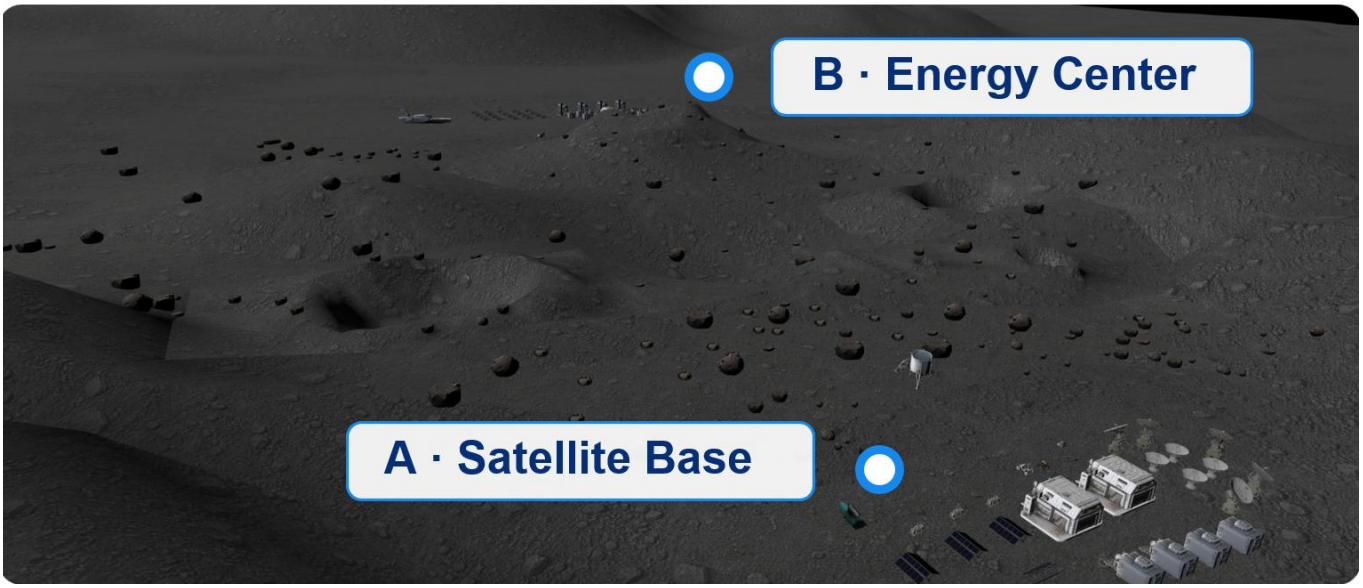
COMPETITION USER MANUAL

Autonomous Lunar Rover Digital Twin

Visual Topological SLAM and Teach-and-Repeat Navigation

Competition Number	**CPWC2026-0025**



Satellite Base A and Energy Center B in the UC-win/Road lunar simulation

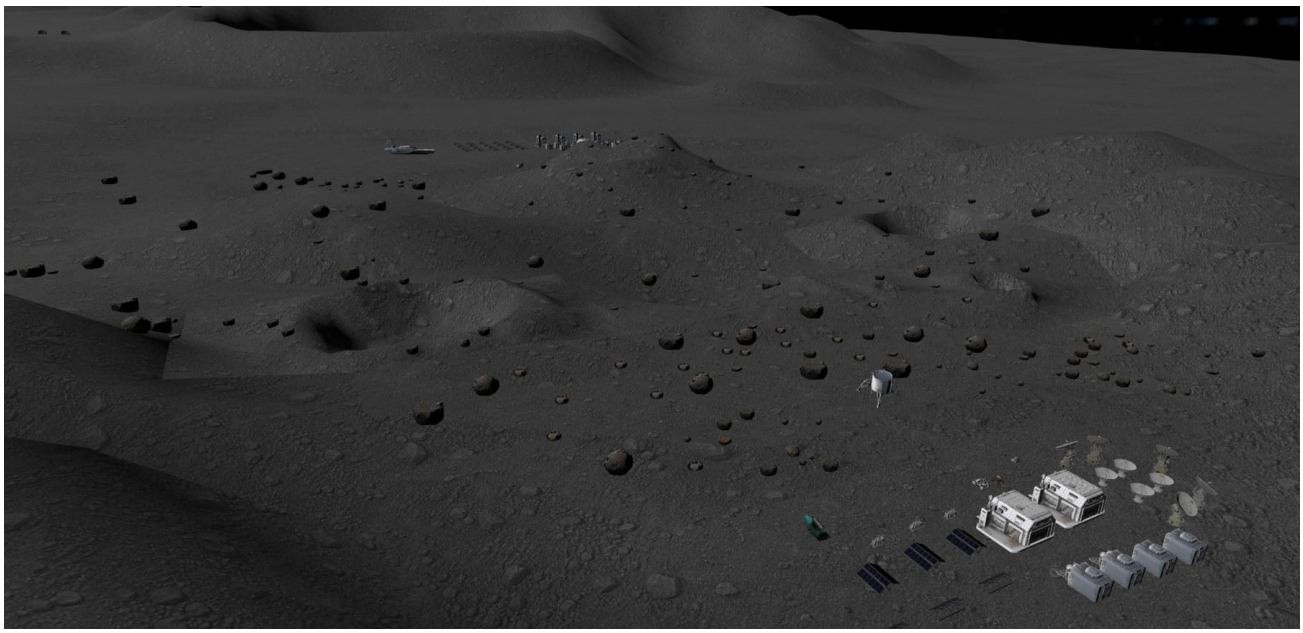
Version 1.0 | September 2026

SECTION 01

System Overview

This manual explains how to run the LunaPilot competition build, teach a safe round trip, save its visual map, and replay the mission autonomously in UC-win/Road.

The mission links Satellite Base A with Energy Center B. During the first run, an operator drives A to B and returns by a different route. The software records visual keyframes, relative motion, arrow-key states, and projected obstacle candidates. During the second run, it localizes against the recorded keyframes and replays the two taught corridors.



Mission layout: A is the satellite base and B is the energy-center sampling site

Implemented Method

Stage	Competition Build	Visible Output
Image capture	Reads the UC-win/Road OpenGL viewport without COM.	Live follow-camera view
Visual tracking	OpenCV ORB features, descriptor matching, and RANSAC motion estimation.	Green features, match and inlier counts
Relative mapping	Fuses visual motion with recorded arrow-key states and projects dark obstacle candidates.	Relative trajectory and red obstacle hits
