IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 10, Issue 3 - V10I3-1146)
Available online at: https://www.ijariit.com

INTELLIGENT NAVIGATION THROUGH RECOMMENDED FLOOR PLANS

Ananya Babu
ananyababu25@gmail.com
Mar Athanasius College of Engineering, Kothamangalam, Kerala

Jawhara Fathima
jawharafathima@gmail.com
Mar Athanasius College of Engineering, Kothamangalam, Kerala

Khrithikesh M U
khrithikesh2002@gmail.com
Mar Athanasius College of Engineering, Kothamangalam, Kerala

Dr. Elizabeth Isaac
elizabethisaac@mace.ac.in
Mar Athanasius College of Engineering, Kothamangalam, Kerala

ABSTRACT

Navigation is essential in the working of a helper robot. The lack of a floor plan hinders the training of the helper bots. A floor plan recommender system will help solve this concern. Creating building floor plans is also very essential in the planning and construction of a building. Moreover, the integration of robotic mobility would provide a real-time understanding of the spatial layouts, which contributes to a more efficient and dynamic design implementation. The goal of this project is to develop a SimGNN-based recommendation system that suggests floor plans that satisfy the client's requirements about the spatial relationship. The SimGNN-based model calculates the similarity between the graphs after transferring the spatial relationship in the floor plan to the graph. We aim to utilize the recommended floor plans to enable the autonomous navigation of a robot. By mentioning the start and the finish point of the robot, a pathis created and the robot will maneuver through the obtained path. Through this integration of a recommendation system with robotic mobility, we aim to optimize the training process of helper bots along with improving the design and construction process of a building.

Keywords: *Floorplan Recommendation, Autonomous Navigation*

I. INTRODUCTION

In the modern world, helper robots play a pivotal role in human assistance. These human-assisting machines have become an important part of our lives. Robot assistance is crucial for automating repetitive and labor-intensive tasks. Helper robots excel in performing tasks with precision and consistency, reducing the burden on human workers. Robots can work tirelessly without fatigue, leading to increased productivity in many industries like manufacturing, logistics, etc. Robot assistance enhances accessibility for individuals with disabilities, providing them with tools to navigate the world more independently. From robotic exoskeletons for mobility support to robotic arms for assisting with daily tasks, these technologies contribute to a more inclusive society.

However, there is one issue that hinders the functionalityof helper bots, the lack of floor plans. The lack of floor plans poses a significant challenge for helper bots operatingin environments like homes, offices, and construction sites where efficient navigation is essential for performing tasks effectively. A floor plan serves as a crucial reference, providinga structured overview of the space and aiding the bots in understanding the spatial layout. Without a predefined floor plan, helper bots lack a systematic understanding of the space they operate in. This can result in inefficient navigation, asthe bots may struggle to identify optimal routes to reach their destinations within the environment. The absence of a floor plan makes it challenging for helper bots to accurately locate specific rooms or areas within a given space. This can leadto delays in task execution, as the bots may need to explorethe environment more extensively to fulfill their objectives.The overall efficiency of tasks performed by helper bots is compromised when navigation is impeded. Delays, errors, and the inability to seamlessly move from one task to another can undermine the productivity gains that helper bots are designedto provide.

The creation of floor plans presents a significant issue in the efficient design and construction of buildings as well. Often, inaccuracies in floor plans can lead to a misalignment with the functional requirements of the structure, resulting in inefficient space utilization. This misalignment not only compromises the building's intended purpose but also translates into wasted square footage and increased construction costs. Comprehending users' needs poses a substantial challenge in the context of floor plan creation for efficient building design and construction. The complexity lies in translating diverse user requirements into a coherent spatial arrangement thataligns with their functional expectations.

This problem of lack of floor plans can be rectified by theuse of a floor plan recommendation system. This approach involves users providing information about the spatial layout, which is then utilized to recommend floor plans. Users can input spatial data into the system. This user-generated input provides valuable information that forms the basis for recommending a personalized and accurate floor plan. This personalized approach ensures that the floor plan aligns closely with the actual layout of the space, enabling effective navigationfor helper bots. Then by mentioning the start and the final points of the bot, a path can be devised. The robot can be madeto follow the path. This can significantly help in the training ofhelper robots. Moreover, a floor plan recommendation system also helps in an efficient house planning process.

