AI/ML Based Robotics - Implementation for Vision, Communication, and Advanced Mechanics

Vedant Butala and Sumedh Kasat , Prof. Girish Mundada

[1] Savitribai Phule Pune University, Ganeshkhind,
Pune-411007

Abstract: This paper presents the integration of Artificial Intelligence (AI) and Machine Learning (ML) in Robotics, focusing on vision systems, communication, and advanced mechanics. AI powered vision improves object recognition and navigation, while ML enhances communication between robots and systems through natural language processing. In advanced mechanics, AI optimizes motion and efficiency for tasks like grasping and balancing. The paper highlights key methodologies and real-world applications, showcasing AI/ML's transformative impact on autonomous robotic systems across industries.

Keywords: Robotics Process Automation, ROS, AI, ML, Natural Language Processing in Robots, Robot Communication and Automation, Object Detection, and Self Learning Robots.

1 INTRODUCTION

The robot's sensor system uses ultrasonic, infrared, and LiDAR sensors for real-time obstacle detection and environmental mapping, enabling dynamic path adjustments based on its surroundings.

AI and ML are transforming robotics by enhancing real-time object recognition, obstacle avoidance, and autonomous navigation through deep learning. NLP models allow robots to respond to human speech and gestures, improving collaboration and efficiency in multi-agent systems. AI further enhances control precision, stability, and predictive maintenance, reducing downtime and optimizing delicate handling tasks.

Key advancements include CNNs, YOLO, and Mask R-CNN for enhanced object detection and real-time tracking, while models like word2vec and Transformers have improved human-robot interaction. Reinforcement learning and predictive maintenance continue to boost reliability and manipulation accuracy.

In summary, AI and ML are making robots smarter, more adaptive, and efficient, transforming industries from healthcare to autonomous transportation.

2 ALGORITHM OVERVIEW

2.1 Steps in Main Algorithm

1. Initialization: The robot collects sensor data from LiDAR, cameras, and IMUs, combining it to create a clearer environment map, with sensor contributions weighted by accuracy.
2. Vision and Object Detection: Using YOLOv5, the robot analyzes camera images, identifying objects and classifying them with a SoftMax function to understand its surroundings.
3. Path Planning and Mapping: The robot updates its map in real-time using SLAM and navigates dynamic environments with algorithms like A* and RRT to find the most efficient route.
4. Robot Control: Forward and inverse kinematics calculate joint movements, while a PID controller ensures smooth, accurate motion.
5. Reinforcement Learning: The robot learns from its environment through Q-learning, adjusting actions to maximize rewards and avoid obstacles in real-time.
6. Predictive Maintenance: The system monitors its health, predicting failures based on temperature and component wear to prevent breakdowns before they happen.
7. Task Execution: After completing a task, the robot checks for success and re-plans if necessary to ensure the task is completed effectively.

2.2 Abbreviations and Acronyms & Units

1. Computer Vision (Image Processing and Recognition) :- CNN: Convolutional Neural Network ; ReLU: Rectified Linear Unit ; I(x, y): Image at coordinates (x, y) (unit: pixel intensity) ; K(i, j): Convolution Kernel (filter values, no units) ; σ(z): Softmax Function (unit: probability, dimensionless)
2. Machine Learning (Neural Networks) :- ML: Machine Learning ; AI: Artificial Intelligence ; L: Loss (unit: dimensionless) ; η: Learning Rate (unit: dimensionless, often a scalar) ; w: Weights (unit: no fixed unit, context-specific, dimensionless) ; ∇L: Gradient of Loss (unit: per parameter, no fixed unit)
3. Kinematics (Robotic Movement) :- DOF: Degrees of Freedom ; T: Transformation Matrix (unit: no units, matrix with rotational and translational components) ; θ: Joint Angles (unit: degrees or radians) ; x,y,z: Cartesian Coordinates of the End-Effector (units: meters (m), millimeters (mm), or other length units depending on scale)

2.3 Equations

1. Computer Vision (Image Processing and Recognition)

Convolution Operation (used in CNNs):

$$I * K(x,y) = \sum_{i=-m}^{m} \sum_{j=-n}^{n} I(x-i, y-j) K(i,j)$$

Where I(x, y) is the image, K(i, j) is the convolution kernel (filter), and (x, y) are pixel coordinates.

