

Nghiên cứu và đề xuất thuật toán lập bản đồ 3D cho robot, áp dụng vào drone và được mô phỏng trên nền tảng ROS

Research and propose a 3D mapping algorithm for robots, applied to drones and simulated on the ROS platform

> TS NGUYỄN HOÀNG THÙY TRANG

Trường Đại học Hàng hải Việt Nam

Email: trangnht@vimaru.edu.vn

TÓM TẮT

Lập bản đồ 3D thời gian thực là yếu tố then chốt giúp drone tự hành định vị và điều hướng an toàn trong môi trường biển. Bài báo đề xuất một phương pháp mới dựa trên máy tính tái cấu hình, thay thế các giải pháp truyền thống như SLAM xác suất hay mạng nơ-ron sâu - vốn tiêu tốn tài nguyên và thiếu tính minh bạch. Phương pháp sử dụng automat tái cấu hình, triển khai trên kiến trúc xử lý song song dạng đường ống, cho phép xử lý dữ liệu LiDAR hiệu quả với chi phí phần cứng thấp. Không gian 3D được biểu diễn theo hai chế độ: Không nhóm và nhóm automat, nhằm cân bằng giữa độ chính xác và hiệu suất tính toán. Mô hình được tích hợp trong ROS/Gazebo (sử dụng drone Hector Quadrotor) và mô phỏng trong Simulink bằng mạch tổ hợp Boolean. Kết quả cho thấy độ chính xác đạt 100% khi mật độ chướng ngại vật $\geq 83\%$; sai số tối đa 52,8% chỉ xuất hiện ở mật độ thấp ($\sim 14\%$) trong chế độ nhóm, đồng thời thời gian xử lý giảm 2,2 lần. Bản đồ đầu ra hoàn toàn khớp với môi trường mô phỏng. Nhờ tính minh bạch, khả năng diễn giải và linh hoạt tái cấu hình, giải pháp này có tiềm năng ứng dụng cao trong giao thông biển, đặc biệt trong khảo sát địa hình, giám sát công trình và cứu hộ trên biển.

Từ khóa: Lập bản đồ 3D; drone; môi trường máy tính tái cấu hình; automat; ROS; Gazebo; xử lý song song; logic Boolean; hệ thống nhúng.

ABSTRACT

Real-time 3D mapping is a key factor in enabling autonomous drones to safely locate and navigate in marine environments. This paper proposes a novel method based on reconfigurable computing, replacing traditional solutions such as probabilistic SLAM or deep neural networks - which are resource-intensive and lack transparency. The method uses reconfigurable automat, implemented on a pipelined parallel processing architecture, allowing for efficient LiDAR data processing at low hardware cost. The 3D space is represented in two modes: non-grouped and grouped automat, aiming to balance accuracy and computational performance. The model is integrated into ROS/Gazebo (using a Hector Quadrotor drone) and simulated in Simulink using Boolean combinatorial circuitry. Results show 100% accuracy when obstacle density $\geq 83\%$. The maximum error of 52.8% only occurs at low density ($\sim 14\%$) in group mode, while processing time is reduced by 2.2 times. The output map perfectly matches the simulated environment. Thanks to its transparency, interpretability and reconfigurability, this solution has high potential for application in maritime transport, especially in topographic surveying, construction monitoring and maritime rescue.

Keywords: 3D mapping; drones; reconfigurable computing environments; automat; ROS; Gazebo; parallel processing; Boolean logic; embedded systems.

1. ĐẶT VẤN ĐỀ

Trong kỷ nguyên số hóa và tự động hóa, drone đã trở thành công cụ không thể thiếu trong nhiều lĩnh vực, từ nông nghiệp, xây dựng đến giao thông vận tải và cứu hộ khẩn cấp. Đặc biệt, tại Việt Nam - quốc gia có bờ biển dài hơn 3.200 km và hệ thống sông ngòi

dày đặc, nhu cầu ứng dụng drone trong giám sát tàu thuyền, khảo sát luồng lạch, phát hiện vật cản trên biển, hay hỗ trợ tìm kiếm cứu nạn ngày càng cấp thiết.

