

PatrolKnight

Intelligent Patrol Security Robotics System

Team E: Sam Dong, Po-Ting Ko, Leo Xu, Pin-Wen Wang, Joshua Tsai

Our Team



Sam Dong | Po-Ting Ko | Leo Xu | Pin-Wen Wang | Joshua Tsai

Table of Contents

- System Engineering
  - Project Description
  - Use Case
  - Requirements Modifications
  - Current System Status
- Project Management
  - Schedule
  - Test Plan
  - Budget
  - Risk



Project Description

User needs

- Continuous patrol coverage for large environments
- Easy waypoint selection and patrol setup for operators
- Real-time anomaly detection and response
- Reduced operator workload with human oversight maintained



Proposed Method

- Data-driven waypoint recommendation for patrol missions
- Operator-in-the-loop patrol mission configuration
- Automated patrol route generation
- Autonomous anomaly response and dynamic rerouting



Use Case

Use cases

- Mission planning with recommended waypoints
- Robot performs autonomous patrol
- Real-time anomaly detection
- Operator selects investigation location
- Robot investigates detected events
- Robot resumes original patrol route





Requirements

Mandatory Functional & Performance Requirements

Mandatory Functional Requirements (The System Shall…)	Mandatory Performance Requirements (The System Will…)
M.F.1: Build Map	M.P.1.1: After loop closure produces at least one loop edge, improve global map consistency without structural misalignment.
	M.P.1.2: Remove 100% of dynamic objects and outliers from the final map, reducing wall misalignment to ≤ 15%.
M.F.2: Plan a collision-free path within lanelet.	M.P.2.1: Keep planned paths inside lanelet.
	M.P.2.2: Produce a collision-free path ≤ 5 seconds.

Mandatory Functional & Performance Requirements

Mandatory Functional Requirements (The System Shall…)	Mandatory Performance Requirements (The System Will…)
M.F.3: Patrol along the planned path	M.P.3.1: Limit lateral overshoot to ≤ 10% of the step offset.
	M.P.3.2: At a constant speed of 1.5 m/s, keep steady-state cross-track error ≤ 5 cm.
M.F.4: Interact with User	M.P.4.1: Establish the connection automatically ≤ 30 seconds of robot-system launch.
	M.P.4.2: Send waypoints to the ROS 2 stack ≤ 50 ms after operator pressing Publish.
	M.P.4.3: Notify operator ≤ 50 ms once robot stack shutdown

Mandatory Functional & Performance Requirements

Mandatory Functional Requirements (The System Shall…)	Mandatory Performance Requirements (The System Will…)
M.F.5: Detect Anomaly	M.P.5.1: F1 score of detecting real anomaly (lying down, loitering, trespassing) ≥ 90%
	M.P.5.2: Notify user with anomaly action alert once detected ≤ 2 seconds.

Mandatory Non-Functional Requirements

M.N.1 The system will allow non-expert users to configure patrol without prior knowledge related to robot.
M.N.2 The system shall provide real-time visualized warnings and status notifications to ensure operator awareness of system conditions.
M.N.3 The system shall provide colored UI waypoints to show users if the point is AI-suggested or manually added

System/Subsystem Descriptions

Subsystem: Prototypes

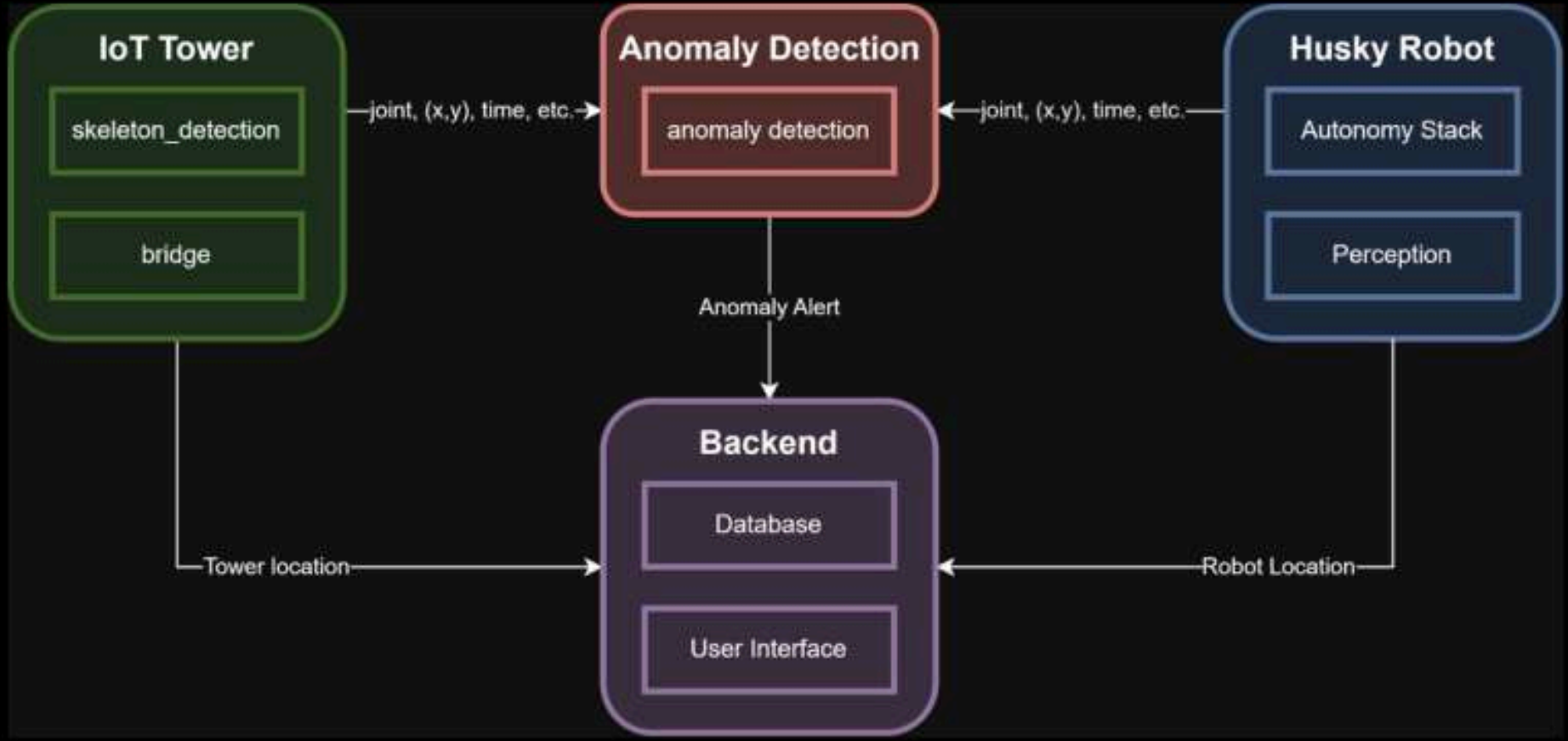


✓ Prototype 1: F1TENTH (Mapping, Localization, Planning, Control)

X Prototype 2: Husky



Subsystem


IoT Tower
skeleton_detection
bridge
joint, (x,y), time, etc.
Anomaly Detection
anomaly detection
joint, (x,y), time, etc.
Husky Robot
Autonomy Stack
Perception
Anomaly Alert
Backend
Database
User Interface
Tower location
Robot Location

IoT Subsystem Functional Description

Input: RGB-D image stream

Functions:

- Human keypoint / skeleton detection
- Persistent person tracking
- Depth-based person localization

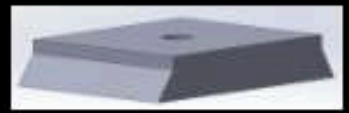




Output:

- person_id
- skeleton keypoints
- bounding box
- person position
- timestamp

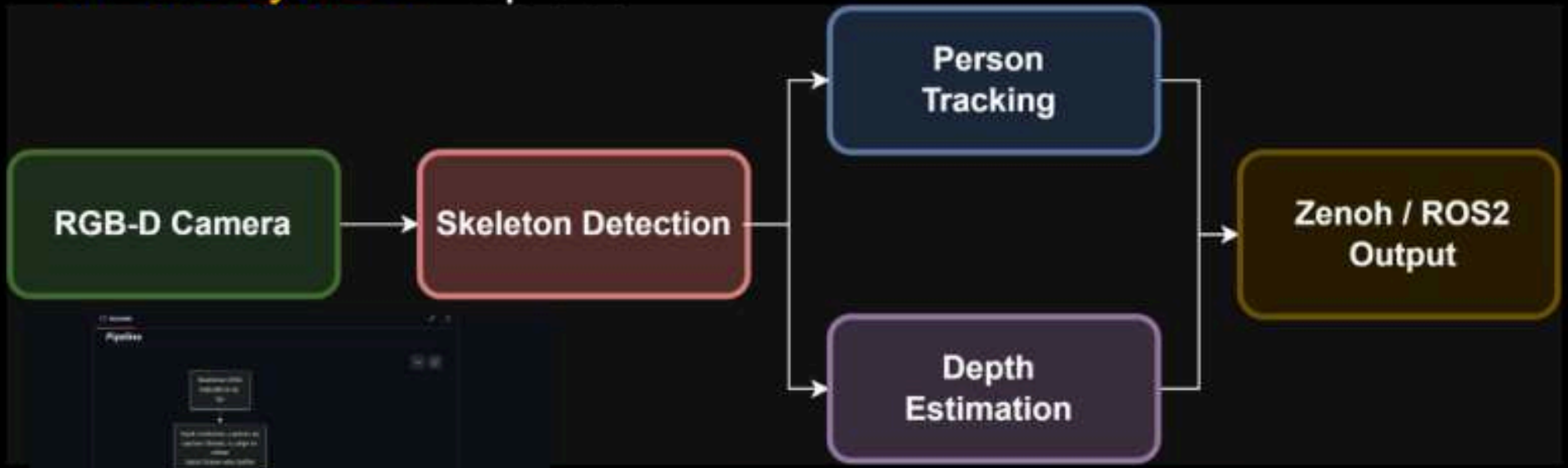








IoT Subsystem Depiction





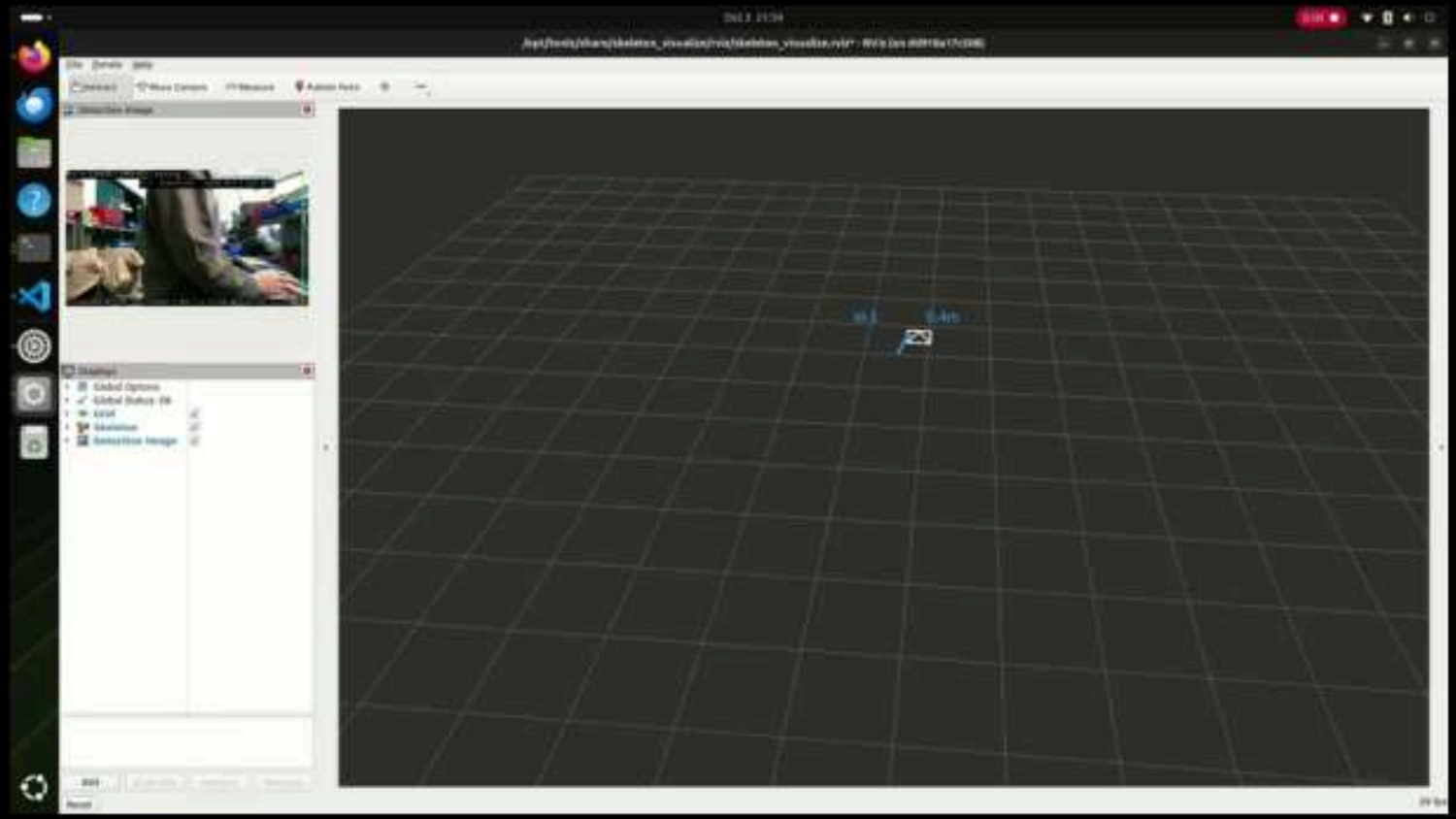
https://github.com/patrol-knight/skeleton_detection



IoT Subsystem Current Level of Functionality

Capability	Status
Skeleton detection (RTMO)	Implemented
Persistent tracking	Implemented
3D localization	Implemented
Occlusion recovery	Under improvement
Outdoor robustness	Under improvement

IoT Subsystem Current Level of Functionality



IoT Subsystem Test Result & Analysis

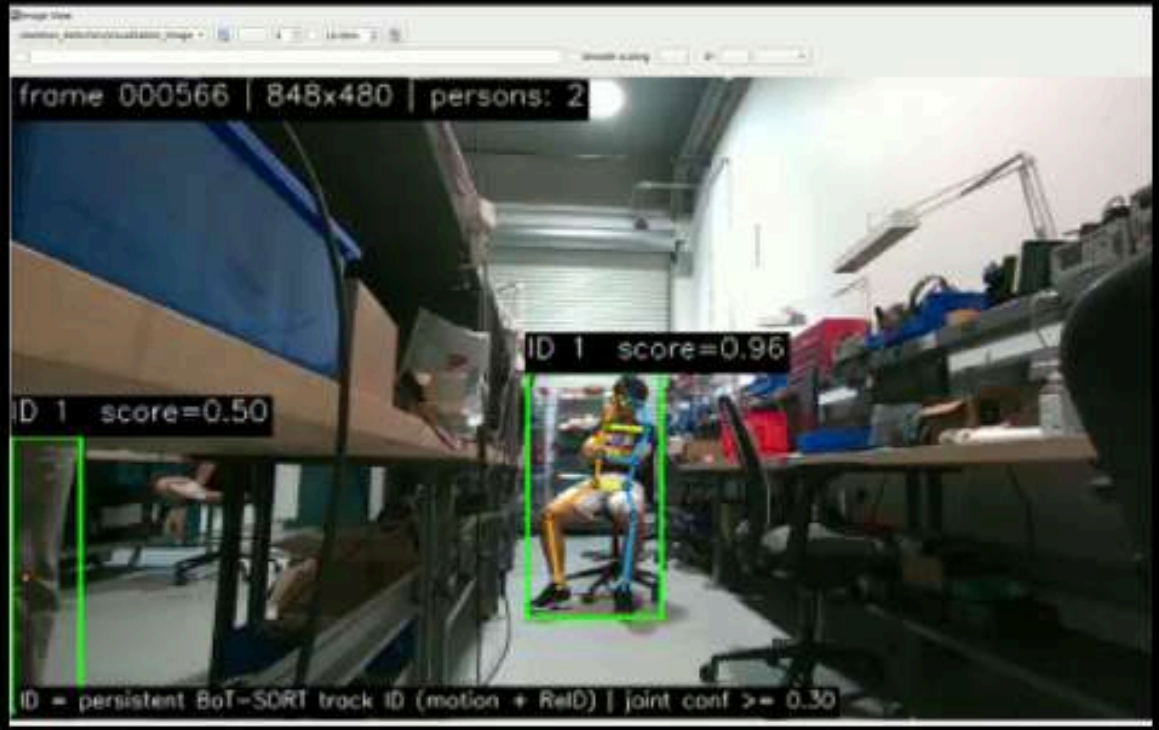
Runtime performance:

- ~30 FPS

- 848x480 resolution

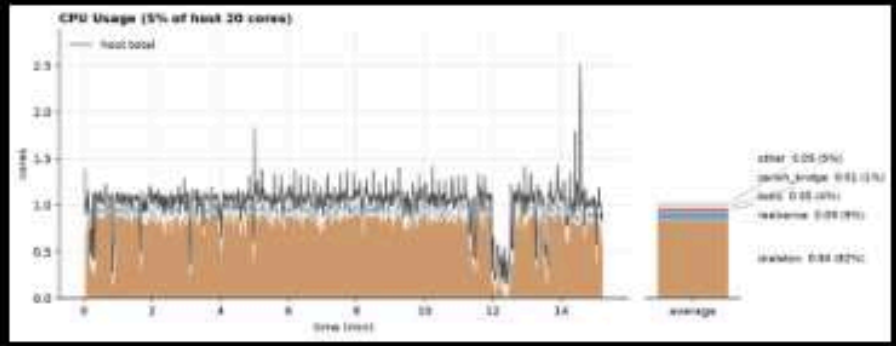
Occlusion recovery:

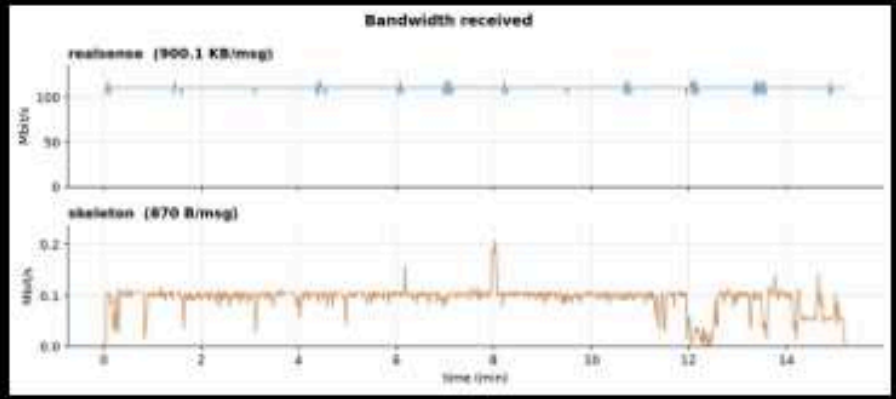
same-ID recovery after ≤3 s occlusion



IoT Subsystem Test Result & Analysis

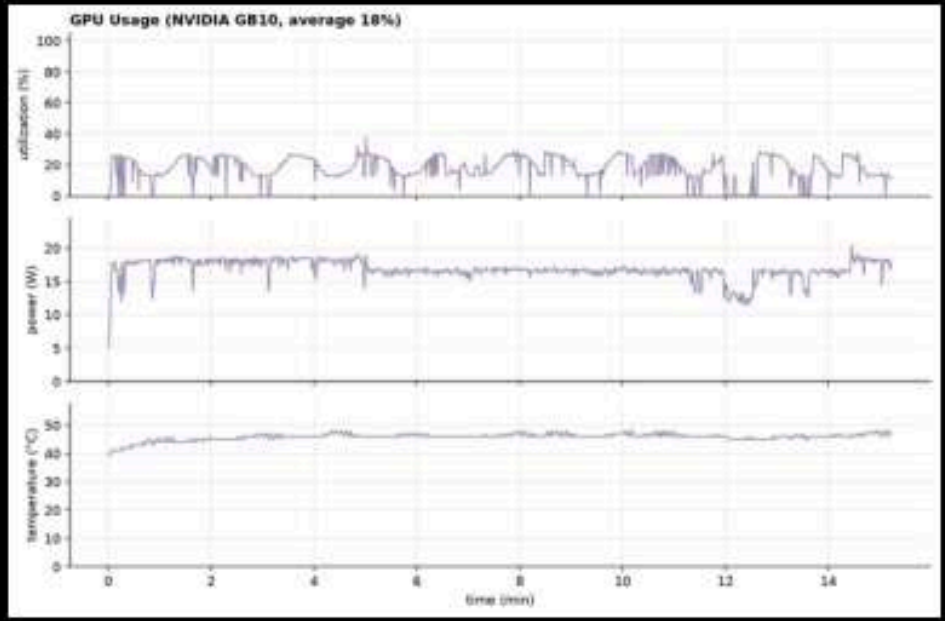
- Average CPU usage: approximately 5% across 20 host cores
- Skeleton processing accounts for approximately 82% of measured CPU usage
- Camera data dominates received bandwidth





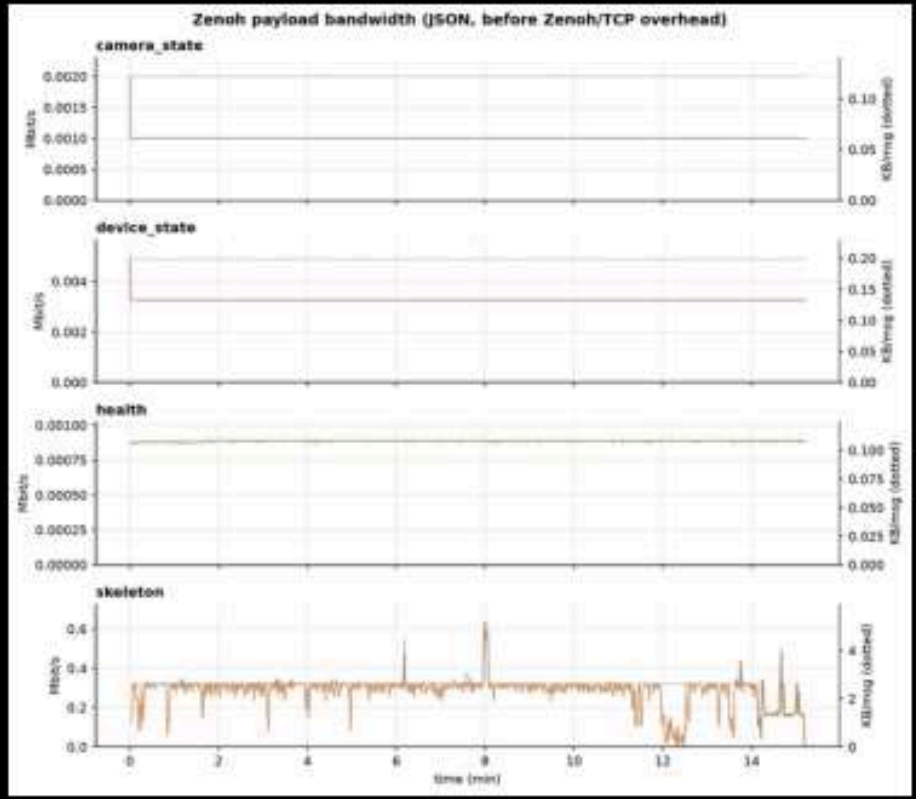
IoT Subsystem Test Result & Analysis

- Average GPU utilization: 18% on NVIDIA GB10
- Power consumption generally remained around 15–20 W
- GPU temperature stabilized below 50°C
- Results suggest capacity for additional GPU workloads



IoT Subsystem Test Result & Analysis

- Camera state, device state, and health messages require little payload bandwidth
- Skeleton data dominates the measured Zenoh payload traffic
- Measurements exclude Zenoh and TCP protocol overhead



IoT Subsystem Challenges Faced

Challenge 1: Temporary ID switching

- Problem: tracker may assign same person different ID in 1s

- Mitigation: temporal ID stabilizer

Challenge 2: Irregular frame timing

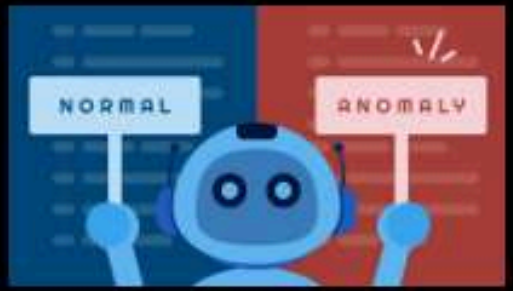
- Problem: downstream temporal detectors need uniform frame interval

- Mitigation: temporal resamplor to fixed interval rate

Anomaly Detection Subsystem Function Description

Input:

- person_id
- skeleton / keypoints
- bounding box
- world position



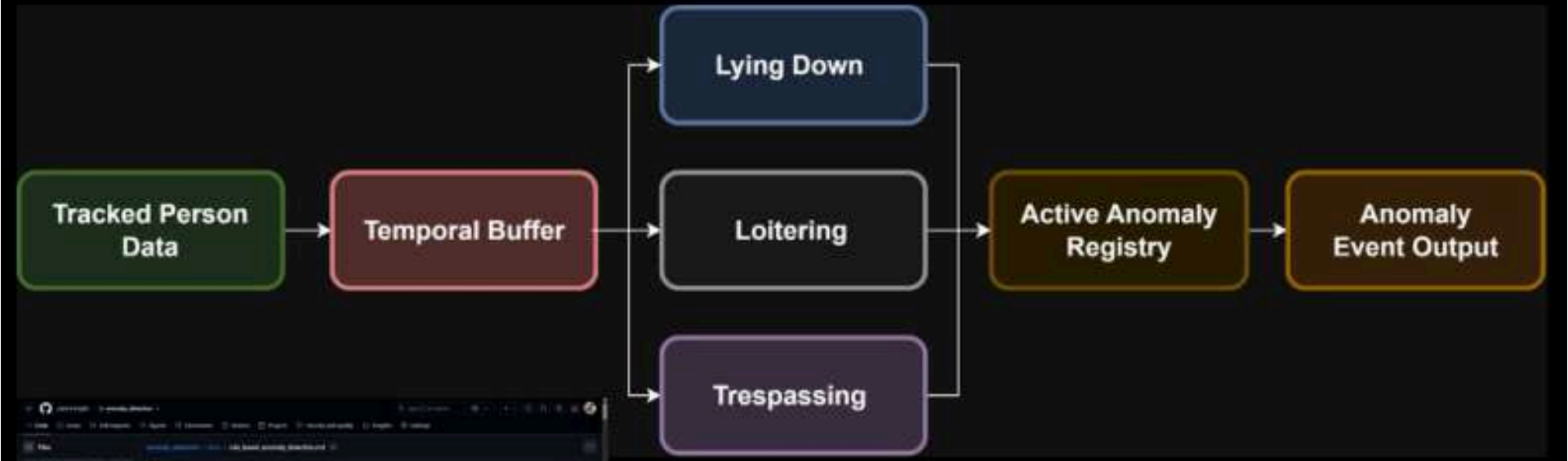
Functions:

- Maintain temporal history
- Evaluate behavior over time
- Detect lying down
- Detect trespassing
- Detect loitering
- Maintain active anomaly state

Output:

- anomaly type
- person ID
- location
- timestamp
- status / confidence

Anomaly Detection Subsystem Depiction





https://github.com/patrol-knight/anomaly_detection/blob/feat/temporal_resampler/docs/rule_based_anomaly_detection.md

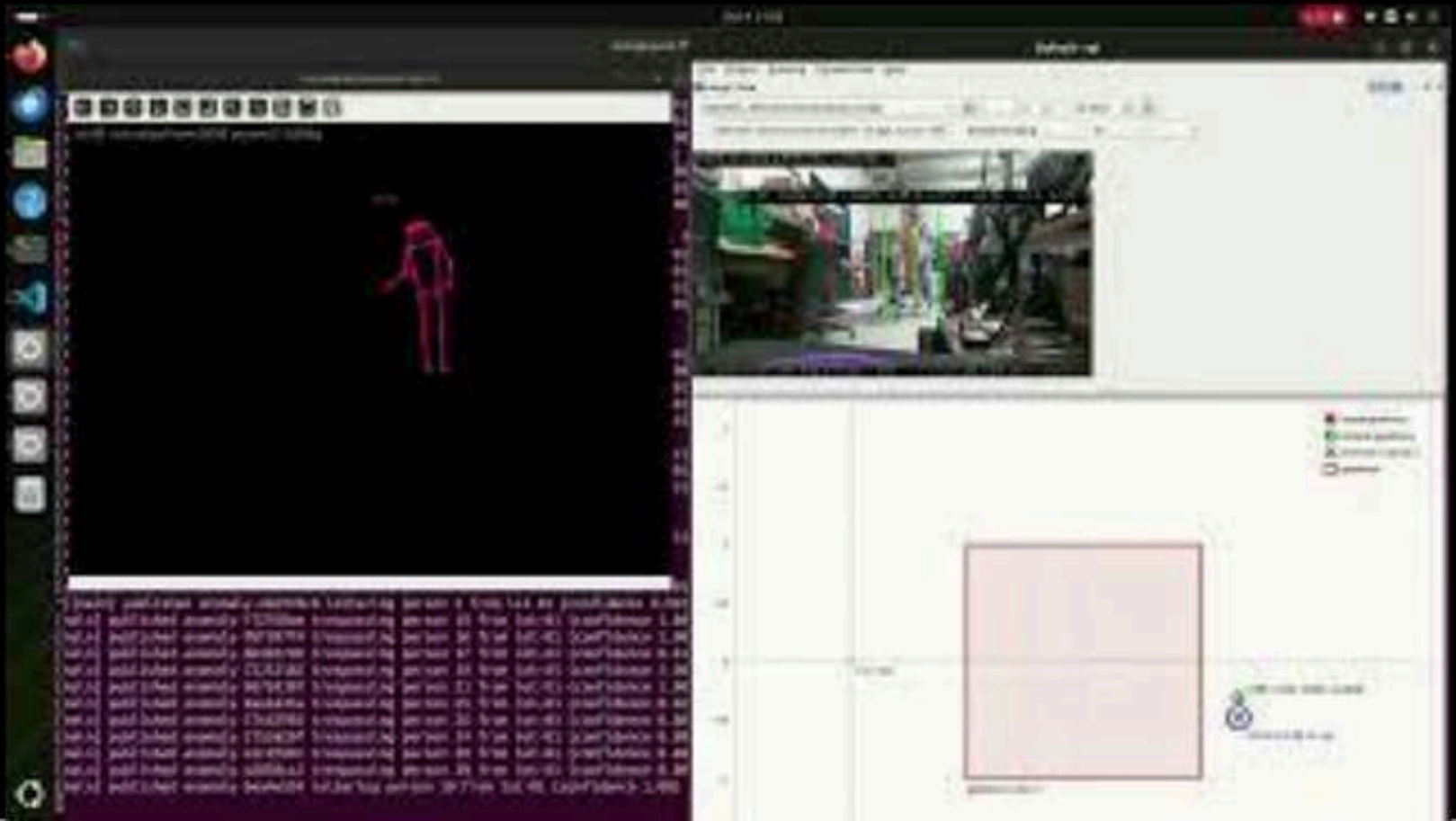
Anomaly Detection Subsystem Current Lvl of Functionality

Capability	Status
Lying-down detector	Implemented
Trespassing detector	Implemented
Loitering detector	Implemented
Temporal resampling	Implemented
Active anomaly registry	In Progress
Zenoh event publishing	Implemented
Learned anomaly detection V2	In Progress

Anomaly Detection Subsystem Test Results



Anomaly Detection Subsystem Test Results



Anomaly Detection Subsystem Challenges Faced

Challenge 1: Rule thresholds are scenario-dependent

- Same threshold may behave differently across people/scenes
- Mitigation: temporal consistency + later learned V2

Challenge 2: Avoiding duplicate anomaly events

- Repeated detector triggers can flood downstream
- Mitigation: active anomaly registry/state tracking

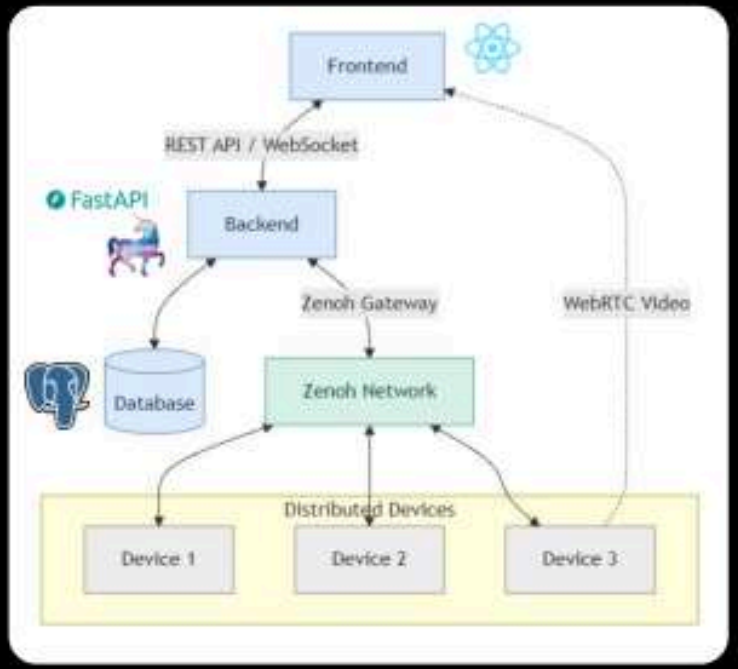
Mission Manager Functional Description

Core Capabilities

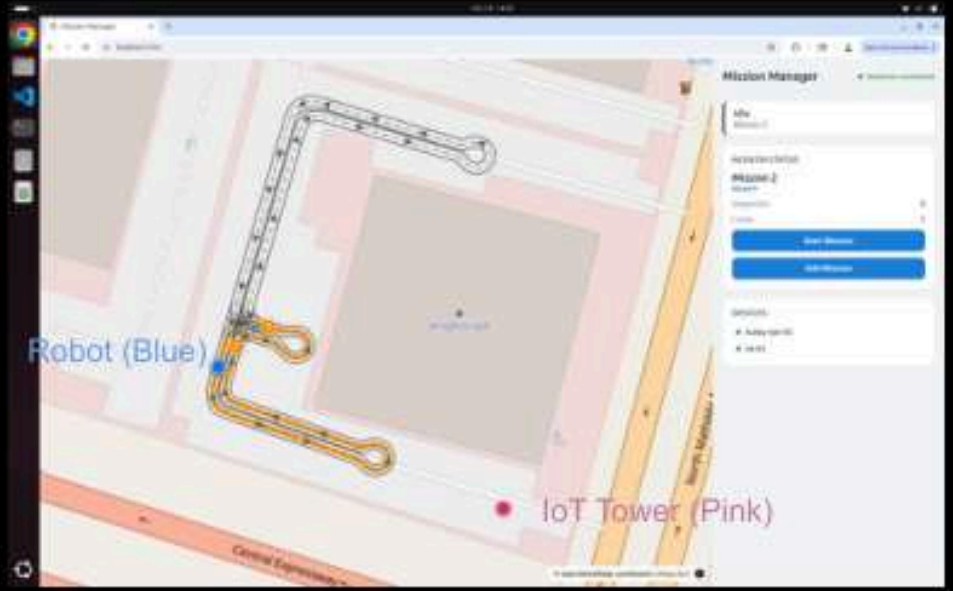
- Device Monitoring: Real-time status and telemetry
- Mission Planning: Robot mission configuration, editing and execution
- Video Streaming: Live camera feeds via WebRTC
- Device Communication: Distributed control and data exchange via Zenoh

