Internship at Laboratory for Electro-Optic Systems, ISRO

"AUTONOMOUS PATH PLANNING FOR ROVERS"

Submitted by:

Utkarsh Sharma 2200911540120

Under the guidance of

Mamta Satyawali
Scientist,
Laboratory For Electro-Optic Systems, ISRO

1ST JULY 2025 – 1ST SEPT 2025

DEPARTMENT OF INFORMATION TECHNOLOG

ABSTRACT

The frontier of planetary exploration increasingly relies on the development of highly autonomous rovers, demanding robust perception and planning systems capable of navigating challenging and unstructured alien environments without human intervention. This project addresses this need by detailing the implementation of a comprehensive, multi-stage processing pipeline. This framework is engineered to convert raw visual sensor data into actionable navigation intelligence, forming the brain of a self-driving robotic system.

At the core of the pipeline is a dual-pronged perception module that intelligently interprets the scene. First, it employs the state-of-the-art **Depth Anything V2** model to perform high-fidelity monocular depth estimation. This deep learning model is crucial for inferring a dense, per-pixel 3D structure of the environment from a single 2D image, a vital capability for understanding terrain geometry. Concurrently, a separate object segmentation model analyse the image to semantically classify different regions, distinguishing between safely traversable **'path'** areas and hazardous **'obstacle's** zones like rocks or steep craters.

The geometric depth data and the semantic class labels are then strategically fused and transformed. Using a standard **pinhole camera model**, this combined information is projected from the rover's first-person perspective into a top-down, 2D **bird's-eye view (BEV)**. This creates an intuitive, map-like representation where the spatial relationships of the terrain are explicitly visualized. A key innovation of this work is its ability to scale this process; it can ingest multiple images from a rover's panoramic cameras of a location and **stitch their individual BEVs into a cohesive, 360-degree environmental map**. This composite view provides complete situational awareness, eliminating blind spots and enabling more informed, holistic navigation decisions.

Finally, this unified BEV is converted into a formal **occupancy grid**. This grid discretizes the continuous environment into a grid of cells, each label as free, occupied, or unknown. This structured format is the ideal input for classical pathfinding algorithms. The **A* search algorithm** is then applied to this grid to efficiently compute an optimal, collision-free trajectory from the rover's current position to a target destination. This complete end-to-end framework represents a significant step towards enabling fully autonomous decision-making in rovers, directly aligning with ISRO’s strategic vision for deploying the next generation of self-sufficient robotic explorers on future celestial missions.

ACKNOWLEDGEMENT

It is with immense gratitude that acknowledgment is extended to the individuals whose guidance and support were instrumental in the successful completion of this internship project. This endeavour would not have been possible without their collective contributions.

Profound appreciation is expressed to the external mentor, **Ms. Mamta Satywali (Sci/Eng-SD, SSSG, LEOS, Indian Space Research Organization, Bangalore)**, for providing the opportunity to engage in this challenging research. Her expert technical counsel, insightful direction, and constructive feedback were pivotal in shaping the project's trajectory and ultimate success. Her steadfast encouragement and patience remained invaluable throughout the entire process.

Sincere thanks are extended to **Dr. Sukhendra Singh, Associate Professor in the Department of Information Technology**, for his continuous support and for inspiring the pursuit of higher standards in this work.

Gratitude is also due to **Dr. Meena Arora, Head of the Department of Information Technology**, whose consistent cooperation and administrative support created a seamless environment for research.

Finally, acknowledgment is made to **Dr. B. Manoj Kumar, Principal of JSS Academy of Technical Education, Noida**, for his commitment to fostering an academic atmosphere that supports and encourages such research-oriented endeavours.

Chapter 1

INTRODUCTION

This internship was conducted as a collaborative project, where a team of interns worked together to design an autonomous path detection system for extraterrestrial rovers. Such rovers are intended to operate in some of the most extreme and unpredictable environments in the solar system, where human control is limited due to long communication delays and environmental uncertainties. Therefore, they must be capable of making independent decisions, detecting obstacles, and identifying safe paths to ensure both mission success and equipment safety.

The project roadmap was structured into multiple phases, each building upon the previous one. Initial phases included panorama creation and image stitching, object detection and segmentation, and monocular depth estimation with 3D reconstruction. These provided the essential perception pipeline for scene understanding, upon which later stages of navigation and path planning were developed.