Route planning	Builds a keyframe topology and runs A* over the taught outbound and return corridors.	Green A-to-B route and orange B-to-A route
Autonomous replay	Matches the live image to nearby route keyframes and replays recorded controls.	Route progress and active control state
Safety	Requires manual arming, supports Esc stop, and stops after persistent localization loss.	STOPPED or ERROR state

SECTION 02

Installation and Setup

Keep the simulator, camera, and Python environment consistent between teaching and autonomous replay.

Item	Required Configuration
Operating system	64-bit Windows
Simulation	UC-win/Road 19.0.2 x64 with the lunar project open
Python	Python 3.9 x64
Packages	numpy 1.26.4, opencv-python 4.10.0.84, Pillow, and pywin32
Camera	Third-person follow camera with the rover near the screen center
Vehicle input	Arrow keys must drive the rover in the active UC-win/Road window
GPU	No GPU is required for the ORB/RANSAC pipeline

First-Time Dependency Check

```
py -3.9 -m pip install -r requirements-visual-slam.txt
```

Start the Dashboard

1. Open the lunar project in UC-win/Road and confirm that the arrow keys control the rover.
2. Select the third-person follow camera. Keep the camera height, pitch, field of view, and simulator window size fixed.
3. From the project folder, double-click start_visual_slam.bat. The command-line alternative is py -3.9 -m vision_nav.slam_app.
4. Confirm that the dashboard shows the live UC-win/Road image and that the state reads READY.

