera will capture an image in an unknown environment, such as the moon's surface, and help in designing and setting up camera parameters for such simulations. The project aims to provide an effective system for simulating images by adjusting parameters like camera position, altitude, roll, yaw, and pitch, which can be used in designing cameras for future lunar missions.

By simulating the moon's surface images based on user input, the system can assist in camera parameter optimization for space missions. It provides an interface where users can input pre-captured images and define specific parameters, and in return, they will get a simulated moon surface image.

## **3.2 PURPOSE**

The primary purpose of this project is to simulate and visualize how the moon's surface will appear based on different camera parameters such as position, altitude, and orientation. The system will serve as a tool for space scientists and engineers involved in the design and calibration of lunar imaging systems.

By offering an interactive platform, the project aims to:

- Provide a realistic simulation of moon surface images based on camera position and settings.
- Aid in studying how camera will capture the images.

## **3.3 OBJECTIVE**

The main objectives of this project are:

- To design a system that can take user input images (pre-captured) and simulate the appearance of the moon's surface under different camera parameters.

- To help in studying for lunar missions by visualizing how the camera would capture images of the moon from different perspectives.
- To develop an intuitive user interface using PyQt5 that allows users to interact with the simulation and adjust parameters like latitude, longitude, altitude, and camera orientation.

### **3.4 SCOPE (WHAT IT CAN AND CAN'T DO)**

#### **Scope of the Project:**

- The system can simulate images of the moon's surface, enabling users to visualize how different camera parameters impact the final image.
- The tool will allow users to input camera settings, such as camera position, altitude, and orientation parameters, and generate simulated images of the lunar surface.

#### **What the Project Can't Do:**

- It cannot provide real-time image capturing or real-world data from lunar surface cameras.
- It cannot generate new images of the moon based on raw data but will only work with pre-captured input images.
- The simulation is limited to a static model and does not include dynamic changes (e.g., changing lunar phases or real-time environmental variations on the moon's surface).

### **3.5 TECHNOLOGY AND LITERATURE REVIEW**

The project uses the following technologies:

- **PyQt5:** For creating the graphical user interface (GUI) that allows users to input and manipulate parameters.
- **PyQtGraph:** Used for visualization, it provides a fast and efficient way to display simulated images and graphical representations of camera settings.
- **GDAL:** A library used to handle geographic data and transform spatial data for accurate simulation of images.
- **NumPy:** Essential for performing mathematical operations and data transformations involved in the image simulation.

## **3.6 PROJECT PLANNING**

### **3.6.1 Project Development Approach and Justification**

This project follows the Iterative Waterfall Model, allowing for incremental development and continuous feedback after each phase. The iterative model provides flexibility to refine the design after each cycle, ensuring that user needs are met and issues are addressed promptly.

**Phase 1: Requirements Gathering and Initial Design:** In this phase, project requirements were gathered from experts in space imaging.

**Phase 2: Prototype Development and User Interface Design:** Based on the requirements, an initial prototype of the GUI was developed using PyQt5. Early simulations with predefined parameters were conducted to verify the system's functionality.

**Phase 3: Iteration and Refinement:** After initial feedback from users, the system was iteratively refined. New features like parameter adjustments and image enhancement tools were added based on user input.

### **3.6.2 Project Effort and Time, Cost Estimation**

The project is expected to take 3-4 months for completion. This includes time for prototype development, testing, and refinement of the system.

#### **Time Estimation:**

- Prototype development: 1 month
- Testing and refinement: 2 months
- Final adjustments and presentation: 1 month

**Cost Estimation:** The project will require already available hardware for testing.

# 4 SYSTEM ANALYSIS

## 4.1 STUDY OF CURRENT SYSTEM

Currently, there is no integrated system that allows users to simulate the appearance of the moon's surface based on varying camera parameters in a user-friendly way. Various image processing and satellite simulation tools exist but often focus on post-processing images or limited camera settings. Therefore, a need exists for a customizable, efficient tool that can provide simulations of lunar surface images with the flexibility to adjust camera positions, orientations, and other parameters.

Existing systems for space imaging simulations may lack features like interactive user inputs or adjustments for environmental factors such as altitude, yaw, and pitch. They also tend to be rigid in their design, focusing mainly on theoretical models rather than offering detailed user-driven simulations.

## 4.2 REQUIREMENTS OF NEW SYSTEM

The new system must fulfill the following requirements:

1. **Customizable Parameters:** The system should allow users to input and modify camera parameters, such as altitude, latitude, longitude, pitch, yaw, and roll.
2. **User-Friendly Interface:** The system must have an intuitive graphical user interface (GUI) developed using PyQt5, enabling easy input of parameters and viewing of simulated results.
3. **Realistic Simulations:** It should simulate realistic moon surface images using pre-captured satellite images.
4. **Support for Lunar Mission Data:** The system must allow integration with data from lunar missions such as Chandrayaan, enabling the simulation of different lunar regions and testing of different camera configurations.

