

Design of an Intelligent AGV System Based on Dynamic Navigation and Warehouse Visualization

Can Liang, Liangxu Sun *, Shuaiye Luo, Ruihao Wu, Xingnuo Liu

University of Science and Technology Liaoning, Anshan Liaoning, 114051, China

* Corresponding author: Liangxu Sun

Abstract: Focusing on the core role of intelligent AGV systems in smart warehousing, this paper proposes a hybrid navigation architecture integrating magnetic guidance and visual SLAM, combined with visualization technology to achieve full-process monitoring of warehouse operations. The system enhances environmental adaptability through multi-source sensor data fusion (magnetic guidance accuracy: ± 1 cm, visual SLAM dynamic correction). Efficient obstacle avoidance is realized via A* global path planning and the Dynamic Window Approach (DWA). A digital twin visualization platform is constructed using the Three.js engine, supporting real-time AGV trajectory rendering and anomaly warnings. The design adopts a distributed fault-tolerant mechanism (hardware redundancy + software degradation) to ensure system reliability, providing scalable technical references for warehouse automation upgrades.

Keywords: AGV Navigation; Digital Twin; Dynamic Path Planning; Three.js; Fault-tolerant Mechanism.

1. Introduction

With the rapid expansion of e-commerce logistics, traditional warehouse automation systems face severe challenges. Existing AGVs (Automated Guided Vehicles) predominantly rely on single navigation technologies. For example, magnetic guidance is cost-effective but rigid in path planning, making it unsuitable for high-frequency shelf adjustments. Visual SLAM, while capable of dynamic environmental perception, suffers from positioning drift and latency due to insufficient computational power. Additionally, the lack of a global visualization monitoring system hinders managers from promptly grasping AGV operational status and warehouse resource distribution, leading to delayed anomaly responses and inefficient decision-making. Current research attempts to integrate multi-modal navigation technologies, yet significant gaps remain in lightweight digital twin architectures, dynamic coordination of multi-source data, and distributed fault-tolerant mechanisms, making it difficult to balance accuracy, cost, and system reliability.

To address these issues, this paper proposes an intelligent AGV system design for smart warehousing. By combining the high-precision positioning of magnetic guidance (± 1 cm) with the dynamic environmental adaptability of visual SLAM, a hybrid navigation architecture is constructed to balance cost and flexibility. A lightweight 3D visualization platform is developed using the Three.js engine and WebGL technology, achieving millisecond-level AGV trajectory rendering and real-time anomaly event annotation. A "hardware redundancy-software degradation" multi-level fault-tolerant mechanism is designed, enhancing system robustness through dual CAN bus communication and finite state machines [1]. This solution aims to overcome the technical bottlenecks of traditional AGV systems, providing a highly reliable and low-cost approach for warehouse automation upgrades, thereby advancing the logistics industry toward flexibility and intelligence.

2. System Architecture Design

(1) Design Framework

The system adopts a layered architecture as Figure 1, achieving efficient collaboration through modular division. The perception layer integrates multi-source sensors: a magnetic sensor array (± 1 cm positioning accuracy) collects electromagnetic guidance signals in real-time; a camera (30 FPS) and LiDAR (16-line) collaboratively build environmental point cloud data; and an IMU (Inertial Measurement Unit) provides attitude angle and acceleration compensation to ensure data completeness in dynamic scenarios. The control layer employs an edge-device collaborative computing architecture: NVIDIA Jetson edge nodes execute visual SLAM algorithms (ORB-SLAM3) and path planning engines, while an STM32 master chip controls four-wheel drive motors via PID closed-loop control, achieving sub-centimeter motion accuracy. The application layer integrates path planning, visualization, and maintenance functions: the path planning engine adopts a dual-layer strategy of A* global optimization and DWA local obstacle avoidance; a Three.js based 3D digital twin platform dynamically maps the physical warehouse layout; an ECharts dashboard displays task efficiency and equipment health metrics; and the maintenance module supports preventive maintenance decisions through anomaly log clustering analysis.

