

Radio Coverage-Aware UAV Planning and Navigation in Urban Scenarios

D. Aláez*, M. Prieto, J. Villadangos, J.J. Astrain
Statistics, Computer Science and Mathematics Department,
Universidad Pública de Navarra, Pamplona 31006, Spain

ABSTRACT

We propose a novel UAV path planning and navigation solution for urban environments, integrating radio path loss into a modified A* path planning algorithm. The proposed algorithm is based on a fast raytracing algorithm, which enables any 3D scenario to be modeled for specific radio link conditions. The radio coverage map is then fed along with an obstacle occupancy grid to the algorithm to find the optimal path that avoids obstacles and minimizes path loss. Additionally, we introduce a smoothing step to generate UAV-suitable paths while maintaining radio performance. To verify the implementation, a test case environment has been proposed in the campus of the Public University of Navarra. According to the proposed test cases, the overall path loss is reduced up to 24.12% by using our methodology.

1 INTRODUCTION AND RELATED WORKS

Autonomous aerial mobility in densely built environments is rapidly transforming urban transportation, providing novel approaches to alleviating traffic congestion and improving logistical efficiency. Integrating Unmanned Aerial Vehicles (UAVs) into urban airspace offers significant potential for accelerating deliveries, transporting medical supplies, and enhancing emergency response capabilities.

Effective UAV navigation in urban scenarios presents significant challenges due to the presence of obstacles such as buildings and the need to maintain reliable radio communication. Ensuring optimal path planning that balances obstacle avoidance and communication quality is critical, especially in communication-sensitive tasks. According to [1], nearly a third of civil UAV accidents experienced communication loss.

A growing body of research examines the integration of collision avoidance strategies within path planning, ensuring safety during UAV operations. López et al. propose collision risk management strategies for multi-UAV systems, where autonomous navigation is critical to avoid obstacles and maintain airspace safety [2]. The utilization of artificial intelligence (AI) in path planning can also enhance

the responsiveness of UAVs to dynamic urban environments through real-time adjustments and risk-aware strategies, as demonstrated in [3]. These advancements are increasingly essential as the complexity of urban environments demands sophisticated algorithms capable of adjusting to sudden changes during flight.

The safety of urban air mobility is further compromised by the risk associated with third-party damage. Path planning must account for third-party risk while ensuring compliance with regulations governing UAV operations, particularly in urban areas crowded with obstacles. Path planning methodologies that consider the avoidance of high-risk zones are gaining attention as effective strategies for mitigating these risks; Tang et al. introduce a risk modeling approach to inform UAV flight paths in urban settings [4].

Traditional path planning algorithms such as Dijkstra's [5], A* [6], and RRT* [7] prioritize spatial optimization, often neglecting radio coverage considerations, which can lead to signal degradation in dense urban areas. A systematic review on recent advances in autonomous mobile robot navigation can be found in [8]. Other authors propose radio map-based UAV path planning [9, 10, 11], but rely on statistical models or only consider topographic data.

To address this gap, we propose a novel modification of the A* algorithm that incorporates radio propagation data, derived from a fast raytracing simulator based on a Multipath Radio Tracer (MURT [12]) for 3D urban photogrammetry data. By integrating path loss (PL) values directly into the cost function of the A* algorithm, our approach generates paths that not only avoid obstacles but also maximize signal strength. In addition, a smoothing step is applied to enhance the trajectory of the UAV while respecting the radio coverage requirements. This method ensures efficient UAV navigation, making it highly suitable for applications such as UAV-assisted wireless networks and urban emergency operations. A simplified overview of the proposed contributions, organized by steps is shown in Figure 1.

2 MATERIALS AND METHODS

Let the environment be represented by a discrete voxel grid $G(i, j, k)$, where each voxel represents a position in the 3D space. Let path loss $PL(i, j, k)$ be a scalar value associated with each voxel. The goal is to find the path that minimizes the overall PL between start location $s = (x_s, y_s, z_s)$ and final location $e = (x_e, y_e, z_e)$.