Technical Stack

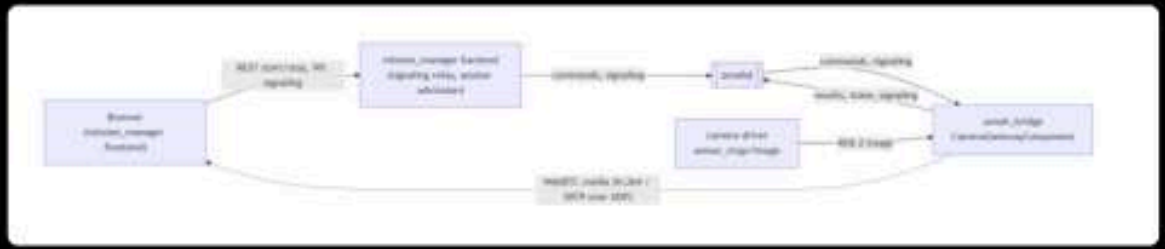
- Frontend: React, TypeScript
- Backend: FastAPI, Uvicorn
- Database: PostgreSQL



Mission Manager Subsystem Depiction



Capability	Status
Device Real-Time Status	Implemented
Mission Editing	Implemented
Mission Execution	Implemented
Video Streaming	Implemented
Anomaly Handling	In Progress
AI Heat Map Generation	In Progress

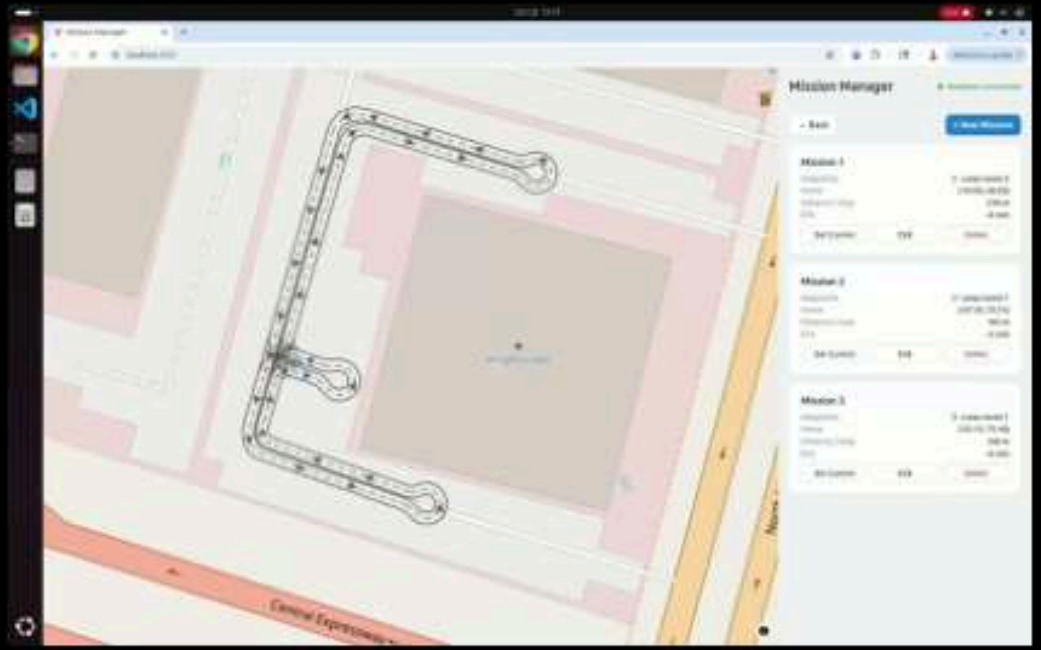


Video Streaming Architecture

Mission Manager Current Level of Functionality

Mission Editing & Device Monitoring

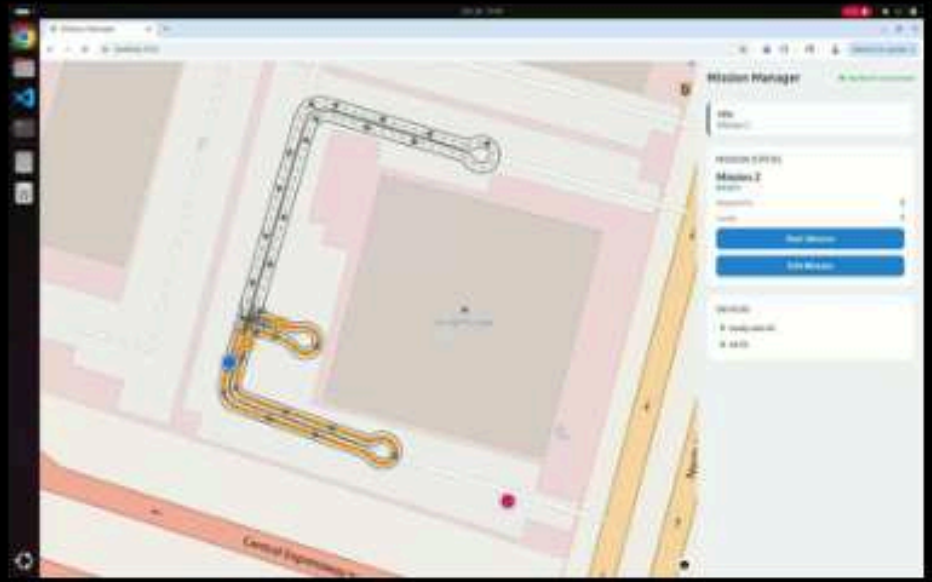
- Mission CRUD and interactive waypoint editing
- Route visualization and mission configuration
- Real-time device status updates



Mission Manager Current Level of Functionality

Video Streaming

- ROS 2 camera integration with GStreamer
- H.264 video encoding and WebRTC transmission
- Browser-based live video playback
- Session-aware streaming control and recovery



Mission Manager Subsystem Test Result & Analysis

Mission Editing

- Mission CRUD Operations
- Saved mission data successfully retrieved and visualized after reload.





Video Streaming

145/145	73	132
C++ Tests Passed	Backend Tests Passed	Frontend Tests Passed

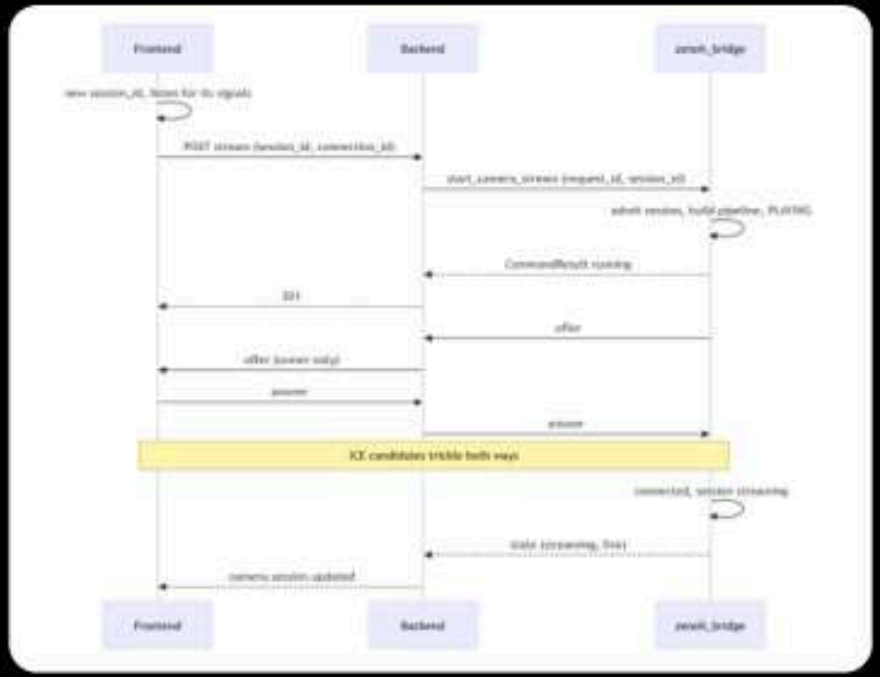
SDP Negotiation	150/150 repeated tests passed
Time to First Frame	~1.07 s
Stream Configuration	848 × 480 @ 30 FPS
Continuous Streaming	60-second test passed
Browser Recovery	Reconnect and restart scenarios passed

Note that:

- 1.07 seconds is the Time to First Frame, not the camera-to-screen latency.
- 30 FPS was the source configuration during testing, so we cannot directly claim that the browser actually maintained a stable 30 FPS.

Mission Manager Subsystem Challenges Face

1. **Video Streaming Concurrency**
   Problem: Race conditions and resource-lifetime risks from concurrent camera callbacks.
   Solution: Implemented safer callback ownership, generation guards, and coordinated async teardown.
2. **H.264 SDP Negotiation Race**
   Problem: Incomplete H.264 parameters from timing-dependent SDP generation caused browser failures.
   Solution: Set deterministic codec preferences and guarded against duplicate SDP Offers.



