ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Design of Real-Time Path Adjustment Mechanism for AGV Obstacle Recognition Based on Visual Navigation System

Houhuang Bi[1,*] *and Jun Huang*[2]

[1] School of Public Administration, Wuhu Vocational and Technical University, Wuhu, Anhui, 241003, China
[2] School of Intelligent Manufacturing, Wuhu Vocational and Technical University, Wuhu 241003, Anhui , China

SUMMARY: *As wheel-driven intelligent robots, Automated Guided Vehicles (AGVs) are increasingly utilized in warehousing, military, and household fields owing to their intelligence, flexibility, and convenience. In this paper, obstacle detection data through obtaining columnar pixels according to the structure of the AGV system from environmental images are presented. The columnar pixel algorithm, which combines conditional probability models and dynamic programming, is used for obstacle zone recognition and navigable area segmentation with the environmental image data. This allows rapid and accurate obstacle recognition and detection under the condition of parallax. Then, using model-based method (the Kalman filter), the trajectory of obstacles can be predicted, which builds a mathematical motion model to forecast the collision point. It will offer references for adjusting the path of AGVs. Taking into account the principle of AGV obstacle avoidance, the path adjustment strategy applies the concept of global guidance combined with local planning. With the help of the advantage of the Q-learning algorithm in adapting the environment, local path planning strategies are proposed. Through the support of the visual navigation system, the path adjustment strategy in this study can obtain the shortest path planning time with average risk being 0.15.*

KEYWORDS: *Q-learning algorithm; AGV obstacle recognition; path planning; motion model; visual navigation system*

1 Introduction

Automated Guided Vehicle (AGV) is an intelligent robot with mobility. During operation, AGV has the capability for automatic navigation, which uses rechargeable batteries such as lead-acid battery systems to supply power [1-3]. Based on the programming command, AGVs follow the predetermined path of the guiding system to work stations, complete the assigned tasks, and then return to the departure point independently [4-6]. When low battery warnings appear during the working process, they will self-navigate to the available charging station and continue working after charging. The functions of self-scheduling, programming, and cooperation make AGVs indispensable production equipment in flexible manufacturing system [7-10].

AGV navigation technology constitutes the most critical technology within AGVs, allowing for affordable, small-sized, and highly reliable navigation systems [11]. Major AGV navigation techniques include magnetic navigation, inertial navigation, visual navigation, laser navigation,

*bhh1988@126.com
https://doi.org/10.65102/is2026105
Published: 30 April, 2026

and ultrasonic navigation. AGVs that depend mainly on magnetic and laser navigation would be ideal for large production assembly workshops; however, the technology would have some challenges in terms of cost and flexibility. Ultrasonic navigation can be affected by environmental factors, whereas inertial navigation is relatively expensive and not accurate enough [12-14]. Recently, visual navigation has become more popular due to its various advantages compared to traditional navigation techniques, including information density, affordability, accuracy, and high levels of intelligence. Obstacle detection and navigation techniques have received significant attention [15-18].

The technology used in visual navigation is generally image processing technology, which includes visual line following navigation and visual obstacle avoidance navigation. In real-time operations, the AGV needs to accomplish the task of path recognition per each sampling interval of the image. Apart from precise path recognition, real-time capability, i.e., fast path image recognition, is equally crucial [19-21]. The working principle of the visual navigation AGV is based on path recognition using machine vision to extract navigation information from an image in the form of parameters that feed into tracking control systems [22, 23].

