Implementation of An Efficient Coverage Method for Irregularly Shaped Terrains with Lunar Terrain

Samara Holmes

Institute for Experiential Robotics
Khoury College of Computer Science
Northeastern University
Boston, MA

Outline

- Introduction
- Problem Description
- Related Work
- Methodology
  - Crater Detection
  - Coverage Path Planning
- Results
  - Crater Detection
  - Coverage Path Planning
- Discussion
- References

Introduction

Coverage path planning (CPP) presents numerous challenges when applied to environments with complex conditions.

Irregular terrain, varying surface features, and the presence of obstacles often make it difficult to create paths that ensure complete and efficient coverage.

Developing robust detection models that leverage image processing techniques can enable high-resolution identification of lunar craters from satellite images.

Problem Description

Lunar Crater Detection:

Developing a reliable method to detect and identify craters on the Moon's surface using high-resolution TIFF images

Robotic Path Planning for Coverage:

"An ideal coverage path should enable a robot to thoroughly cover the area of interest while minimizing revisits to previously covered zones" [2]

Related Work

LunaNet has been proposed as a system that uses CNN to detect craters [8]. Fig. 1 shows the image processing that LunaNet uses to predict craters.

It's input is a grayscale image, then a prediction mask is outputted by the neural network, a binary threshold is applied to the prediction, the contours are fitted with ellipses to form circular crater detections, then the contours are applied to the original grayscale image.

[9] uses the Segment Anything Model (SAM) from META AI to obtain segmentation masks from lunar surface images. The presented crater detection algorithm (CDA) does a similar process to LunaNet, where a mask is fitted from the input image and the craters are then drawn and applied (Fig. 2).

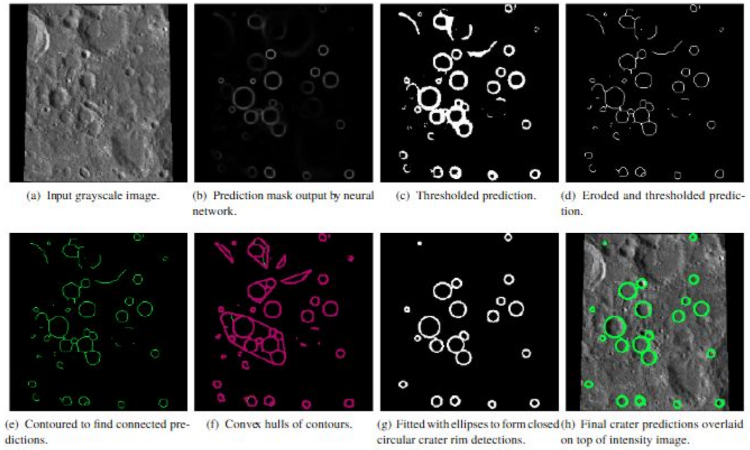


Fig 1. LunaNet workflow

Fig 2. Segment Anything Model

Methodology - Crater Detection

Edge detection:

1. Apply a Sobel filter along the X and Y axes
2. Use the gradient magnitude to normalize the image
3. Apply a binary threshold
4. Use a morphological filter to clean the image up
5. Use skimage.measure.regionprops to identify crater regions

Relative depth:

1. Apply DepthAnythingV2 model to crop from lunar data

YOLO:

1. Apply pre-trained YOLO model to crop from lunar data
2. Get the bounding box information from the detected craters

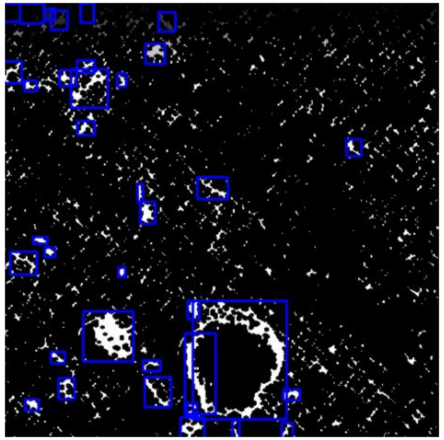


Fig 1. Edge detection

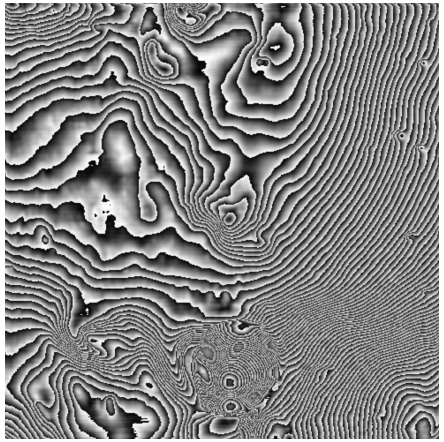


Fig 2. DepthAnythingV2

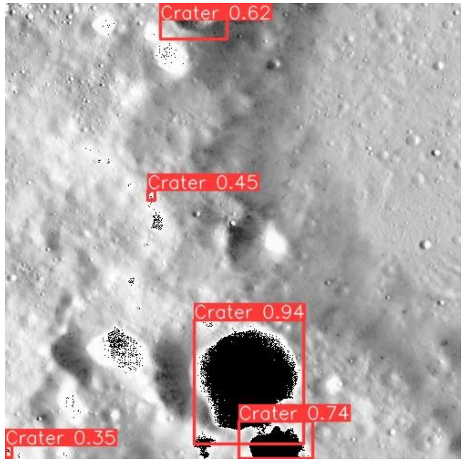


Fig 3. YOLO

Methodology - Coverage Path Planning

Using the algorithm presented in [2], we identify the following steps:

1. Create a grid mapping of the target area using square cells
2. Conduct cell classification, which will be discussed more below
3. Identify Hamiltonian cycles in the trunk graph using the STC algorithm
4. Identify Hamiltonian paths in the branch graph
5. Combine the cycles and the paths

Algorithm 1: Low-repetitive coverage path planner

```
Input: 𝒯, 𝒮, ℋ, 𝒞
Output: A low-repetitive path that traverses all path points
k ← number of Hamiltonian structures in ℋ
ℰ ← ∅ ;        // rules of tree formation
do
    for j ← 1 to k do
        if B'_j ∉ 𝒯 and l(B'_j) fits (9) then
            𝒯 ← 𝒯 ∪ B'_j
            E_j ← extra rules due to the link
while 𝒯 has been augmented with new parts;
ST ← an MST formed by PRIM in the mega-cell grid graph, prioritizing all rules in ℰ;
while any E_x in ℰ mandates using cells that are already utilized in the MST. do
    𝒯 ← 𝒯 − B'_x
    𝒮 ← cells in B'_x
HC_t ← circular path in 𝒯 obtained through ST
HC_t ← connect all Hamiltonian paths complying with PRIM to HC_t according to (8)
L ← ℒ(HC_t)
m ← number of remained path points
for n ← 1 to m do
    Δ(L, r_{s_n}) ← argmin (‖Δ(L, r_{s_n})‖ − ‖L‖)
    L ← Δ(L, r_{s_n})
return L
```

