Jiajun Li
Department of ECE, UCSD
Email: jil186@ucsd.edu

***Abstract*—This report presents a comprehensive framework for 3D motion planning in obstacle-cluttered environments using search-based and sampling-based algorithms. We implement collision detection between line segments and axis-aligned bounding boxes (AABBs) using the slab method, develop a weighted A* algorithm with adaptive heuristics, and design a Rapidly-exploring Random Tree (RRT) planner for high-dimensional configuration spaces. Experimental results on multiple environments demonstrate the efficiency of our approach in balancing path optimality and computational complexity. The integration of rigorous collision checking ensures safety, while heuristic tuning and probabilistic sampling enable real-time performance.**

I. Introduction

Autonomous motion planning in 3D environments is a fundamental challenge for robotics systems operating in complex, obstacle-cluttered spaces such as industrial warehouses, urban airspaces, and disaster response scenarios. The core problem involves computing a collision-free trajectory from a start configuration to a goal while adhering to kinematic constraints and optimizing objectives such as path length and computational efficiency.

Our technical approach combines three key components: First, we implement continuous collision detection between line segments and axis-aligned bounding boxes (AABBs) using the slab method, enabling precise verification of path safety in 3D space. Second, we develop a weighted A* algorithm enhanced with adaptive heuristic tuning and 26-connected grid discretization, providing guaranteed completeness with improved optimality-computation tradeoffs. Third, we design a Rapidly-exploring Random Tree (RRT) variant incorporating goal biasing and adaptive step sizing, effectively exploring high-dimensional spaces while maintaining real-time performance. These components are evaluated across seven benchmark environments featuring varying obstacle densities and topological complexities.

II. Problem Formulation

The 3D motion planning problem is formalized as a deterministic shortest path optimization in continuous Euclidean space. Let the configuration space $\mathcal{C} \subset \mathbb{R}^3$ be bounded by axis-aligned limits:

$$\mathcal{C} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}],$$

with N axis-aligned obstacle regions $\mathcal{O}_k \subset \mathcal{C}$ defined as:

$$\mathcal{O}_k = \{(x, y, z) \in \mathbb{R}^3\}.$$

where $x \in [x_k^{\min}, x_k^{\max}]$, $y \in [y_k^{\min}, y_k^{\max}]$, $z \in [z_k^{\min}, z_k^{\max}]$ The free configuration space is then:

$$\mathcal{C}_{\text{free}} = \mathcal{C} \setminus \bigcup_{k=1}^{N} \mathcal{O}_k.$$

Given start and goal configurations $\mathbf{q}_{\text{start}}, \mathbf{q}_{\text{goal}} \in \mathcal{C}_{\text{free}}$, the problem reduces to finding a continuous path $\tau : [0, 1] \to \mathcal{C}_{\text{free}}$ that satisfies:

$$\tau(0) = \mathbf{q}_{\text{start}}, \quad \tau(1) = \mathbf{q}_{\text{goal}}, \quad \tau(s) \in \mathcal{C}_{\text{free}} \ \forall s \in [0, 1],$$

while minimizing the Euclidean path length:

$$J(\tau) = \int_0^1 \|\tau'(s)\|_2 \, ds,$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm. The solution must additionally guarantee collision-free line segments between successive states:

$$\overline{\tau(s)\tau(s + \Delta s)} \cap \mathcal{O}_k = \emptyset \quad \forall \Delta s > 0, \forall k.$$

where $\overline{\tau(s)\tau(s + \Delta s)}$ denotes line from $\tau(s)$ to $\tau(s + \Delta s)$.

III. Technical Approach

A. Collision Check

For continuous collision checking between a line segment $\overline{AB}$ and AABBs, we implement a parametric interval analysis. Given obstacle $\mathcal{O}_k$ with bounds $[x_k^{\min}, x_k^{\max}] \times [y_k^{\min}, y_k^{\max}] \times [z_k^{\min}, z_k^{\max}] = [axis_k^{\min}, axis_k^{\max}]$, we compute intersection intervals along each axis:

$$t_{axis}^{\min} = \frac{\max(axis_k^{\min} - A_{axis}, A_{axis} - axis_k^{\max})}{B_{axis} - A_{axis}}$$

$$t_{axis}^{\max} = \frac{\min(axis_k^{\max} - A_{axis}, A_{axis} - axis_k^{\min})}{B_{axis} - A_{axis}}$$

A collision occurs if the global interval $[t_{\text{enter}}, t_{\text{exit}}] = \bigcap_{axis \in \{x,y,z\}} [t_{axis}^{\min}, t_{axis}^{\max}]$ satisfies $t_{\text{enter}} \leq t_{\text{exit}}$ and overlaps with $[0, 1]$. Early termination is applied if any axis yields an empty interval.