The contribution began from Phase 4, focusing on generating the Bird's Eye View (BEV) map. This involved projecting depth information into a top-down perspective using the pinhole camera model and addressing challenges such as misalignment between YOLO detections and depth projections. Multiple approaches were explored to improve the accuracy and consistency of the BEV map, since errors at this stage could directly affect navigation performance.

The work primarily centred on creating occupancy overlays from the BEV map. This step was critical in distinguishing between free space, obstacles, and uncertain regions, thereby providing the rover with a reliable representation of its surroundings. By combining object detection results with depth-based spatial mapping, clear occupancy maps were generated to serve as the foundation for safe navigation planning.

Finally, methods for shortest path inference were implemented using the occupancy maps. The objective was to determine safe and efficient trajectories across rugged surfaces while balancing safety and path optimality. This enabled the system to provide autonomous decision-making capabilities, allowing rovers to navigate effectively in complex terrains without constant human intervention.

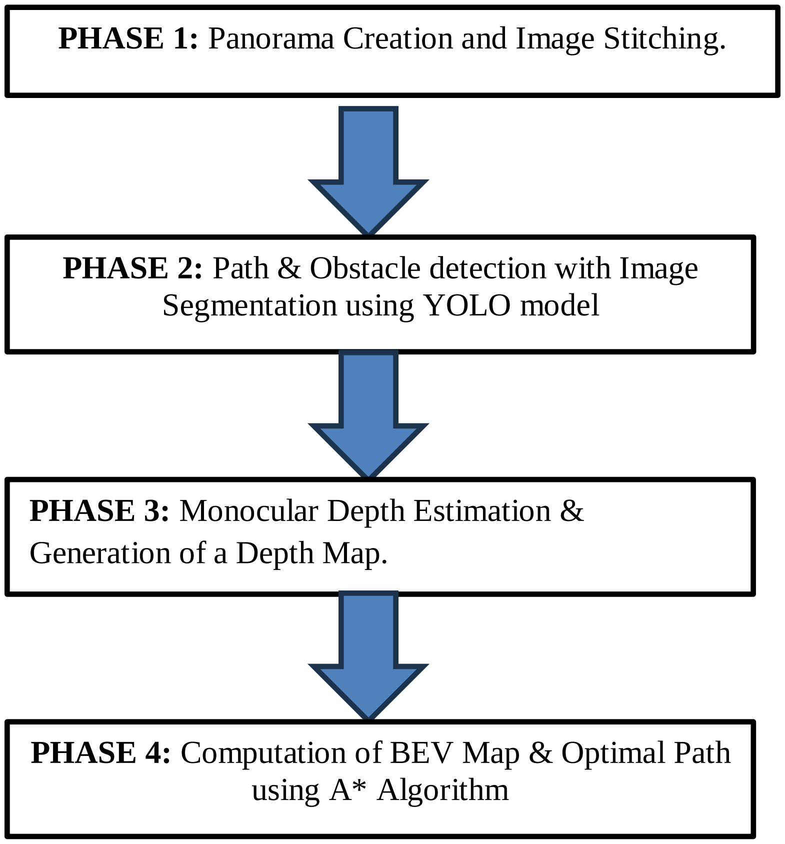
PHASE 1: Panorama Creation and Image Stitching.
PHASE 2: Path & Obstacle detection with Image Segmentation using YOLO model
PHASE 3: Monocular Depth Estimation & Generation of a Depth Map.
PHASE 4: Computation of BEV Map & Optimal Path using A* Algorithm

Fig: 1.1

The Fig 1.1 illustrates the seven major phases of the autonomous navigation pipeline.

- Phase 1 shows panorama creation and image stitching.
- Phase 2 highlights path and obstacle detection with image segmentation.
- Phase 3 demonstrates monocular depth estimation and generation of a depth map.
- Phase 4 depicts the computation of BEV map and optimal path using the A* algorithm.

Contributions to this project primarily began from Phase 4 onward, focusing on BEV computation, occupancy mapping, and optimal path detection using A* algorithm.

Project Pipeline: From BEV Map to Path Planning

The project was executed through a series of distinct, yet interconnected, phases. Each phase built upon the successes and lessons of the previous one, leading to the final robust system for indoor spatial understanding.