Fig 1. CPP algorithm

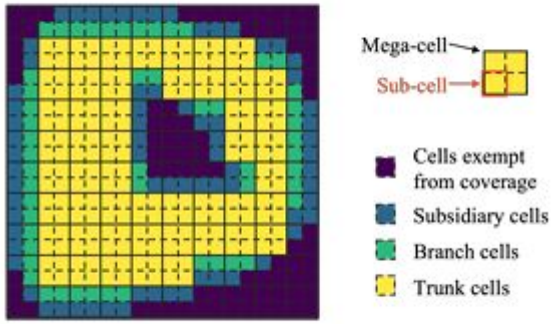


Fig 2. Cell classification

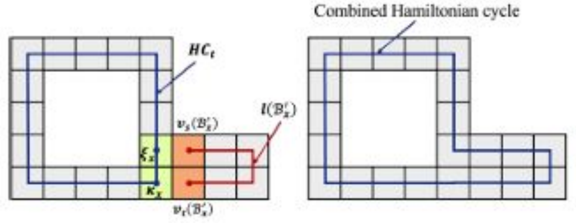
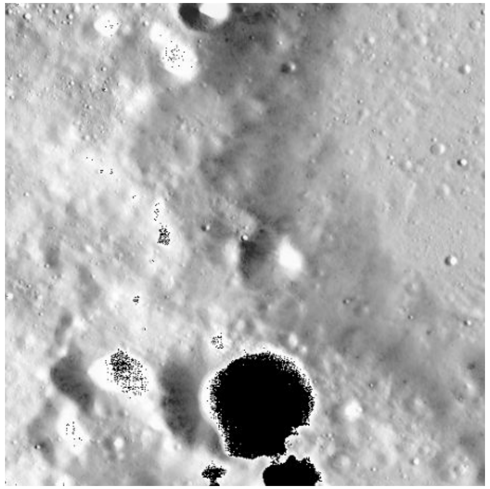
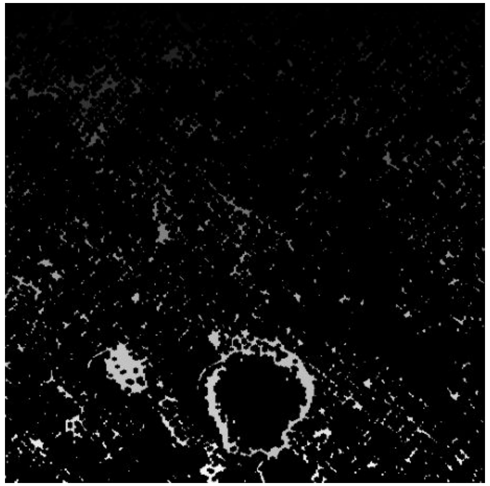
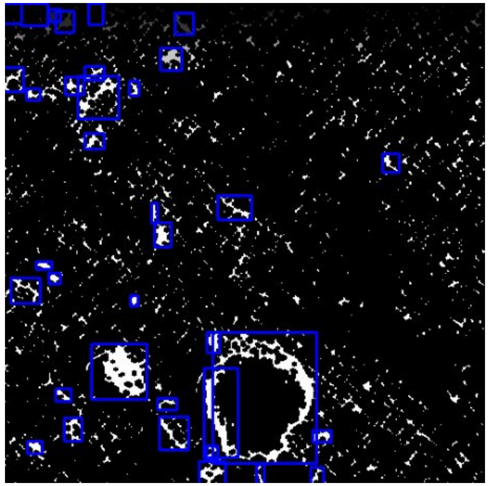