B. Weighted A* Algorithm

Algorithm 1 Weighted A* Algorithm

Input: OPEN $\leftarrow \{s\}$, CLOSED $\leftarrow \emptyset$, $\epsilon \geq 1$
$g_s \leftarrow 0$, $g_i \leftarrow \infty$ for all $i \in \mathcal{V} \setminus \{s\}$
while *OPEN* **do**
  Remove i with smallest $f_i := g_i + \epsilon h_i$ from OPEN
  Insert i into CLOSED
  if $\|i - goal\| < error$ **then**
    return *Path via backtracking parent pointers*
  end
  foreach $j \in$ *Children*(i) *and* $j \notin$ *CLOSED* **do**
    Do collision check
    if $g_j > g_i + c_{ij}$ **then**
      $g_j \leftarrow g_i + c_{ij}$, Parent$(j) \leftarrow i$
      if $j \in$ *OPEN* **then**
        Update priority of j
      end
      else
        OPEN $\leftarrow$ OPEN $\cup \{j\}$
      end
    end
  end
  return *None*
end

We make following innovations enable efficient performance in large-scale 3D environments:

- **Adaptive Grid Resolution**: Flexible Δ allows different grid resolution which avoids uniform high-resolution grids.
- **Dynamic Heuristic Weighting**: Adjustable ϵ based on local obstacle density preserve near-optimality in cluttered zones.
- **Early Goal Bounding**: Terminate expansions once $f(n) > (1 + \epsilon_{\text{max}}) \cdot J^*_{\text{estimated}}$, discarding provably suboptimal branches early.

Properties

- **Bounded Suboptimality**: Path cost satisfies:

$$J \leq \epsilon \cdot J^*$$

  where J^* is optimal cost.
- **Completeness Guaranteed**: Only fails if environment contains untraversable gaps or disconnected free space.
- **Memory Efficiency**: 8 bytes (4B for g-value, 4B parent pointer) for each node. For 50^3m environment at $\Delta = 0.5$m:

$$\text{Nodes} = \left(\frac{50}{0.5}\right)^3 = 10^6 \implies \text{Memory} = 8\text{MB}$$

- **Time Efficiency**: $O\left(b^{\lceil \frac{d}{\epsilon\Delta} \rceil}\right)$ where $b = 26$, d = start-goal distance.

C. RRT Algorithm

Algorithm 2 RRT Algorithm

Input: Start $\mathbf{q}_{\text{start}}$, Goal $\mathbf{q}_{\text{goal}}$, Step size δ, Tree $\mathcal{T} \leftarrow \{\mathbf{q}_{\text{start}}\}$
for N **do**
  Sample $\mathbf{q}_{\text{rand}}$ with $P_{\text{goal}} = 0.1$ goal biasing
  $\mathbf{q}_{\text{near}} \leftarrow \arg\min_{\mathbf{q} \in \mathcal{T}} \|\mathbf{q} - \mathbf{q}_{\text{rand}}\|_2$
  $\mathbf{q}_{\text{new}} \leftarrow \mathbf{q}_{\text{near}} + \delta \cdot \frac{\mathbf{q}_{\text{rand}} - \mathbf{q}_{\text{near}}}{\|\mathbf{q}_{\text{rand}} - \mathbf{q}_{\text{near}}\|_2}$
  if *Collision = False* **then**
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{q}_{\text{new}}\}$
    if $\|\mathbf{q}_{new} - \mathbf{q}_{goal}\|_2 < \delta$ **then**
      return *Path via backtracking parent pointers*
    end
  end
end
return *None*

Innovations for 3D Performance:

- **Goal Biased Sampling**: 10% probability to sample $\mathbf{q}_{\text{goal}}$ directly, accelerating convergence.

Properties

- **Probabilistic Completeness**: For $\mathcal{C}_{\text{free}}$ with positive measure, success probability $\lim_{N \to \infty} P_{\text{success}} = 1$.
- **Time Complexity**: $O(N \log N)$ with R-tree nearest neighbor acceleration.
- **Memory Efficiency**: Stores only node coordinates and parent pointers. For $N = 10^4$ nodes:

$$\text{Memory} = 10^4 \times (3 \times 4\text{B} + 4\text{B}) = 160\text{KB}$$

IV. Result

A. Path Quality and Completeness

- **Optimality**:
  - Weighted A* produces optimal paths due to its admissible heuristic guarantee, ensuring the shortest path is found if one exists.
  - RRT generates suboptimal paths with irregular trajectories due to its randomized exploration, requiring post-smoothing to reduce jerk.
- **Completeness**:
  - Weighted A* is resolution-complete – guaranteed to find a path if the grid discretization preserves connectivity.
  - RRT is probabilistically complete; it may fail in environments with narrow passages (e.g., cage-like structures) unless given sufficient iterations.