Phase 4.1: BEV MAP generation through Pinhole Projection

- **Action:** The project returned to the pinhole projection method. Multiple images (five per environment) with a 72∘Field of View (FOV) were used. BEV maps were generated with colour encoding directly from the pixel-level segmentation results of the YOLO model.
- **Result:** This provided more precise object placement and clearer scene interpretation.

Phase 4.2: 360° Room Visualization

- **Action:** The generated BEV maps were rotated, aligned and stitched to construct a complete 360∘ visualization of the room.
- **Result:** Full spatial coverage was achieved, providing a comprehensive representation of the environment.

Phase 4.3: Occupancy Grid Overlay

- **Action:** A grayscale occupancy overlay was introduced, marking occupied, free, and unknown cells on the map.
- **Result:** A navigationally meaningful representation of the environment was created.

Phase 4.4: Path Planning with A*

- **Action:** An A* path planning module was implemented to find optimal paths within the occupancy grid. Start and goal points were defined for the algorithm.
- **Result:** The system successfully integrated environment perception with an autonomous navigation capability.

Chapter 2

LITERATURE SURVEY

Research in computer vision, robotics, and autonomous navigation has advanced rapidly, providing many established methods that form the basis of this project. This chapter reviews key works on object detection, depth estimation, map construction, and path planning, highlighting concepts adapted for the system pipeline.

2.1 Path and Obstacle Detection – YOLO

Paper: *You Only Look Once: Unified, Real-Time Object Detection* [1]

The advantage of YOLO is its speed and ability to process frames in real time without sacrificing much accuracy. For autonomous navigation, especially in unknown environments, fast and reliable obstacle detection is crucial. This is why YOLO and its successors have become standard in robotics pipelines where both efficiency and accuracy are required. In this project, YOLO forms the base for detecting possible navigable paths and obstacles from the stitched panoramic image input.

2.2 Bird's Eye View (BEV) Computation

Paper: *Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D* [3]

Bird's-eye view (BEV) maps give a top-down view of the scene, which is useful for navigation. The Lift-Splat-Shoot (LSS) method creates BEV by lifting image pixels into 3D using depth, then projecting them onto a 2D plane. This makes path planning easier, like reading a map. In this project, BEV maps are used to build occupancy grids and plan safe routes.

2.3 Occupancy Map Construction

Paper: *Using Occupancy Grids for Mobile Robot Perception and Navigation* [4]

Occupancy grids divide the environment into small cells, each showing if an area is free, occupied, or unknown. This makes complex 3D spaces easier to handle in 2D for navigation. In this project, BEV maps are turned into grayscale occupancy grids, where dark areas show obstacles and light areas show free paths. These grids are then used for pathfinding.

2.4 Optimal Path Detection – A*

Paper: *A Formal Basis for the Heuristic Determination of Minimum Cost Paths* [5]

A* is a path planning algorithm that finds the shortest and safest route. It uses a cost function **$f(n) = g(n) + h(n)$**, where $g(n)$ is the cost so far and $h(n)$ is the estimated cost to the goal. If the estimate is accurate, A* guarantees the optimal path. In this project, it is applied on the occupancy grid to plan safe navigation around obstacles.

Chapter 3
METHODOLOGY

This chapter presents a step-by-step account of the implementation process. For each stage, the chosen methodologies, algorithms, and tools are described along with the reasons for their selection. It also outlines model comparisons, evaluation methods, and practical solutions to challenges encountered during implementation.

The overall project was divided into six phases. Since my contribution focused on Phases 5 and 6, the following sections of Chapter 3 describe these parts in detail. To avoid confusion with report chapter numbering, each section is renumbered sequentially within this chapter, while also noting the original project phase in parentheses.

- Section 3.1: Pinhole Projection (corresponding to Project Phase 4.1)
- Section 3.2: 360° Room Visualization (corresponding to Project Phase 4.2)
- Section 3.3: Occupancy Grid Overlay (corresponding to Project Phase 4.3)
- Section 3.4: Path Planning with A (corresponding to Project Phase 4.4)

3.1: **<u>Pinhole Projection</u>**

3.1.1 - Depth Estimation per Camera View

- Camera (with ~72° FOV) captures images of its local sector around the robot.
- Using **DepthAnythingV2**, you compute a dense depth map:
  - Each pixel (u, v) → depth Z (distance along the optical axis).
- This gives a 2.5D perception for **one pinhole camera view**.

