Chapter Outline

Localization, Mapping, Prediction, and Planning

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.3.1 Simultaneous Localization and Mapping

Textbook: Sec. 8.1.2.4

8.3.2 Behavior Prediction and Interaction Modeling

Textbook: Sec. 8.1.2.5

8.3.3 Path Planning and Dynamic Obstacle Avoidance

Textbook: Sec. 8.1.2.6

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making

Textbook: Sec. 8.1.2.7

8.3.1 Simultaneous Localization and Mapping: Joint Estimation

Locating the Target and the Vehicle

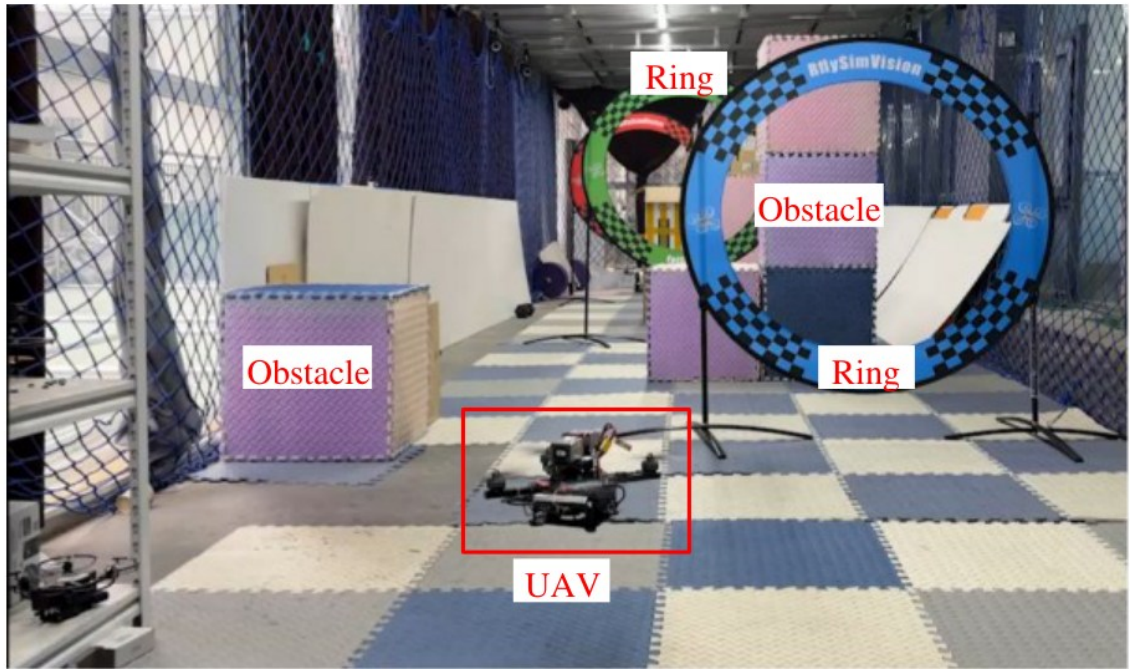
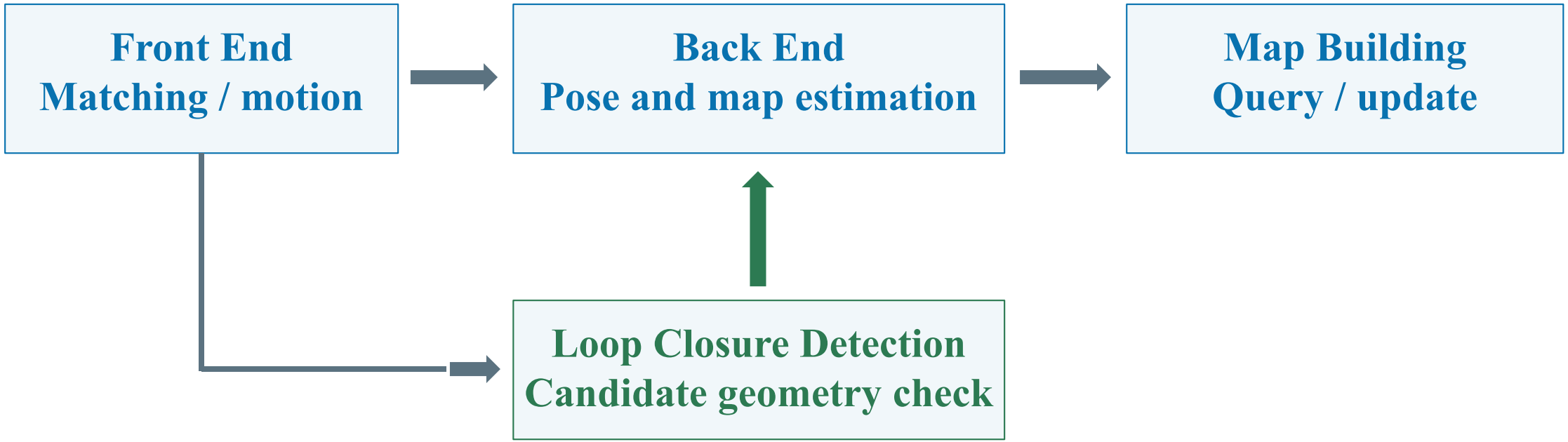


Fig. 8.3(a) Target, Obstacles, and Vehicle

- **Simultaneous localization and mapping (SLAM):** jointly estimates the vehicle pose and an environment map.
- **Visual odometry (VO):** mainly estimates successive motion. It may maintain a local feature map.
- **Different outputs:** a target box is in the current image. The vehicle pose is in an agreed map frame.

8.3.1 Simultaneous Localization and Mapping: Four Core Modules

Front-End Constraints and Back-End Consistency



The front end supplies local relations. Loop closure links observations across time. The back end jointly adjusts past poses and the map.

8.3.1 Simultaneous Localization and Mapping: Sensor Approaches

Visual Feature Matching and LiDAR Geometry

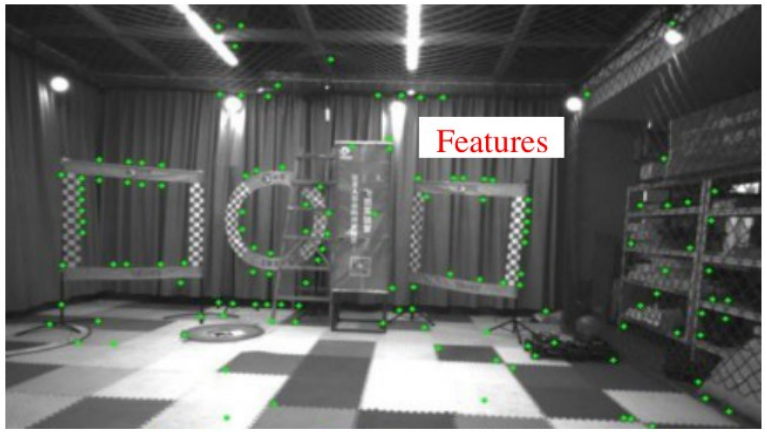


Fig. 8.3(d) ORB-SLAM Features

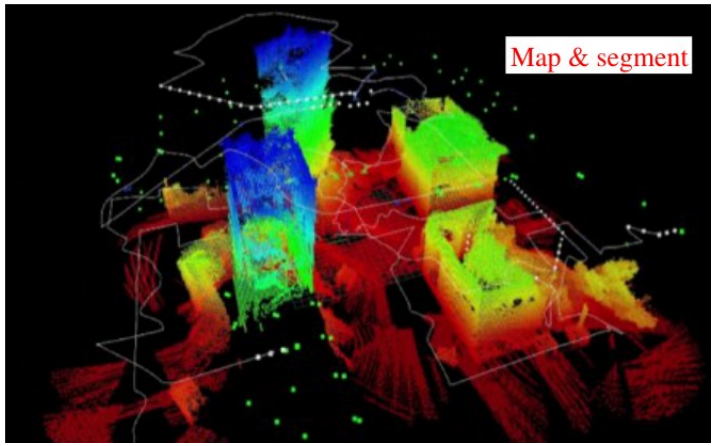


Fig. 8.3(f) FAST-LIO Map

Visual SLAM: feature / pixel matching.
ORB-SLAM: sparse feature constraints.
Weak texture and blur reduce valid observations.