*Email address(es): contact_author@mail.com

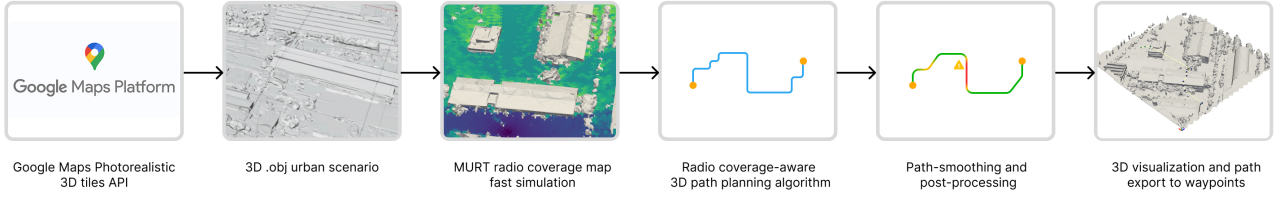


Figure 1: Step by step process diagram for obtaining a radio coverage-aware path plan for UAV flight in dense urban scenarios.

PL is the attenuation of the signal as it propagates through space and is affected by: (a) grid location: the position of the voxel (i, j, k) ; (b) transmitter location: the position of the transmitter (x_{tx}, y_{tx}, z_{tx}) as the source of the radio control signal; (c) frequency: the frequency of the signal impacts how it interacts with materials and how it propagates through space; (d) material properties: different material properties cause different levels of absorption, reflection, and diffraction of the signal, based on their electromagnetic properties (dielectric constant, permeability, conductivity); (e) obstacles: obstacles between transmitter and receiver (e.g. walls, buildings) affect the direct line-of-sight path, leading to additional losses due to shadowing effects and multipath; (f) interferences caused by other emitting sources such as other aircrafts.

2.1 Obtaining a PL model for a given scenario

Using Google's Photorealistic 3D tiles API¹, urban scenarios can be downloaded for a set of coordinates as .obj files that may be directly imported into the simulator. The mesh is then voxelized to calculate the number of points to be simulated using trimesh [13]. This voxelized map facilitates obtaining the binary occupancy grid that will later be used to search for optimal collision-free paths. For fast computation of PL in all voxels, a modified version of MURT is used [12]. MURT is a Python ray-tracing engine for multipath propagation of radio waves, with the ray-tracing core engine implemented in Python C++ Extension. This ray tracing algorithm computes the PL for the most common propagation mechanisms: direct line-of-sight, reflection, and diffraction [14].

The **direct PL** calculation is based on the Friis transmission equation[15], and is used to compute the free-space path loss (FSPL) over a direct line-of-sight path between the transmitter (x_{tx}, y_{tx}, z_{tx}) and the receiver (x_{rx}, y_{rx}, z_{rx}) :

$$PL_{LOS}(d, f) = 20 \log_{10}(d) + 20 \log_{10}(f) - 20 \log_{10} \left(\frac{4\pi}{c} \right), \quad (1)$$

where d is the Euclidean distance between the transmitter and the receiver, f is the transmission frequency in Hz, and c is the speed of light in m/s.

¹Google Photorealistic 3D Tiles official website: <https://developers.google.com/maps/documentation/tile/3d-tiles>

The **reflection PL** accounts for the reflection of the signal off a surface, using Fresnel reflection coefficients. The reflection coefficient depends on the angle of incidence computed using Snell's law: $n_1 \sin \theta_1 = n_2 \sin \theta_2$, where n_1 and n_2 are the refractive indices of air and the material, and θ_1 and θ_2 are the angles of incidence and transmission, respectively. The reflection coefficient also depends on whether the wave is transverse electric (TE) or transverse magnetic (TM) polarized:

$$R_{\parallel} = \frac{n_1 \cos \theta_1 - n_2 \cos \theta_2}{n_1 \cos \theta_1 + n_2 \cos \theta_2}, \quad R_{\perp} = \frac{n_2 \cos \theta_1 - n_1 \cos \theta_2}{n_2 \cos \theta_1 + n_1 \cos \theta_2} \quad (2)$$