Tuy nhiên, để drone có thể hoạt động độc lập trong môi trường thực, hệ thống phải có khả năng xây dựng bản đồ 3D chính

xác, thời gian thực, từ đó thực hiện định vị và lập kế hoạch đường đi. Các phương pháp phổ biến như Gmapping, Cartographer hay ORB-SLAM tuy mạnh mẽ nhưng thường dựa trên ước lượng xác suất, tối ưu hóa phi tuyến hoặc học sâu dẫn đến tải tính toán lớn, tiêu thụ năng lượng cao và khó kiểm soát lỗi trên nền tảng nhúng.

Trong bối cảnh đó, nghiên cứu này tiếp cận bài toán từ góc độ kiến trúc tính toán tái cấu hình, nhằm giảm phụ thuộc vào mô hình xác suất và tăng tính minh bạch trong quá trình ra quyết định. Thay vì biểu diễn môi trường dưới dạng phân bố xác suất, chúng tôi sử dụng logic Boolean kết hợp với ma trận trạng thái rời rạc để mô tả không gian. Cách tiếp cận này không chỉ đơn giản hóa mô hình tính toán mà còn cho phép triển khai linh hoạt trên cả nền tảng phần mềm (CPU trong mô phỏng) lẫn phần cứng chuyên dụng (FPGA/ASIC), đáp ứng yêu cầu về hiệu suất và tiết kiệm năng lượng cho các hệ thống drone thực tế.

2. CƠ SỞ LÝ THUYẾT VÀ PHƯƠNG PHÁP ĐỀ XUẤT

Trong vài thập kỷ qua, Cellular Automata (CA) đã được sử dụng rộng rãi trong robot mapping và path planning, tiêu biểu như các thuật toán Wavefront [5], Brushfire [6] hay các mô hình swarm robotics [7]. Các CA thường biểu diễn môi trường dưới dạng lưới nhị phân, trong đó mỗi ô (cell) cập nhật trạng thái dựa trên quy tắc cục bộ từ các ô lân cận - tương tự với tiếp cận của bài báo.

Tuy nhiên, phần lớn các nghiên cứu trước đây tập trung vào môi trường 2D hoặc sử dụng CA như một lớp hậu xử lý (post-processing) cho bản đồ occupancy grid [8]. Gần đây, OctoMap [9] đã trở thành tiêu chuẩn trong biểu diễn bản đồ 3D dạng octree, nhưng vẫn phụ thuộc vào mô hình xác suất và yêu cầu phần cứng mạnh.

Về mặt phần cứng, nhiều công trình đã đề xuất triển khai SLAM trên FPGA để tăng tốc độ [3, 4, 10], nhưng chủ yếu vẫn giữ nguyên logic xác suất. Trong khi đó, tiếp cận thuần Boolean và tái cấu hình như trong bài báo nhằm giảm tải tính toán triệt để - một hướng ít được khai thác trong bối cảnh hiện đại.

Ngoài ra, cơ chế "gộp nhóm" (grouped) có thể liên hệ với kỹ thuật "obstacle inflation" trong costmap của ROS Navigation Stack [11], tuy nhiên khác biệt ở chỗ: Inflation là bước xử lý sau khi bản đồ đã được xây dựng, còn trong bài báo, việc gộp nhóm là một phần của quá trình tạo bản đồ nguyên thủy, giúp giảm số lượng đơn vị xử lý ngay từ đầu.

2.1. Mô hình môi trường máy tính tái cấu hình

Môi trường máy tính tái cấu hình là một kiến trúc đồng nhất, bao gồm tập hợp các automat có cấu trúc hoàn toàn giống nhau và được liên kết theo một quy tắc kết nối nhất quán. Mỗi automat hoạt động như một đơn vị xử lý logic cơ bản, nhận dữ liệu từ cảm biến và các automat lân cận để ra quyết định về sự hiện diện của chướng ngại vật trong ô không gian tương ứng.

Cụ thể, mỗi automat được trang bị 10 đầu vào chính, bao gồm: x_1 : Nhận dữ liệu từ cảm biến Lidar, phản ánh sự hiện diện của chướng ngại vật tại vị trí tương ứng;

x_2-x_4 : Kết nối các automat lân cận trong cùng một nhóm của cùng 1 mặt phẳng;

x_5-x_8 : Kết nối các automat lân cận tới các nhóm khác của cùng 1 mặt phẳng;

x_9-x_{10} : Mở rộng trong không gian 3D, cho phép trao đổi thông tin với các lớp không gian liền kề trên dưới.