3.1.2 - Pinhole Camera Model Back-Projection

- You assume a standard pinhole projection with intrinsics (fx, fy, cx, cy).
- For every pixel (u, v) with depth Z, 3D coordinates are reconstructed:

$$X = \frac{(u - c_x) \cdot Z}{f_x}, \quad Y = \frac{(v - c_y) \cdot Z}{f_y}, \quad Z = \text{depth}(u, v)$$

- Here:
  - X = horizontal axis (left-right).
  - Z = forward axis (depth along ground plane).
  - Y = vertical axis (height, often ignored in BEV).

3.1.3 - Projection onto BEV Plane

- Instead of using bounding boxes only, now you **paint BEV pixel-wise**:
  - Each (X, Z) point is mapped into a 2D BEV grid:

$$bx = \frac{X}{\text{resolution}} + \frac{\text{bev\_width}}{2}, \quad bz = \frac{Z}{\text{resolution}}$$

- Where resolution = meters per BEV cell (e.g., 0.02 m/pixel).
- This creates a **dense occupancy projection** (no longer sparse bounding-box rectangles).

3.1.4 - Pixel-Wise Colouring with YOLO Integration

- **Previously:** obstacles vs path regions were filled only at **bounding-box level**.
- **Now:** YOLO provides masks → **pixel-level precision**:
  - For each pixel classified by YOLO:
    - **Path / road pixels** → Green.
    - **Obstacle pixels** → Red (or blue).
- This prevents "flood-filling" large bounding boxes, keeping obstacles sharper.

3.1.5 - Multi-Camera Surround Coverage

- Since one pinhole camera has only ~72° FOV, you deploy **multiple cameras (e.g., 5)** around the robot.
- For each camera view:
  1. Depth estimation → 3D back-projection → BEV slice.
  2. YOLO detections mask + pixel-wise colouring → path/obstacle map.
  3. Save individual BEV slice (bev_1.jpg, bev_2.jpg, …).
- End result: **5 BEV maps**, each covering ~72° → combined they give **360° coverage**.

3.2: <u>**360° Room Visualization**</u>

To combine BEV maps generated from multiple cameras into a unified **360° spatial representation** of the room, enabling complete environment coverage for navigation and analysis, key steps involved are:

3.2.1 - Camera Pose Calibration

- Each camera has a **known extrinsic pose** relative to the robot:
  - Translation: (tx, ty, tz)
  - Rotation (yaw, pitch, roll)
- Since cameras are mounted around the robot, the most relevant is **yaw rotation** (horizontal orientation).
- Example for 5 cameras covering 360°:
  - Camera 1 → 0∘
  - Camera 2 → 72∘
  - Camera 3 → 144∘
  - Camera 4 → 216∘
  - Camera 5 → 288∘

3.2.2 - BEV Slice Generation (Local Frame)

- Each camera produces a **BEV slice** in its **local forward-facing coordinate system**.

- In local coordinates, the forward axis is **+Z**, sideways axis is **+X**.
- Each pixel (u,v)(u,v) is projected to BEV (x, z) using the pinhole model:

$$x = \frac{(u - c_x)}{f_x} \cdot d, \quad z = d$$

where d = depth of pixel

3.2.3 - Coordinate Transformation to Global Frame

- To place each BEV slice correctly, a **2D rigid body transformation** is applied.
- For each local point (x,z)(x,z):

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

- Where:
  - (X, Y) → global BEV coordinates
  - θ → yaw angle of camera
  - (tx, ty) → translation offset from robot centre (often zero if all cameras are equidistant)

This ensures each BEV slice is **rotated & aligned** into a common global map.

3.2.4 - Image Stitching

- Once all BEV slices are transformed into global coordinates:
  - They are placed on a **shared canvas**.
  - Overlapping regions are merged using:
    - **Max-pooling of obstacle probability** (keep obstacle if any slice detects it).
    - **Confidence-weighted averaging** for path regions.
- This creates a **seamless stitched BEV map**.

3.2.5 - 360° Coverage Validation

- After stitching, the final BEV map forms a **full circular coverage** around the robot.
- The **robot lies at the centre**, and the environment is represented in all directions.
- Grid lines (1m major, 0.5m minor) are overlaid for scale reference.