LiDAR SLAM: matches scan geometry.
Cartographer: scans and submaps.
FAST-LIO: LiDAR-inertial odometry.

8.3.1 Simultaneous Localization and Mapping: Inertial Fusion

High-Rate Propagation and Geometric Corrections

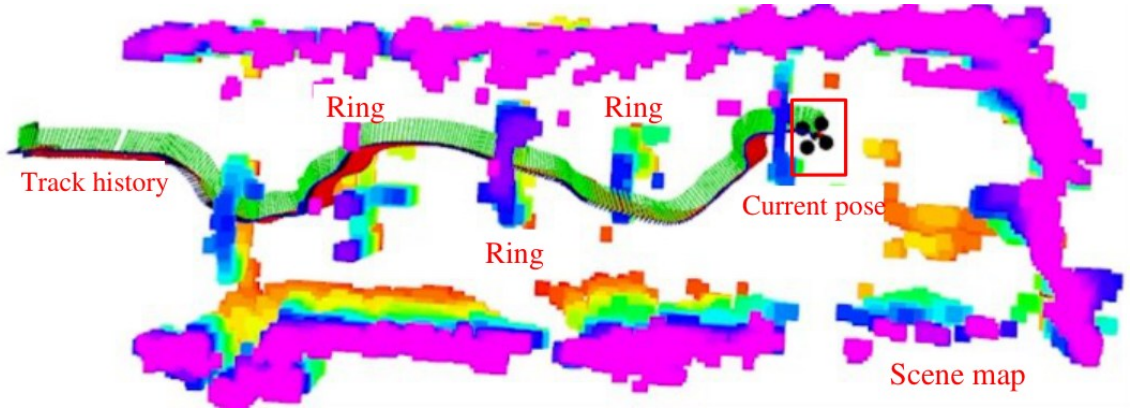


Fig. 8.3(c) Pose, Map, and Trajectory

Fusion	Meaning
VIO / VINS	Visual-inertial odometry Visual-inertial navigation system
LIO	LiDAR-inertial odometry
LVI	LiDAR, visual, and inertial fusion

Inertial measurement unit (IMU): high-rate propagation. Vision / LiDAR limits drift.
VINS-Mono: mono + IMU. VINS-Fusion: mono / stereo + IMU, or stereo alone.
Example rates: geometry 10–30 Hz, IMU 100–1000 Hz. Set rates per algorithm.

8.3.1 Simultaneous Localization and Mapping: Map Representations

Different Maps Support Different Queries

Representation	Stored Information	Supported Tasks
Sparse feature map	Feature positions, descriptors, and observation links	Matching and localization. No direct clearance query.
Dense occupancy grid	Free, occupied, unknown, and occupancy probabilities	Collision queries. Finer grids need more memory.
Semantic map	Geometry + classes, instances, or regions	Traversability and object-level tasks

Map and query must match: feature matching, occupancy, and object semantics are not interchangeable.

8.3.1 Simultaneous Localization and Mapping: Distance Representations

TSDF Surfaces and ESDF Clearance

TSDF: truncated signed distance function. ESDF: Euclidean signed distance field.
$\mathbf{x}$ is the query point. $\mathbf{q}$ lies on obstacle boundary $\partial\mathcal{O}$. Distances are in m.

Representation	Geometric Meaning	Direct Use
TSDF	Truncated surface distance along a sensor ray	Zero-crossing surfaces / depth fusion
ESDF	Euclidean distance to the nearest obstacle boundary	Clearance queries / trajectory constraints

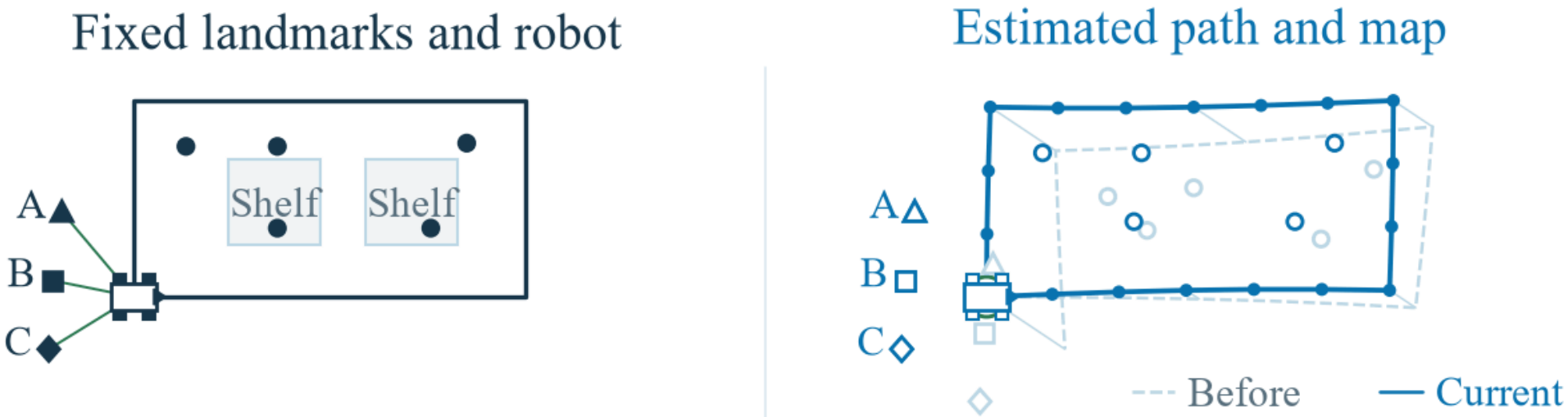
$$d_{\mathrm{ESDF}}(\mathbf{x}) = \pm \min_{\mathbf{q}\in\partial\mathcal{O}} \|\mathbf{x}-\mathbf{q}\|$$

Outside: positive. Inside: negative. Boundary: zero. Unknown space is not assumed free.

8.3.1 Simultaneous Localization and Mapping: Loop Closure Detection

Verified Revisits Correct the Whole Trajectory

Accepted loop: path and map corrected



Preset pairs: 3
Same pose; fixed heading

Loop gap: 1.44 → 0.00 m
Equal weights; fixed start

Same-pose revisit example: after verification, the trajectory and map are corrected together.

8.3.1 Simultaneous Localization and Mapping: Loop Closure Calculation

Four Odometry Segments Accumulate 0.4 m of Error

Measured increments: 10.1, 10.0, −9.8, −9.9 m. Sum: +0.4 m.
Fixed start, equal edge variances, and an exact closure constraint at the start.

$$\min_{\delta_1,\ldots,\delta_4} \sum_{i=1}^{4} \delta_i^2 \quad \text{s.t.} \quad \sum_{i=1}^{4} \delta_i = -0.4$$

$$\delta_i = -\frac{0.4}{4} = -0.1\ \text{m}$$

δ_i corrects increment i (m). The four segment errors are independent.
Corrected increments: 10.0, 9.9, −9.9, −10.0 m. Sum: 0. All four edges share the correction.

8.3.1 Simultaneous Localization and Mapping: Full Trajectory Result

Loop Closure Also Adjusts Intermediate Poses

Time Step	Accumulated Pose (m)	Pose After Closure (m)
0	0	0
1	10.1	10.0
2	20.1	19.9
3	10.3	10.0
4	0.4	0

$$J = 4 \times (0.1)^2 = 0.04\ \mathrm{m}^2$$

Loop closure improves relative consistency. Global pose needs an anchor.

8.3.1 Simultaneous Localization and Mapping: SLAM and Satellite Positioning

A Consistent Local Map Still Needs a Global Reference

Aspect	SLAM: Local Reference	GNSS: Global Reference
Information source	Relative scene observations and past constraints	Satellite observations and a coordinate datum
Error factors	Initialization, scale, and accumulated drift	Occlusion, multipath, and observation geometry
Setting	Indoors / tunnels Needs useful texture or geometry	Open outdoors Needs satellite observations

Global navigation satellite system (GNSS): a global coordinate reference.
map: agreed map frame. body: current body frame. Shared positions need a common time.