II. RELATED WORKS

Currently, there are some approaches and methodologies forfloor plan recommendation and intelligent navigation systems but most of them are either time-consuming or expensive.

A. Graph2Plan

This innovative learning framework introduces an automated approach to generating floor plans by combining deep neural networks with user-in-the-loop design inputs, allowing users to provide sparse design constraints in the form of a layout graph. These constraints are pivotal in representing user preferences and spatial requirements. The central element of this framework is a deep neural network called Graph2Plan,designed to transform a layout graph alongside a buildingboundary into a coherent floor plan that adheres to specified constraints. When presented with a building boundary input, Graph2Plan enables users to define room counts and other layout specifics, leveraging a database to retrieve a selectionof floor plans matched to the constraints. For each layout graph retrieved from the database and combined with the input boundary, Graph2Plan executes a two-step process: firstly, generating a raster floor plan image, and subsequently refining this image into boxes that represent individual rooms. Graph2Plan's training is conducted on RPlan, a substantial dataset comprising 80,000 annotated floor plans. The neural network primarily utilizes convolutional processing, leveraging a graph neural network (GNN) for the layout graph and a conventional image convolution for raster floor plan images. This framework's adaptability and quality in generating floor plans across diverse user inputs are demonstrated through comprehensive evaluations, including qualitative and quantitative assessments, ablation studies, and comparisons against state-of-the-art methodologies. These evaluations underscore the efficacy and versatility ofthe proposed floor plan generation approach, illustrating its potential for various design scenarios and user preferences. [**?**].

B. LayoutGMN

LayoutGMN is an innovative system designed to assessstructural similarities between 2D layouts through the utilization of graph matching networks (GMN). The primary goal of LayoutGMN is to undergo training using floor plans that have been labeled as either similar or dissimilar based on intersection-over-union (IOU) metrics, employing a method of weakly supervised learning with labeled positive and negative examples determined by a specific threshold. This approach stands distinct from the recommendation system outlined in the study, primarily due to the methodology utilized by LayoutGMN. Specifically, LayoutGMN adopts weak labels derived from pixel-wise IOUs for graph neural network (GNN) training, in contrast to the supervised learning technique employed by the recommendation system, which relies on ground truth data. The core methodology of LayoutGMN revolves around the assessment of similarity by evaluating the direct comparison of graphs extracted from floor plans. This process involves leveraging GMNs to establish the similarity between layout structures based onspecific learned features and metrics. Through this approach, LayoutGMN aims to contribute to the field of layout analysis and similarity assessment, offering a novel perspective on leveraging weakly supervised learning for graph-based layout comparisons.

C. Rescue Bot

Rescue Bot is an autonomous ground rover designed to navigate post-disaster terrain, locate survivors, and enhance disaster response. Invented by Muhammad Zain and Vishnuraj A. It is an innovative search and rescue robot designed for swift and effective emergency response. Boasting a durable chassis with high-traction treads, the robot excels in navigating challenging terrains. Equipped with infrared cameras, gas detectors, and a versatile robotic arm, the Rescue Bot enhances its perception capabilities and aids in object manipulation, obstacle clearance, and victim assistance. Its user-friendly remote control interface, coupled with an intelligent software architecture featuring AI algorithms, enables autonomous navigation and decision-making. Rigorously tested in simulated disaster scenarios, the Rescue Bot proves its adaptability and reliability in dynamic environments. With a modular design for easy customization, this robot stands as a cutting-edge solution, providing invaluable support to emergency responders in their mission to swiftly and efficiently save lives.

III. PROPOSED MODEL

The project consists of two modules which comprise the floor plan recommendation system and the autonomous navigating robot. Module 1 which comprises the software part, involves the dataset collection, training, and testing process. We use a graphical neural network namely SimGNN for the recommendation system. Module 2 consists of the hardware of the robot and the ROS library for the control and coordination of the robot.