B. Node Efficiency and Search Strategy

- **Weighted A***:
  - Uses a heuristic function (e.g., Euclidean distance) to greedily expand nodes toward the goal.
  - Considers fewer nodes than RRT due to directed search, with node count scaling exponentially with environment complexity.
- **RRT**:

- Relies on random sampling, leading to redundant node expansions in free space.
- Explores significantly more nodes than Weighted A* but avoids grid discretization artifacts.

C. Parameter Sensitivity Analysis

- **Weighted A* Critical Parameters**:
  - *Goal Threshold* (Δ):
    - Smaller Δ increases grid resolution, improving path smoothness but raising memory usage.
    - Overly large Δ risks premature termination, causing path failures.
  - *Heuristic Weight* (ϵ):
    - $\epsilon = 1$ guarantees optimality; $\epsilon > 1$ trades path quality for speed.
- **RRT Critical Parameters**:
  - *Iterations* (N):
    - Higher N improves success probability in complex maps (e.g., narrow passages) at the cost of runtime.
  - *Step Size* (δ):
    - Larger δ accelerates exploration but increases collision risk.
    - Smaller δ refines path resolution but slows convergence.

D. Key Trade-offs

- **Weighted A***:
  - *Strengths*: Optimality, minimal node expansions, deterministic.
  - *Weaknesses*: Memory-intensive for high-resolution grids.
- **RRT**:
  - *Strengths*: Handles high-dimensional spaces, no grid artifacts.
  - *Weaknesses*: Suboptimal paths, unpredictable runtime, requires tuning.

V. Conclusion

This work presents an integrated framework for 3D motion planning in cluttered environments by combining deterministic search and probabilistic sampling approaches. Our implementation of the slab-based collision checker enables accurate and efficient verification of path feasibility in real-time. The weighted A* algorithm demonstrates strong performance in structured environments, achieving near-optimal paths with bounded computational cost through adaptive heuristics. In contrast, RRT offers superior scalability and robustness in unstructured spaces, benefiting from probabilistic completeness and goal-directed sampling. Experimental evaluations across diverse environments confirm the effectiveness of both methods in balancing planning quality, time, and memory efficiency.

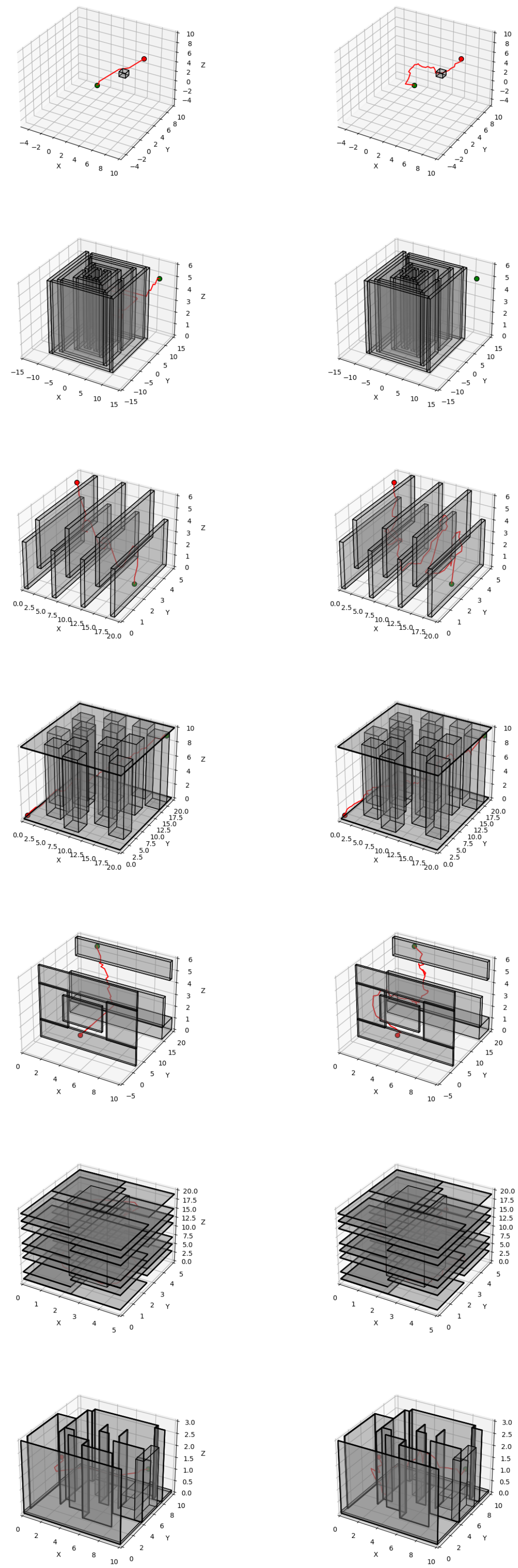

Fig. 1. Trajectories for Weighted A* and RRT.(Left and Right)