8.3.1 Simultaneous Localization and Mapping: RTK and Heading

Corrections and Antenna Baselines Provide Different Data

Base: known position
Shared satellite data

→ **Rover A**
Observations + timely corrections

→ **Rover B**
Observations + timely corrections

Real-time kinematic (RTK) positioning:
Centimeter accuracy needs fixed ambiguities and suitable observations.
One base can support multiple rovers.

Baseline: Δx north, Δy east (m).
Zero at north, positive toward east, in rad.

$$\psi_{\text{baseline}} = atan2(\Delta y, \Delta x)$$

Baseline heading: dual antennas. Body yaw: mounting extrinsics. Multipath still matters.

8.3.1 Simultaneous Localization and Mapping: Approaches and Fusion Conditions

Texture, Geometry, and Compute Load Shape the Choice

Approach	Observable Information	Degradation / Cost
Vision-led	Color, texture, classes, and geometric cues	Weak texture, blur, lighting. Training and computation.
LiDAR fusion	Direct range and spatial geometry	Occlusion, rain / fog, geometric degeneracy. Fusion overhead.

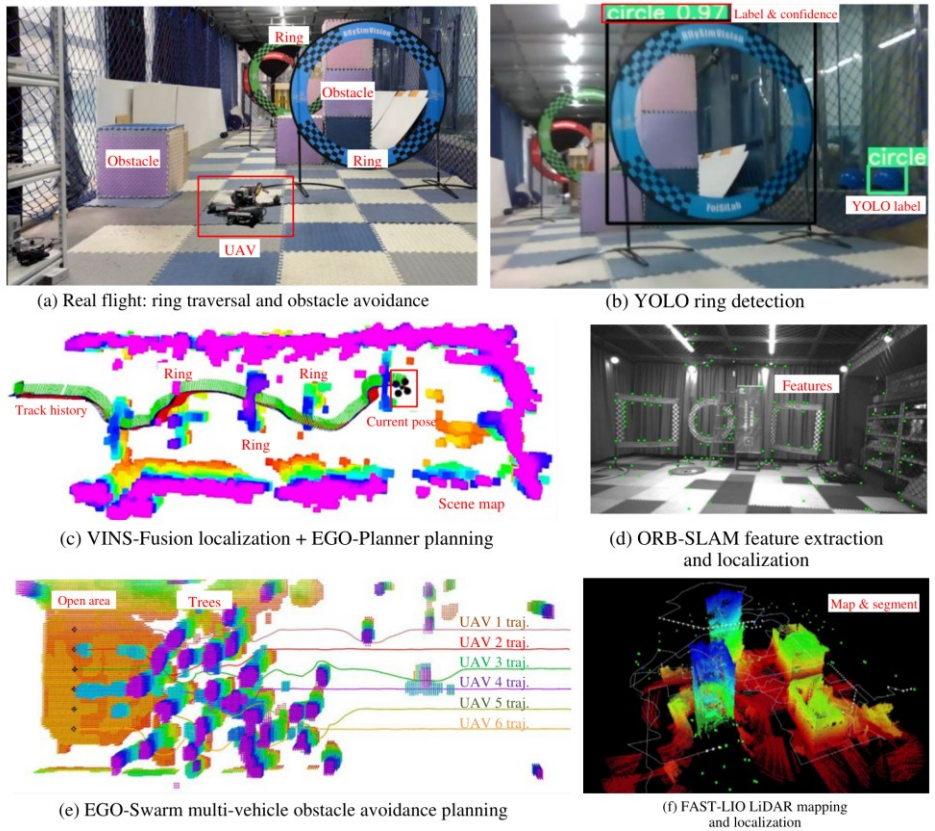
Fusion checks: time synchronization, extrinsics, compute load, failover, and cost.
$v = 1.0\ \mathrm{m/s}$, $\Delta t = 50\ \mathrm{ms} = 0.050\ \mathrm{s}$. Δp is the position offset.

$$\Delta p = v\Delta t = 1.0 \times 0.050 = 0.050\ \mathrm{m}$$

Textbook examples: Tesla vision; sensor fusion at Baidu, NIO, and Xiaomi.

8.3.1 Simultaneous Localization and Mapping: Textbook Figure Overview

Different Algorithms Produce Different Levels of Output



(a) Real flight: ring traversal and obstacle avoidance

(b) YOLO ring detection

(c) VINS-Fusion localization + EGO-Planner planning

(d) ORB-SLAM feature extraction and localization

(e) EGO-Swarm multi-vehicle obstacle avoidance planning

(f) FAST-LIO LiDAR mapping and localization

- **a / b:** real scene / object detection (Section 8.2, slide 10).
- **c / d / f:** visual-inertial localization and planning / visual features / LiDAR-inertial map.
- **e:** multi-vehicle planned trajectories (enlarged on slide 46).

Chapter Outline

Localization, Mapping, Prediction, and Planning

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.3.1 Simultaneous Localization and Mapping

Textbook: Sec. 8.1.2.4

8.3.2 Behavior Prediction and Interaction Modeling

Textbook: Sec. 8.1.2.5

8.3.3 Path Planning and Dynamic Obstacle Avoidance

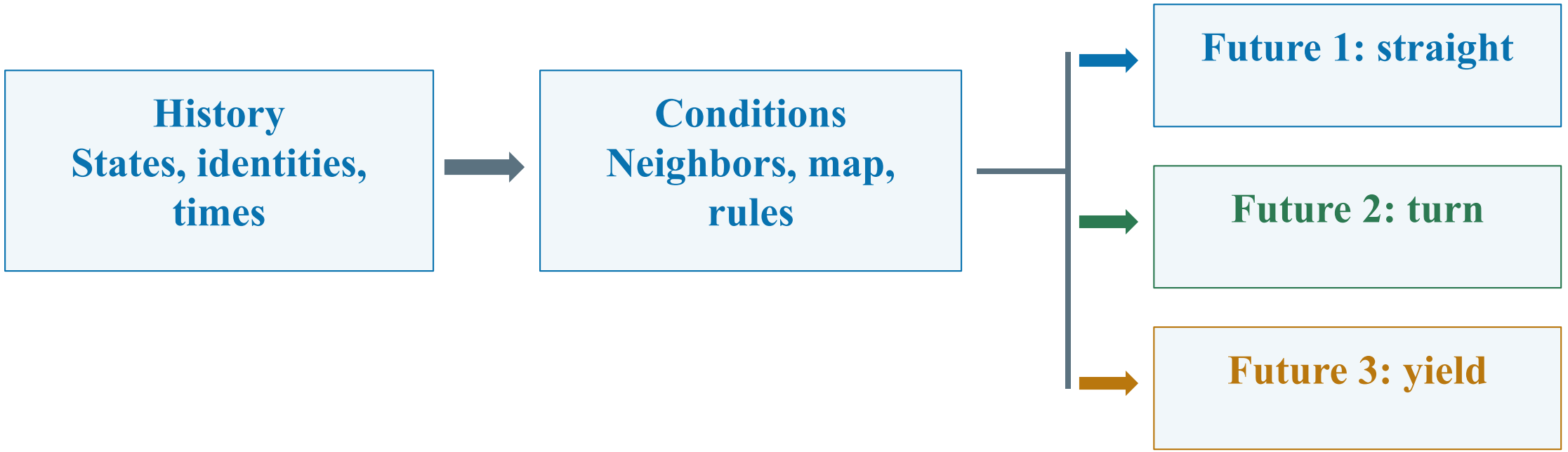
Textbook: Sec. 8.1.2.6

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making

Textbook: Sec. 8.1.2.7

8.3.2 Behavior Prediction and Interaction Modeling: Inputs and Prediction Horizon

History Conditions Several Possible Futures



The prediction horizon sets how far ahead to look. Multiple futures represent possible branches; validate their weights against observed outcomes.

