Urban Mission Planning Challenge

February 25, 2026

1 Problem

Given a high-resolution satellite image and two pixel coordinates, output a path between them that travels along roads.

1.1 Input

Each test case provides:

- **Satellite image:** RGB TIFF (2048×2048 to 8192×8192 pixels)
- **Start coordinate:** $[x_{\text{start}}, y_{\text{start}}]$
- **Goal coordinate:** $[x_{\text{goal}}, y_{\text{goal}}]$

Coordinate system:

- Origin at top-left: $(0, 0)$
- x increases right, y increases down
- Integer pixel coordinates

Example: `test_001_mission.json`

```
{
  "id": "test_001",
  "image": "test_001.tif",
  "start": [523, 1847],
  "goal": [3201, 456]
}
```

1.2 Output

Submit a JSON file with a sequence of waypoints:

```
{
  "id": "test_001",
  "path": [[x1, y1], [x2, y2], ..., [xn, yn]]
}
```

Requirements:

- Minimum 2 waypoints
- All coordinates within image bounds
- Path segments should stay on roads

2 Dataset

Directory	Files	Purpose
`reference/sats/`	50 TIFFs	Training images
`reference/maps/`	50 masks	Ground truth roads (for training)
`reference/solutions/`	50 JSONs	Example valid paths
`test/sats/`	10 TIFFs	Evaluation images
`organizers/`	hidden	Private evaluation data

Note: Road masks for test images are *not provided*. You must extract roads from satellite imagery.

3 Scoring

$$\text{Score} = 1000 - \text{PathLength} - 50 \times \text{Violations}$$

PathLength: Euclidean distance in pixels

$$\sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

Violations: Number of segments that:

- Pass through non-road pixels
- Go out of image bounds
- Have invalid format

Segment checking: Each line segment is rasterized and every pixel is checked against the ground truth road mask.

Path type	Length	Violations	Score
Optimal	1200	0	−200
Short but off-road	800	5	−50
Long but valid	2500	0	−1500
Invalid	1000	20	−1000

Table 1: Higher scores are better. Goal: shortest valid path with zero violations.

4 Deliverables

4.1 Required Files

1. **Solution JSONs:** One file for all the test images combined
   - Containing {`"id": "...", "path": [[x,y], ...]`}
2. **Code:** Your implementation
   - Python script or Jupyter notebook
   - Must be runnable in Google Colab
   - Include `requirements.txt` with dependencies
3. **README.txt:** Execution instructions
   - How to install dependencies
   - How to run your code
   - Any required API keys
   - Expected runtime

4.2 Submission Format

To be Announced

5 Rules

- Use any publicly available models or libraries
- Training on reference data is allowed
- Manual annotation of test images is **not** allowed
- External road databases (OpenStreetMap, etc.) are **not** allowed
- Code must be reproducible

6 Evaluation Process

1. Load submitted JSON files
2. For each path, compute PathLength
3. Rasterize each segment and check against ground truth mask
4. Count violations (off-road pixels)
5. Calculate score: $1000 - \text{PathLength} - 50 \times \text{Violations}$
6. Sum scores across all 10 test images
7. Rank submissions by total score

7 Getting Started

1. Clone repository: `git clone https://github.com/SamyakSS83/ump_data`
2. Explore reference samples in `reference/`
3. Develop a solution using reference data
4. Package and submit on the evaluation (test) data

8 FAQ

Q: What if start/goal are not on a road?
A: Snap them to the nearest road pixel (within reasonable distance, e.g., 50px).

Q: Can I use external road data?
A: No. Extract roads from the satellite images only.

Q: What if there's no valid path?
A: Submit the best path you can find. It will incur violations but is better than nothing.

Q: Can I downsample images?
A: Yes, but you may lose road details.

Q: How is "road" defined?
A: Any pixel marked as road (value = 255) in the ground truth binary mask.

9 Contact

- GitHub: `https://github.com/SamyakSS83/ump_data`
- Issues: `https://github.com/SamyakSS83/ump_data/issues`