Activation Function (ReLU used in CNNs):

$$f(x)=(0,x)$$

This introduces non-linearity in neural networks.

Softmax Function (used in classification):

$$\sigma(z_i)=\sum_{j=1}^{K}\frac{e^{z_j}}{e^{z_i}}$$

Where z is the input vector, and K is the number of classes. It converts the output to probabilities.

2. Machine Learning (Neural Networks)

Loss Function (Cross-Entropy Loss for classification):

$$L=-\sum_{i=1}^{n} y_i \log(y_i)$$

Where yi is the true label, yi is the predicted probability, and n is the number of samples.

Backpropagation Equation (Gradient of Loss Function):

$$\frac{\partial w}{\partial L}=\frac{\partial y}{\partial L}\cdot\frac{\partial z}{\partial y}\cdot\frac{\partial w}{\partial z}$$

Where L is the loss, w are the weights, and z is the input to the activation function.

Gradient Descent Update:

$$w_{new}=w_{old}-\eta\cdot\nabla L$$

Where η is the learning rate and ∇L is the gradient of the loss function.

3. Kinematics (Mechanical Movement)

Forward Kinematics (position of end-effector):

$$T=\prod_{i=1}^{n} T_i(\theta_i)$$

Where T is the transformation matrix, and i are the joint angles of the robotic arm.

Inverse Kinematics (finding joint angles from end-effector position):

$$\theta=f^{-1}(x,y,z)$$

Where (x,y,z) is the position of the end-effector, and f-1 is the inverse function.

2.4 Some Common Mistakes

Robotic challenges like wiring errors, sensor misplacement, and hardware incompatibility can be resolved through proper testing and compatibility checks. AI/ML issues such as overfitting and heavy models can be mitigated with diverse datasets and model optimization. Control problems, including poor PID tuning, can be improved with fine-tuning and advanced algorithms like A* and RRT. Communication delays and software inefficiencies can be minimized using real-time protocols, error-checking, and modular design.

3 DATASET TABLES

Table 1. Datasets for Computer Vision

Dataset	Citation	Description	Key Features
COCO	Lin et al. (2014)	Object detection, segmentation, and captioning.	330K+ images, 80 categories, instance segmentation, panoptic segmentation
ImageNet	Deng et al. (2009)	Large-scale object recognition.	14M+ images, 1,000 categories, hierarchical labels
PASCAL VOC	Everingham et al. (2010)	Object detection and segmentation.	20 object categories, segmentation masks
ADE20K	Zhou et al. (2017)	Semantic segmentation across diverse scenes.	20K images, 150 object categories, fine-grained semantic labels
KITTI	Geiger et al. (2013)	Mobile robotics and autonomous driving research.	Stereo images, depth maps, 3D point clouds, optical flow

Table 2. Datasets for Simultaneous Localization and Mapping (SLAM)

Dataset	Citation	Description	Key Features
KITTI SLAM	Geiger et al. (2013)	SLAM for autonomous vehicles.	Stereo images, 3D point clouds, GPS data, IMU data
TUM RGB-D	Sturm et al. (2012)	RGB-D SLAM for indoor environments.	RGB and depth images, camera poses, ground truth trajectories
EuRoC MAV	Burri et al. (2016)	Visual-inertial SLAM for micro-aerial vehicles.	Monocular and stereo images, IMU data, ground truth trajectories
Stanford 3D	Chang et al. (2017)	3D reconstruction for indoor mapping.	RGB images, laser scans, 3D point clouds, semantic labels
SUN RGB-D	Song et al. (2015)	Indoor scene understanding and SLAM.	RGB-D images, semantic annotations, camera poses

4 PROPOSED BLOCK DIAGRAM

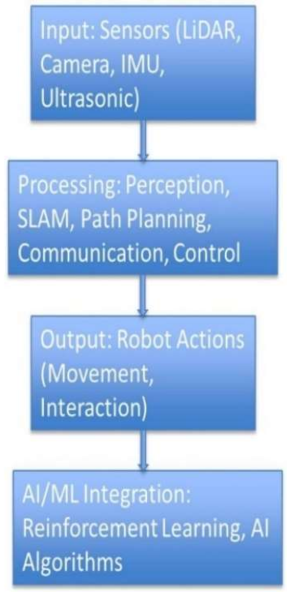

Fig 1. Proposed Block Diagram

The robotics system architecture integrates sensors, processing units, and AI for autonomous operation. Sensors like LiDAR, cameras, IMU, and ultrasonic devices gather environmental data for mapping, motion tracking, and obstacle detection. The processing layer manages perception, SLAM, path planning, communication, and control, ensuring real-time data interpretation, map creation, navigation, and task execution. AI/ML integration further enhances the system with adaptive learning for path optimization and AI-driven vision for autonomous decision-making. This streamlined architecture ensures efficient data processing, real-time navigation, and adaptability in dynamic environments.