8.3.2 Behavior Prediction and Interaction Modeling: Time Series Models

Hidden States Accumulate History and Gates Control Memory

Model	Full Name	State Mechanism
RNN	Recurrent neural network	Previous hidden state + current observation
LSTM	Long short-term memory network	Memory cell with input, forget, and output gates
GRU	Gated recurrent unit	Update and reset gates control the state

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t)$$

t: observation index. $\mathbf{x}_t$: current observation. $\mathbf{h}_t$: hidden state. f: learned update function.

8.3.2 Behavior Prediction and Interaction Modeling: One-Step Prediction Calculation

A Constant-Velocity Baseline Extends the Latest Speed

History Time (s)	0	1	2
Horizontal x (m)	0	1	2

$$\hat{v} = \frac{2-1}{2-1} = 1 \text{ m/s}$$

$$\hat{x}(2+2) = 2 + 1 \times 2 = 4 \text{ m}$$

Full baseline: 3 m at 1 s ahead, 4 m at 2 s ahead. This assumes constant velocity.

8.3.2 Behavior Prediction and Interaction Modeling: Multi-Target Interaction

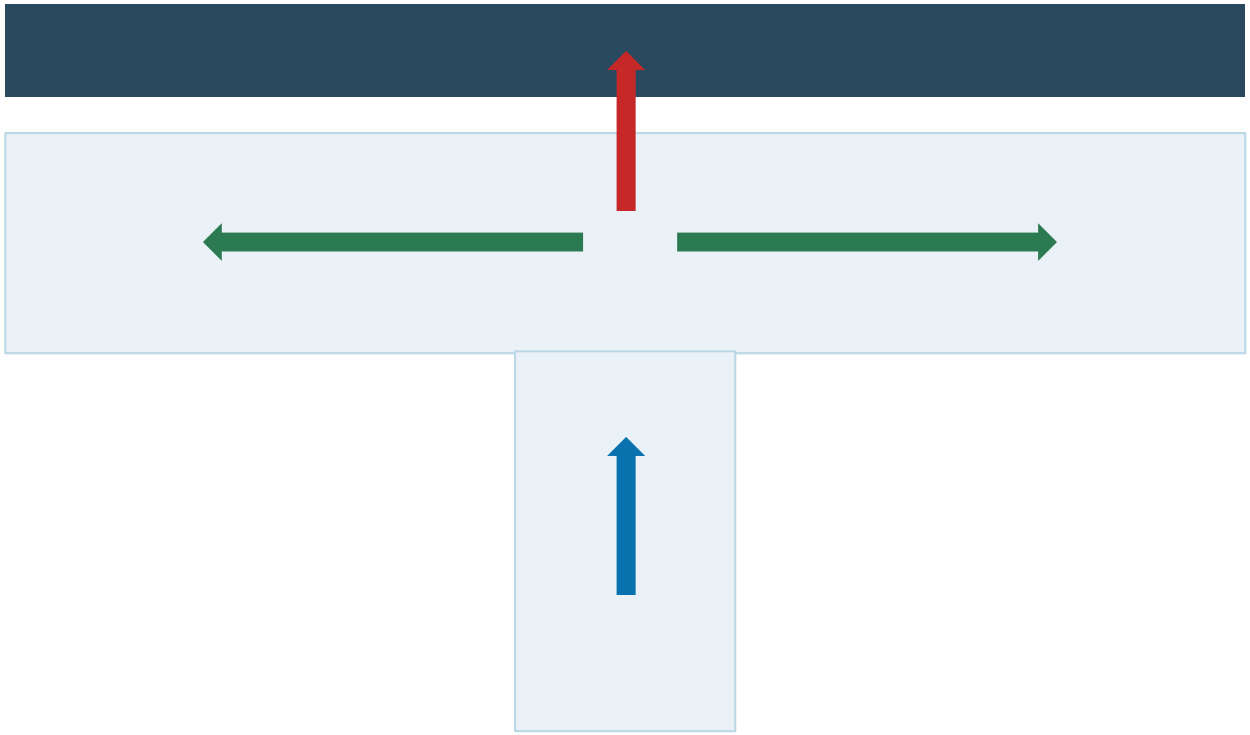
Neighbors Change an Individual Target's Future

Mechanism	Input Relations	Aggregation
Social-LSTM	Past states in a spatial neighborhood	Social pooling of neighbors' hidden states
GNN / GAT	Object nodes and interaction edges	Graph neural / graph attention network: aggregate neighbors with attention weights.
GCN	Graph structure and node states	Graph convolutional network aggregates neighbors

Interaction models change the future distribution. Neighbor weights or predicted trajectories do not prove collision safety.

8.3.2 Behavior Prediction and Interaction Modeling: Scene Constraints

Feasible Futures Stay in Traversable Regions



Green: valid branches. Red: through-wall candidate.

- **Geometry:** candidates cannot pass through obstacles.
- **Rules:** regions and travel directions constrain behavior.
- **Dynamics:** speed, acceleration, and turning constraints still apply.

8.3.2 Behavior Prediction and Interaction Modeling: Multiple Futures and Uncertainty

Parameter Uncertainty and Behavioral Branches

Method	Main Mechanism	Output Meaning
BDL	Bayesian deep learning models parameter distributions	Model uncertainty and predictive distributions
GAN	Generative adversarial network learns diverse samples	Diverse trajectories. Probabilities need calibration.
CVAE	Conditional variational autoencoder uses conditional latent variables	Multiple future modes under given conditions

Multimodal means multiple future modes. Diverse samples do not automatically provide usable risk probabilities.

8.3.2 Behavior Prediction and Interaction Modeling: Risk Calculation

A Lower-Probability Branch Can Determine the Action

Action Cost	Straight: 0.7	Turn: 0.3
Cross now	0	100
Wait	2	2

$$\mathbb{E}[C_{\text{cross}}] = 0.7 \times 0 + 0.3 \times 100 = 30$$

$$\mathbb{E}[C_{\text{wait}}] = 0.7 \times 2 + 0.3 \times 2 = 2$$

Branch probabilities are given. All costs use the same dimensionless scale. Choose to wait.

8.3.2 Behavior Prediction and Interaction Modeling: Results Across Time

New Observations Update Futures and Error Measures

Future Mode	Position at 3 s (m)	Position at 4 s (m)	Weight Update
Straight	(3,0)	(4,0)	0.7→0.2
Left turn	(2,1)	(2,2)	0.3→0.8

Prediction starts at 2 s. A turn observed at 3 s updates the weights (teaching assumption). Average displacement error (ADE): mean pointwise error. Final displacement error (FDE): endpoint error.

Given errors for one candidate at 3 s and 4 s: 0.2 m and 0.4 m.

$$\mathrm{ADE} = \frac{0.2 + 0.4}{2} = 0.3\ \mathrm{m}, \quad \mathrm{FDE} = 0.4\ \mathrm{m}$$

Chapter Outline

Localization, Mapping, Prediction, and Planning

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.3.1 Simultaneous Localization and Mapping

Textbook: Sec. 8.1.2.4

8.3.2 Behavior Prediction and Interaction Modeling

Textbook: Sec. 8.1.2.5

8.3.3 Path Planning and Dynamic Obstacle Avoidance

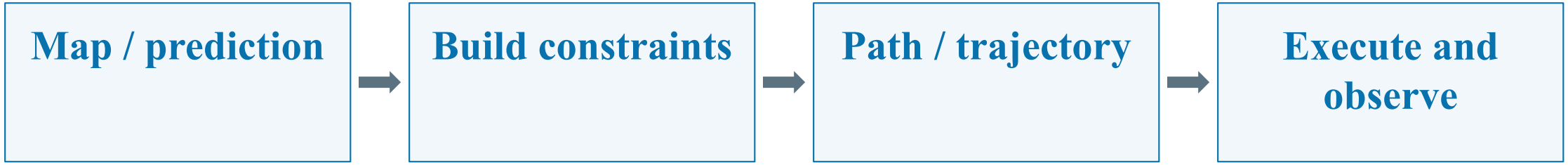
Textbook: Sec. 8.1.2.6

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making

Textbook: Sec. 8.1.2.7

8.3.3 Path Planning and Dynamic Obstacle Avoidance: Paths and Trajectories

Perception Becomes Geometry and Timing Constraints



- **Path:** a sequence of geometric positions. Does it cross obstacles or leave enough clearance?
- **Trajectory:** positions with timing. Do position, velocity, and acceleration meet execution limits?
- **Constraint sources:** occupancy / distance fields, target futures, vehicle radius, control limits, and update delay.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: Rules and Reactive Methods