Bên cạnh đó, automat có 5 đầu ra:

f_1-f_3 : Dùng để truyền tín hiệu nội bộ giữa các phần tử trong cùng nhóm;

f_4 : Để chuyển kết quả xử lý sang nhóm automat lân cận.

Để đảm bảo tính đồng bộ trong xử lý không gian đa lớp, các

đầu vào từ x_5 đến x_{10} được đưa qua phần tử bộ nhớ trễ một chu kỳ xung nhịp. Mỗi quan hệ giữa tín hiệu đầu vào và giá trị lưu trữ được mô tả bởi biểu thức:

$$q_{i,k} = x_{i,k-1}, \text{ với } i = 5, \dots, 10, \text{ trong đó } k \text{ là chu kỳ tính toán hiện tại.}$$

Cơ chế trễ này cho phép hệ thống duy trì tính nhất quán thông tin khi cập nhật bản đồ trong môi trường 3D.

2.2. Biểu diễn không gian 3D

Trong không gian ba chiều, môi trường được phân vùng thành các khối lập phương và tổ chức theo cấu trúc octree - một dạng cây mà mỗi nút cha có tối đa 8 nút con, tương ứng với 8 phần vùng con của khối không gian hiện tại. Phương pháp này cho phép biểu diễn linh hoạt và hiệu quả mật độ thông tin theo từng vùng.

- Hệ thống hỗ trợ hai chế độ biểu diễn:

+ Chế độ không nhóm: Mỗi khối không gian tương ứng với một automat, tạo ra bản đồ chi tiết với độ chính xác cao;

+ Chế độ nhóm: 8 khối kế nhau được gộp thành 1 nhóm, tạo ra bản đồ thô nhưng giúp tăng đáng kể tốc độ xử lý.

Theo giả định hợp lý trong bối cảnh drone bay ở độ cao trung bình - nơi chướng ngại vật (như công trình, cây cối) thường chiếm giữ khối không gian liên tục - nếu bất kỳ khối con nào trong nhóm phát hiện chướng ngại vật, toàn bộ nhóm sẽ được đánh dấu là có vật cản. Cách tiếp cận này không chỉ giảm tải tính toán mà còn đảm bảo an toàn cho quá trình điều hướng, phù hợp với các ứng dụng thời gian thực trên nền tảng tài nguyên hạn chế.

2.3. Hàm logic và mã cấu hình

Hoạt động của automat được điều khiển bởi mã cấu hình ($z_1, z_2, \dots, z_k$), cho phép tái cấu hình hành vi mà không thay đổi phần cứng. Hệ công thức logic tổng quát cho không gian 3D được xây dựng như sau:

$$1) \text{ Khi } z_1 = z_2 = z_3 = z_4 = z_5 = z_6 = 0$$

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_7 \vee q_8 \vee q_9 \\ f_5 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_7 \vee q_8 \vee q_9 \end{cases}$$

$$2) \text{ Khi } z_1 = z_2 = z_3 = z_4 = z_5 = 0, z_6 = 1$$

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_8 \vee q_{10} \\ f_5 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_8 \vee q_{10} \end{cases}$$

$$3) \text{ Khi } z_1 = z_2 = z_3 = z_4 = 0, z_5 = 1, z_6 = 0$$

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_{10} \\ f_5 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_{10} \end{cases}$$

$$4) \text{ Khi } z_1 = z_2 = z_3 = z_4 = 0, z_5 = 1, z_6 = 1$$

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \vee q_7 \vee q_9 \vee q_{11} \\ f_5 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \vee q_7 \vee q_9 \vee q_{11} \end{cases}$$

$$5) \text{ Khi } z_1 = z_2 = z_3 = 0, z_4 = 1, z_5 = z_6 = 0$$

$$\begin{cases} f_1 = x_1 \\ f_2 = x_1 \\ f_3 = x_1 \\ f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_8 \vee q_{10} \\ f_5 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_8 \vee q_{10} \end{cases}$$