3.3 - Occupancy Grid Overlay

To transform the stitched BEV map into a **grayscale occupancy grid**, where each cell in the grid encodes whether it is **occupied**, **free**, or **unknown**. This makes the map directly usable by path-planning algorithms (A*), key steps involved are:

3.3.1 - Define Grid Resolution & Size

- Decide **map resolution** r (meters per cell). Example: r=0.05 m/cell (5 cm).
- Choose **map dimensions**:
    - Width W and Height H in cells.
    - Centre of grid corresponds to robot position.

So each cell (i, j)maps to global coordinates:

$$X = (i - \frac{W}{2}) \cdot r, \quad Y = j \cdot r$$

3.3.2 - Classify Pixels in BEV Map

From the stitched BEV image:

- **Obstacle pixels** (e.g., red) → Occupied.
- **Path pixels** (e.g., green) → Free.
- **Unknown regions** (black/empty) → Unknown.

This can be done by checking pixel colour or class mask.

3.3.3 - Cell Occupancy Assignment

For each grid cell G(i, j):

1. Collect all BEV pixels that fall inside the cell's spatial bounds.
2. Count:
    - N occ: number of obstacle pixels.
    - N free: number of path pixels.
    - N unk: number of unknown pixels.
3. Assign cell state based on majority rule (or probability threshold):
    - If N occ / N total > T occ = occupied
    - Else if N free / N total > T free = free
    - Else = Unknown

Typical Thresholds: T occ = 0.6, T free = 0.6

3.3.4 - Grayscale Encoding

Encode each state in **grayscale intensity**:

- **Occupied** → 0 (black).
- **Free** → 255 (white).
- **Unknown** → 127 (gray).

So, the occupancy grid is:

$$O(i,j) = \begin{cases} 0 & \text{if occupied} \\ 255 & \text{if free} \\ 127 & \text{if unknown} \end{cases}$$

3.3.5 - Post-Processing

- Apply **morphological closing** → remove small gaps in obstacles.
- Apply **dilation** → expand obstacles by robot radius, to ensure safety margins.
- Smooth free-space regions for cleaner path planning.

3.4: **Path Planning with A***

The **A*** algorithm is used to find the shortest and safest path from a **start point** to a **goal point**. It is applied on an **occupancy grid map**, where:

- **White/empty cells** = free space (can move).
- **Black/filled cells** = obstacles (cannot move).

Core Idea

A* is like a smart version of Google Maps for robots:

- It uses **Dijkstra's Algorithm** (always finds the shortest path).
- Plus **Greedy Search** (uses a guess to make the search faster).
- It decides the best next step using:

$$f(n) = g(n) + h(n)$$

Where:

- **g(n)** = distance already travelled from start to current point.
- **h(n)** = estimated distance from current point to goal.
- **f(n)** = total estimated path cost.

How A* Works (Steps)

1. **Start** → Put the starting cell in a list called **Open Set**.
2. **Pick Best Cell** → Choose the cell with the **lowest f(n)** (shortest predicted path).
3. **Check Goal** → If this cell is the goal → path found .
4. **Explore Neighbours** → Look at nearby cells (up, down, left, right, diagonals).
   - Ignore if it's an obstacle.
   - Calculate new cost (**g**) for moving there.
   - Update if this route is shorter.
5. **Repeat** until the goal is reached or no path exists.

Chapter 4

RESULTS

This report summarizes the results of a multi-phase vision pipeline designed for autonomous navigation. The system integrates **image stitching**, **semantic segmentation**, **depth estimation**, and **pathfinding** to generate a safe, traversable route in an indoor environment.

4.1 <u>Semantic Segmentation for Path and Obstacle Detection</u>

The pipeline applies a YOLO-based semantic segmentation model to classify raw images into **path (traversable ground)**and **obstacles**, with confidence scores indicating accuracy. This step is essential for hazard detection before path planning.

- Fig: 4.1.1, 4.1.2: A cabinet as obstacle and path detected; confidence 0.67 (obstacle) and 0.46 (path) with the depth map.

- Fig: 4.1.3, 4.1.4: Wider view with multiple obstacles; clear path segmented with high confidence of 0.90 with the depth map.

- Fig: 4.1.5, 4.1.6: Chair and cabinet identified as obstacles; path confidence 0.39 in a cluttered scene with the depth map.

- Fig: 4.1.7, 4.1.8: Path detected at 0.62 confidence; obstacles (chair, door) also correctly segmented with the depth map.

- Fig: 4.1.9, 4.1.10: Complex scene with multiple obstacles; path confidently segmented at 0.89 with the depth map.