Husky Robot Subsystem Functional Description

Input:

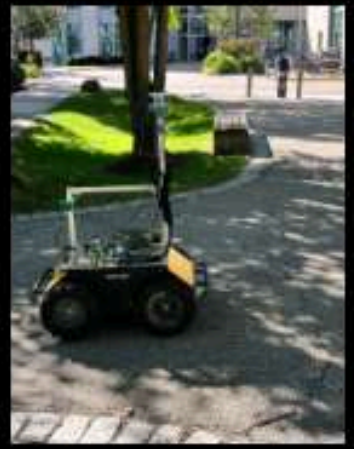
- LiDAR / GNSS / IMU / Wheel Odometry
- Point Cloud Map / Lanelet2 Map / Mission Goal

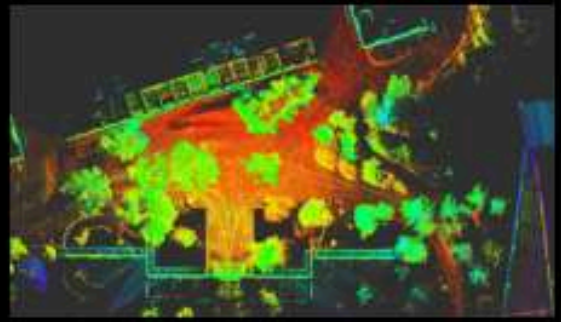
Functions:

- Mapping and Localization
- Obstacle Avoidance
- Lanelet2-based Planning
- Path Following and Motion Control

Output:

- Robot Pose
- Planned Trajectory
- Navigation / Mission Status





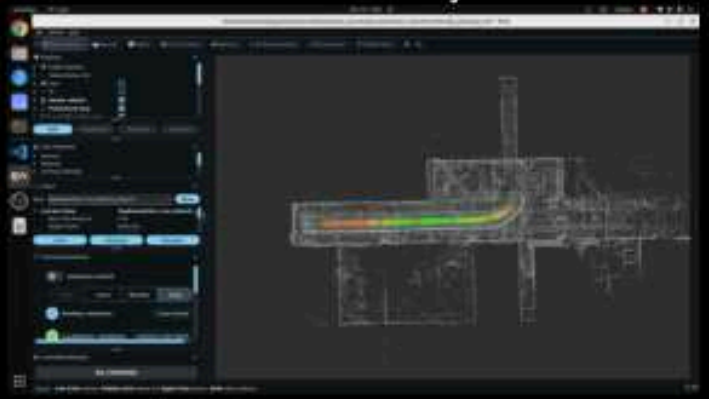


Husky Robot Subsystem Depiction

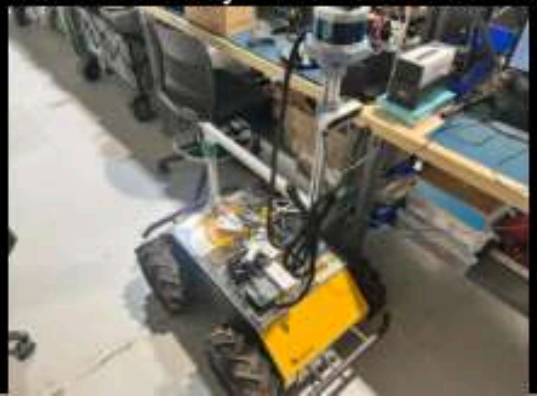
Sensor & Hardware Architecture



Autoware Autonomy Stack



Hardware Assembly and Sensor Installation



Outdoor Autonomous Navigation Testing



Husky Robot Subsystem Current Level of Functionality

Implemented & Functional

- Point Cloud Mapping & Map-based Localization
- Lanelet2-based Route Planning
- Autonomous Path Following & Goal Arrival