$$6) \text{ Khi } z_1 = z_2 = z_3 = 0, z_4 = 1, z_5 = 0, z_6 = 1$$

7) Khi $z_1 = z_2 = z_3 = 0, z_4 = z_5 = 1, z_6 = 0$

8) Khi $z_1 = z_2 = z_3 = 0, z_4 = z_5 = z_6 = 1$

9) Khi $z_1 = z_2 = 0, z_3 = 1, z_4 = z_5 = z_6 = 0$

10) Khi $z_1 = z_2 = 0, z_3 = 1, z_4 = z_5 = 0, z_6 = 1$

11) Khi $z_1 = z_2 = 0, z_3 = 1, z_4 = 0, z_5 = 1, z_6 = 0$

12) Khi $z_1 = z_2 = 0, z_3 = 1, z_4 = 0, z_5 = 1, z_6 = 1$

13) Khi $z_1 = z_2 = 0, z_3 = 1, z_4 = 1, z_5 = z_6 = 0$

14) Khi $z_1 = z_2 = 0, z_3 = z_4 = 1, z_5 = 0, z_6 = 1$

15) Khi $z_1 = z_2 = 0, z_3 = z_4 = z_5 = 1, z_6 = 0$

$$\begin{cases} f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_0 \\ f_5 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_0 \end{cases}$$

$$\left\{ \begin{array}{l} f_2 = x_1 \\ f_3 = x_2 \end{array} \right.$$

17) Khi $z_1 = 0, z_2 = 1, z_3 = z_4 = z_5 = z_6 = 0$

$$\begin{cases} f_2 = x_1 \\ f_3 = x_1 \end{cases}$$

18) Khi $z_1 = 0, z_2 = 1, z_3 = z_4 = z_5 = 0, z_6 = 1 = 1, z_6 = 0$

$$\begin{cases} f_3 = x_1 \\ f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_9 \vee q_{10} \end{cases}$$

19) Khi $z_1 = 0, z_2 = 1, z_3 = z_4 = 0, z_5 = 1, z_6 = 0$

$$f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7$$

20) Khi $z_1 = 0, z_2 = 1, z_3 = z_4 = 0, z_5 = z_6 = 1$

21) Khi $z_1 = 0, z_2 = 1, z_3 = 0, z_4 = 1, z_5 = z_6 = 0$

22) Khi $z_1 = 0, z_2 = 1, z_3 = 0, z_4 = 1, z_5 = 0, z_6 = 1$

23) Khi $z_1 = 0, z_2 = 1, z_3 = 0, z_4 = 1, z_5 = 1, z_6 = 0$

24) Khi $z_1 = 0, z_2 = 1, z_3 = 0, z_4 = 1, z_5 = 1, z_6 = 1$

25) Khi $z_1 = 0, z_2 = 1, z_3 = 1, z_4 = z_5 = z_6 = 0$

26) Khi $z_1 = 0, z_2 = 1, z_3 = 1, z_4 = z_5 = 0, z_6 = 1$

27) Khi $z_1 = 0, z_2 = z_3 = 1, z_4 = 0, z_5 = 1, z_6 = 0$

28) Khi $z = 0, z = z = 1, z = 0, z = z = 1$

$$\begin{cases} f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \\ f_5 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_6 \end{cases}$$

$$\begin{cases} f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \\ f_5 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_7 \end{cases}$$

$$\left\{ \begin{array}{l} f_3 = x_1 \\ f_4 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee q_5 \vee q_6 \vee q_0 \end{array} \right.$$

$$\left\{ \begin{array}{l} f_2 = x_1 \\ f_3 = x_1 \\ f_4 = x_1 M x_1 M x_1 M x_1 M x_1 M x_1 M x_1 M x_1 \end{array} \right.$$

$$\int \begin{matrix} f_1 = x_1 \\ f_2 = x_1 \\ \vdots \\ f_r = x_1 \end{matrix}$$

$$f_1 = x_1$$

100 90 80 70 60 50 40 30 20 10 0

kể số lượng automat cần xử lý, từ đó cắt giảm thời gian tính toán - đúng như phân tích về cân bằng an toàn/hiệu suất tại Mục 2.2.