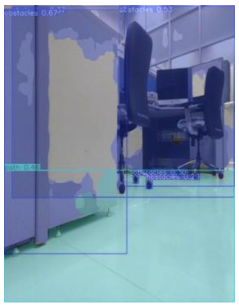
Fig: 4.1.1(Segmentation Overlay(1))

Fig: 4.1.2(Depth Map(1))

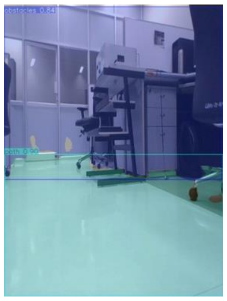
Fig: 4.1.3(Segmentation Overlay(2))

Fig: 4.1.4(Depth Map(2))

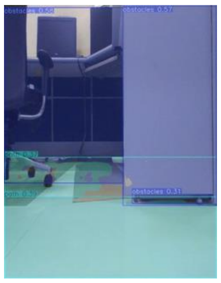
Fig: 4.1.5(Segmentation Overlay(3))

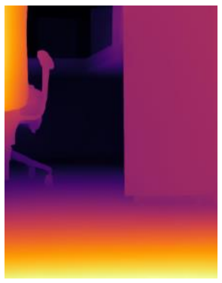
Fig: 4.1.6(Depth Map(3))

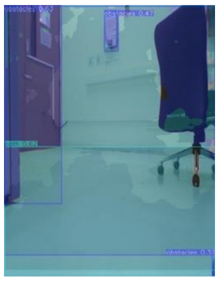
Fig: 4.1.7(Segmentation Overlay(4))

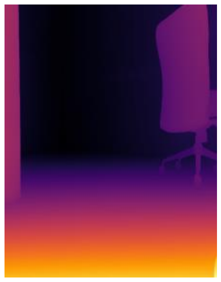
Fig: 4.1.8(Depth Map(4))

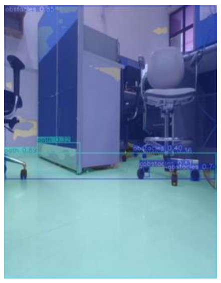
Fig: 4.1.9(Segmentation Overlay(5))

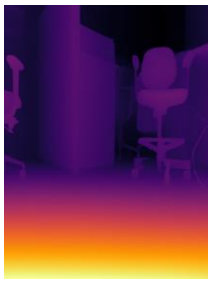
Fig: 4.1.10(Depth Map(5))

4.2 <u>**BEV MAP (TOP – DOWN VIEW)**</u>

The Bird’s-Eye-View (BEV) maps are generated by projecting semantic segmentation outputs into a top-down view using pinhole projection, making them suitable for path planning. In these maps, **green** represents traversable paths, **red** indicates obstacles, and **blue** denotes unmapped regions outside the sensor’s view. a comprehensive representation of the scene's layout from above.

- **Fig: 4.2.1 (BEV Map 1)**: This map shows a large obstacle on the left, corresponding to a cabinet or wall, with a small obstacle in the foreground. The red area accurately reflects the segmentation from the raw image.
- **Fig: 4.2.2 (BEV Map 2)**: This view shows a complex distribution of obstacles, including the legs of a chair and part of a desk. The fragmented red areas accurately represent these smaller obstacles in a top-down view.
- **Fig: 4.2.3 (BEV Map 3)**: This map captures a small, distinct obstacle in the foreground. The red area represents the chair and cabinet legs that were segmented in the original image.
- **Fig: 4.2.4 (BEV Map 4)**: This map primarily shows a large, clear green area, indicating a wide, unobstructed path. The small red area on the far right represents a minor obstacle at the edge of the view.
- **Fig: 4.2.5 (BEV Map 5)**: This map features multiple obstacles, including the legs of chairs and a cabinet, shown as two distinct red areas. The **contour lines** within the green area provide a visual representation of the ground surface or topology.