Potential Fields, Velocity Obstacles, and Roadmaps

Method	What It Acts On	Main Limitations
Artificial potential field (APF)	Attraction / repulsion gives a local motion direction	Local minima, narrow passages, and oscillation
Velocity obstacle (VO)	Relative velocities causing collision within a horizon	Prediction, delay, and dynamic feasibility
Dynamic motion roadmap	Roadmap edges / nodes as collisions change	Needs collision geometry and an update mechanism

Velocity obstacle VO differs from visual odometry VO. Local sensing may omit a global map, but still needs obstacle geometry.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: Grid Search

Accumulated Costs, Heuristics, and Incremental Repair

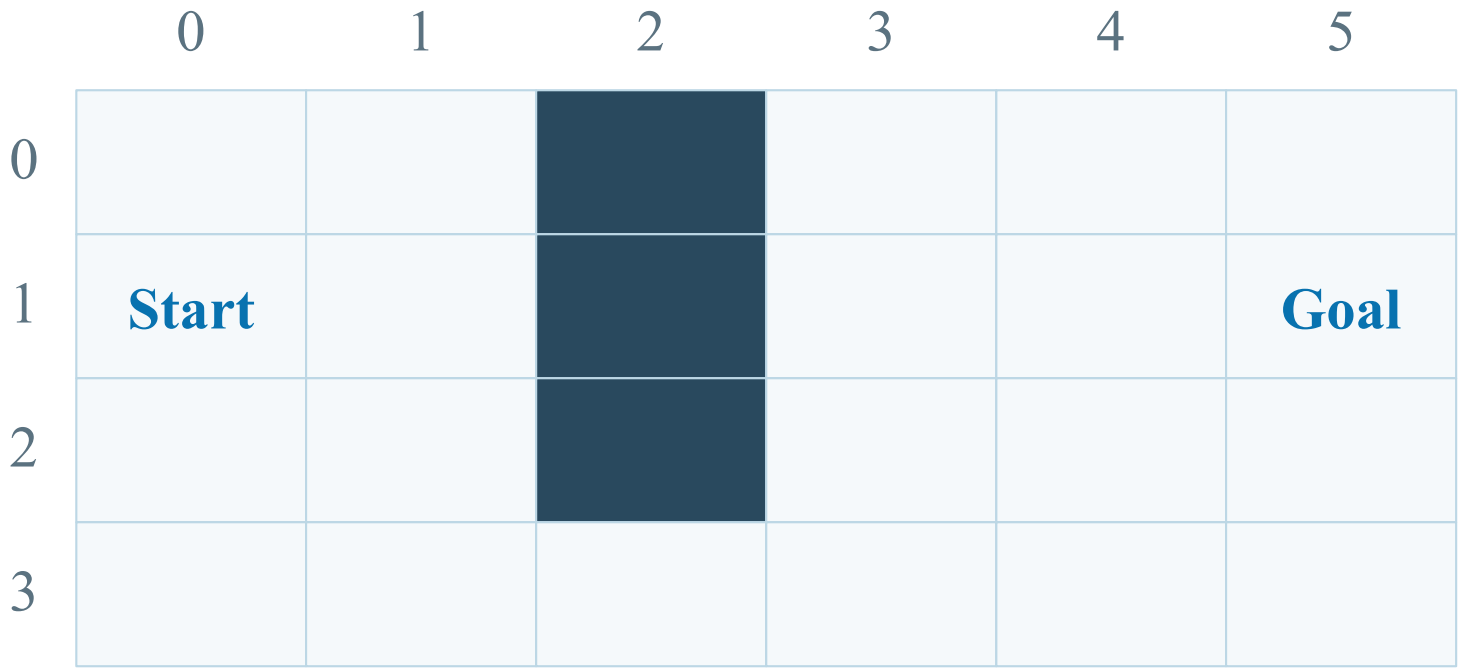
Method	Priority / Mechanism	Conditions / Use
Dijkstra	Expand the lowest accumulated cost g first	Nonnegative edge costs
A*	Accumulated cost + remaining-cost heuristic	No overestimation. Reopen nodes if inconsistent.
D* variants	Incremental repair after graph costs change	Reuses search. No dynamic safety guarantee.

$$f(n) = g(n) + h(n)$$

n: node. g: accumulated cost from the start. h: remaining-cost estimate. f: their sum.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: A* Calculation

Comparing Accumulated Cost and Remaining Estimates

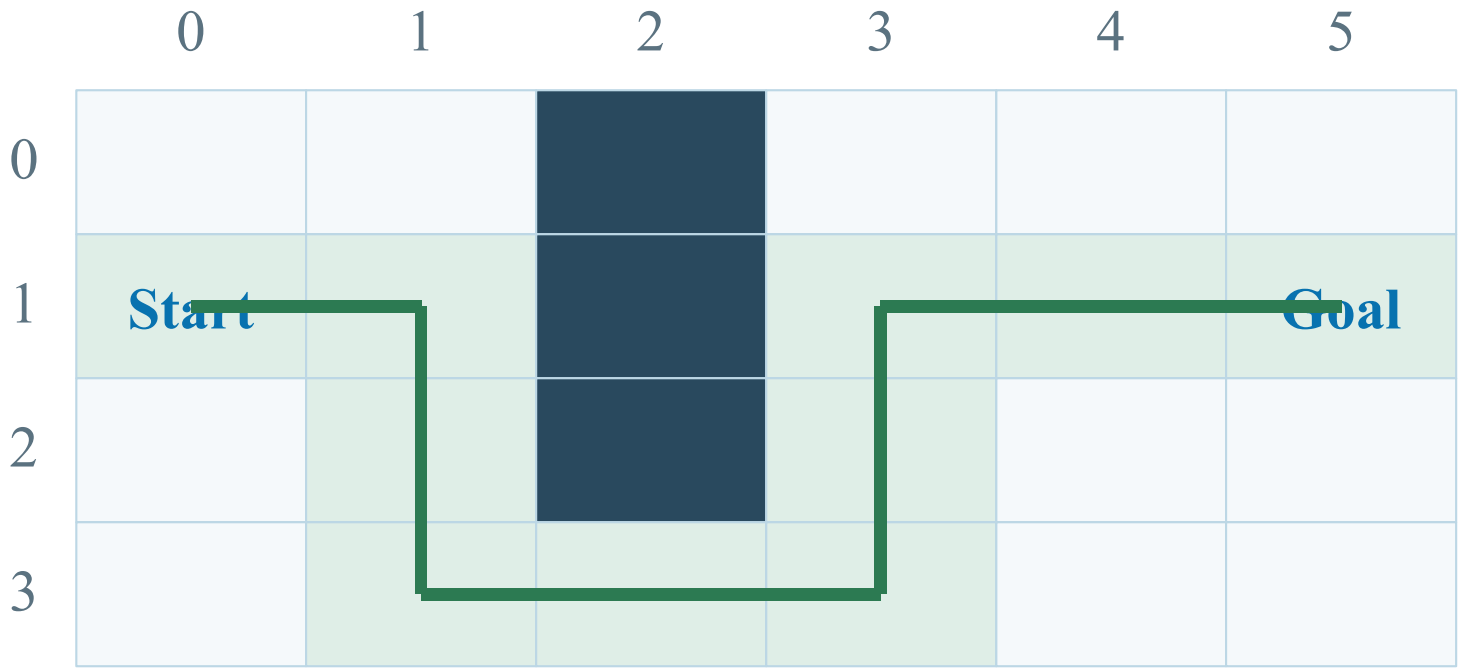


Candidate	g	h	f
(1,1)	1	4	5
(0,0)	1	6	7
(0,2)	1	6	7

Four neighbors, 1 m edges, coordinates (column, row). Dark cells are obstacles.
Manhattan heuristic: remaining horizontal + vertical steps. Expand (1,1), total 5, first.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: Full Search Result

A Nine-Step Detour Is Shortest Under This Cost Model



Detour below the obstacles. Search again after inflating obstacles for the vehicle radius.