The reflection coefficient is used to adjust the PL equation:

$$PL_r(d_{tot}, f, R) = 20 \log_{10}(d_{tot}) + 20 \log_{10}(f) - 20 \log_{10}(|R|) - 20 \log_{10} \left(\frac{4\pi}{c} \right), \quad (3)$$

where d_{tot} is the total distance traveled by the reflected path.

The **diffraction PL** is computed using the knife-edge diffraction model, which approximates diffraction around obstacles like edges. The diffraction parameter ν is computed as:

$$\nu = h \sqrt{\frac{2d_{tx-rx}}{\lambda d_{tx-edge} d_{edge-rx}}}, \quad (4)$$

where h is the effective height of the obstacle relative to the LOS path and λ is the wavelength of the transmitted signal. The diffraction loss uses the Fresnel integral approximation:

$$PL_d(\nu) = 6.9 + 20 \log_{10} \left(\sqrt{(\nu - 0.1)^2 + 1} + \nu - 0.1 \right). \quad (5)$$

Lastly, the total PL is computed considering the contributions of multiple propagation paths and summing their effects in the linear domain:

$$PL_{total} = -10 \log_{10} \left(\sum_{i=1}^n 10^{-\frac{PL_i}{10}} \right) \quad (6)$$

Summing PL directly (even in the linear domain) is an oversimplification that ignores the phase-dependent nature of

multipath propagation (particularly constructive and destructive interference effects). Averaging out multipath interference effects over time when dealing with large distances or in highly scattered environments is far less precise than conventional raytracing simulators and neglects detailed fading patterns [16]. However, it allows simulation times to be significantly reduced to almost real time. Studies are currently being conducted to compare the applicability and limitations of this algorithm against a proven ray tracing simulator [17].

2.2 Finding the optimal path

Given a path P , which is a sequence of voxels $P = \{p_0, p_1, \dots, p_n\}$, where $p_0 = s$ (start) and $p_n = e$ (end), the total PL can be defined as the sum of PL through all the nodes in the path:

$$\mathcal{L}(P) = \sum_{i=0}^{i=n} PL(p_i), \text{ s.t. } p_0 = s, p_n = e \quad (7)$$

This is the cost function which we mainly aim to minimize. The problem is subject to two constraints:

1. Occupancy constraint: the path can only pass through free spaces, i.e., voxels that are not occupied by obstacles. This is typically represented by a binary occupancy grid:

$$G(p_i) = 0, \forall i = 0, 1, \dots, n \quad (8)$$

2. Movement constraint: The movement between adjacent voxels should follow one of the valid moves (e.g., six connected neighbors in a 3D space).

Each voxel of the binary occupancy grid can be associated to a scalar value representing the path loss for a given radio setup.

There are numerous path finding algorithms in the literature [18], but the classical A* algorithm [6] generally provides an intuitive well-performing solution to this problem. The A* algorithm is a heuristic search algorithm that finds the path with the lowest cost between two points. Each node p_i is associated with a cost function $f(p_i)$, which is the sum of two components:

1. Cost of travelling from the start node s to the current node p_i : $g(p_i)$.
2. Heuristic: an estimate of the remaining cost to reach the goal node e from the current node p_i , denoted as $h(p_i)$. A common choice is the Manhattan distance: $h(p_i) = |x_i - x_e| + |y_i - y_e| + |z_i - z_e|$

Thus, the total cost function at node p_i is given by:

$$f(p_i) = g(p_i) + h(p_i), \quad (9)$$

where $g(p_i)$ represents a combination of the path loss (PL) and the distance traveled from the start node to the current

node, and $h(p_i)$ is the estimated cost to reach the goal node. We have opted for modifying the conventional step cost $g(x)$ to consider a weighted combination of distance and PL:

$$g(x_{i+1}) = g(x_i) + (1 - \lambda) - \lambda PL_{norm} \quad (10)$$

where $\lambda \in [0, 1]$ is the coverage-over-distance weight parameter, and PL_{norm} is the normalized path loss reward, where $PL_{norm} = 1$ means best path loss and $PL_{norm} = 0$ is the worst-possible path loss value.