## 4.3 SYSTEM FEASIBILITY

### 4.3.1 Does the system contribute to the overall objectives of the organization?

Yes, the system contributes to the overall objectives of ISRO, especially in the area of satellite imaging and space exploration. By helping engineers design and set up optimal camera parameters for lunar missions, the tool supports space mission readiness and camera configuration optimization. It aligns with the vision of advancing space technology and enhancing satellite-based remote sensing for lunar exploration.

### 4.3.2 Can the system be implemented using the current technology and schedule constraints?

Yes, the system can be implemented using current technologies such as PyQt5, NumPy, PyQtGraph, and GDAL. These tools are highly capable for developing the GUI, performing mathematical computations, and processing satellite image data. Additionally, the iterative waterfall model allows for efficient development within the time constraints, as each phase will build upon feedback from the previous iteration.

### 4.3.3 Can the system be integrated with other systems already in place?

Yes, the system is designed to integrate with existing simulation modules

## 4.4 ACTIVITY / PROCESS IN NEW SYSTEM / PROPOSED SYSTEM

The proposed system will include the following activities:

- **Input Data:** Users will provide pre-captured lunar images and set parameters such as latitude, longitude, altitude, camera orientation, etc.
- **Image Simulation:** Based on the input parameters, the system will simulate a new image of the moon's surface, reflecting how the camera would capture the scene in those specific parameters.
- **Visualization:** Simulated images will be visualized on a graphical interface, and users can interact with the interface to adjust parameters.
- **Export:** After running the simulation, users can export the generated images or data for further analysis.

## 4.5 FEATURES OF NEW SYSTEM / PROPOSED SYSTEM

- **Interactive User Interface:** Developed using PyQt5, allowing users to easily input and modify camera parameters and view simulated images.
- **Simulation:** Based on pre-defined camera parameters and the moon's surface data.
- **Parameter Control:** Includes controls for various camera settings (e.g., latitude, longitude, altitude, pitch, yaw, roll).
- **Export and Report Generation:** Users can save simulated images and generate detailed reports for mission planning.

## 4.6 LIST OF MAIN MODULES / COMPONENTS / PROCESSES / TECHNIQUES OF NEW SYSTEM / PROPOSED SYSTEM

The main modules/components of the system will include:

- **User Interface Module:** Developed with PyQt5 for user input and interaction.
- **Simulation:** Handles image processing and simulation based on camera parameters using NumPy and GDAL.
- **Visualization Module:** Uses PyQtGraph to display the simulated image and allow interactive viewing.
- **Data Integration Module:** Interfaces with satellite data for real-time simulation.
- **Export/Reporting Module:** Allows users to export simulated images for analysis.

## 4.7 SELECTION OF HARDWARE / SOFTWARE / ALGORITHMS / METHODOLOGY / TECHNIQUES / APPROACHES AND JUSTIFICATION

### 4.7.1 Hardware:

The project will be developed on standard computers with sufficient computational power to handle image processing. Systems should have good graphic processing capabilities for efficient simulation and rendering.

**4.7.2 Software:**

- PyQt5 for GUI development
- NumPy for mathematical operations
- PyQtGraph for image visualization
- GDAL for handling geospatial data

**4.7.3 Algorithms:**

Parallel processing algorithms were implemented for fast processing.

**4.7.4 Methodology:**

The project will follow the Iterative Waterfall Model, ensuring that each phase undergoes continuous refinement through feedback loops.

**4.7.5 Techniques:**

The system will employ camera parameter optimization and image simulation techniques to provide users with a realistic simulation of lunar surface imagery.

# 5 SYSTEM DESIGN

## 5.1 SYSTEM DESIGN & METHODOLOGY

The image simulation system is designed to take a GeoTIFF image as input and simulate how a camera would capture it in an unknown environment. The methodology follows a structured pipeline consisting of image loading, parameter configuration, marker placement, and simulation output.

The key components of the system design are:

### 5.1.1 Graphical User Interface (GUI)

Built using PyQt and PyQtGraph, providing an interactive environment for parameter adjustments and image visualization.

### 5.1.2 Camera Parameters Handling

Users can set parameters such as detector height, width, FOV, focal length, and altitude to define the camera specifications.

### 5.1.3 Marker-Based Simulation

Users can place markers using latitude-longitude coordinates to define the simulation area. Single-point mode allows one marker, while multi-point mode enables simulation between two points.