IV. SOFTWARE DETAILS

The methodology of the study involved the development and implementation of a similarity-based recommendation system for house floor plans. This system utilized the SimGNN model, an AI technology capable of learning graphs, to analyze and recommend floor plan alternatives based on spatial relationships. The researchers established 120 spatial relationship types for house floor plans, expanding the range of options for users and enabling a more accurate reflection of their requirements. The system was trained and tested using various parameter configurations to maximize the performance of the SimGNN model, resulting in high accuracy in predicting spatial relationships. To implement SimGNN in a software application, the process begins with the preparation of the dataset. This involves collecting and structuring the graph data that represent spatial relationships or any relevant information for the intended application. The dataset is then used to train

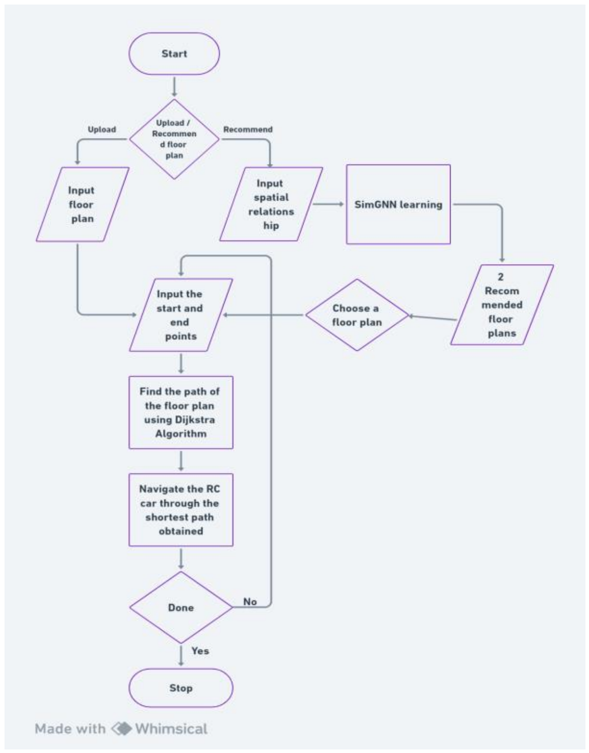
Start
Upload / Recommend floor plan
Upload
Recommend
Input floor plan
Input spatial relationship
SimGNN learning
2 Recommended floor plans
Choose a floor plan
Input the start and end points
Find the path of the floor plan using Dijkstra Algorithm
Navigate the RC car through the shortest path obtained
Done
No
Yes
Stop
Made with Whimsical

Fig. 1. Flow Chart

the SimGNN model, a critical step that involves feeding the graph data into the model and optimizing its parameters. During training, experimentation with various configurations and hyperparameters is essential to find the most effective settings for the model. This iterative process may require adjustments to ensure that the model accurately learns thepatterns and relationships within the data. Once the model is trained, it needs to be integrated into the software application. This integration typically involves developing an interface or API that allows the software to interact with the model for graph similarity computation. Thorough testing and validationof the integrated software are crucial to ensure that the SimGNN model functions as expected within the application. This includes validating its performance by comparing thecomputed graph similarities with known ground truth data. After rigorous testing, the software is optimized for efficiency and scalability before being deployed for practical use. Optimization may involve considerations for computational resources, response time, user access, system compatibility, and performance monitoring. Throughout the implementation process, documentation, adherence to software development best practices, and considerations for data privacy and security are essential. By following these steps, the implementation of SimGNN in a software application can effectively leverage itscapabilities for graph similarity computation and other relevanttasks