Hình trên minh họa bản đồ 3D do máy tính tái cấu trúc tạo ra (màu đậm cho chướng ngại vật, màu sáng cho vùng trống) và xác nhận rằng bản đồ này khớp hoàn toàn với môi trường thực tế trong Gazebo, chứng tỏ tính chính xác và khả năng tái tạo môi trường của phương pháp đề xuất.

4. ỨNG DỤNG TRONG GIAO THÔNG VẬN TẢI

Công nghệ drone hiện đại đã và đang khẳng định vai trò then chốt trong nhiều lĩnh vực của ngành Giao thông vận tải, nhờ vào khả năng thu thập dữ liệu không gian với độ chính xác cao, tốc độ nhanh và tính chất phi tiếp xúc. Nhờ những ưu điểm này, drone không chỉ hỗ trợ hiệu quả trong quản lý, giám sát mà còn góp phần nâng cao năng lực phản ứng trong các tình huống khẩn cấp cũng như đổi mới phương pháp đào tạo kỹ thuật.

Trong khảo sát địa hình và giám sát kết cấu hạ tầng giao thông, drone được sử dụng để xây dựng các mô hình 3D chi tiết của các công trình chiến lược như hầm, cầu và tuyến đường đèo. Dữ liệu thu thập được cung cấp cơ sở định lượng cho việc phát hiện sớm các nguy cơ như sạt lở, rạn nứt kết cấu, từ đó hỗ trợ ra quyết định bảo trì chủ động và đảm bảo an toàn vận hành.

Đối với lĩnh vực hàng hải, các hệ thống drone tích hợp cảm biến chuyên dụng cho phép phát hiện chướng ngại vật nổi hoặc trôi dạt trên luồng tuyến, đồng thời tự động cập nhật cơ sở dữ liệu bản đồ hàng hải. Giải pháp này không chỉ nâng cao độ tin cậy trong dẫn luồng tàu mà còn góp phần tối ưu hóa luồng vận chuyển, giảm thiểu rủi ro va chạm và gián đoạn luồng giao thông đường thủy.

Trong các tình huống thiên tai như sạt lở đất hoặc lũ lụt, drone có thể được triển khai nhanh chóng để khảo sát khu vực bị ảnh hưởng, từ đó tái tạo mô hình 3D hiện trạng và hỗ trợ lực lượng cứu hộ xác định vị trí nạn nhân cũng như lập kế hoạch tiếp cận hiệu quả, an toàn và kịp thời.

Bên cạnh các ứng dụng thực tiễn, dữ liệu không gian và mô hình 3D do drone thu thập còn được chuyển hóa thành tài liệu học tập thực hành trực quan, phục vụ giảng dạy trong các học phần chuyên ngành như Hệ thống nhúng, Robot thông minh và Phát triển ứng dụng mã nguồn mở. Điều này góp phần thúc đẩy đổi mới phương pháp đào tạo theo hướng liên ngành, tích hợp giữa lý thuyết và ứng dụng thực tiễn.

5. KẾT LUẬN

Nghiên cứu này đã đề xuất và thực nghiệm thành công một thuật toán lập bản đồ 3D hiệu quả, được xây dựng trên nền tảng máy tính tái cấu trúc và dựa vào logic Boolean. Thuật toán thể hiện những ưu điểm vượt trội so với các phương pháp SLAM (Simultaneous Localization and Mapping) truyền thống trong các môi trường có cấu trúc rõ ràng, nhờ vào khả năng xử lý dữ liệu không gian với độ phức tạp tính toán thấp hơn và hiệu suất thời gian thực cao hơn.

Một trong những đặc điểm nổi bật của phương pháp là khả năng tái cấu trúc hình ảnh hoạt động thông qua mã điều khiển z, cho phép thích ứng nhanh chóng với các yêu cầu thay đổi của nhiệm vụ hoặc môi trường. Bên cạnh đó, kiến trúc của thuật toán được thiết kế theo hướng mô-đun, hỗ trợ triển khai đồng thời trên cả nền tảng phần mềm (ví dụ: ROS - Robot Operating System) và phần cứng nhúng (ví dụ: FPGA - Field-Programmable Gate Array), tạo điều kiện thuận lợi cho việc tích hợp vào các hệ thống nhúng tài nguyên hạn chế.