Full result: 9 edges, 9 m.
At least 4 vertical steps and 5 horizontal steps.

$$L_{\min} = 4 + 5 = 9\ \mathrm{m}$$

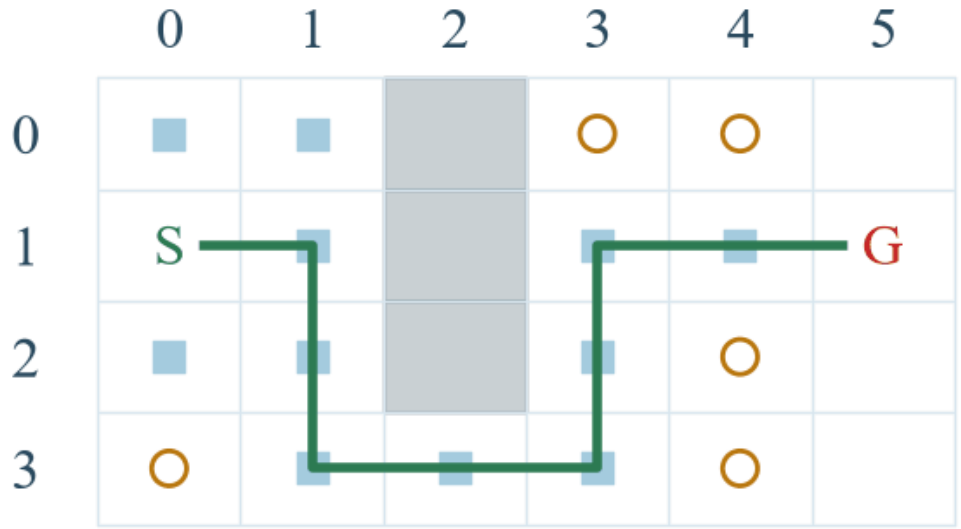
Green: path. Dark: obstacles.
Shortest for this unit-cost grid.
Dynamic feasibility unverified.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: Search Process

A* Expands the Open Set to Find the Nine-Step Detour

Priority: cost + Manhattan heuristic. Pop the goal, then backtrack via parents.

Goal popped; backtracked path is 9 m



Path

$9 = 5 + 4$ m

(column, row); each edge 1 m ■ Expanded ● Open

Step-by-step search on this chapter's 6 × 4 grid. Each edge costs 1 m.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: RRT and RRT*

Tree Expansion Finds Connections and Rewiring Improves Cost

Random sample → **Nearest node** → **Feasible extension** → **Add to tree**

RRT: add an edge after collision checks

RRT*: choose parent + rewire to reduce cost

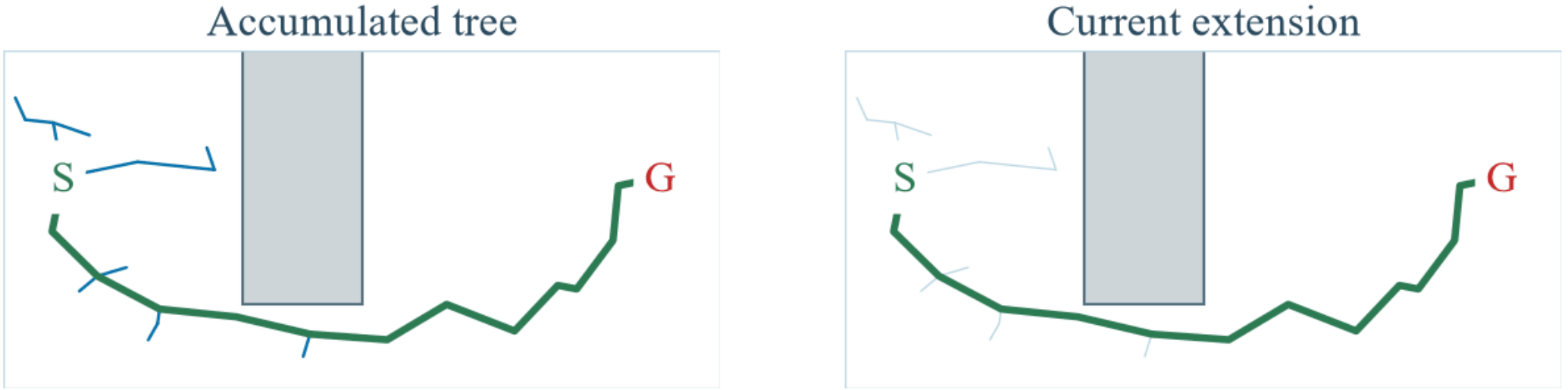
Rapidly exploring random tree (RRT): repeated sampling and extension.
RRT* needs sampling / connection conditions for asymptotic optimality, not finite-trial optimality.
Edges need state-space collision / dynamics checks, beyond straight-line wall tests.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: Random Tree Geometry

RRT Checks the Whole Candidate Segment

Sample, extend from the nearest node, and reject any segment that hits an obstacle.

Connected around the wall; every edge checked



Feasible path 7.68 m

Offline geometric RRT for a point robot on the same obstacle map.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: Trajectory Optimization

Smoothness, Time, and Clearance in Constrained Optimization

$$\min_{\mathbf{p}(t),T} \int_0^T \|\ddot{\mathbf{p}}(t)\|^2 \, dt + \lambda T$$

$$d_{\mathrm{ESDF}}(\mathbf{p}(t)) \geq r + m, \quad \|\dot{\mathbf{p}}(t)\| \leq v_{\max}, \quad \|\ddot{\mathbf{p}}(t)\| \leq a_{\max}$$

Position in m, time in s. $\dot{\mathbf{p}}$ is velocity and $\ddot{\mathbf{p}}$ is acceleration.

$\mathbf{p}(t)$: position trajectory. T: duration. r: vehicle radius. m: safety margin.
$v_{\max}$, $a_{\max}$: speed and acceleration limits. d_{ESDF}: nearest-obstacle distance.
$\lambda \geq 0$: time-cost weight. Larger values favor shorter duration over smoothness.

Given endpoints. Local optimization needs an initial guess. Verify full-trajectory clearance.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: Ant Colony and Particle Swarm

Population Search Still Requires Constraint Checks

Ant colony optimization (ACO)

Construct candidate routes
→ Update pheromones by cost
→ Evaporate and select again
→ Check collisions / limits

Particle swarm optimization (PSO)

Particle position = candidate parameters
→ Personal best + swarm best
→ Update velocity / position
→ Check collisions / limits

Hybrid search changes candidate generation. Safety conditions still come from geometry, timing, and dynamics.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: Clearance and Dynamic Occupancy