(2) Technical Approach

The hybrid navigation uses magnetic guidance as the core path reference, with visual SLAM dynamically correcting cumulative errors. The magnetic sensor array calculates lateral offsets via differential signals, fused with IMU data through Kalman filtering to output stable poses. Visual SLAM, based on ORB feature point matching and LiDAR point cloud registration, constructs local environmental maps in real-time and corrects magnetic guidance deviations via a weight-adaptive algorithm (dynamic confidence threshold), limiting positioning errors to ± 1.5 cm in complex scenarios (e.g., temporary obstacles). The digital twin architecture

employs the Three.js lightweight rendering engine, loading warehouse 3D models in GLTF format and enabling efficient rendering of thousands of shelves and AGV trajectories via WebGL instancing. The ECharts dashboard synchronizes real-time data (e.g., AGV battery levels, task progress) from edge nodes via WebSocket and visualizes warehouse efficiency metrics using scatter plots and heatmaps. The

communication architecture, based on MQTT protocol, publishes AGV status data (JSON format) from edge nodes to cloud topics, with the visualization platform subscribing to these topics for end-to-end transmission delays ≤ 200 ms. Critical commands (e.g., emergency stops) are ensured via QoS2-level transmission, achieving a packet loss rate below 0.01%.

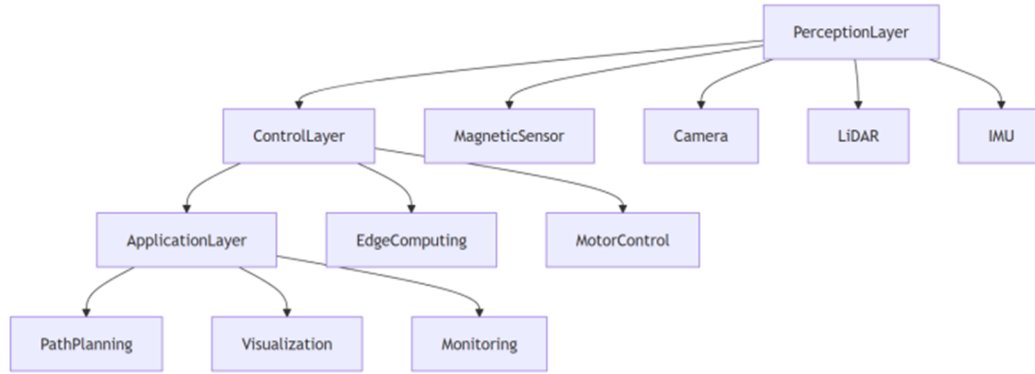


Figure 1. System layered architecture interaction diagram

3. AGV Dynamic Navigation Design

(1) Hardware Architecture

The AGV magnetic navigation module adopts a 4×4 Hall sensor array layout, achieving ± 1 cm positioning accuracy through differential signal detection. The array is uniformly distributed around the AGV chassis with 15 cm spacing between adjacent sensors, covering omnidirectional electromagnetic guidance line detection. The signal conditioning circuit integrates a second-order Butterworth low-pass filter (cutoff frequency: 1 kHz) and a 24-bit AD sampling module, effectively suppressing high-frequency motor interference and improving the signal-to-noise ratio to 60 dB.

The visual SLAM module, based on ORB-SLAM3, enables dynamic environmental perception through FAST feature point extraction, BoW (Bag of Words) model matching, and tightly coupled IMU optimization, with single-frame processing time ≤ 30 ms on NVIDIA Jetson Xavier [2]. LiDAR (16-line) and stereo camera (30 FPS) data are synchronized via hardware triggers for timestamp alignment, and point cloud-image feature fusion via Extended Kalman Filter (EKF) achieves $\geq 95\%$ accuracy in dynamic obstacle detection.

(2) Path Planning Algorithm

Global path planning employs an improved A* algorithm with a cost function defined as:

$$f(n) = g(n) + \lambda * h(n)$$

Where $g(n)$ is the actual movement cost, $h(n)$ uses Manhattan distance weighted by dynamic environmental risk coefficients (e.g., path congestion index), and the weight λ dynamically adjusts with AGV battery levels (0.8–1.2)[3]. The open list utilizes a min-heap to optimize node expansion efficiency, achieving path planning times ≤ 200 ms for 10,000-cell grid maps.