V. HARDWARE DETAILS

Implementing robot navigation involves integrating several key components seamlessly. The primary focus of this project is navigating a helper robot. Visualization of the generated model floor plan is achieved using a prototype RC car controlled by the ROS library through a Raspberry Pi. Robot navigation integrates key components like Encoder Motors and Raspberry Pi for precise movement and control. The encoder motor, chosen for accuracy, converts electrical pulses intomechanical motion. A motor driver interprets digital signals from the Arduino Mega to regulate speed, direction, and micro-stepping. Here Arduino Mega acts as a microcontroller for regulating wheel movement and mapping the encoder readings. With a stable power supply and ROS integration, the system enables autonomous navigation in dynamic environments, with intelligent robot behavior. The components we used for the implementation of hardware are Raspberry Pi 4 model B for processing the encoder readings, implementation of ROS Libraries, RPLiDar A1 for mapping the environment, 12V 1200mAh Battery, 12V to 5V buck converter to step down the voltage for connecting to Raspberry Pi, Arduino MEGA,2 Encoder motor (RMCS-5013) and RMCS-2303 Drivers for controlling the motor. The wheel movement is controlled using askid steer mechanism and differential drive. Here we use the Ubuntu 20.04 Operating system to install the ROS 1 noetic version. By using Rosserial for interface drivers attached to Arduino in the ROS environment. The ROS1 installed in the Raspberry Pi acts as the ROS master node and the Arduino actsas the ROS subscriber node. Hector mapping in SLAM is usedfor mapping the environment. Then the helper bot runs throw the map with the help of the A* algorithm.

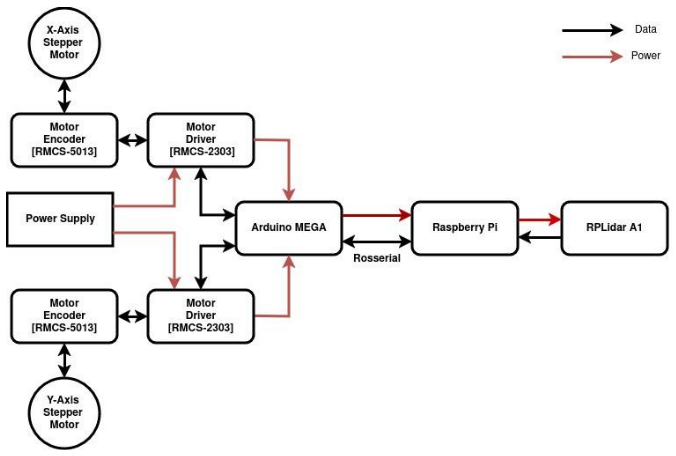

Fig. 2. Block diagram representation of the model

The hardware requirements for the project are as follows:

A. A* algorithm

In computer science and artificial intelligence, the A* method is a basic pathfinding technique that finds the shortest route between nodes in a weighted graph. By iteratively evaluating nodes with the lowest total cost estimate of achieving the objective, A* efficiently navigates from a defined start nodeto a goal node by combining aspects of Dijkstra's algorithm and heuristic search. Through a loop, the algorithm selects the node with the lowest estimated cost, evaluates its neighbors, and updates its costs accordingly. Termination occurs either upon reaching the goal or exhausting the open list without success. A* employs a heuristic function to estimate the cost from each node to the goal, guiding the search efficiently. The function f(n)=g(n)+h(n)f(n)=g(n)+h(n) represents the total estimated cost from the start node to the goal node through node nn, where g(n)g(n) is the actual cost from the start node to node nn and h(n)h(n) is the estimated cost from node nnto the goal. A* guarantees finding the shortest path when an admissible heuristic is used, although an inadmissible heuristic may still yield a path, albeit not necessarily the shortest.

B. Raspberry Pi

In an autonomous navigation system using the ROS library. The Raspberry Pi serves as the central processor, managing data from sensors and translating decisions into control signals for navigation. Through ROS, the Raspberry Pi facilitates communication and coordination among different components,allowing the RC car to navigate autonomously.

C. LiDAR Sensor

It functions by emitting laser pulses and gauging the duration for them to return after interacting with nearby objects. Lidar furnishes highly precise three-dimensional layouts of the surroundings, empowering autonomous vehicles to grasp and move through their environment with accuracy. By detecting obstacles, determining distances, and creating detailed mapsin real time, lidar plays a vital role in ensuring the safety and efficiency of autonomous navigation systems. Its ability to work effectively in various lighting conditions and environments makes it an indispensable tool for achieving reliable autonomous navigation.