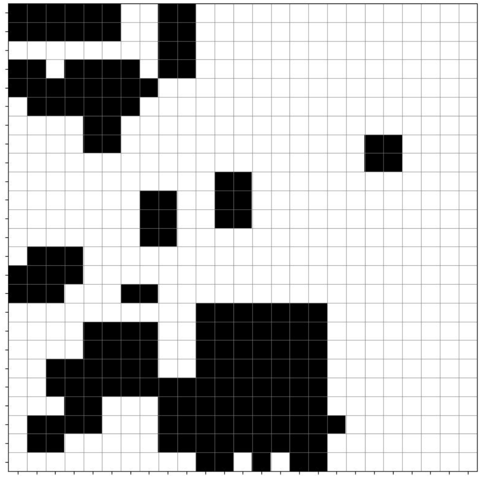
Fig 3. Hamiltonian cycle extension

Results - Crater Detection









Random Crop → Edge Detection → Regions → Grid Map

Results - Coverage Path Planning

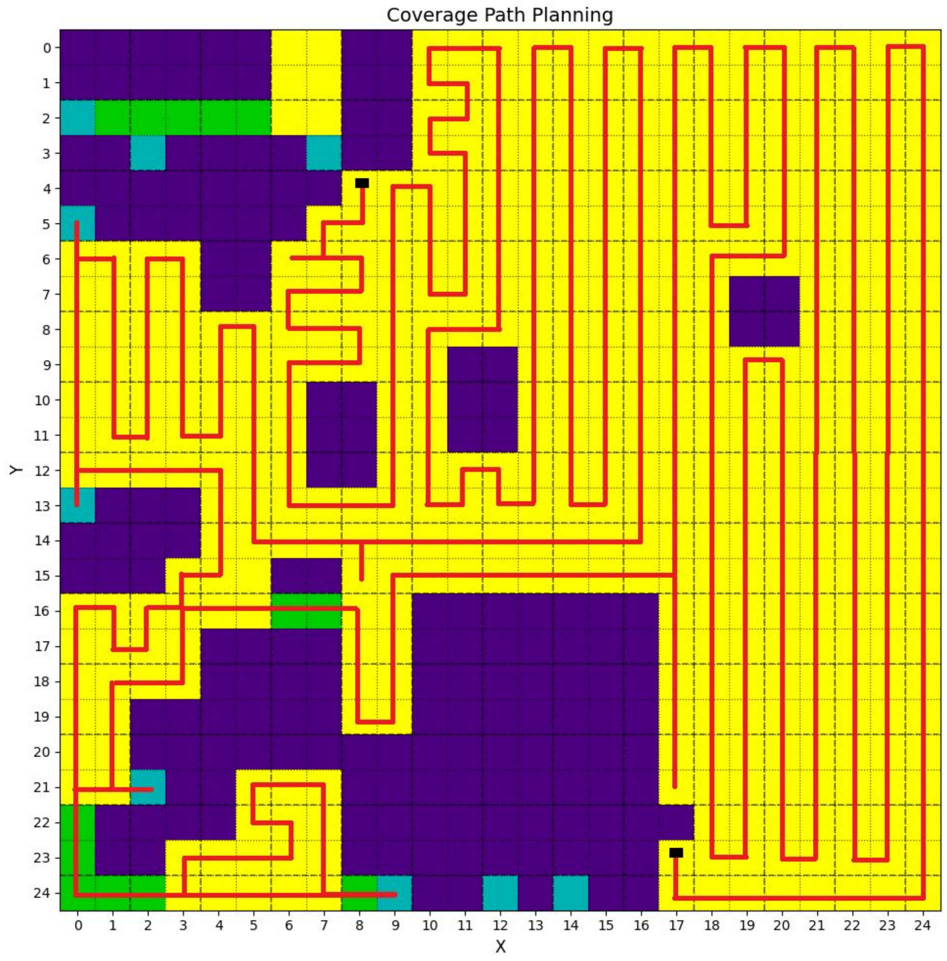
In our simulation, we create a 25m x 25m grid.

The number of cells exempt from coverage is 170 (27.2%).

The number of cells that are inaccessible to the robot is 16.

The number of cells covered is 439. We determine that the maximum amount of coverage that the robot can do is 96.48%, which is 439 out of the 455 cells.

The path length is 460 meters for a 625m^2 grid.



Discussion - Crater Detection

We found that using the 3 different crater detection methods (edge detection, relative depth, and YOLO) generated 3 significantly different grid maps.

We found that with the grid map created using Sobel edge detection provided the easiest lunar crater coverage path planning.

The DepthAnythingV2 grid result provided too little crater detections, while the YOLO model provided better results.

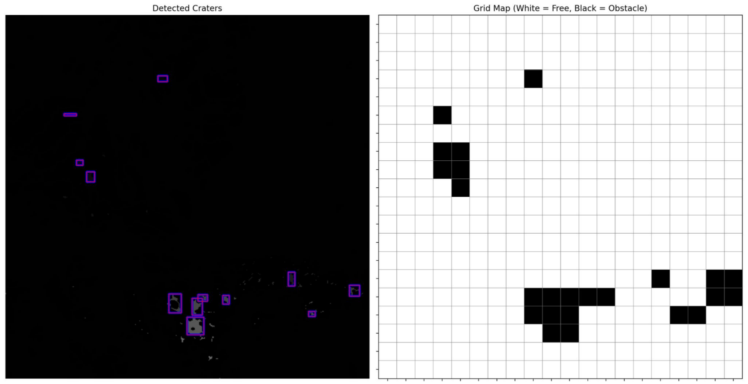


Fig 1. DepthAnythingV2 grid result

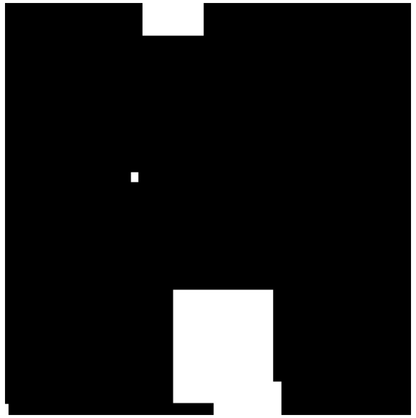


Fig 2. YOLO grid result

Discussion - Coverage Path Planning

It is difficult to compare the resulting paths (see Fig. 1) used in [2] and our implementation because the test areas are generated differently.

Not only are the test areas different, but they are also of different sizes. In [2], these use four 6m x 6m test areas and one 20m x 20m that is used for a multi-robot system test.