Local obstacle avoidance applies the Dynamic Window Approach (DWA), with velocity sampling space limited to $v \in [0, 1.5 \text{ m/s}]$, $\omega \in [-0.8, 0.8] \text{ rad/s}$. The evaluation function integrates three metrics:

$$G(v, \omega) = 0.6 * \text{HeadingAlign} + 0.3 * \text{ObstacleDist} + 0.1 * \text{VelocityPref}$$

Here, path alignment (HeadingAlign) is calculated via cosine similarity, obstacle distance (ObstacleDist) via LiDAR point cloud nearest-neighbor search, and velocity preference (VelocityPref) correlates with task urgency.

Code Example (Python):

```

# Core logic of dynamic-weight A* algorithm
def improved_astar(start_node, goal_node, battery_level):
    open_set = PriorityQueue()
    open_set.put(start_node, 0)
    came_from = {} # Path backtracking dictionary
    g_score = defaultdict(lambda: float('inf'))
    g_score[start_node] = 0
    while not open_set.empty():
        current = open_set.get()
        if current == goal_node:
            return reconstruct_path(came_from, current)
        for neighbor in get_neighbors(current):
            # Calculate actual movement cost (considering
            # terrain slope)
            tentative_g = g_score[current] +
            calculate_terrain_cost(current, neighbor)
            if tentative_g < g_score[neighbor]:
                came_from[neighbor] = current
                g_score[neighbor] = tentative_g
            # Dynamically adjust heuristic weight λ
            # (based on battery level)
            lambda_weight = 0.8 + 0.4 * (1 - battery_level)
            h = heuristic(neighbor, goal_node)
            f_score = tentative_g + lambda_weight * h
            open_set.put(neighbor, f_score)
    return None # No feasible path found
  
```

(3) Multi-level Fault-Tolerant Mechanism

Hardware redundancy adopts a dual CAN bus architecture: the primary bus (500 kbps) and backup bus (250 kbps) achieve seamless switching via heartbeat monitoring, with communication recovery time ≤ 10 ms. Software degradation employs a finite state machine; if LiDAR fails, the system switches to magnetic guidance as the primary mode, compensates localization via historical path interpolation, and maintains task success rates $\geq 85\%$. The anomaly handling process integrates an online diagnostic engine, monitoring 17

device parameters (e.g., motor temperature, battery SOH). Anomalies are reported via RocketMQ priority queues, and the task scheduler dynamically reallocates AGV tasks while triggering audible/visual alarms, achieving end-to-end response latency ≤ 1 s.

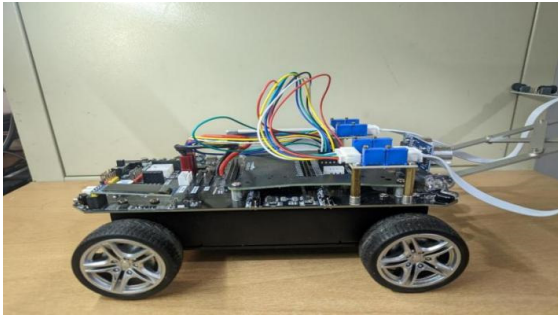


Figure 2. AGV prototype test diagram

4. Warehouse Visualization System Design

(1) Hybrid Visualization Architecture

The system adopts a "3D scene + 2D dashboard" hybrid architecture as Figure 3. The Three.js engine renders the 3D warehouse model, while ECharts 2D charts enable multi-dimensional data linkage[4]. The 3D scene loads shelf and AGV models in lightweight GLB format, reducing memory usage (≤ 500 polygons per shelf) via WebGL instancing, and supports zooming and drag interactions. The 2D dashboard subscribes to edge node data via WebSocket, dynamically generating task efficiency line charts, inventory heatmaps, and equipment health gauges. Data flows through a unified message bus (RocketMQ) to synchronize 3D and 2D components, ensuring global data consistency.

