

MoonSafe: Collision Avoidance System for Lunar Construction Sites

STRL Team

CPWC2026-0053

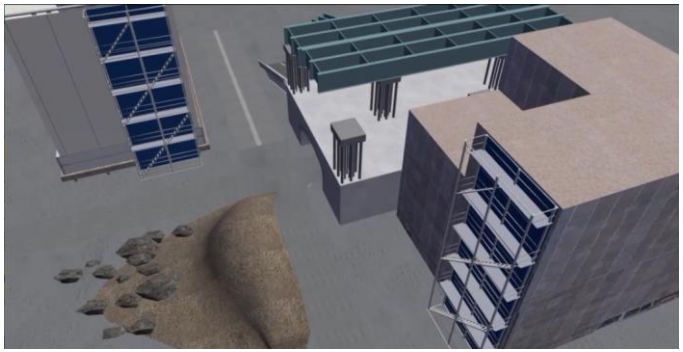


Development Details



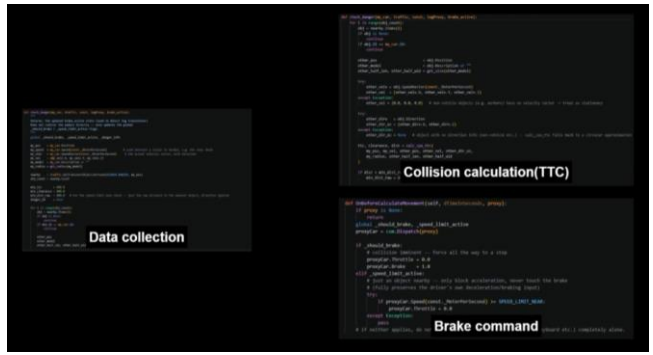
System Architecture

Experimental Network Setup



UC-win/Road

Python Development Environment Setup



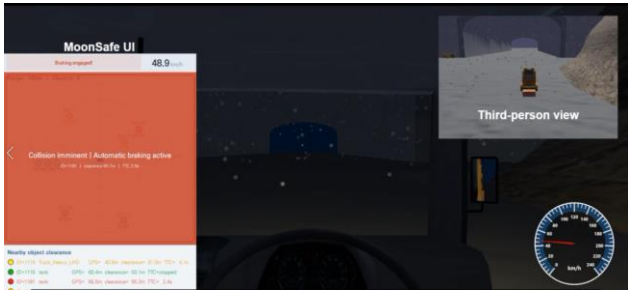
Use of python libraries

GPS-based object detection program



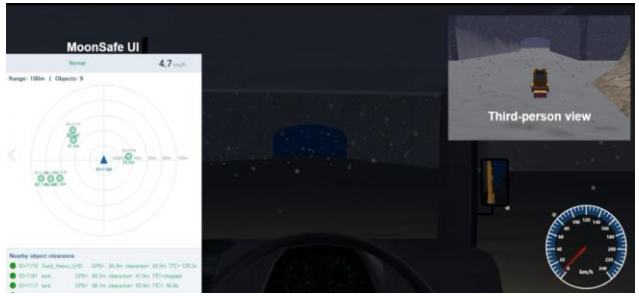
Python-based User Interphase radar

Automatic braking system based on TTC(Time to Collision)



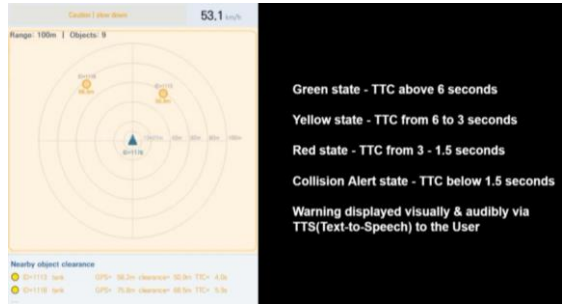
PythonAPIPlugin

Development of GPS-based autonomous driving capabilities for the lunar surface



Sensing/Perception/Localization
Prediction/Path Planning/Control/Actuation

Calculation of distance between vehicle and surrounding objects



TTC based alert state system

Establishment of alert stage criteria



TTC Calculation

1 TTC exceeded by 6s

☑ Alert Level “Green”