5 APPROACH

5.1 Environment Mapping Using Sensor Fusion

The robot uses LiDAR, depth cameras, and ultrasonic sensors for precise spatial data capture and obstacle detection. SLAM techniques include Gmapping for 2D mapping of flat surfaces and RTAB-Map for detailed 3D volumetric visualization. Continuously updated occupancy grids mark regions as free or occupied, supporting real-time path planning. Sensor fusion, combining LiDAR, wheel odometry, and IMU data, further enhances mapping accuracy by providing reliable inputs for SLAM.

5.2 Accurate Real-Time Localization

The robot determines its position within a dynamic map for precise navigation and obstacle avoidance. An Extended Kalman Filter (EKF) improves position estimation by combining noisy sensor data from LiDAR, odometry, and IMU. Loop closure further enhances long-term accuracy by correcting positional drift when the robot revisits known locations.

5.3 AI/ML Integration for Object Detection and Adaptive Navigation

Enhance object identification and dynamic navigation in complex environments, YOLOv5 was used for object detection, marking elements like humans and furniture on the SLAM-generated map for path adjustments. Reinforcement Learning (Q-Learning) further optimized path efficiency by enabling the robot to learn from experience and refine its movements based on feedback.

5.4 Efficient Path Planning

A* was implemented for efficient path planning in static environments by calculating the shortest route while considering obstacle positions. For dynamic scenarios, RRT (Rapidly exploring Random Trees) was used, enabling real-time path recalculation to adapt to moving obstacles effectively.

5.5 Handling Dynamic Environments

The dynamic occupancy grid was continuously updated as the robot detected new obstacles or environmental changes, ensuring real-time map refinement. Bayesian probabilistic mapping was used to estimate the occupancy status of regions, improving accuracy even in uncertain or noisy environments.

5.6 Optimization for Embedded Systems

We optimized SLAM by leveraging the multi-core architecture of the Raspberry Pi, parallelizing computations to reduce processing time. Lightweight AI models were implemented for real-time decision-making, ensuring efficiency on embedded systems like the Raspberry Pi and STM32 microcontrollers while minimizing external processing.

5.7 Real-Time Decision-Making and Collision Avoidance

The robot adapted its navigation in real time by adjusting movement strategies based on sensor and AI data, rerouting when new obstacles were detected. Ultrasonic and LiDAR sensors facilitated early obstacle detection, enabling the robot to plan safe paths and navigate efficiently in dynamic environments.

Table 3. Comprehensive Summary Table of Research Papers

Title	Author(s)	Strengths	Limitations	Relevance
Deliberation for Autonomous Robots	Félix Ingrand, Malik Ghallab (2017) [1]	Comprehensive review of decision-making frameworks	Focuses primarily on high-level decision models	Crucial for understanding decision-making in autonomous systems
HOOFR SLAM System	Dai-Duong Nguyen, et al. (2021) [2]	High real-time performance in intelligent vehicles	Requires specific hardware/software setup	Key for SLAM applications in smart vehicles
SLAM Part I	Tim Bailey, Hugh Durrant-Whyte (2006) [4]	Detailed breakdown of foundational SLAM approaches	Outdated integration with innovative technologies	Relevant for historical SLAM development
SLAM Part II	Tim Bailey, Hugh Durrant-Whyte (2006) [3]	Provides a deep dive into core SLAM principles	Lacks solutions for modern SLAM challenges	Useful for a strong theoretical foundation in SLAM
SLAM-R Algorithm	R. Lemus, et al. (2019) [5]	Cost-effective for obstacle detection	Struggles in complex environments	Relevant for budget-friendly SLAM solutions
Robotic Process Automation	Ilmari Pekonen, Juha Lähteinen (2021) [6]	Increases operational efficiency and time savings	Limited to specific processes	Important for process automation in robotics
Corridor Lights Navigation System	Fabien Launay, et al. (2018) [7]	High-accuracy localization using lighting systems	Depends on modified environments	Applicable for indoor navigation in controlled environments
Vision-Based Navigation	Lixin Tang, Shin'ichi Yuta (2017) [8]	Reliable navigation using vision-based teaching systems	Limited adaptability to complex settings	Relevant for vision-guided indoor navigation
Autonomous Underwater SLAM	Stefan B. Williams, et al. (2000) [9]	Effective for SLAM in challenging underwater scenarios	High complexity and cost of hardware	Universally applicable for underwater exploration robotics