Pre-Run Checklist

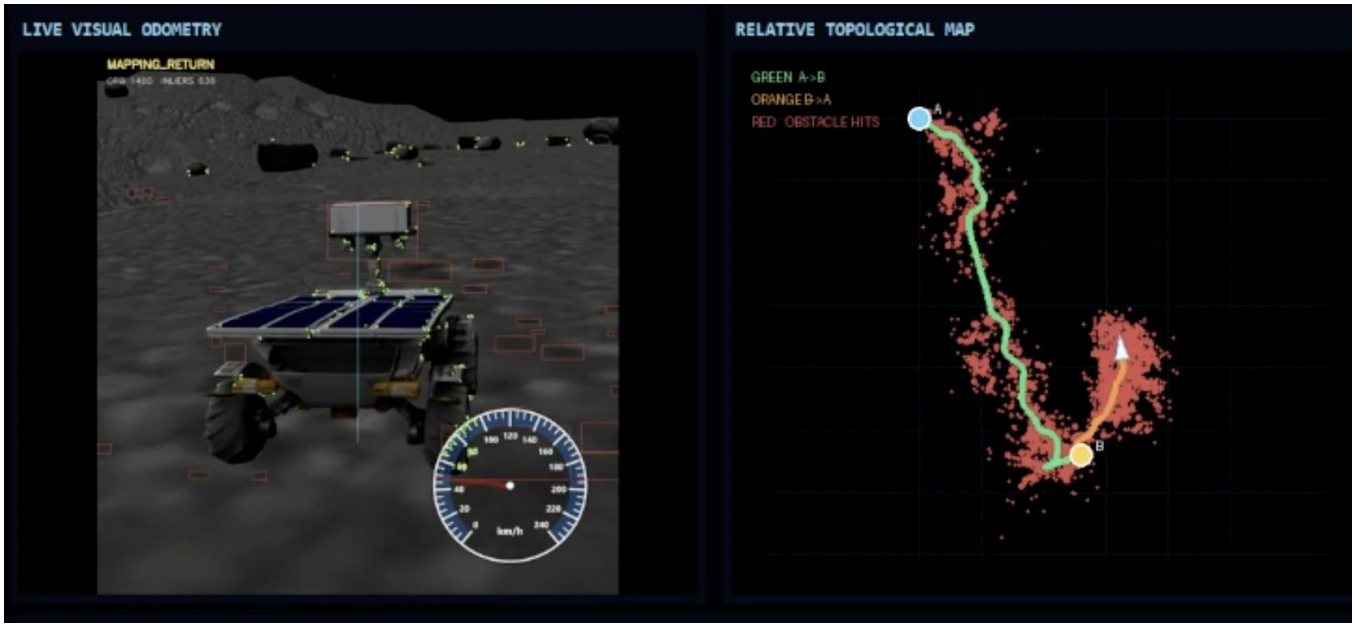
- Only one visible UC-win/Road main window is open.
- The rover remains visible in the follow-camera image.
- The simulator and SLAM dashboard remain restored. Do not minimize either window.
- Lighting, weather, and exposure remain stable during teaching and replay.
- The rover starts from Satellite Base A with enough open space for a short validation route.

Important: COM registration is not required. The dashboard captures the TF8OpenGL viewport directly and keeps vehicle control locked until the operator explicitly enables it.

SECTION 03

Dashboard Interface

The left pane shows visual tracking. The right pane shows the relative topology, mission points, routes, and obstacle candidates.



LunaPilot dashboard during return-route mapping

Display	Meaning
Green image points	ORB features used for visual matching
Red image boxes	Dark obstacle candidates detected in the ground region
Green map line	Recorded outbound corridor from A to B
Orange map line	Recorded return corridor from B to A
Red map points	Repeated obstacle observations projected into the relative map
A and B markers	Satellite Base A and Energy Center B
FPS / ORB / Matches / Inliers	Capture speed and current visual-matching quality
Keyframes / Route	Saved references and current autonomous route progress

Reading the map: The map uses relative units. Shape and route separation matter more than physical distance. A small amount of drift is normal in a single-camera prototype.

SECTION 04

Teach and Map the Round Trip

The first run records one safe outbound corridor and a different return corridor. The software observes the operator and does not send vehicle controls during this phase.

Step	Operator Action	Expected System Response
1	Place the rover at A and focus the UC-win/Road driving window.	Dashboard remains in READY.
2	Press F5, then drive from A to B with the arrow keys.	State changes to MAPPING_OUTBOUND. The green route grows.
3	Stop at B and press F6.	The system stores the B keyframe and changes to MAPPING_RETURN.
4	Drive from B to A along a visibly different safe route.	The orange return route grows and obstacle hits accumulate.
5	Stop at A and press F7.	The system saves map.json, frames, features, and generates topology routes.

Teaching Quality

- Drive smoothly and avoid rapid camera changes. Make turns gradually so adjacent keyframes retain visual overlap.
- Keep rocks, terrain texture, and permanent structures in view. These provide stable features for replay localization.
- Pause briefly at B before pressing F6 and at A before pressing F7.
- Use a genuinely different return corridor so the green and orange routes remain visually distinct.
- For the first test, teach a short route with generous obstacle clearance.

Teaching Hotkeys

Key	Action	Control Output
F5	Start A-to-B teaching	No vehicle command
F6	Mark B and begin return teaching	No vehicle command
F7	Finish at A and save	No vehicle command
Esc	Stop the active operation	Releases all arrow keys

Operator note: Do not press F8 during teaching. If the map contains too few keyframes or B has not been marked, the system refuses to save the session.