Weckx et al. (2020) introduced model predictive control and dynamic maximum speed approaches in AGVs, where maximum speed limits can be dynamically set for cornering and obstacles conditions, leading to emergency braking of vehicles [24]. Miao et al. (2021) devised a fuzzy control and an adaptive neural fuzzy inference system for the obstacle avoidance of AGVs with regard to moving obstacles. This helps in achieving effective dynamic obstacle avoidance under complex environmental conditions [25]. Wu et al. (2022) proposed a dynamic obstacle avoidance plan for forklift AGVs with the occurrence of sudden obstacles. This is achieved using an improved A* algorithm for global path planning and a dynamic window algorithm for local path planning near global paths. This does not only allow for dynamic obstacle avoidance but also avoids deviations from the globally planned optimal paths after avoiding obstacles [26]. Chaudhuri et al. (2022) proposed a sector-priority obstacle avoidance technique by combining 3D depth-based SLAM with trajectory mapping in-depth and optimized the AGV obstacle avoidance path in indoor (signal-blocked) and outdoor settings [27]. Yuan et al. (2022) have developed an obstacle avoidance velocity approach that uses Kalman filtering with forward simulation to minimize collision rates. The algorithm utilizes a target function representing efficiency and safety aspects to determine avoidance velocities [28]. Li et al. (2023) used the A* algorithm for global path planning and a timed elastic band algorithm for local dynamic obstacle avoidance path planning of AGVs using laser navigation technology. Based on point cloud and grid maps references, static and dynamic obstacle avoidance planning are carried out [29]. Zhang et al. (2024) improved the safety margin of obstacles avoidance in AGVs through a safety-distance based Lyapunov function. Errors of operation and speeds of AGVs in obstacle avoidance range were minimized by avoiding control using exponential and saturation functions. Disturbance observers and command filtering techniques helped in proposing a novel approach of AGV obstacle avoidance including dead zone consideration [30].

In order to improve the AGV autonomous navigation ability and obstacle avoidance capability, Huang et al. (2025) introduced a composite reward function that integrates enhanced deep reinforcement learning with distance-based obstacle penalties [31]. The authors of the paper by Zhang et al. (2025) have proposed a local path tracking algorithm for AGV operation among dense dynamic obstacles. Implemented using model predictive control and dynamic control barrier functions with the use of slack variables, this algorithm demonstrates significant efficiency and safety properties [32]. To provide an efficient navigation of AGVs in multi-AGV rail logistics transportation systems, Wu and Li (2025) used the A* algorithm to compute global routes for every AGV. Using a multisensor feature fusion environmental perception algorithm,

both dynamic and static obstacles were detected in the logistic environment. With the help of a combination of environmental perception capabilities with deep Q-networks, they managed to develop a dynamic obstacle avoidance algorithm for a multi-AGV system [33]. In their research work, Chen et al. (2025) have developed a time-consuming vertical segmentation optimization scheduling algorithm as well as a dynamic obstacle avoidance algorithm on the basis of the improved BUG2 algorithm and A* algorithm [34]. Hu et al. (2025) have developed a bio-inspired obstacle avoidance technique for AGVs by improving artificial potential field and dynamic window approaches to expand the window size and add additional potential field points [35].

Wu et al. (2019) applied four methods including Kernel Principal Component Analysis, Backpropagation Neural Networks, Improved Particle Swarm Optimization, and Binary Tree-Guided Window Partitioning to achieve better visual guidance and intelligent path recognition for AGVs in complicated interference situations, which consist of random illumination, obstruction of line-of-sight view, and object destruction [36]. In another study, Sun et al. (2022) combined a neural autoregressive model with the A* algorithm and carried out moving target velocity prediction and visual AGV path planning, respectively, through using past and current information to obtain dynamic AGV path planning for non-point moving targets [37]. Wang et al. (2023) changed the images in complicated light condition to an invariant image by using logarithm chromaticity space and minimum mean angular entropy [38]. Zhu et al. (2023) employed visual sensors to capture global visual information for AGVs. Under obstacle detection algorithms and AGV position/attitude recognition, path planning algorithms created virtual environments. Reinforcement learning then transmitted virtual AGV visual information to real AGVs for path tracking [39]. Jiang et al. (2023) replaced the 2D simultaneous localization and mapping (SLAM) system with a lidar-based and vision-based RGB-D camera system. The resulting AGV demonstrated good adaptability in environments containing three types of obstacles: sparse, thin structures, and suspended objects [40]. Nie et al. (2024) developed a vision-aware obstacle detection system using radar and RGB-D cameras, incorporating action value estimation to enhance obstacle detection. They controlled the AGV's obstacle avoidance angle through a dual-layer decision-making process combining deep deterministic policy gradient and deep Q-networks [41].

This paper first briefly outlines the system architecture of AGVs and the operational functions of their primary subsystems. Based on the integrated navigation system, it proposes a method for identifying and detecting environmental obstacles using parallax information. The next part covers the process of applying Kalman filter algorithm to define the stages of creating the dynamic obstacles movement mathematical model; thus, developing an approach for the prediction of the movement of these obstacles. Taking into consideration both global guidance and local planning approach and considering the fundamental laws of operation of the Q-learning algorithm, the real-time adjustment system of path planning according to the results of the detection system is created. Further on, the performance of obstacle recognition and detection system is analyzed via experimental research. Lastly, simulation experiment is carried out in order to estimate the performance of real-time path adjustment and comparison of prediction errors of similar approaches.