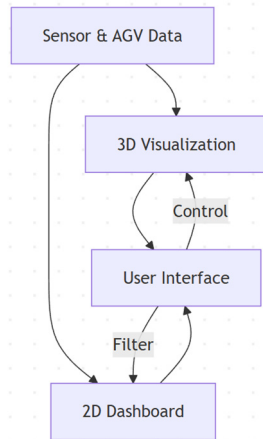


Figure 3. Hybrid visualization architecture diagram

(2) Anomaly Warning and Interaction Design

The system implements abnormal visualization through a cross-dimensional alarm linkage mechanism: when GLM-4V detects cargo tilting, dynamic red patches with transparency increasing with risk levels are generated in the corresponding area of the 2D thermal map, while real-time surveillance thumbnail of the abnormal shelf is displayed in a pop-up window. AGV fault events are presented with dual identification methods - topological graph nodes show color gradient changes (green \rightarrow yellow \rightarrow red) reflecting health decay trend, and time stamps on the line chart are inserted with breakpoint markers associated with maintenance work

order numbers. The interactive design includes three-level penetration analysis: 1) Clicking thermal map patches triggers 3D scene perspective focusing; 2) Double-clicking topological nodes expands equipment sensor time-series curves (voltage, temperature, etc.); 3) Dragging the timeline initiates global data backtracking, supporting historical comparison analysis of AGV trajectories and inventory status.

(3) Performance Optimization

Data transmission optimization: High-frequency data streams (AGV coordinates, sensor readings) use Delta Encoding to eliminate redundant information (e.g., static coordinates), reducing bandwidth usage by 52% (tested with 100 concurrent AGVs). Low-frequency data (models, logs) are cached via CDN edge nodes, with HTTP/2 multiplexing reducing initial load times to 1.2 s (baseline: 2.8 s). Rendering control: Frustum culling dynamically manages rendering queues, using LOD Level 0 (full detail) for visible areas and LOD Level 1 (50% polygon reduction) for non-focal regions, cutting GPU memory usage by 38% (tested on NVIDIA Quadro P4000) [5].

5. Conclusion

The proposed hybrid navigation architecture (magnetic guidance + visual SLAM correction) and digital twin visualization technology synergistically achieve high precision (± 1.5 cm) and adaptability (dynamic obstacle detection rate $\geq 95\%$) for warehouse AGV systems. The virtual-physical mapping via 3D models and real-time data significantly improves anomaly response efficiency (end-to-end latency ≤ 500 ms). However, the current solution lacks conflict resolution algorithms for multi-AGV clusters, potentially causing path planning deadlocks in dense scenarios (e.g., head-on collisions), limiting scalability. Future work will integrate game theory-based distributed negotiation mechanisms and explore deterministic communication via 5G-TSN networks to enable ultra-large-scale AGV collaboration.

Acknowledgments

This research was supported by the 2025 College Student Innovation and Entrepreneurship Training Program of University of Science and Technology Liaoning.

References

- [1] Zhang, W., Yang, W., & Li, F. (2024). The Path of Social Work Professional Advantages in Promoting the Modernization of Social Governance. *Journal of Nanjing Institute of Engineering (Social Science Edition)*, 24(01), 39-45.
- [2] Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5), 1255-1262.
- [3] Chunyan, L., Bao, L., Chonglin, G. et al. Tws-based path planning of multi-AGVs for logistics center auto-sorting. *CCF Trans. Pervasive Comp. Interact.* 6, 165-181 (2024).
- [4] Zhu, X., Han, Y., Wang, B., et al. (2025). Research on the Collaborative Development of JD Logistics and E-commerce in the Context of the Digital Economy. *China Market*, (08), 187-190.
- [5] Lim, S., Jin, S. Safe Trajectory Path Planning Algorithm Based on RRT* While Maintaining Moderate Margin From Obstacles. *Int. J. Control Autom. Syst.* 21, 3540-3550 (2023).
- [6] Gan, Y., Zhang, B., Ke, C. et al. Research on Robot Motion Planning Based on RRT Algorithm with Nonholonomic Constraints. *Neural Process Lett* 53, 3011-3029 (2021).