Safe Geometry Also Needs Separation in Time

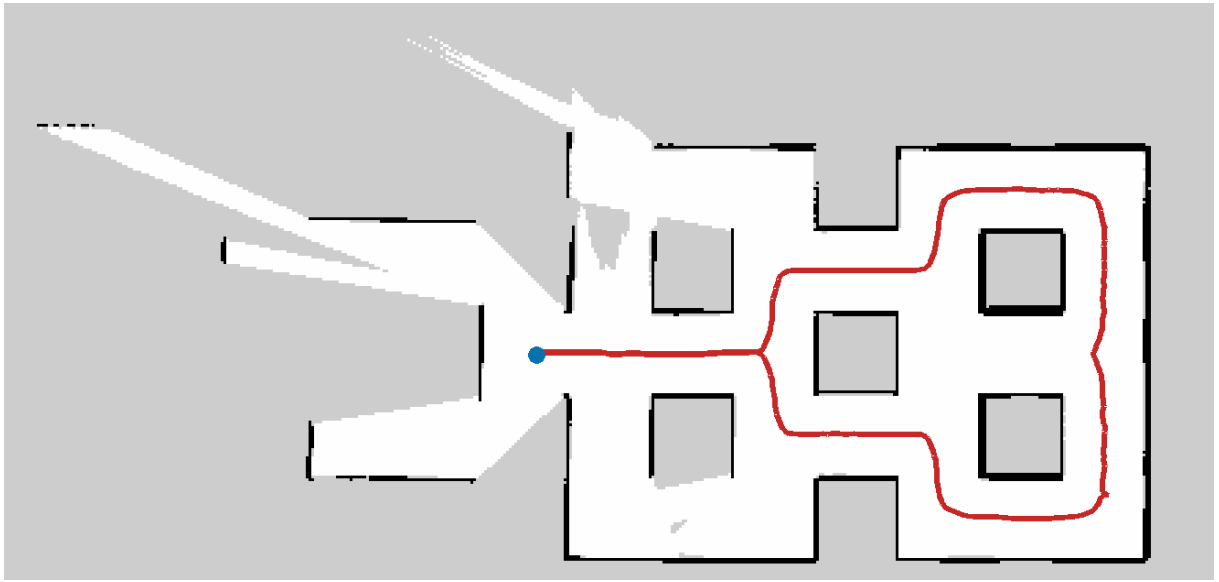
$$d_{\text{safe}} = r + m = 0.25 + 0.15 = 0.40 \text{ m}$$

Current map	→	Predicted arrival time	→	Future occupancy check

- **Static corridor:** for a centered circular footprint, clear width must be at least 0.80 m. Check 3D space and errors too.
- **Moving obstacles:** compare both trajectories at the same time. Include update delay and braking conditions.

8.3.3 Path Planning and Dynamic Obstacle Avoidance: ESDF Example Assets

From Map Coordinates to an Executable Path



Existing map with the saved 448-point path.

Map: map_cropped.pgm
Path: planned_path.npy
Array order: (row, column)
Resolution: 0.05 m/pixel
Not yet in the control frame.

Student entry: [installation directory]/RflySimAPIs/8.RflySimVision/2.AdvExps/e16_ESDFPathPlan/maps/OGM2ESDF.py

Chapter Outline

Localization, Mapping, Prediction, and Planning

Chapter 8 Multimodal Perception and Intelligent Decision-Making

8.1 Image Processing Fundamentals

8.2 Recognition, Tracking, Optical Flow

8.3 Localization, Mapping, and Planning

8.4 Common Perception Sensors

8.5 Coordinate Transformations

8.6 Fusion and Control Decision-Making

8.7 Tools and Framework Interfaces

8.8 Experiments and Real-World Transfer

8.9 Exercises and Classroom Discussion

4 Units in This Section

8.3.1 Simultaneous Localization and Mapping

Textbook: Sec. 8.1.2.4

8.3.2 Behavior Prediction and Interaction Modeling

Textbook: Sec. 8.1.2.5

8.3.3 Path Planning and Dynamic Obstacle Avoidance

Textbook: Sec. 8.1.2.6

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making

Textbook: Sec. 8.1.2.7

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Four Information Roles

Localization, Mutual Perception, Maps, and Decisions

Role	Shared Information	Problem Addressed
Common GNSS reference	Global positions and coordinate datum	Initial position agreement over a wide area
Visual mutual perception	Neighbor relative poses and observations	Short-range relative constraints
Cross-vehicle graph optimization	Keyframes, cross-vehicle edges, local maps	Map alignment and consistency
Collaborative decision-making	Obstacles, targets, and task costs	Allocation / space-time conflicts / control

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Frame Alignment Calculation

Transform a Shared Target into Common Coordinates

2D: x right, y up, counterclockwise positive. Not north-east-down (NED) heading.
Origin of B: (10,2) m. Local x-axis: 90° counterclockwise from the common x-axis.
B's local target is (2,1) m. Rotate first, then translate.

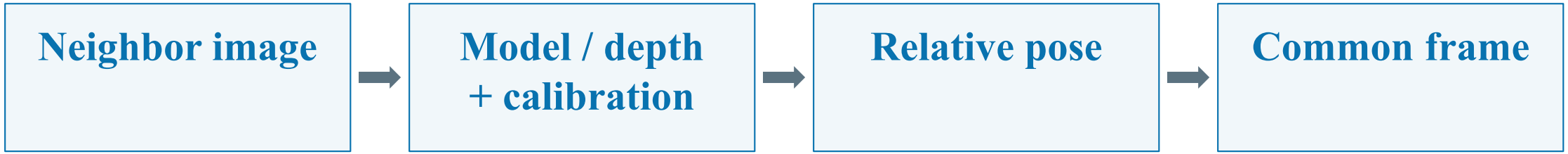
$$\mathbf{p}^{W} = R(90^\circ)\mathbf{p}^{B} + \mathbf{t}_{B}^{W}$$

$$(2,1) \overset{R(90^\circ)}{\rightarrow} (-1,2) \overset{+(10,2)}{\rightarrow} (9,4)\ \mathrm{m}$$

Ignoring rotation gives the wrong result, (12,3) m. Common coordinates also require common time and scale.

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Visual Mutual Perception

Relative Pose Constraints Connect Two Vehicles



$$T_B^W = T_A^W T_c^A T_B^c$$

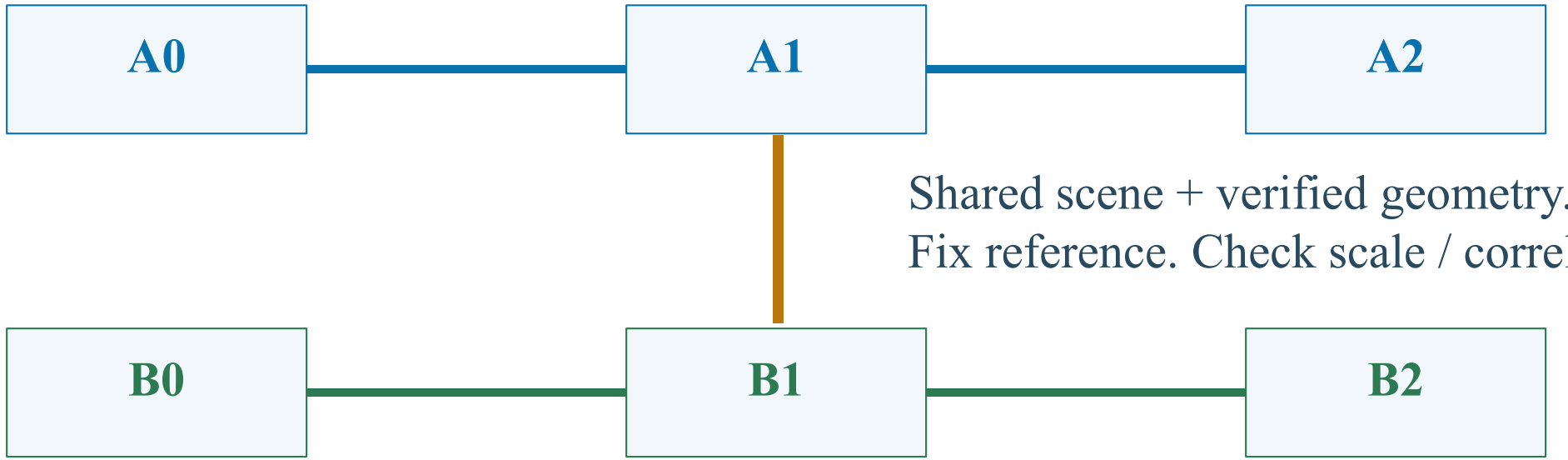
T_b^a maps frame b to a. w: common frame. A/B: body frames. c: camera on A.