This is a classical graph problem of heuristic search. Dealing with multiple optimization criteria consists of finding paths that are optimal according to more than one attribute. A common approach to include multi-criteria into the search is to define edge costs $\sum_i \lambda_i w_i$, with $\sum_i \lambda_i = 1$. Albeit being a popular approach in videogames, the optimization cost algebra must be generalized to ensure the admissibility of the exploration algorithm, i.e., the fact that it solves the optimality problem. As depicted in [19], a heuristic function $h : V \rightarrow A$ with $h(t) = 1$ for each goal node $t \in T$ is

- admissible, if for all $u \in V$ we have $h(u) \preceq \delta(u, T)$, i.e., h is a lower bound, denoting $\delta(u, V)$ as the cost of the optimal path starting at node u and reaching node v in a set V .
- consistent, if for each $u, v \in V$ and $e \in E$, such that $u \xrightarrow{e} v$, we have $h(u) \preceq \omega(e) \times h(v)$.

Therefore, the coverage-over-distance weight parameter λ should be selected such that both conditions are met to ensure admissibility of the optimal path planning algorithm.

The algorithm uses an open list to add nodes from the start node s up to the goal node e by selecting nodes with the lowest $f(p_i)$. The path is then reconstructed by following the parent pointers of each node.

The generated path based on contiguous movements in a grid differs from the usual waypoint-based navigation employed with UAV. To reduce unnecessary waypoints and allow movement in any direction, a path smoothing algorithm is used. The path smoothing algorithm works by iterating over the given path and trying to remove unnecessary intermediate points. For each point p_i , it checks if the direct line between p_i and a later point p_j is obstacle-free and that the deviation of intermediate points between p_i and p_j from the straight line is less than a specified maximum deviation d_{max} . The pseudocode is detailed in Algorithm 1.

Algorithm 1 Path Smoothing Algorithm

Require: Path P , grid G , maximum deviation $\delta_{\max}$, buffer $b \leftarrow 0$

Ensure: Smoothed path S

```

1:  $S \leftarrow [P_0]$ 
2:  $i \leftarrow 0$ 
3: while  $i < |P| - 1$  do
4:    $j \leftarrow i + 1$ 
5:   while  $j < |P|$  do
6:     if  $\text{LINEOF SIGHT}(G, P_i, P_j, b)$  then
7:       if  $j - i > 1$  then
8:          $\delta \leftarrow \max_{k=i+1}^{j-1} \text{DISTFROMLINE}(P_k, P_i, P_j)$ 
9:         if  $\delta > \delta_{\max}$  then
10:          break
11:        end if
12:      end if
13:       $j \leftarrow j + 1$ 
14:    else
15:      break
16:    end if
17:  end while
18:  Append  $P_{j-1}$  to  $S$ 
19:   $i \leftarrow j - 1$ 
20: end while
21: return  $S$ 

```

The straight line is discretized using Bresenham's 3D Line algorithm [20], obtaining all grid points that lie approximately along the straight line between two selected nodes. The differences in each axis are calculated as follows: $\Delta x = |x_2 - x_1|$, $\Delta y = |y_2 - y_1|$, $\Delta z = |z_2 - z_1|$. The dominant axis is incremented step by step, while the other axes are incremented according to a continuous error term to approximate the straight line. Line-of-sight check verifies that for all points $p_i(x_i, y_i, z_i)$ along the Bresenham-generated line, the grid value at that point is: $G_i(x_i, y_i, z_i) \neq 1$ (obstacle). Lastly, the distance of a point from a line segment is used to ensure that the new path does not differ from the optimal PL path, so that the PL estimations remain valid through the smoothed path.