In Progress

- GNSS-based Pose Initialization
- Complete Patrol Area Mapping

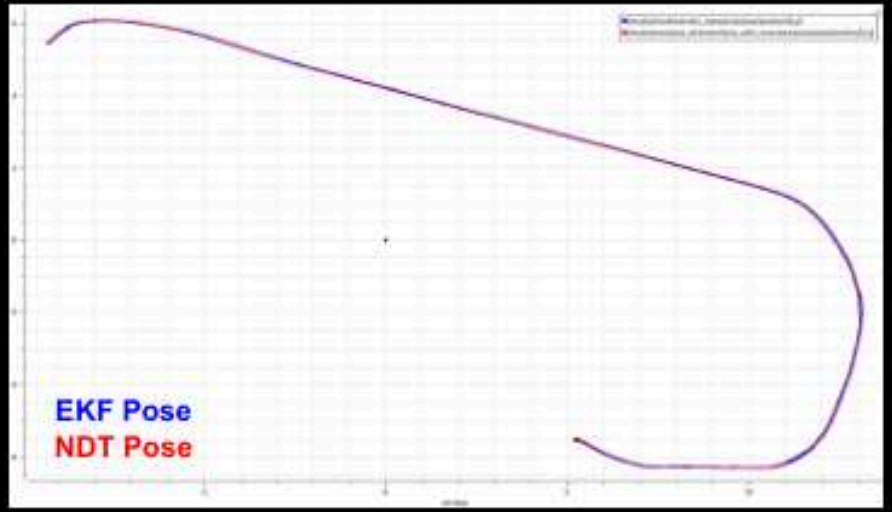
Not Yet Implemented

- Simple Obstacle Avoidance

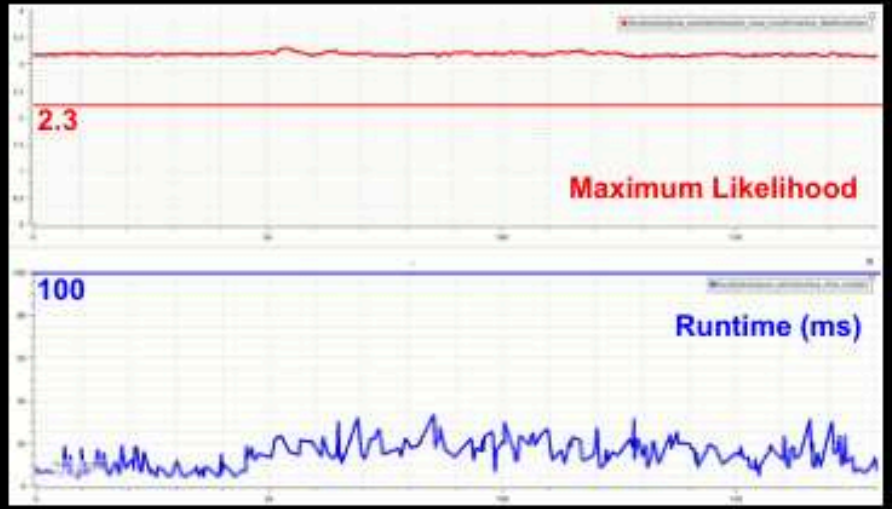


Husky Robot Subsystem Test Results

Localization Metrics



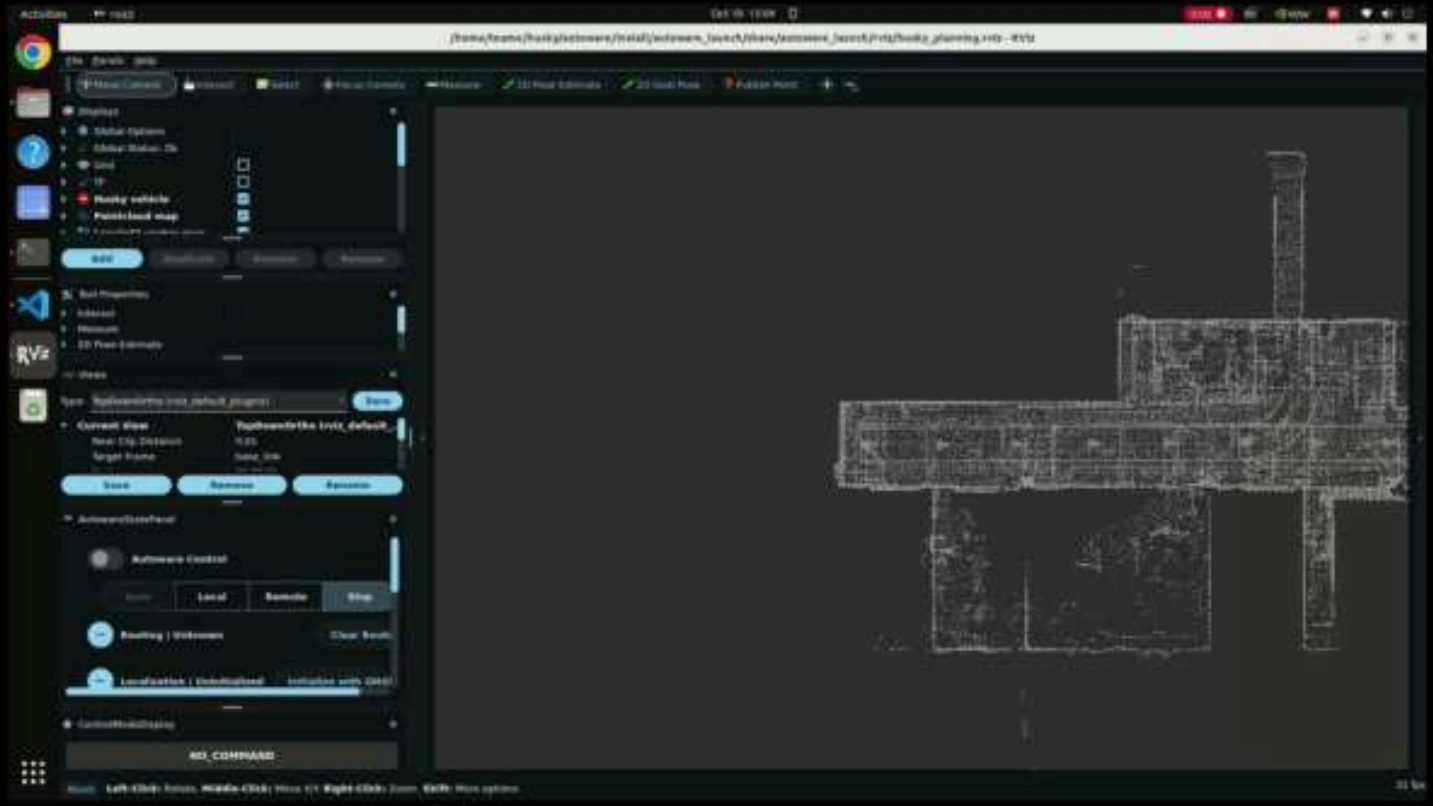
NDT & EKF Localization Trajectories



NDT Localization Metrics

Husky Robot Subsystem Test Results

Initialization -> Localization -> Planning -> Control



Husky Robot Subsystem Challenges faced

Technical Challenges

1. Sensor Time Synchronization

- Problem: Inconsistent sensor timestamps

- Mitigation: Network time synchronization

2. ROS 2 DDS Communication

- Problem: Joystick commands not received

- Mitigation: Align Fast DDS configurations

Husky Robot Subsystem Challenges faced

Non-Technical Challenges

1. Limited Battery Runtime

- Problem: Insufficient runtime for testing

- Mitigation: Replacement battery + external power

2. Outdoor Computer Access

- Problem: Difficult to visualize Orin computer outdoors

- Mitigation: AnyDesk + portable monitor

All Subsystems Major Remaining Challenges

Technical Challenges

1. Robot and Mission Manager Integration

- Problem: Husky state transitions (patrolling, stopping, anomaly investigation) remain unimplemented.

Non-Technical Challenges

1. Unstable Outdoor Wi-Fi Connection

- Problem: Weak and unstable Wi-Fi signal during outdoor testing

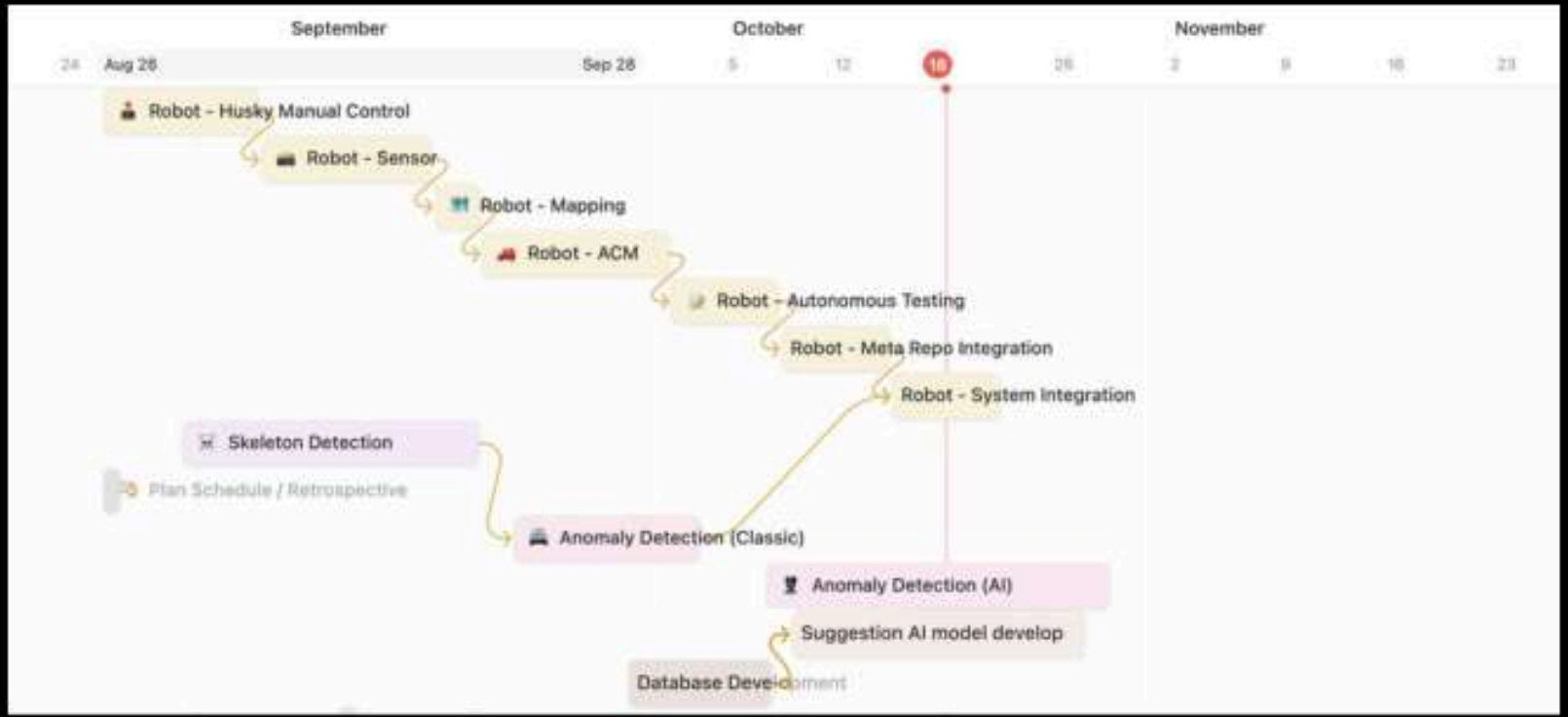
2. Computer Edge Computing Bottleneck

- Problem: IoT Subsystem + Autoware Autonomy Stack require a lot of computation power

Project Management

Schedule

Schedule


September
October
November
24
Aug 28
Sep 28
5
12
19
26
2
9
16
23
Robot - Husky Manual Control
Robot - Sensor
Robot - Mapping
Robot - ACM
Robot - Autonomous Testing
Robot - Meta Repo Integration
Robot - System Integration
Skeleton Detection
Plan Schedule / Retrospective
Anomaly Detection (Classic)
Anomaly Detection (AI)
Suggestion AI model develop
Database Development



Budget

Budget

- Budget used: 3768.96 / 5000 (75%)
- We separate travel expense from other expenses
- Expense breakdown (Top 5)
  - Jetson Orin (1694.40)
  - Zed Camera (651.63)
  - Husky Charger (280)
  - T-Slotted Framing (183.24)
  - Power Station Battery (134.99)
- All major purchases have been completed

reComputer Super J4012 - Advanced Edge AI Computer with NVIDIA® Jetson™ Orin™ NX 16GB in Super Mode	$1,449.00	$1,694.40
ZED 2i Stereo Camera	$579.00	$651.63
Husky SLA Charger	$195	$280
Silver Gusset Bracket 2" Long for 1" High Rail T-Slotted Framing	$15.27	$183.24
Dynesspower PPS300 300W Portable Power Station with 256Wh LiFePO4 Battery	$134.99	$134.99



Test Plan

PR 10

- Milestones:
  - Anomaly heatmap model that suggests waypoints to visit
  - Anomaly detection with live streaming
  - Dashboard and backend reacts correctly according to user input toward anomaly detection.

- Test:
  - AI-Assisted Anomaly Heatmap (T.P.03)
  - Anomaly Detection V2 (T.P.07)
  - Live Investigation and Patrol Resumption (T.P.09)

PR 11

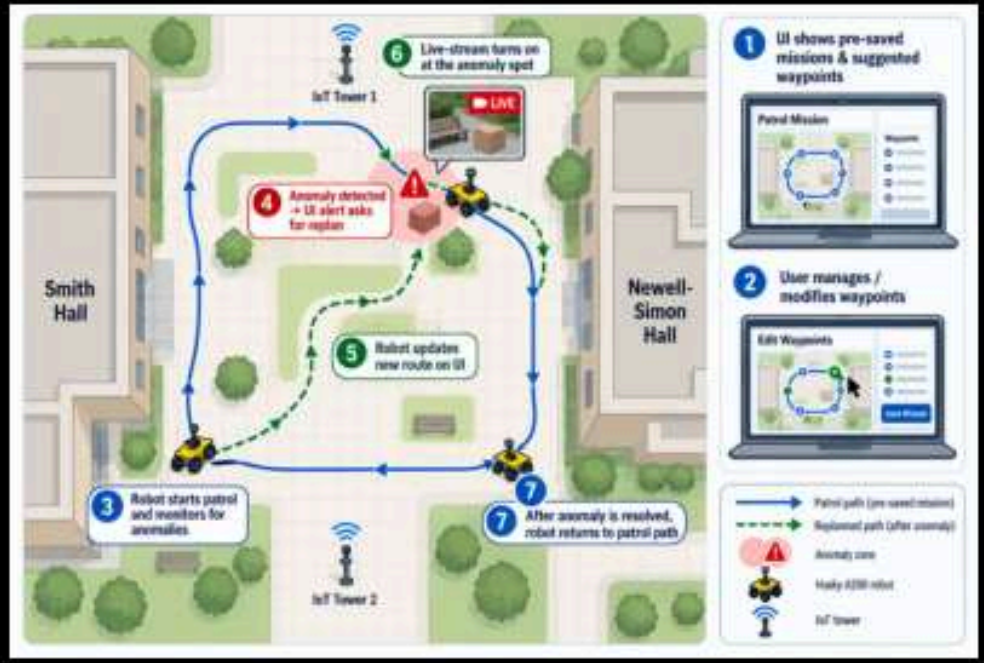
- Milestones:
  - Heatmap-assisted mission selection
  - Anomaly detection and investigation
  - Full FVD rehearsal

- Test:
  - FVD test plan (T.P.10)

FVD

Goals:

1. Missions load based on the AI heatmap within 2s
2. Anomaly detection F1 score is greater than 0.9
3. Investigation route starts within 5s
4. Robot reaches the closest lanelet point next to warning site within 1m
5. The live stream appears within 5s after arrival.