- **Conditions:** known model or dimensions, scale, extrinsics, and time. Occlusion can interrupt observations.
- **Output limits:** a 2D box center does not give six degrees of freedom (3 translation + 3 rotation). Relative constraints need error and quality measures.

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Multi-Vehicle Graph Optimization

Cross-Vehicle Edges Align Local Maps

Nodes: keyframe poses. Horizontal: odometry. Vertical: verified cross-vehicle constraints.



Shared scene + verified geometry.
Fix reference. Check scale / correlation.

Sharing keyframes does not guarantee map overlap. An incorrect cross-vehicle edge affects both vehicles' joint estimate.

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Fusion Calculation

Weight Independent Estimates by Their Error Scale

Given: 9.0 m with standard deviation 0.2 m, and 9.4 m with standard deviation 0.4 m. Same quantity and time. Unbiased and mutually independent.

$$\hat{x} = \frac{9.0/0.2^2 + 9.4/0.4^2}{1/0.2^2 + 1/0.4^2} = 9.08\ \text{m}$$

$$\sigma = \sqrt{\frac{1}{1/0.2^2 + 1/0.4^2}} \approx 0.179\ \text{m}$$

Data from a common source or previous fusion are correlated. Do not count them again as new independent observations.

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Timing and Stale Information

Common Coordinates Also Need a Common Time

$$\Delta p = v\Delta t = 2 \times 0.100 = 0.20 \text{ m}$$

At 2 m/s, a state aged 100 ms differs by 0.20 m without propagation.

Observation time → **Timestamp and state propagation** → **Common current time**

- **Message fields:** frame / scale, observation timestamp, source / covariance or quality, and validity period.
- Check transmission delay, clock offset, and shared-data correlation. Apply expiry rules after communication loss.

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Task Allocation Calculation

Joint Allocation Can Outperform Vehicle-by-Vehicle Greed

Task Cost	Task T1	Task T2
Vehicle A	2	3
Vehicle B	2.5	100

$$C_{\text{greedy}} = 2 + 100 = 102$$

$$C_{\text{joint}} = 3 + 2.5 = 5.5$$

One task per vehicle; dimensionless costs. Greedy: A first. Joint: compare both assignments.

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Decisions Across Time

Assigned Goals Still Require Intersection Coordination

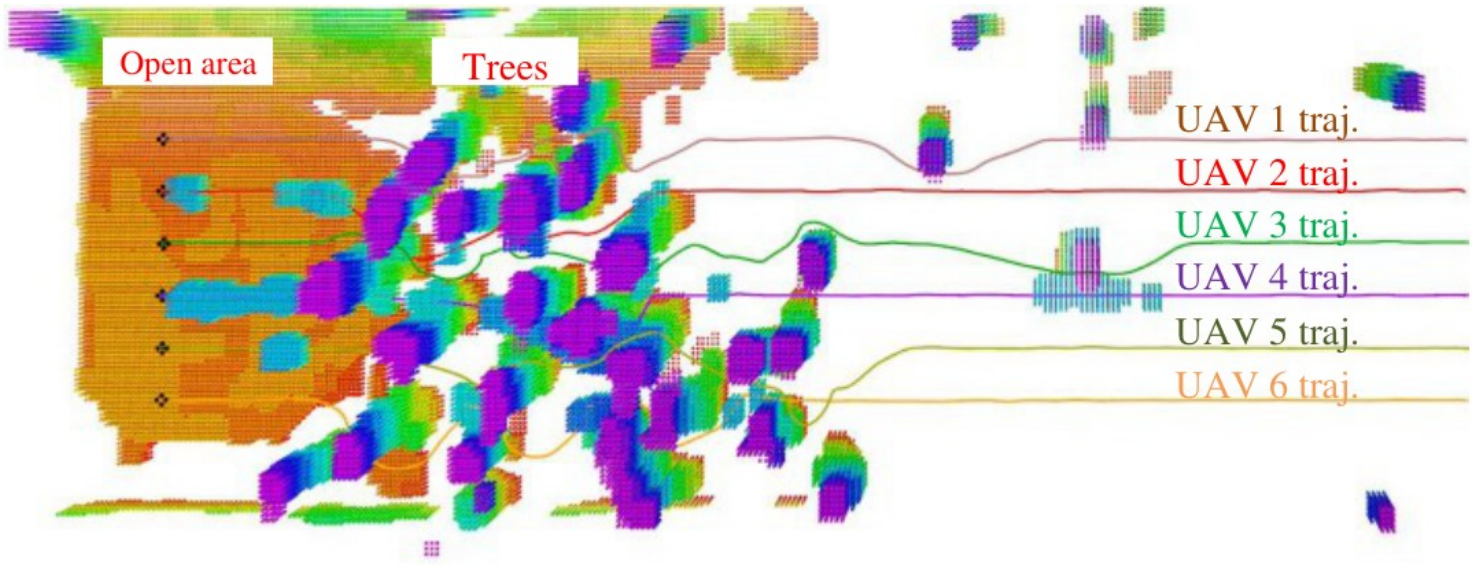
Share states → **Allocate tasks** → **Plan paths** → **Time windows** → **Feedback update**

Time (s)	Vehicle A	Vehicle B
0	Approaching intersection	Approaching hold point
1	Occupying intersection	Waiting outside intersection
2	Has exited	Entering intersection
3	Continuing to goal	Has exited

Exit before entry. Set safety gaps from vehicle size, braking, and information validity.

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Textbook Multi-Vehicle Figure

Multiple Trajectories Show Multi-Vehicle Avoidance Planning



Shared map: geometry.
Relative states: vehicle relations.
Common time: occupancy conflicts.
Limits: each vehicle's capability.

Fig. 8.3(e) Multi-vehicle trajectories. Full figure: slide 14.

A planning image does not verify collision-free execution. Evaluation also needs execution states, timing, and separation.

8.3.4 Multi-Vehicle Perception Fusion and Decision-Making: Centralized and Distributed Architectures

Communication and Solver Location Shape Collaboration

Aspect	Centralized	Distributed
Solver location	Center handles joint allocation / scheduling	Nodes solve locally and exchange constraints
Information	Uploaded states. Center combines group information.	Neighbor information. Asynchronous, partial observations.
Main risks	Communication / compute bottlenecks, center failure	Consistency, local conflicts, stale information

Both need timestamps, expiry, conflict rules, and communication-loss behavior. Model communication conditions in decisions.

Exercises and Discussion

Conditions for Maps, Futures, and Collaboration

- **Question 1:** a similar corridor triggers a loop closure candidate. Can the trajectory be corrected immediately? What evidence is still needed?
- **Question 2:** obstacle avoidance needs clearance to the nearest obstacle. Should it query TSDF or ESDF?
- **Question 3:** straight / turn probabilities are 0.7 / 0.3. Crossing costs 0 / 100. Waiting costs 2 in either case. Which action?
- **Question 4:** x right, y up, counterclockwise positive (slide 39). B's origin is (10,2) m, local x-axis rotated 90°, target (2,1) m. Find common coordinates. Compare assignments using slide 44.

Answer Points

Explaining Results with Observations, Frames, and Constraints

- **Answer 1:** verify geometry before adding the loop closure constraint. Repeated texture can create false candidates.
- **Answer 2:** query ESDF for nearest-obstacle distance. TSDF reconstructs surfaces and does not replace clearance over all space.
- **Answer 3:** expected crossing cost is 30, waiting cost is 2. The most likely future omits a lower-probability, high-loss branch.
- **Answer 4:** common target (9,4) m. A → T2, B → T1 costs 5.5 in total, versus 102 for greedy. Timing still needs coordination.

Section Summary

Local Observations Support Feasible Joint Action

- **Localization and maps:** joint estimation, map representations, and loop closure constraints. Local consistency does not provide global coordinates.
- **Behavior prediction:** history, interactions, scene, and multiple futures. Samples do not automatically give calibrated probabilities.
- **Planning and avoidance:** geometric paths plus time and limits. Graph search, sampling, and optimization have different conditions.
- **Multi-vehicle collaboration:** common frames, scale, time, and provenance. Joint allocation still needs path and occupancy coordination.