2 Obstacle Recognition and Detection for AGVs Based on Visual Navigation Systems

2.1 System Composition of AGVs

AGVs are mechatronic products with relatively complex system structures. Their system composition is shown in Figure 1.

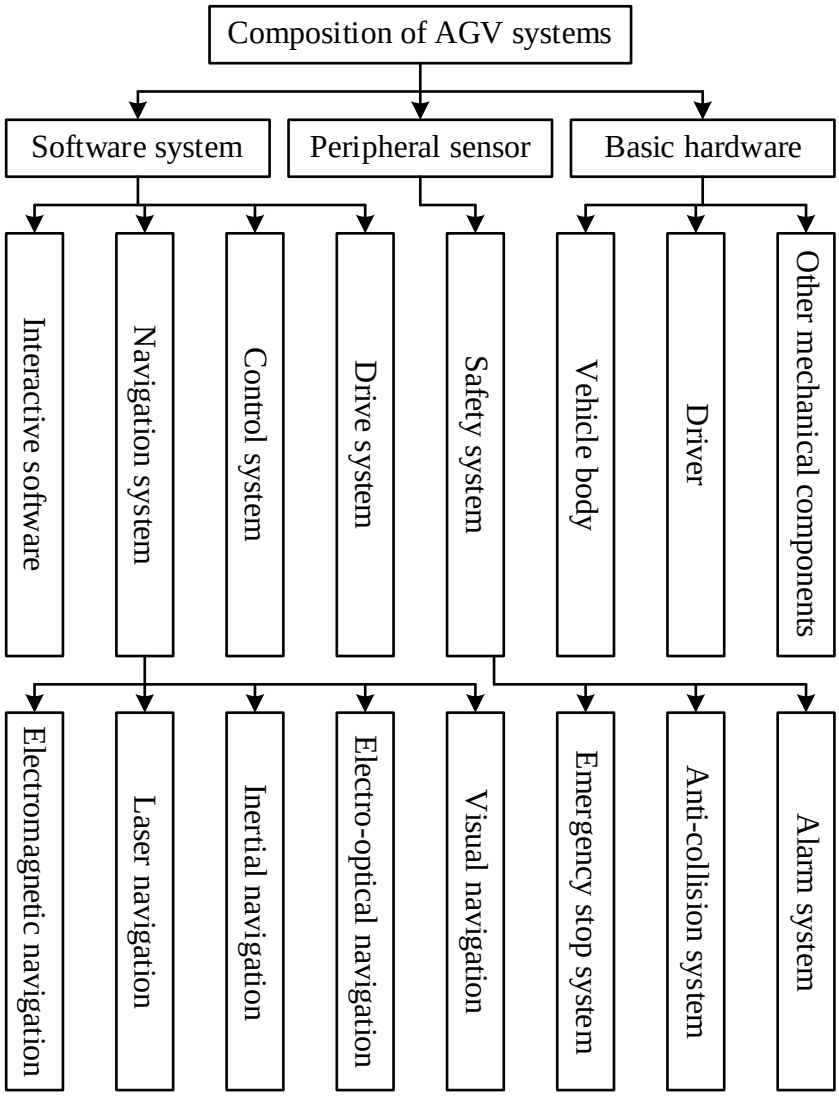


Figure 1: System Composition of AGV

(1) Navigation System

There are different kinds of navigation systems depending on the type of navigation used. These include electromagnetic navigation, laser navigation, inertial navigation, photoelectric navigation, and visual navigation among others. The main constituents of visual navigation system include a camera and image acquisition board.

(2) Control System

Control system forms the main part of AGV system and comprises the host computer and controller. The host computer gathers information about other systems, while the controller transmits the information from the host computer to various systems to enable their operations.

(3) Drive System

Drive system works as described below: With the starting of the DC motor, the transmission unit and reducer start working transmitting kinetic energy to the wheels that help the vehicle to move. In addition, the rotational speed of the DC motor is monitored using the encoder of drive system.

(4) Safety System

This system helps to avoid accidents during the operation of AGVs. It usually entails fitting sensors and protection equipment on the body surface of the AGVs. The former detects any dangerous situations in the surroundings whereas the latter forms a second line of defense.