D. Arduino Mega

The ATmega2560 microprocessor serves as the foundation for the Arduino Mega microcontroller board. It is a member of the open-source Arduino hardware platform family, which is renowned for its ease of use and accessibility for DIY projects and electronics prototyping. With so many digital and analog input/output ports, the Arduino Mega offers a lot of options for attaching sensors, actuators, and other electrical parts. It also has built-in capabilities including power and programming USB connectivity, along with a range of communication interfaces like SPI, I2C, and UART that allow for easy communication with external devices. Robotics, automation, data logging, and control systems are just a few of the many applications that the board's adaptability and expandability make it ideal for. Additionally, the Arduino development environment provides a simplified version of the C++ programming language together with an intuitive interface for programming the board. When it comes to designing and creating electronic projects, professionals, educators, and hobbyists all like the Arduino Mega because of its cost, ease of use, and versatility.

E. Encoder

The RMCS-5013 motor encoder stands out as a reliable solution for accurately measuring a motor shaft's rotational position and speed, making it indispensable in various fieldssuch as motion control systems, CNC machines, and robotics. At the heart of the RMCS-5013 are its rotary encoder and associated electronics. As the motor shaft rotates, the rotary encoder generates electrical signals, which are thenprocessed by the electronics to determine the motor's position and velocity with precision.

One of the standout features of the RMCS-5013 is its exceptional resolution, achieved through a combination of a high-resolution optical or magnetic encoder disk and sophisticated signal processing techniques. This ensures that it delivers accurate feedback on motor position and velocity, even indemanding applications.

Moreover, the RMCS-5013 offers versatility in integration with various motor control systems, supporting multiple output formats including quadrature signals and serial communication protocols like SPI or I2C. This simplifies the integration process and enhances compatibility across different platforms. Additionally, the RMCS-5013 may incorporate advanced features such as integrated error detection and correctionsystems. These capabilities contribute to ensuring precise andreliable performance, even in challenging environments where accuracy is paramount.

F. Motor Driver

The RMCS-2303 motor driver module is engineered to deliver precise control over the speed and direction of DC motors, making it an essential component in robotics, automation, and motor control applications where accuracy is paramount. Notably, the RMCS-2303 is designed to accommodate a wide range of input voltages, ensuring compatibility with various power sources. Its bidirectional control capability allows motors to rotate in both forward and reverse directions, offering flexibility in motion control tasks. Additionally, the module enables smooth speed control of connected motors through pulse-width modulation (PWM) control, ensuring efficient and precise motor operation.

To safeguard both the motors and the driver module from potential damage, the RMCS-2303 may incorporate integratedprotections such as overcurrent prevention mechanisms.

Thanks to its digital communication interface, typically interfacing with a microcontroller or computer, the RMCS-2303 motor driver module seamlessly integrates into existing electronic systems. By sending appropriate control signals to the module, users can execute a variety of motion controltasks with precision, including fine-tuning motor speed and direction.

VI. RESULT

In this project, we developed a system utilizing the SimGNNmodel to recommend house floor plans, with the goal of enhancing architectural design processes. We identified 120 unique spatial relationships and optimized the system with various configurations to accelerate floor plan generation and offer more objective suggestions. Additionally, we integrated a hardware robot to provide users with visual representations of these recommended floor plans. This integration of advanced machine learning and physical technology aims to transform how architects and users collaborate on architectural designs, simplifying the process and improving efficiency. Lookingahead, our plans involve further enhancing the system’s performance, exploring additional spatial relationships, andrefining the hardware interface to enhance user-friendlinessfor architectural design tasks. These future developments will contribute to advancing the capabilities of our recommendation system, making architectural design more accessible and streamlined for professionals and users alike.

VII. FUTURE SCOPE

1) Expansion of Spatial Relationships: We can identify and include more types of spatial relationships to provide even more accurate and personalized floor plan recommendations.

2) Integration of User Preferences: By incorporating user preferences and feedback, we can personalize the recommendations further. This could involve creating user-friendly interfaces for architects and users to refine suggested floor plans based on specific needs and preferences.

3) Enhanced Hardware Integration: The hardware component, such as the helper robot for navigation, can be enhanced to offer additional features. For instance, integrating augmented reality (AR) or virtual reality (VR) technologies could allow users to explore recommendedfloor plans in immersive ways.