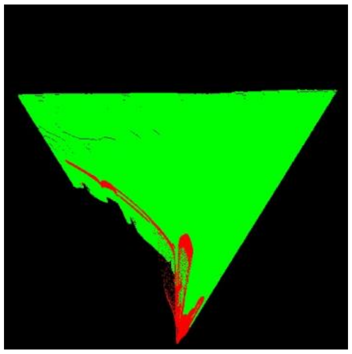
Fig: 4.2.1

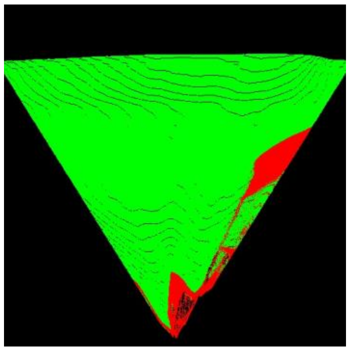
Fig: 4.2.2

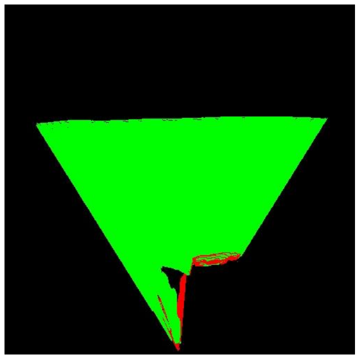
Fig: 4.2.3

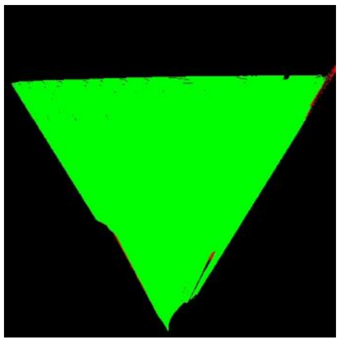
Fig: 4.2.4

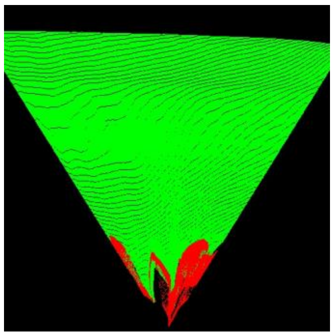
Fig: 4.2.5

4.3 <u>360° Bird's-Eye-View (BEV) Map Stitching</u>

Figure 4.3.1 illustrates the stitched 360° BEV map obtained by combining individual BEV maps from multiple viewpoints into a unified top-down representation of the environment. This stitched map serves as a comprehensive occupancy grid for navigation and path planning. The colour coding in the map is as follows:

- **Green regions**: Traversable paths.
- **Red regions**: Detected obstacles.
- **Blue regions**: Unmapped or unknown areas lying outside the collective field of view of the cameras.

This integrated representation provides complete environmental awareness and serves as the final input for the path planning algorithm, ensuring safe and efficient trajectory computation.

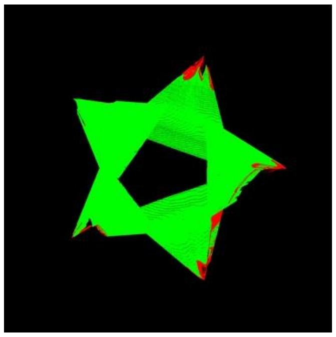

Fig: 4.3.1 (Stitched 360° BEV MAP)

4.4 <u>Occupancy Grid Generation from 360° BEV Map</u>

The image on the left depicts a **stitched 360° Bird's-Eye-View (BEV) map**, which combines multiple individual BEV maps into a single, comprehensive representation of the environment. The different colours represent distinct classifications: **green** for traversable path, **red** for obstacles, and **blue** for unmapped areas. This stitched map provides a unified perspective for the next stage of the pipeline.

The image on the right shows the **occupancy grid** generated from the stitched BEV map. The process involves converting the multi - coloured BEV map into a **grayscale representation**. In this new format, **white** areas represent free space, **black** areas represent obstacles, and **gray** signifies unknown space. This conversion simplifies the environment into a binary grid, making it an ideal input for path planning algorithms like A*, which can then efficiently identify the shortest, most optimal path by avoiding the black-occupied pixels.

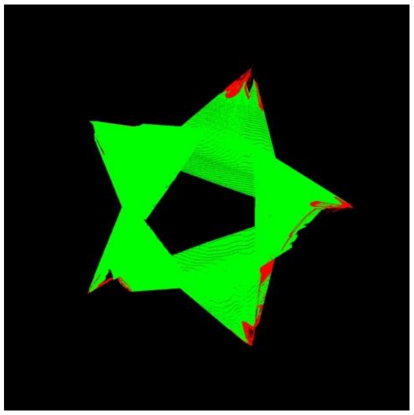

Fig: 4.4.1(Coloured BEV MAP as INPUT)