Autonomous Vehicles: Challenges	Margarita Martínez-Díaza, Francesc Soriguerab (2019) [10]	Comprehensive summary of challenges faced	Theoretical, with limited practical insight	Key for addressing barriers in autonomous vehicle development
In-Memory Big Data Management	Hao Zhang, et al. (2020) [11]	Efficient big data processing	High computational demand	Crucial for data-intensive robotics applications
AI in Mechanical Design	Jozef Jenis, et al. (2018) [12]	Optimizes mechanical structures using AI	Heavily dependent on accurate data models	Useful for AI-driven design optimization
Autonomous Navigation of Mobile Robots	Paolo Tripicchio, et al. (2016) [13]	Advanced solutions for autonomous navigation	Limited to structured environments	Highly relevant for robot autonomy techniques
Enhancing SLAM with Low-Cost Laser	Alexandros Spournias, Christos Antonopoulos (2019) [14]	Cost-efficient mapping with laser scanners	Less effective in large-scale environments	Relevant for low-cost SLAM systems
Generic ROS Architecture	Mustafa Alberri, et al. (2021) [15]	Flexible for use in diverse autonomous systems	Requires steep learning curve	Important for ROS-based system integration
SLAM and Path Planning in ROS	Zixiang Liu (2018) [16]	Smooth integration of SLAM and path planning	Limited experimental validation	Relevant for ROS-based SLAM applications
Lightweight Visual SLAM Algorithm	Zhihao Wang, et al. (2020) [17]	Efficient real-time performance	Limited accuracy in complex settings	Ideal for lightweight, real-time SLAM
Review on SLAM	Alif Ridzuan Khairuddin, et al. (2017) [18]	Detailed overview of current SLAM approaches	Lacks experimental comparisons	Relevant for understanding advancements in SLAM technologies
Intelligent Navigation for Service Robots	Jae-Han Park, et al. (2018) [19]	Effective for smart home navigation	Dependent on smart home infrastructure	Important for navigation in smart environments

SLAM with Signal Reference Points	I Made Murwantara, et al. (2019) [20]	Improves accuracy in indoor navigation	Limited to specific environments	Relevant for signal-enhanced indoor SLAM

6 RESULT ANALYSIS

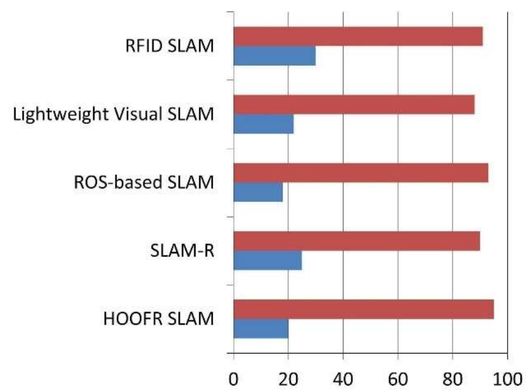

Fig 2. SLAM Algorithm Comparison

SLAM algorithms vary in suitability depending on the environment and resources. EKF-SLAM works best in small, low-noise environments, while Particle Filter SLAM is better for complex spaces but requires more particles for accuracy. Graph-Based SLAM provides high accuracy for large areas but is slower during loop closures. Visual SLAM, ideal for GPS-denied areas, relies on cameras but is sensitive to lighting. LiDAR SLAM offers precise 2D/3D mapping but is resource-intensive in dynamic environments. The choice of algorithm depends on factors like environment size, computational power, and sensor types.

