

# Frontier Selection for Unknown-World Navigation — One Pager

**Context.** Single-loop autonomous navigation with ICP-aided localization, OGM mapping, and frontier-based exploration. This note summarizes the frontier selector implemented only inside `determine_frontier_path(...)`.

## 1) Key Mechanisms & Rationale

**Forward progression bias.** Prefer frontiers within a front cone (60° primary, 90° relaxed) around the robot heading to avoid backward oscillations.

**Goal-seeking when visible.** If the goal is line-of-sight clear in the OGM, directly select the goal (A\* still respects obstacles).

**Stability.** Sticky target (keep previous frontier until close) and a short cooldown (8 steps) penalize immediate backtracking.

**Determinism & safety.** Ties are broken lexicographically; paths are always planned on the OGM.

## 2) Scoring & Weights (condensed)

Term	Meaning	Weight (↓ better)
d_goal	Distance from candidate to goal (meters / cell_size)	0.70
bfs	BFS steps from robot to candidate (reachability cost)	0.30
$\Delta\theta$	Heading change toward the candidate (radians / $\pi$ )	0.15
d_robot	Distance from robot to candidate (meters / cell_size)	0.10
penalty	Cooldown if revisiting same frontier within 8 steps	+ up to 5.0

Objective:  **$S = 0.70 \cdot d\_goal + 0.30 \cdot bfs + 0.15 \cdot \Delta\theta + 0.10 \cdot d\_robot + penalty$** . Candidates are filtered by the forward-cone (prefer 60°, fallback 90°, then any). Choose the minimum score; break ties by coordinates.

## 3) Selection Flow (pseudocode)

```
if LOS_to_goal_OGM() and start_cell != goal_cell: target = goal else: # keep
previous target until close if prev_target in frontiers and dist(robot,
prev_target) > 0.35·cell_size: target = prev_target else: buckets =
{primary_60°, relax_90°, any} for f in frontiers: if f == start_cell: continue
s = 0.70·d_goal + 0.30·bfs + 0.15· $\Delta\theta$  + 0.10·d_robot + cooldown_penalty(f) put
(s,f) into bucket by heading  $\Delta$  target = argmin_score(primary or relax or any)
record_visit(target); state.frontier_goal = target state.path =
A_star_unknown_world(start_cell, target)
```

## 4) Complexity & Limitations

**Complexity.** Frontier detection BFS:  $O(V)$ . Scoring adds  $O(|F|)$  with constant-time terms. Overall linear in explored free cells.

**Limitations.** Cone angles and weights are heuristic; extreme narrow corridors may still cause brief stalls. A learning-based selector or frontier clustering could further improve decisions.