Đặc biệt, nhờ yêu cầu tài nguyên tính toán thấp và khả năng

xử lý thời gian thực, thuật toán rất phù hợp để triển khai trên các phương tiện bay không người lái (drone), nơi yêu cầu nghiêm ngặt về hiệu quả năng lượng và độ trễ xử lý.

Hướng phát triển trong tương lai bao gồm: (1) Tích hợp dữ liệu đa cảm biến (multimodal sensing), kết hợp Lidar và camera để nâng cao độ chính xác và độ tin cậy của bản đồ 3D; (2) mở rộng khả năng hoạt động sang các môi trường động, nơi có sự hiện diện của đối tượng di chuyển; và (3) triển khai thực nghiệm trên các nền tảng drone thương mại như DJI Matrice và hệ thống điều khiển bay Pixhawk nhằm đánh giá hiệu năng trong điều kiện thực địa.

TÀI LIỆU THAM KHẢO

- [1] I. O. Shepel. Modified algorithm for constructing an occupancy map based on a point cloud from several lidars. *T.*, 2, no 33, pp. 257-265, 2020, doi: 10.15827/0236-235X.130.257-265.
- [2] Xiaojie Niu, Tong Yu, Jian Tang, Le Chang. An Online Solution of LiDAR Scan Matching Aided Inertial Navigation System for Indoor Mobile Mapping. *Mobile Information Systems*, no 2, pp. 1-11, 2017, doi: 10.1155/2017/4802159.
- [3] Mizuki Yasuda, Keisuke Sugiyama, Ryuto Kojima, Hiroki Matsutani. An Edge-Server Partitioning Method for 3D LiDAR SLAM on FPGAs. *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2023, doi: 10.1109/IPDPSW59300.2023.00030.
- [4] Qi Deng, Hao Sun, Fupeng Chen, Yuhao Shu, Hui Wang, Yajun Ha. An Optimized FPGA-Based Real-Time NDT for 3D-LiDAR Localization in Smart Vehicles. *IEEE Transactions on Circuits and Systems II: Express Briefs*, T. 68, no. 9, pp. 3167-3171, 2021, doi: 10.1109/TCSII.2021.3095764.
- [5] Robert Tarjan. Depth-first search and linear graph algorithms. *12th Annual Symposium on Switching and Automata Theory*, 1971, doi: 10.1109/SWAT.1971.10.
- [6] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, vol. 2, no. 1, pp. 14-23, 1986, doi: 10.1109/JRA.1986.1087032.
- [7] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, Marco Dorigo. Swarm Robotics: A Review from the Swarm Engineering Perspective. *Swarm Intell.*, vol. 7, no. 1, pp. 1-41, 2013, doi: 10.1007/s11721-012-0075-2.
- [8] Howie Choset. Coverage for robotics - A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113-126, 2001.
- [9] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, Wolfram Burgard. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Auton Robot.*, vol. 34, pp. 189-206, 2013.
- [10] Marcel Flottmann, Marc Eisoldt, Julian Gaal, Marc Rothmann, Marco Tassemeier, Thomas Wiemann, Mario Pormann. Energy-efficient FPGA-accelerated LiDAR-based SLAM for embedded robotics. *2021 International Conference on Field-Programmable Technology (ICFPT)*, 2021, doi: 10.1109/ICFPT52863.2021.9609934.
- [11] ROS navigation. Internet: <https://wiki.ros.org/navigation>.
- [12] Sebastian Thrun. Probabilistic Algorithms in Robotics. pp. 1-20, 2000.
- [13] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots, 1985.
- [14] Junhui Wang, Dongjie Huo, Yongliang Shi, Chao Gao, Yan Qiao, Guyue Zhou. OPEN: Lightweight Map-Based Semantic Navigation for GPS-Free Last-Mile Delivery. *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 20713-20725, 2025.
- [15] Johann Schmidt, Charlotte Fischer, Sebastian Weber. Autonomous Systems and Robotics Using Reconfigurable Computing. *SCCTS Transactions on Reconfigurable Computing*, vol. 2, no 2, pp. 25-30, 2025.