2.2 Obstacle Recognition and Detection Based on Parallax Information

Real-world scenes mostly consist of horizontal and vertical planes, where horizontal planes usually represent the floor, while vertical planes can be seen as representations of objects. Using the features of these real-world scenes, columnar pixels were developed. In describing real-world scenes, columnar pixels use rectangular boxes to represent the environment. These pixel structures have properties like easy accessibility, high compactness, and durability, and they only require two parameters for description: depth and height. The columnar pixel approach minimizes the volume of data required, compressing tens of thousands of disparity points down to a few hundred points. Consequently, this paper employs a columnar pixel model based on probabilistic methods and global optimization to extract columnar pixels and represent obstacle detection results.

Considering that obstacles of varying depths may exist within the same column, after extracting columnar pixels via the columnar pixel model, multi-layer columnar pixels are employed to represent obstacles in the image. Typically, different colors represent obstacle distances, transitioning from warm to cool tones to indicate increasing distance. Simultaneously, based on depth value differences, obstacles within the same column are described using multi-layer columnar pixels, segmenting potential obstacles into multi-layer columnar pixel segments.

The multi-layer columnar pixel method for obstacle representation utilizes the disparity map as its foundation. Based on the depth attributes of each pixel point, different object segments within the image (as demonstrated in the following formula using ground, object, and sky as three distinct segments) are described on the left image of the image pair using distinct label sets. The relationship between these label sets is defined by Equation (1).

$$\begin{cases} L = \{L_u\}, 0 \le u \le w \\ L_u = \{s_n\}, 1 \le n \le N_u \le h \\ s_n = \{v_n^b, v_n^t, c_n, f_n(v)\}, 0 \le v_n^b \le v_n^t \le h \end{cases} \tag{1}$$

In formula (1), s_n represents a segmentation segment, where objects with different depth values are represented by s_n with distinct attribute labels; N_u denotes the number of segmentation segments in each column; v_n^b denotes the number of bottom rows for each segment s_n of a category within a column; v_n^i denotes the number of top rows for each segment s_n of a category within a column; $c_n = \{g, o, s\}$ represents ground, object, and sky, respectively; $f_n(v)$ is the function calculating the disparity of the vth row between the top row and bottom row of the segment.

Additionally, to efficiently solve the segmentation task, $f_n(v)$ is also categorized into ground, object, and sky. This yields formula (2):

$$\begin{cases} f_n^g(v) = \alpha(v_{hor} - v) \\ f_n^o(v) = d_n \\ f_n^s(v) = d_n \end{cases} \tag{2}$$

The first term in formula (2) represents the ground segmentation, where α denotes the ground parallax gradient and v_{hor} is the horizontal line's row coordinate. The second term represents the average parallax value of s_n within the object segmentation d_n. The third term represents the sky segment. Since the sky is at an extremely distant location, $f_n^s(v)$ approaches 0.

The extraction of columnar pixels must adhere to fundamental assumptions such as the Bayesian principle, gravity principle, and sequential constraints. Consequently, columnar pixel extraction transforms into an optimal segmentation problem for the input data. Thus, the label L^* with the maximum search probability can be obtained as shown in Equation (3):

$$L^* = \arg\max_L P(L \mid D) \tag{3}$$

Applying Bayesian theory's posterior probability yields formula (4):

$$P(L \mid D) \sim \prod_{u=0}^{w-1} P(D_u \mid L_u) \cdot P(L_u) \tag{4}$$

In equation (4), $P(D_u \mid L_u)$ represents the conditional probability density of input data D_u given label L_u, constituting the conditional probability model; $P(L_u)$ denotes the overall probability of each individual column label L_u, representing the prior knowledge model; D_u denotes the vertical parallax vector; L_u denotes the column label. Further decomposing the two terms $P(D_u \mid L_u)$ and $P(L_u)$ yields Equations (5)-(6):

$$P(D_u \mid L_u) = \prod_{n=1}^{N_a} \prod_{v=v_n^b}^{v_n'} P_D(d_v \mid s_n, v) \tag{5}$$

$$P(L_a) = P(s_1) \cdot \prod_{n=2}^{N_a} P(s_n \mid s_{n-1}) \tag{6}$$

For the conditional probability model, the actual likelihood function is provided for scoring each disparity value across all three hypotheses. It consists of a mixture of two distributions: a normal distribution that penalizes deviations from the class-dependent model expectations, and a uniform distribution that ensures robustness to outliers. This is expressed in Equations (7)-(8):