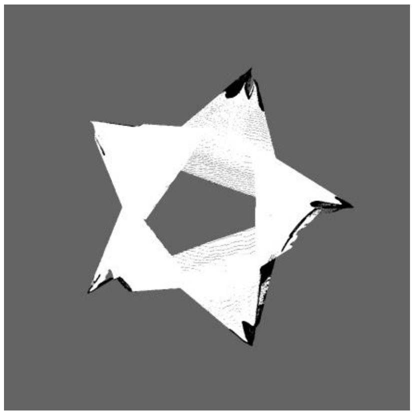

Fig: 4.4.2(Gray Scale Occupancy Grid)

4.5 A* Path Planning on Occupancy Grid

Figure 4.5.1 shows the implementation of the A* algorithm on the grayscale occupancy grid for determining the optimal shortest path. The algorithm computes a collision-free trajectory by simultaneously considering the actual distance traversed from the start point and the estimated cost to reach the goal point.

The procedure can be summarized as follows:

- **Input:** The grayscale occupancy grid serves as the navigation map.
- **Start and End Points:** The start point is defined at pixel coordinates **(1214, 696)**, while the end point is defined at **(421, 688)**.
- **Path Calculation:** The A* algorithm determines the shortest feasible path between the two points by avoiding the black pixels, which represent obstacles, and traversing through the white pixels, which represent free space.

The resulting path, visualized as a blue line superimposed on the occupancy grid, confirms the algorithm's capability to generate an efficient and collision-free route.

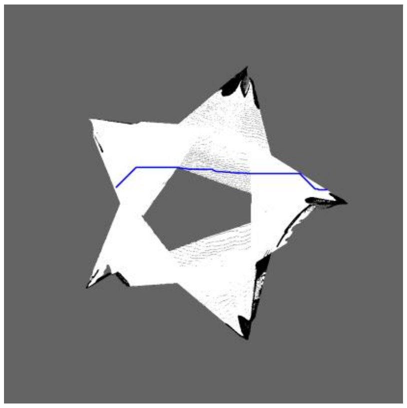

Fig: 4.5.1 (Blue Coloured Path on Gray Scale Occupancy Grid)

Chapter 5

CHALLENGES

- **Misalignment:** The edges of the green (path) and red (obstacle) segments do not align correctly. This creates gaps and overlaps, making the map appear disjointed. For example, some green areas end abruptly against a blue or black background, rather than connecting with the adjacent mapped area. This is a common issue with **image stitching** and **monocular depth estimation** errors, as any inaccuracies in the perspective transformation will become more apparent when views are combined.

- **Fragmentation:** The map is not a continuous, single representation of the environment. The individual BEV projections are visible as separate star-shaped segments, with large **unmapped areas** (blue and black) in between them. This fragmentation reduces the utility of the map for navigation, as a path planning algorithm cannot see a clear, continuous path or obstacle field.

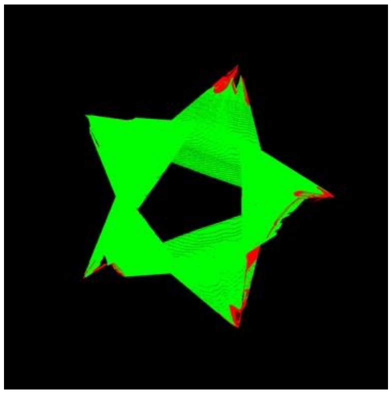

Fig: 5.1 Coloured BEV MAP (top-down view)

- **Incomplete Information:** Large portions of the map are classified as unknown (blue and black), which means there is no data to support navigation in those areas. This limits the robot's ability to plan a full path and navigate safely. A robust BEV map should cover the entire 360-degree environment with as little unknown space as possible.

- **Visual Noise:** The noisy, grainy texture within the blue and gray areas could indicate errors in the depth estimation or projection process, further reducing the reliability of the map.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.

[2] L. Yang, et al., “Depth Anything V2: Consistent Monocular Depth Estimation across Diverse Domains,” *arXiv preprint arXiv:2406.09414*, 2024.

[3] J. Philion and S. Fidler, “Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 194–210.

[4] A. Elfes, “Using Occupancy Grids for Mobile Robot Perception and Navigation,” *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[5] P. Hart, N. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.

[6] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3D: A Modern Library for 3D Data Processing,” *arXiv preprint arXiv:1801.09847*, 2018.