Path length is used only for the 6m x 6m test areas. While, coverage rate and path length are used for the multi-robot system test.

It would be interesting to expand our implementation to multi-robot systems.

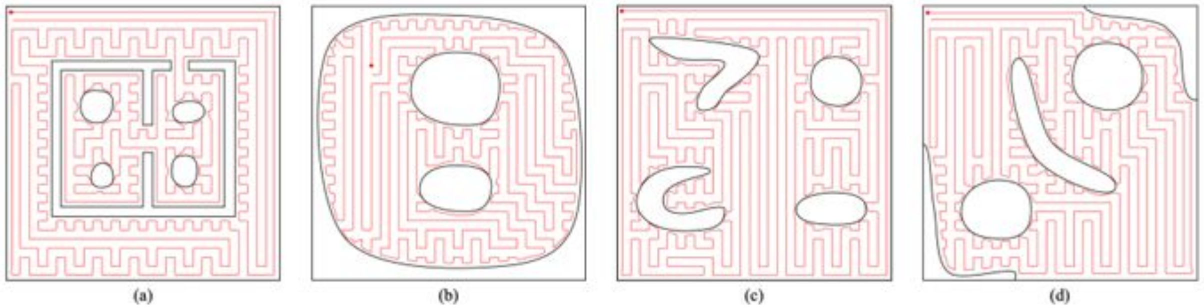


Fig 1. Single-robot coverage paths from [2]

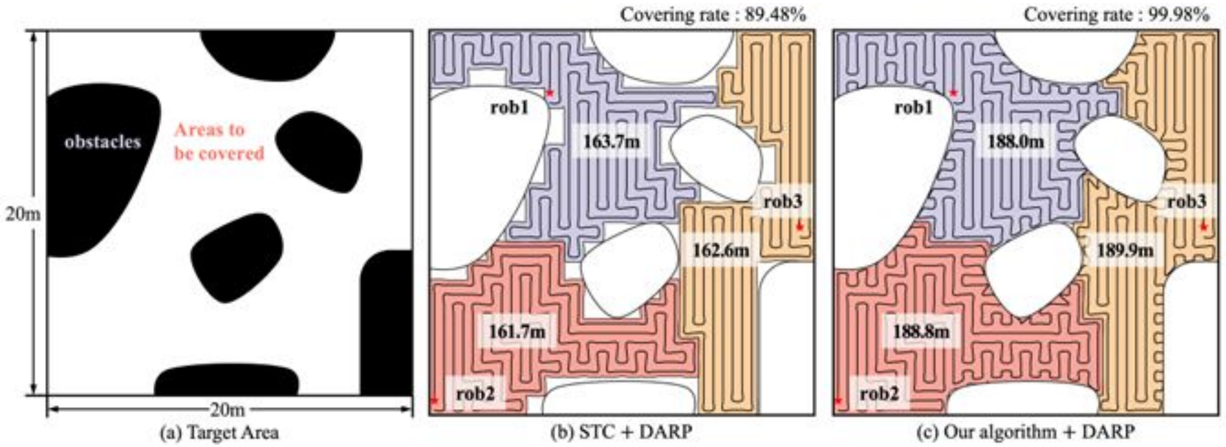


Fig 2. Multi-robot coverage paths from [2]

References

[2] Y. Tang, Q. Wu, C. Zhu, and L. Chen, “An Efficient Coverage Method for Irregularly Shaped Terrains,” in 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), ISSN: 2153-0866, Oct. 2024, pp. 7641–7647. DOI: 10.1109/IROS58592.2024. 10801856. [Online]. Available: https://ieeexplore.ieee.org/document/10801856 (visited on 04/05/2025).

[8] L. M. Downes, T. J. Steiner, and J. P. How, “Deep Learning Crater Detection for Lunar Terrain Relative Navigation,” en, Other repository, Jan. 2020. [Online]. Available: https://dspace.mit.edu/handle /1721.1/137175.2 (visited on 04/05/2025).

[9] I. Giannakis, A. Bhardwaj, L. Sam, and G. Leontidis, Deep learning universal crater detection using Segment Anything Model (SAM), 2023. DOI: 10.48550/ARXIV.2304.07764. [Online]. Available: https://arxiv.org/abs/2304.07764 (visited on 04/08/2025).