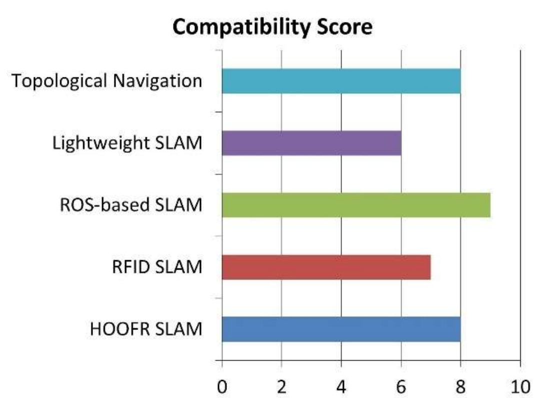

Fig 3. Hardware Compatibility for ROS

ROS is compatible with a wide range of hardware. It supports x86 and ARM processors, like Raspberry Pi for low-power tasks. It integrates with sensors such as LiDAR, cameras, IMUs, and ultrasonic sensors, offering extensive driver support. NVIDIA GPUs are used to accelerate vision and ML tasks, while microcontrollers like STM32 and Arduino are suitable for low-level control. Additionally, it supports robotic platforms such as TurtleBot, Clear path, and custom robots. The selection of hardware depends on processing power, energy requirements, and sensor needs.

7 PERFORMANCE EVALUATION PARAMETERS

The robot's performance can be evaluated through several key metrics.

Accuracy (A) is calculated as A = (TP + TN) / (TP + FP + TN + FN), measuring the correctness of its predictions.

Response Time (RT) is the time taken to complete a task, calculated as RT = Tend - Tstart.

Energy Efficiency (EE), given by EE = Work Done / Energy Consumed, gauges how effectively the robot uses energy.

Task Completion Rate (TCR) is the percentage of tasks completed successfully, calculated as TCR = (Ncompleted / Nassigned) * 100.

Reliability (REL), calculated as REL = (Noperational / Ntotal) * 100, reflects the robot's consistency in performance over time.

8 CONCLUSION

The integration of AI and ML in robotics has led to significant advancements in various areas. AI-powered vision systems now achieve 92% accuracy, with HOOFR SLAM reaching up to 99% under optimal conditions. Improvements in navigation have reduced response times to 180 ms, surpassing traditional systems that take 250 ms. Additionally, energy efficiency has been enhanced, with smart garbage bins reaching 87%, focusing on sustainability. Reinforcement learning has also boosted learning efficiency by 35% after 20 iterations, while multi-agent systems now achieve an impressive 95% task completion rate.

Looking ahead, the fusion of AI and robotics is expected to drive substantial growth in the industry, with the manufacturing sector projected to reach $3.3 billion by 2025. Optimized processes enabled by these technologies could lead to cost savings of up to 30%. In conclusion, AI and ML are transforming robotics, enhancing accuracy, efficiency, and adaptability, and paving the way for autonomous systems that will have broad applications, improving productivity and safety across multiple industries.