### 5.1.4 Footprint Preview

The system provides a footprint visualization before actual simulation to help users understand the expected output coverage.

### 5.1.5 Image Processing

The core simulation logic processes the input GeoTIFF based on selected camera parameters, generating the simulated image.

This structured approach ensures that the system effectively meets the objective of simulating camera behavior for testing and design purposes.

## 5.2 DATA STRUCTURE DESIGN / PROCESS DESIGN / STRUCTURE DESIGN

### 5.2.1 Data Structure Design

The system primarily works with the following data structures:

- **Camera Parameter Dictionary:** Stores values such as detector size, FOV, focal length, and altitude.
- **GeoTIFF Image Data:** Handled as multi-dimensional arrays for processing.
- **Marker Data:** Stores latitude-longitude pairs for single and multi-point modes.

### 5.2.2 Process Design

The image simulation follows these steps:

1. **Load Image:** User selects a GeoTIFF image.
2. **Set Camera Parameters:** Parameters are either manually entered or selected from the preloaded camera types via a combo box.
3. **Mark Position(s):** User marks a single or two-point location using the mouse or by entering coordinates.
4. **Preview Footprint:** The system calculates and displays the expected coverage.
5. **Simulate Image:** The system processes the image according to camera parameters and outputs the simulated image.
6. **Save or Clear:** Users can save results or reset parameters and markers.

## 5.3 INPUT / OUTPUT AND INTERFACE DESIGN

### 5.3.1 State Transition Diagram (Optional)

If required, a state transition diagram can be included to depict the various states of the system, such as:

Idle → Image Loaded → Parameters Set → Simulation Performed → Output Displayed

### **5.3.2 Samples of Forms, Reports, and Interface**

- **Forms & Inputs:** The GUI provides QLineEdit fields for numerical input and a QComboBox for selecting preloaded camera types.
- **Interactive Controls:** Buttons for loading, saving, and clearing data, along with QDials for adjusting roll, pitch, yaw, and tilt angles.
- **Visualization:** The PyQtGraph canvas displays the input and output images with an interactive histogram.

### **5.3.3 Access Control / Security (If Applicable)**

Since this is a local simulation tool, access control is minimal. However, basic safeguards such as input validation and error handling are implemented to prevent incorrect data entry.

# **6 IMPLEMENTATION**

## **6.1 IMPLEMENTATION PLATFORM / ENVIRONMENT**

The image simulation system is developed using Python and implemented on a Windows/Linux environment. The key components of the implementation platform are:

- **Programming Language:** Python
- **GUI Framework:** PyQt5 (for creating an interactive graphical interface)
- **Image Processing Library:** GDAL (for handling GeoTIFF images)
- **Visualization Library:** PyQtGraph (for displaying images and histograms)
- **Mathematical Computation:** NumPy (for handling array-based image transformations)
- **Development Environment:** Spyder
- **Hardware Requirements:** A system with a minimum of 8GB RAM, multi-core CPU, and GPU acceleration

The system runs efficiently on standard computing hardware, with optimizations to handle large GeoTIFF files.

## **6.2 PROCESS / PROGRAM / TECHNOLOGY / MODULES**

### **SPECIFICATION(S)**

#### **6.2.1 Process Flow**

The image simulation follows these steps:

- 1. Loading the Input Image:**

- Users load a GeoTIFF file using a file selection dialog.

- The image data is read using the GDAL library and processed into an array for simulation.

## **2. Setting Camera Parameters:**

- Users manually enter or select a predefined camera type from a QComboBox.
- Parameters such as detector size, focal length, field of view, and altitude are applied.

## **3. Marking Positions (Single or Multi-Point Mode):**

- Users click on the image or enter latitude-longitude coordinates to set markers.
- The system converts these to pixel coordinates for processing.

## **4. Footprint Visualization:**

- A footprint button calculates and overlays the expected simulated area on the original image.

## **5. Simulating the Image:**

- The system applies geometric transformations based on camera parameters.
- The modified image is displayed on the PyQtGraph canvas.

## **6. Saving / Resetting Data:**

- Users can save results, reset inputs, or clear markers as needed.

### **6.2.2 Technologies & Modules Used**

The following technologies and modules are used in the implementation:

- **PyQt5:** GUI framework for interactive controls.
- **PyQtGraph:** Efficient image rendering and histogram display.
- **GDAL:** Reads and processes GeoTIFF images.
- **NumPy:** Handles mathematical transformations for image simulation.

### **6.2.3 Findings / Results / Outcomes**

The implementation of the system resulted in the following outcomes:

- **Accurate Image Simulation:** The system successfully simulates how a camera would capture an image in an unknown environment.