Normal operation,
no special measures taken

```
TTC_GREEN = 6.0 # above 6s -> green
    def _ttc_level(self, ttc):
return “green” # 6s over
        COLOR_GREEN = “#27ae60”
```

2 TTC exceeded by 3~6s

☑ Alert Level “Yellow”



Display caution messages,
output safe-driving voice guidance.

```
TTC_YELLOW = 3.0  # 3 to 6s -> yellow
    def _ttc_level(self, ttc):
if ttc <= TTC_GREEN:  return "yellow" # 3 ~ 6s
        COLOR_YELLOW = "#f39c12"
```

3 TTC exceeded by 1.5~3s

☑ Alert Level “Red”

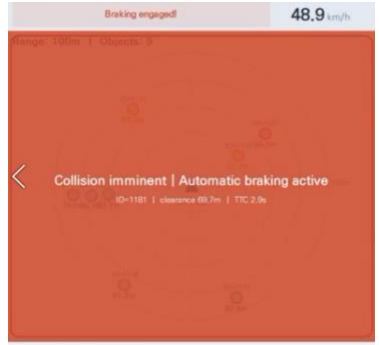


Displays warning messages, outputs
safe-driving voice guidance and alerts.

```
TTC_RED   = 1.5  # 1.5 to 3s -> red
    def _ttc_level(self, ttc):
if ttc <= TTC_RED:   return “red”  # 1~3s
        COLOR_RED = "#e74c3c"
```

4 TTC less then 1.5s

☑ Alert Level “Stop”



System engaged brake command
PythonAPIPlugin

```
TTC_STOP   = 1.5 # 1.5s or below -> collision imminent
    def _ttc_level(self, ttc):
if ttc <= TTC_RED:   return "stop"  # 1s under
        COLOR_STOP = "#c0392b"
```

▶ Calculation of TTC Correction Factor

☑ Correction of moving object targets

```
CPA_SAFETY_MARGIN_BASE = 2.0          # base margin
CPA_SAFETY_MARGIN_PER_SPEED = 0.6     # additional margin
```

```
other_speed_mag = math.sqrt(other_vel[0]**2 + other_vel[1]**2 + other_vel[2]**2)
margin = CPA_SAFETY_MARGIN_BASE + CPA_SAFETY_MARGIN_PER_SPEED * other_speed_mag
```

Future Task



Extending Collision Prediction Technology to Autonomous Lunar Surface Navigation

Collision prediction and braking algorithm enhancement

Environmental variable compensation logic expansion

- ✓ Prediction-time margin proportional to communication latency
- ✓ Refinement of braking distance and inertial dynamics models based on lunar gravity
- ✓ Remodeling of friction coefficients for regolith surfaces

Contingency response planning

- ✓ LIDAR-based emergency backup protocol for GPS transmitter failure
- ✓ Sensor redundancy architecture for visibility-limited conditions

Infrastructure integration and perception expansion

- ✓ Integration with traffic signal systems with development of Lunar infrastructure
- ✓ Expansion from object detection to comprehensive terrain recognition
- ✓ Evolution from 2D to 3D radar representation



Full-autonomy evaluation framework and outcomes

Risk Scenario Refinement

- ✓ Development of crater, slope, and inter-rover collision scenarios
- ✓ Development of sensor malfunction scenarios

Evaluation Metric Diversification

- ✓ Risk exposure quantification via TTC probability distribution (cumulative time below threshold)
- ✓ Driving efficiency evaluation via braking error and detour rates

Expected Outcomes

- ✓ Reduced collision rate via latency, gravity, and friction compensation
- ✓ Generalized framework applicable for future lunar surface projects
- ✓ Unmanned validation before manned application

MoonSafe Road Map

- ✓ Stage 1: Driver safety net, intervening only in critical situations
- ✓ Stage 2: Continuous driver assistance, with expanded intervention
- ✓ Stage 3: Integration as the core decision module of end-to-end (E2E) autonomous driving

Thank You
for Your Attention

Safer Tomorrow,
on Earth and Beyond