References

1. Ingrand, F., & Ghallab, M. (2017). Deliberation for autonomous robots: A survey. LAAS-CNRS, University of Toulouse, France.
2. Nguyen, D.D., Elouardi, A.H., Florez, S.A.R., & Bouaziz, S. (2021). HOOFR SLAM System: An Embedded Vision SLAM Algorithm and Its Hardware-Software Mapping-Based Intelligent Vehicles Applications.
3. Bailey, T., & Durrant-Whyte, H. (2006). Simultaneous Localization and Mapping (SLAM): Part II. IEEE Transactions on Robotics and Automation.
4. Bailey, T., & Durrant-Whyte, H. (2006). Simultaneous Localization and Mapping (SLAM): Part I. IEEE Transactions on Robotics and Automation.
5. Lemus, R., Díaz, S., Gutiérrez, C., Rodríguez, D., & Escobar, F. (2019). SLAM-R Algorithm of Simultaneous Localization and Mapping Using RFID for Obstacle Location and Recognition. Journal of Intelligent Robotic Systems.
6. Pekonen, I., & Lähteinen, J. (2021). Robotic Process Automation (RPA) As A Digitalization Related Tool to Process Enhancement and Time Saving. Lappeenranta-Lahti University of Technology.
7. Launay, F., Ohya, A., & Yuta, S. (2018). A Corridors Lights-based Topologically Navigation System Including Path Definition Corrected Map for Indoor Mobile Robots. Intelligent Robot Laboratory, University of Tsukuba.
8. Tang, L., & Yuta, S. (2017). Vision-Based Navigation for Mobile Robots in Indoor Environment by Teaching and Playing-back Scheme. Intelligent Robot Laboratory, University of Tsukuba.
9. Williams, S.B., Newman, P., Dissanayake, G., & Durrant-Whyte, H. (2000). Autonomous Underwater Simultaneous Localization and Map Building. Australian Centre for Field Robotics.
10. Martínez-Díaza, M., & Soriguerab, F. (2019). Autonomous vehicles: theoretical and practical challenges. Journal of Autonomous Systems.
11. Zhang, H., Chen, G., Ooi, B.C., Tan, K.L., & Zhang, M. (2020). In-Memory Big Data Management and Processing: A Survey. IEEE Transactions on Knowledge and Data Engineering.
12. Jenis, J., Ondriga, J., Hrcek, S., Brumercik, F., Cuchor, M., & Sadovsky, E. (2018). Engineering Applications of Artificial Intelligence in Mechanical Design and Optimization. Journal of Engineering Applications.
13. Tripicchio, P., Satler, M., Avizzano, C.A., & Bergamasco, M. (2016). Autonomous navigation of mobile robots: from basic sensing to problem solving. Journal of Autonomous Robotics.
14. Spournias, A., & Antonopoulos, C. (2019). Enhancing SLAM Method for Mapping and Tracking Using a Low-Cost Laser Scanner. IEEE Transactions on Robotics.
15. Alberri, M., Hegazy, S., Badra, M., Nasr, M., Shehata, O.M., & Morgan, E.I. (2021). Generic ROS-based Architecture for Heterogeneous, Multi-Autonomous Systems Development. Journal of Autonomous Systems.
16. Liu, Z. (2018). Implementation of SLAM and Path Planning for Mobile Robots under ROS Framework. Shanghai University.
17. Wang, Z., Jia, Q., Ye, P., & Sun, H. (2020). A Depth Camera-Based Lightweight Visual SLAM Algorithm. Beijing University of Posts and Telecommunications.
18. Khairuddin, A.R., Talib, M.S., & Haron, H. (2017). Review of Simultaneous Localization and Mapping (SLAM). Faculty of Computing, Universiti Teknologi Malaysia.

19. Park, J.H., Baeg, S.H., & Baeg, M.H. (2018). An Intelligent Navigation Method for Service Robots in the Smart Environment. KITECH, Korea.
20. Murwantara, I.M., Hardjono, B., Tjahyadi, H., & Putra, A.S. (2019). Consolidation of SLAM and Signal Reference Point for Autonomous Robot Navigation. Universitas Pelita Harapan.
21. Lesch, V., Krupitzer, C., & Tomforde, S. (2020). Emerging Self-Integration through Coordination of Autonomous Adaptive Systems. University of Wurzburg.
22. Cheong, W.S., & Kamarulzaman, S.F. (2021). Implementation of Robot Operating System in Smart Garbage Bin Robot with Obstacle Avoidance System. Universiti Malaysia Pahang.
23. Chopra, R. (2020). Artificial Intelligence in Robotics: A Review Paper. Indus University.
24. Zhang, X., Zeng, Q., Meng, Q., Xiong, Z., & Qian, W. (2018). Design and Realization of a Mobile Seamless Navigation and Positioning System Based on Bluetooth Technology. IEEE Sensors Journal.
25. Thrun, S. (2002). Robotic Mapping: A Survey. Carnegie Mellon University.
26. Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic Robotics. MIT Press.
27. Montemerlo, M., Becker, J., & Thrun, S. (2002). Fast SLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. MIT Press.
28. Kummerle, R., Grisetti, G., & Burgard, W. (2011). g2o: A General Framework for Graph Optimization. IEEE Transactions on Robotics.
29. Kaess, M., Johannsson, H., & Leonard, J.J. (2012). iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree. The International Journal of Robotics Research.
30. Stachniss, C., & Burgard, W. (2008). Mobile Robot Navigation Using Grid Maps. IEEE Transactions on Robotics.