- **Parameter Adjustment:** Users can modify camera parameters and observe changes dynamically.
- **Footprint Visualization:** The footprint feature helps users understand how the final image will appear before simulation.
- **Multi-Mode Simulation:** Both single-point and multi-point simulation modes function as expected.
- **Efficient Processing:** The use of NumPy and GDAL ensures that large GeoTIFF images are processed efficiently.

#### **6.2.4 Result Analysis / Comparison / Deliberations**

The system was tested with various input images and camera parameters to evaluate its performance.

The key observations include:

- **Comparison with Theoretical Expectations:** The simulated images align with expected outputs based on camera equations.
- **Performance Metrics:** Processing time varies based on image resolution and selected parameters. Higher-resolution images take longer to process.
- **User Interaction:** The GUI provides an intuitive interface for parameter adjustments, making the system user-friendly.
- **Challenges:** Some minor challenges were faced in marker precision, which were mitigated using coordinate transformations.

# 7 TESTING

## 7.1 TESTING PLAN / STRATEGY

The testing phase ensures that the image simulation system functions correctly and meets the intended objectives. The strategy involves:

### 7.1.1 Testing Types Used

- **Unit Testing** – Individual components (e.g., image loading, parameter inputs, footprint preview) are tested separately.
- **Integration Testing** – Ensures that different modules (e.g., GUI, parameter selection, simulation engine) work together correctly.
- **Functional Testing** – Verifies that all features (e.g., single/multi-point simulation, marker placement) perform as expected.
- **Performance Testing** – Checks system efficiency with different image sizes and parameter variations.
- **User Acceptance Testing (UAT)** – Ensures the system meets user requirements for camera testing and design.

### 7.1.2 Testing Environment

- **Platform:** Linux
- **Software Dependencies:** Python, PyQt5, PyQtGraph, GDAL, NumPy
- **Hardware:** System with at least 8GB RAM and multi-core processor

## 7.2 TEST RESULTS AND ANALYSIS

Multiple test cases were executed to verify the system's performance and accuracy. The following table summarizes key test cases:

### 7.2.1 Test Cases

Test ID	Test Condition	Expected Output	Actual Output	Remarks
TC-01	Load GeoTIFF image	Image loads successfully	Image loads correctly	Pass
TC-02	Enter valid camera parameters	Parameters accepted and applied	Parameters stored and used	Pass
TC-03	Enter invalid camera parameters	Error message displayed	Error handled correctly	Pass
TC-04	Single-point marker placement	Marker appears at correct lat-long	Marker placed accurately	Pass
TC-05	Multi-point marker placement	Two markers placed correctly	Markers appear as expected	Pass
TC-06	Footprint preview	Correct footprint displayed	Footprint aligns with expected region	Pass
TC-07	Run image simulation	Simulated image is generated	Image output is accurate	Pass
TC-08	Save simulated image	Image saved in directory	File saved successfully	Pass
TC-09	Clear parameters	All fields reset	Fields cleared properly	Pass
TC-10	Performance with high-res image	Processing within reasonable time	Acceptable performance	Pass (minor delay)

Table 7.1: Test Cases for Image Simulation System

## 7.3 ANALYSIS OF TEST RESULTS

- **Functionality:** All core features performed as expected, with accurate image simulation.
- **Usability:** The GUI is intuitive, with smooth user interactions.
- **Performance:** The system handles various image resolutions well, but higher-resolution images slightly increase processing time.
- **Error Handling:** Proper validation prevents invalid inputs, improving system robustness.

# 8 CONCLUSION AND SUMMARY

## 8.1 CONCLUSION AND SUMMARY

The image simulation system was developed to evaluate how a camera would capture images in an unknown environment. By incorporating camera-specific parameters such as detector height, width, field of view (FOV), focal length, and altitude, the system provides an effective way to analyze and design cameras before actual deployment.

The system meets its objectives by providing a useful tool for Image simulation and being helpful for studying images.

## 8.2 LIMITATIONS

- **Processing Time for High-Resolution Images:** When working with very large GeoTIFF images, processing can take longer than expected. Optimizations in computation can help reduce delays.
- **Limited Camera Parameter Customization:** The system currently supports predefined parameters and manual input. Future versions could allow users to import real-world camera calibration data.
- **No Real-Time 3D Visualization:** The system operates in a 2D simulation environment. Adding a 3D rendering option could improve visualization and understanding of camera perspectives.
- **Georeferencing Accuracy:** While the system accurately places markers based on latitude-longitude coordinates, minor discrepancies may arise due to projection transformations.
- **No External Data Integration:** Currently, the system processes only user-provided images. Future enhancements could support integration with external satellite data sources.