4) Performance Optimization: Continuous improvement of the system’s performance is crucial as architectural designs become more complex. This includes refining training processes and exploring advanced parameter-tuning techniques.

5) Application in Other Domains: The techniques and technologies used in this project can be adapted for other areas beyond architectural design, such as urban planning or interior design.

In summary, future improvements to this project involve refining the recommendation system, exploring new applications, and addressing ethical considerations. These advancements aim to make architectural design methodologies more accessible, efficient, and user-centric.

VIII. CONCLUSION

In conclusion, this project signifies a substantial leap forward in architectural design methodologies through the implementation of a recommendation system for house floor plans using the SimGNN model. We meticulously defined a comprehensive set of 120 spatial relationship types and fine-tuned thesystem with various parameter configurations to significantly enhance performance and accuracy. This innovative approach not only streamlines the generation of floor plan alternatives but also delivers more objective recommendations grounded in spatial insights, thus greatly enhancing the efficiency and effectiveness of architectural design processes.

Incorporating a hardware robot for navigation enhances userinteraction and visualization of floor plans. By combiningtechnology and physical interfaces, we aim to simplify architectural design and make it more efficient for architects and users. This project sets a new standard for integrating technology into architectural workflows, showcasing the potential for innovation in the field. The successful implementationhighlights how technology can transform traditional practices in architecture. This project inspires similar initiatives, pushingarchitectural design toward greater efficiency and creativity inthe digital era.

REFERENCES

[1] Hyejin Park, Hyegyo Suh, Jaeil Kim, Seungyeon Choo, “Floor plan recommendation system using graph neural network with spatial relationship dataset”, Journal of Building Engineering 71, March 2023, 106378, Elsevier.

[2] Ahmet Vatan, Seden Dogan, “What do hotel employees think about service robots? A qualitative study in Turkey”, Tourism Management Perspectives, January 2021, 100775, Elsevier.

[3] Alexey S. Matveev, Andrey V. Savkin, Michael Hoy, Chao Wang, “4 - Shortest path algorithm for navigation of wheeled mobile robots among steady obstacles”, Safe Robot Navigation Among Moving and Steady Obstacles, 2016, Elsevier.

[4] Zhudan Chen, Dazi Li, Minghui Liu, Jun Liu, “Graph neural net- works with molecular segmentation for property prediction and struc- ture–property relationship discovery”, Computers and Chemical Engi- neering, November 2023, Elsevier.

[5] Linus Nwankwo, Clemens Fritze, Konrad Bartsch, Elmar Rueckert, “ROMR: A ROS-based open-source mobile robot”, HardwareX, 2023, Elsevier.

[6] Pradumn Mishra, Urja Jain, Siddharth Choudhury, Surjeet Singh, Anish Pandey, Abhishek Sharma, Ramanpreet Singh, Vimal Kumar Pathak, Kuldeep K. Saxena, Anita Gehlot, “Footstep planning of humanoid robot in ROS environment using Generative Adversarial Networks (GANs) deep learning”, Robotics and Autonomous Systems, December 2022, Elsevier.

[7] A.G. Patil, M. Li, M. Fisher, M. Savva, H. Zhang, ”LayoutGMN: Neural graph matching for structural layout similarity”, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, pp. 11043–11052, 2021

[8] N. Nauata, K.H. Chang, C.Y. Cheng, G. Mori, Y. Furukawa, House- gan: Relational generative adversarial networks for graph-constrained house layout generation, in: European Conference on Computer Vision, (ECCV), pp. 162–17, 2020

[9] Muhammed Zain, Vishnuraj A, ”Rescue Bot”, hackster.io, December 2023

[10] R. Hu, Z. Huang, Y. Tang, O. van Kaick, H. Zhang, and H. Huang, ”Graph2Plan: Learning Floorplan Generation from Layout Graphs”, ACM Trans. Graph., Vol. 39, No. 4, Article 1., July 2020.

[11] Jianfeng Zheng , Shuren Mao, Zhenyu Wu, Pengcheng Kong and Hao Qiang, ”Improved Path Planning for Indoor Patrol Robot Based on Deep Reinforcement Learning”, Symmetry, 14, pp. 132, 2022