2.3 Calculating conflicting path segments for enhanced decision making

Given a PL-optimal smoothed path, the PL of each voxel the path goes through can be obtained thanks to the Bresenham algorithm. The PL for the maximum range can be expressed as:

$$PL = G_{Tx} + G_{Rx} + P_{Tx} - S_{Rx} - LM, \quad (11)$$

where G_{Tx} is the transmitter gain, G_{Rx} the receiver gain, P_{Tx} the transmitter power, S_{Rx} the receiver sensitivity, and LM the link margin. Since the link margin, transmitter gain, and receiver gain are typically fixed during flight, the trans-

mitter power and receiver sensitivity can be dynamically adjusted at specific locations where PL is expected to exceed the threshold that allows direct communication with the aircraft. By having this information before the flight, the transmission and reception power needs can be intelligently planned to adapt it dynamically during the operation, avoiding saturating the radio channel in urban environments and increasing power efficiency.

3 RESULTS AND DISCUSSION

The proposed system has been implemented in Python and is publicly accessible from the Github repository (<https://github.com/UPNAdrone/UrbanPathPlanner>). The PL at each position is obtained from the ray launching simulation and converted into a point cloud matching the voxel grid coordinates. The binary occupancy grid is also extracted from the ray launching simulation setup. The voxel mesh is generated using trimesh. Lastly, pyvista is used for interactive 3D visualization of the original mesh and an interactive slice map of the radio coverage for the selected height.

To verify the implementation, a test case environment has been proposed in the campus of the Public University of Navarra. Following the steps described in the previous Sections, the coverage map for a 868 MHz transmitter at different locations has been obtained, and the optimal path has been estimated for different paths between different locations of the campus. Four randomized transmitter locations are selected, along with two random initial and final locations from a pool of flyable take-off and landing zones. The results are shown in Table 4. As can be seen, the overall PL is reduced from 6.21% up to 24.12% by using our methodology.

For reference, in Figure 2 the path selected by A* using the conventional cost function where only distance is considered is shown on the left, the path selected using our proposed cost function (considering PL) is shown on the middle, and the coverage map for each experiment is shown on the right.

4 CONCLUSION

Throughout this paper, we have proposed a novel UAV path planning and navigation solution for urban environments, integrating radio PL into a modified A* path planning algorithm, including a fast raytracing algorithm, a binary occupancy grid, and a path-smoothing algorithm. As a complement to this work, we propose using the expected coverage information at each point of the path to adjust the transmitter and receiver power in complex urban scenarios in real time, providing the aircraft with greater intelligence and autonomy. Although this work is based on a modification of A*, the algorithm is currently being generalized to any heuristics-based path planning algorithm.

For future works, we propose using real-time planning algorithms that allow the flight path to be adapted based on changing conditions and new information that may be avail-