SECTION 05

Autonomous Round-Trip Replay

Autonomous replay loads the most recent saved session, localizes at A, drives the outbound corridor, pauses at B, and returns along the taught orange corridor.



Third-person follow view used as the visual-navigation input

1. Reset the rover to the same A position used at the start of teaching.
2. Restore the same camera height, pitch, field of view, simulator window size, and lighting.
3. In the dashboard, select Enable Real Arrow-Key Control. Keep Visual Correction disabled for the first short validation run.
4. Return focus to the UC-win/Road driving window and press F8.
5. Watch the route index and inlier count. The system pauses at B before starting the return leg.
6. Press Esc immediately if the rover departs from the demonstrated corridor.

Control and Safety Behavior

Condition	System Behavior
Control lock remains off	F8 does not start vehicle control.
Reliable keyframe match	The system advances route progress and sends a short recorded arrow-key pulse.
Arrival at B	The system releases the keys, pauses, then loads the return route.
Ten consecutive localization misses	The system releases all arrow keys and enters ERROR.
Esc pressed	The system immediately releases all arrow keys and enters STOPPED.

Optional correction: Visual Correction is optional and camera-dependent. Enable it only after a short validation run. If its steering direction is consistently reversed, select Reverse Visual Correction.

SECTION 06

OBS Recording and Saved Data

The dashboard includes the live simulator image and the evolving relative map, so it can serve as a complete algorithm view for the competition video.

OBS Setup

1. Add UC-win/Road with Game Capture or Window Capture as the main scene source.
2. Add a Window Capture source for [python.exe]: SLAM / UC-win/Road (Lunar Rover Visual SLAM).
3. Use the Windows 10 (1903 and later) capture method first. Try BitBlt if the source remains black.
4. Keep both UC-win/Road and the SLAM dashboard restored. Windows and OBS cannot reliably capture a minimized Tk window.
5. Record teaching and autonomous replay as separate takes. Preserve the F6 transition at B, the automatic pause at B, and the route-color change on return.

Saved Session Structure

```
vision_nav\sessions\<timestamp>\
```

Path	Contents
map.json	Session metadata, B index, keyframe poses, controls, and occupancy hits
frames*.jpg	Visual keyframes used for replay localization
features*.npz	ORB points and binary descriptors
sessions\latest.json	Pointer to the most recently completed session

Performance Guidance

- The default 960-pixel processing width balances visual detail with real-time CPU performance.
- ORB, RANSAC, and obstacle-candidate mapping run on the CPU. The Laptop RTX 4070 is not required for this build.
- A stable feature count and at least 12 inliers support route localization. More inliers usually indicate a stronger match.
- Close unnecessary high-load applications if capture speed becomes unstable during OBS recording.

SECTION 07

Troubleshooting and Quick Reference

Most failures come from a minimized window, changed camera settings, missing keyframes, or starting autonomous control without the safety lock enabled.

Symptom	Likely Cause	Action
Dashboard cannot start	No unique visible UC-win/Road window	Open one UC-win/Road project window and retry.
Live view is black	UC-win/Road is minimized	Restore the simulator. It may remain covered, but not minimized.
OBS cannot capture dashboard	The Tk window is minimized or the source is stale	Restore the dashboard and reselect the python.exe Window Capture source.
No green route appears	F5 was not accepted or the arrow keys are not being read	Confirm MAPPING_OUTBOUND and drive with the standard arrow keys.
F7 refuses to save	B was not marked or too few keyframes exist	Press F6 at B and complete a longer teaching run.
F8 does not move the rover	Actual control remains locked	Select Enable Real Arrow-Key Control, focus UC-win/Road, then press F8.
Localization fails immediately	A position, camera, window size, or lighting changed	Restore the teaching configuration and reset the rover precisely at A.
Rover steers away from route	Correction sign or taught turn sequence is unsuitable	Press Esc. Retest on a short route with correction off; invert only after validation.

Quick Reference

Key	Function
F5	Start teaching at A
F6	Mark B and switch to return teaching
F7	Finish at A and save the session
F8	Start autonomous round-trip replay
Esc	Emergency stop and release all direction keys

Operational limitation: this prototype repeats previously demonstrated corridors and uses a relative obstacle visualization. It does not provide metric terrain reconstruction or guaranteed replanning around a new, unseen obstacle. Maintain operator supervision throughout the run.