Risk Management

Risk Management

		Old			New	
Risk Number	Description	Consequence	Likelihood	Mitigation	Consequence	Likelihood
1	K7 Robot Availability Issue	5	4	Use other prototypes (Husky) for onboard testing	2	**1***
6	System Integration Blockage	5	3	Create dummy topics to simulate the input and output of the incomplete module	2	3
11*	Outdoor Lighting Degrades Skeleton Detection	5	3	Tune D435i RGB exposure settings and camera orientation; add a sunshade if necessary.	3	2

* marks the changed risks

Risk Management

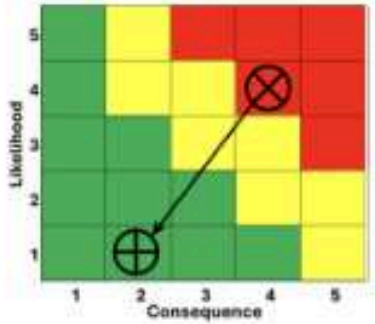
		Old			New	
Risk Number	Description	Consequence	Likelihood	Mitigation	Consequence	Likelihood
13*	Unstable WiFi Connection	4	4	Use AP to strength connectivity	2	3
14*	Computer Edge Computing Bottleneck	5	3	Use light weighted model Replace autoware with customized functions	2	2

* marks the changed risks

KNIGHTSCOPE

R1: K7 Robot Availability Issue

Risk ID	Risk Title	Risk Owner
R1	K7 Robot Availability Issue	Leo Xu
Description		**Date Added**
The K7 Patrol Robot in California may not be available due to contract delays, scheduling constraints, and limited travel budget.		11/26/2025
Consequence		**Date Updated**
No platform available for testing our system.		02/15/2026



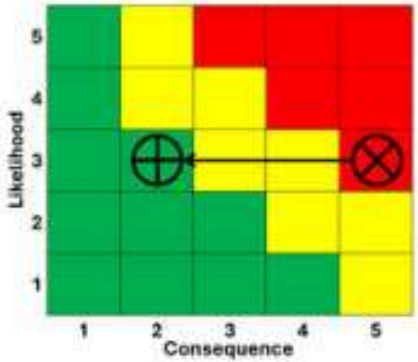
Risk Reduction Plan Summary		Risk Type:	Schedule, Cost
Action/Milestone	**Success Criteria**	**Date Planned**	**Date Implemented**
Use other prototypes (F1 Tenth, Golf Cart) for onboard testing.	Successfully run our system on the prototype.	11/28/2025	02/05/2026
Use simulators (CARLA, Gazebo, Rviz) for offline testing.	Successfully test developed features on the simulator.	11/28/2025	02/15/2026

Comments

KNIGHTSCOPE

R6: System Integration Blockage by Incomplete Modules

Risk ID	Risk Title	Risk Owner
R6	System Integration Blockage by Incomplete Modules	Joshua Tsai
Description		**Date Added**
During the system integration process, the schedule may be delayed by individual modules that have not yet completed development and validation.		02/15/2026
Consequence		**Date Updated**
Because the modules in our system are highly dependent on the outputs of other modules, incomplete functionality in one module can prevent us from testing downstream modules, thereby delaying the overall system integration timeline.		02/20/2026



Risk Reduction Plan Summary		Risk Type:	Programmatic
Action/Milestone	**Success Criteria**	**Date Planned**	**Date Implemented**
Create dummy topics to simulate the input and output of the incomplete module.	Dataflows are validated by generating expected outputs.	02/15/2026	02/20/2026

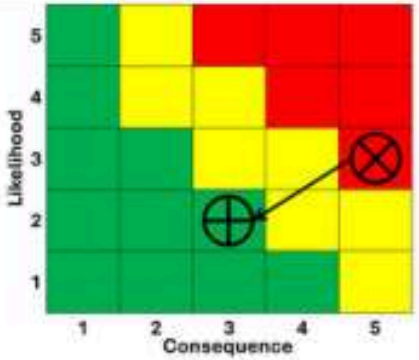
Comments

KNIGHTSCOPE

R11: Outdoor Lighting Degrades Skeleton Detection

Risk ID	Risk Title	Risk Owner
R11	Outdoor Lighting Degrades Skeleton Detection	Sam Dong

Description	Date Added
The Intel RealSense D435i RGB camera has primarily been tested indoors. Strong sunlight, backlighting, shadows, and rapid illumination changes outdoors may cause poor exposure and reduce skeleton detection performance.	08/15/2026
Consequence	**Date Updated**
Skeleton keypoints may be missed or incorrectly detected, reducing the reliability of human detection and downstream system functions during outdoor operation.	

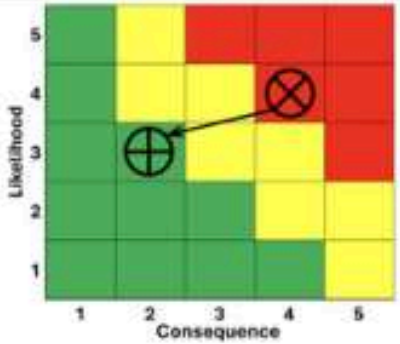


Risk Reduction Plan Summary		Risk Type:	Technical
Action/Milestone	**Success Criteria**	**Date Planned**	**Date Implemented**
Test the current skeleton detection pipeline under representative outdoor light conditions, including direct sunlight, backlighting, shadows, and transitions between shade and sunlight.	Skeleton detection maintains acceptable detection rate and keypoint confidence across representative outdoor conditions.	08/24/2026	
Tune D435i RGB exposure settings and camera orientation; add a sunshade if necessary.	Exposure remains stable enough for reliable skeleton detection without significant overexposure or underexposure.	08/24/2026	

Comments

R13: Unstable WiFi Connectivity

Risk ID	Risk Title	Risk Owner
R13	Unstable WiFi Connection	Po-Ting Ko
Description		**Date Added**
Weak and unstable Wifi signal during outdoor testing.		09/15/2026
Consequence		**Date Updated**
Intermittent connectivity may disrupt remote monitoring, data transmission, and robot operation during outdoor testing, resulting in testing delays and potential loss of communication with the robot.		

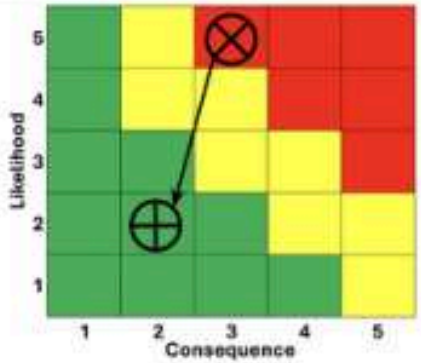


Risk Reduction Plan Summary		Risk Type:	Technical
Action/Milestone	**Success Criteria**	**Date Planned**	**Date Implemented**
Deploy an additional Access Point (AP) to improve WiFi coverage and signal strength during outdoor testing.	Stable WiFi connectivity is maintained throughout the designated testing area with minimal connection interruptions.	09/15/2026	
Conduct outdoor connectivity tests to evaluate network stability and optimize AP placement.	Robot communication and data transmission remain reliable during testing, with no significant disruptions to robot operations.	09/15/2026	

Comments

R14: Computer Edge Computing Bottleneck

Risk ID	Risk Title	Risk Owner
R14	Computer Edge Computing Bottleneck	Joshua Tsai
Description		**Date Added**
IoT Subsystem + Autoware Autonomy Stack require a lot of computation power		09/15/2026
Consequence		**Date Updated**
Limited edge computing resources may cause high CPU/GPU utilization, increased inference latency, and reduced system responsiveness, potentially affecting real-time perception, navigation, and overall robot performance.		



Risk Reduction Plan Summary		Risk Type:	Technical
Action/Milestone	**Success Criteria**	**Date Planned**	**Date Implemented**
Deploy lightweight AI models to reduce computational overhead and improve inference efficiency.	Reduced CPU/GPU utilization and inference latency while maintaining acceptable model accuracy.	09/15/2026	
Conduct integrated system performance tests under representative operating conditions.	Stable real-time operation without significant processing delays, resource exhaustion, or system crashes.	09/15/2026	

Comments


Thank You