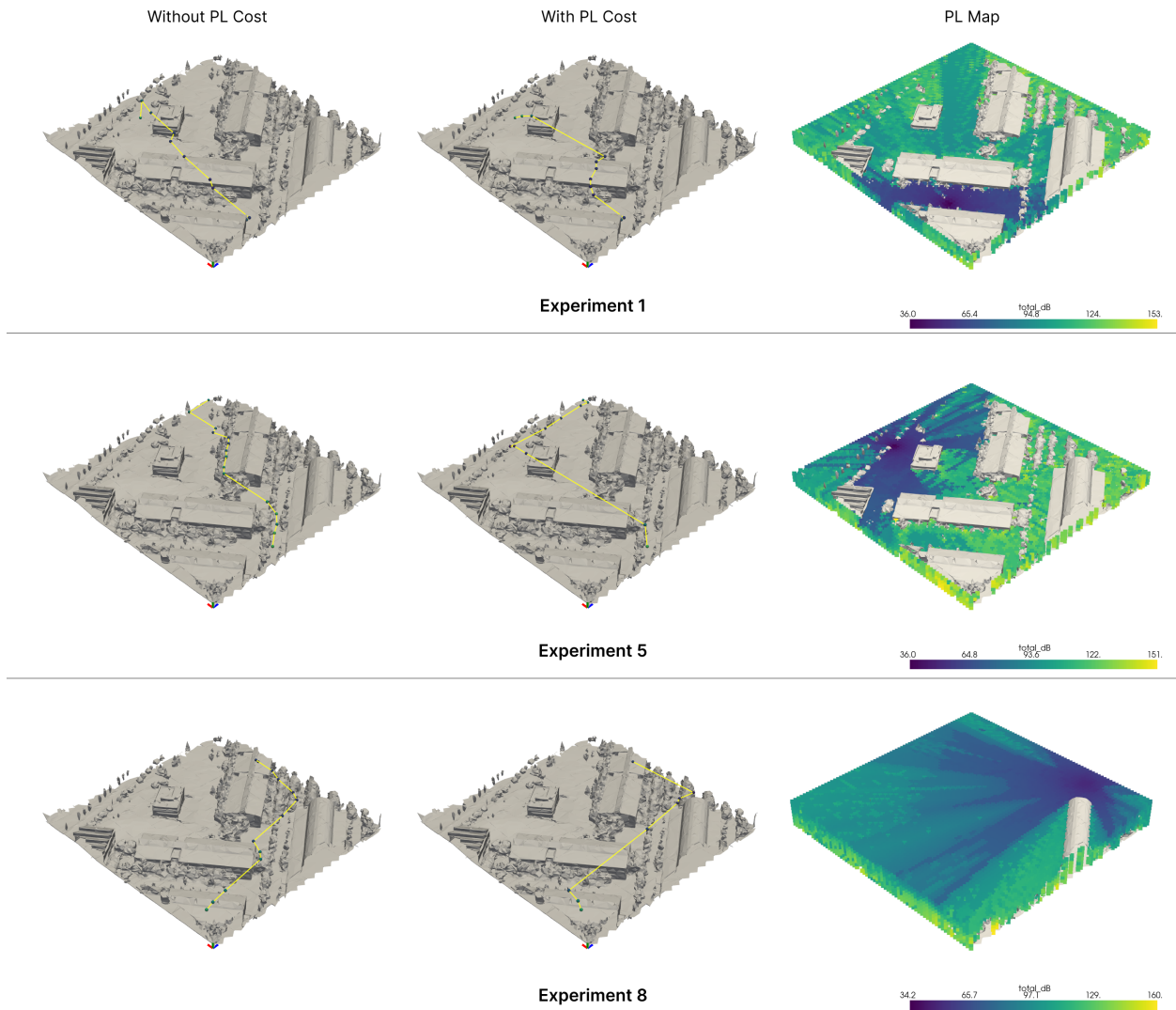


Figure 2: Experiments 1, 5 and 8. On the left: path generated by A* without considering PL. On the middle: path generated considering PL. On the right: PL map slice for reference.

able. Finally, we propose replacing the simplified ray tracing algorithm with one that considers aspects such as different materials, interference effects for multipath, and other complex phenomena.

REFERENCES

- [1] Graham Wild, John Murray, and Glenn Baxter. Exploring civil drone accidents and incidents to help prevent potential air disasters. *Aerospace*, (3), 2016.
- [2] B. M. López, J. Muñoz, F. Quevedo, C. A. Monje, S. Garrido, and L. Moreno. Path planning and collision risk management strategy for multi-uav systems in 3d environments. *Sensors*, 21:4414, 2021.
- [3] S. Primatesta, G. Guglieri, and A. Rizzo. A risk-aware path planning strategy for uavs in urban environments. *Journal of Intelligent & Robotic Systems*, 95:629–643, 2018.
- [4] H. Tang, Q. Zhu, B. Qin, R. Song, and Z. Li. Uav path planning based on third-party risk modeling. *Scientific Reports*, 13, 2023.
- [5] Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy*, pages 287–290. 2022.
- [6] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum

Table 1: Path planning algorithm comparison between standard A* and our modified version including PL.

ID	Tx Position [m]	Start [m]	End [m]	Total PL (standard)	Total PL (ours)	% Variation
1	49, 10, 26	53, 2, 25	12, 6, 45	5293.30	4964.52	-6.21
2	49, 10, 26	53, 2, 25	15, 6, 174	9016.10	7605.66	-15.64
3	70, 12, 80	7, 3, 21	162, 9, 12	5846.50	4805.77	-17.80
4	70, 12, 80	21, 2, 40	156, 6, 165	5904.62	4571.16	-22.58
5	170, 8, 80	3, 2, 23	186, 6, 180	9868.78	7766.59	-21.30
6	170, 8, 80	60, 3, 2	54, 12, 183	9885.72	7500.93	-24.12
7	50, 13, 180	56, 4, 12	33, 9, 102	6752.15	6078.43	-9.98
8	50, 13, 180	6,5,4	117, 18, 171	7002.20	6224.48	-11.11

cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

- [7] Sertac Karaman, Matthew R. Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime motion planning using the rrt*. In *2011 IEEE International Conference on Robotics and Automation*, pages 1478–1483, 2011.
- [8] Anbalagan Loganathan and Nur Syazreen Ahmad. A systematic review on recent advances in autonomous mobile robot navigation. *Engineering Science and Technology, an International Journal*, 40:101343, 2023.
- [9] Shuowen Zhang and Rui Zhang. Radio map-based 3d path planning for cellular-connected uav. *IEEE Transactions on Wireless Communications*, 20(3):1975–1989, 2021.
- [10] Behzad Khamidehi and Elvino S. Sousa. Federated learning for cellular-connected uavs: Radio mapping and path planning. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, 2020.
- [11] Yangrui Dong, Chen He, Ziteng Wang, and Lin Zhang. Radio map assisted path planning for uav anti-jamming communications. *IEEE Signal Processing Letters*, 29:607–611, 2022.
- [12] Supawat Tamsri. Murt, 4 2021.
- [13] Dawson-Haggerty et al. trimesh, 12 2019.
- [14] Franco Fuschini, Hassan El-Sallabi, Vittorio Degli-Esposti, Lasse Vuokko, Doriana Guiducci, and Pertti Vainikainen. Analysis of multipath propagation in urban environment through multidimensional measurements and advanced ray tracing simulation. *IEEE Transactions on Antennas and Propagation*, 56(3):848–857, 2008.
- [15] Harald T Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, 1946.
- [16] Bernard Tamba Sandouno, Yamen Alsaba, Chadi Barakat, Walid Dabbous, and Thierry Turetletti. A novel approach for ray tracing optimization in wireless communication. *Computer Communications*, 209:309–319, 2023.
- [17] L Azpilicueta, Meenakshi Rawat, Karun Rawat, F Ghannouchi, and F Falcone. Convergence analysis in deterministic 3d ray launching radio channel estimation in complex environments. *The Applied Computational Electromagnetics Society Journal (ACES)*, pages 256–271, 2014.
- [18] Azad Noori and Farzad Moradi. Simulation and comparison of efficiency in pathfinding algorithms in games. *Ciência e Natura*, 37(6-2):230–238, 2015.
- [19] Stefan Edelkamp, Shahid Jabbar, and Alberto Lluch-Lafuente. Cost-algebraic heuristic search. In *AAAI*, volume 5, pages 1362–1367, 2005.
- [20] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Syst. J.*, 4(1):25–30, March 1965.