$$P_D(d_v \mid s_n, v) = \frac{P_{out}}{d_{\max} - d_{\min}} + \frac{1 - p_{out}}{A_{rouge}} \cdot \frac{1}{\sigma^{C_n}(f_n, v) \cdot \sqrt{2\pi}} \cdot \exp\left[-\frac{1}{2} \left(\frac{d_v - f_n(v)}{\sigma^{C_n}(f_n, v)} \right)^2 \right] \tag{7}$$

$$A_{nnge} = \frac{1}{2} \cdot \left(erf\left(\frac{d_{\max} - f_n(v)}{\sqrt{2} \cdot \sigma^{C_n}(f_n, v)} \right) - erf\left(\frac{d_{\min} - f_n(v)}{\sqrt{2} \cdot \sigma^{C_n}(f_n, v)} \right) \right) \tag{8}$$

In equation (7), $P_D(d_v \mid s_n, v)$ denotes the probability of the parallax measurement d_v occurring at column v within the fixed segment s_n; P_{out} denotes the outlier rate, representing the probability of detecting anomalous values in the computational model; $\sigma^{C_n}(f_n, v)$ denotes Gaussian parameters combined with a noise model for disparity measurement; A_{range} denotes the normalization term for integral computation.

Regarding the prior knowledge model $P(L_u)$: considerable content exists. For instance, a ground segment is more likely to occur in front of an obstacle than behind it. To quantify the former, the vertical transition probability $P(s_n \mid s_{n-1})$ from each segment to the preceding segment is evaluated, independent of any input data.

Finally, applying dynamic programming concepts, the optimal segmentation of ground, object, and sky segments is computed. Dynamic programming is frequently employed as a solution for large-scale optimization problems; it constructs global optimal solutions for discrete problems by non-iteratively obtaining several optimal subproblems within polynomial time. Thus, dynamic programming enables real-time computation of the energy function L^*. Employing the negative log-likelihood approach for probabilistic values, the recursive objective function to be minimized is presented here. C^v denotes a column in row v of the table containing the minimum aggregated cost for ground, object, and sky segments, as defined in Equation (9):

$$C^v = \{G^v, O^v, S^v\} \tag{9}$$

The entry with the minimum cost in C^v then defines the segment class ending at position v. For example, equation (10) provides the recursive function for the minimum ground cost up to position v.

$$G^n = \min \begin{cases} g_0^n + c(g_0^n) \\ g_1^n + c(g_1^n, G^0) + G^0 \\ g_1^n + c(g_1^n, O^0) + O^0 \\ g_1^n + c(g_1^n, S^0) + S^0 \\ \vdots \\ g_n^n + c(g_n^n, G^{n-1}) + G^{n-1} \\ g_n^n + c(g_n^n, O^{n-1}) + O^{n-1} \\ g_n^n + c(g_n^n, S^{n-1}) + S^{n-1} \end{cases} \tag{10}$$

3 Prediction and Real-Time Adjustment of Path Obstacles

In the previous section, this paper proposed an obstacle recognition and detection mechanism based on the AGV system architecture using parallax information. Building upon this foundation, this section introduces the Kalman filter algorithm for predicting the motion trajectories of dynamic obstacles. It further proposes a path planning method based on the Q-

learning algorithm, establishing an AGV obstacle recognition and real-time path adjustment mechanism following the workflow: “image recognition → detection determination → trajectory prediction → path adjustment.”

3.1 Establishing a Motion Model Based on Kalman Filtering

To predict the position of an obstacle in the next frame, we first need to define the state of the dynamic obstacle. By obtaining the obstacle's position, velocity, and other information through 3D Fidelity and substituting these values, we derive the state vector: $X=[x,y,v^x,v^y]$, where x,y denote the obstacle's current position on the map, and v^x,v^y represent the velocity components in the x,y directions, respectively. Therefore, the state transition matrix is given by Equation (11):

$$F=\begin{bmatrix}1&0&T&0\\0&1&0&T\\0&0&1&0\\0&0&0&1\end{bmatrix} \tag{11}$$

The predicted state equation is given by Equation (12):

$$\begin{bmatrix}x_t\\y_t\\v_t^x\\v_t^y\end{bmatrix}=\begin{bmatrix}1&0&T&0\\0&1&0&T\\0&0&1&0\\0&0&0&1\end{bmatrix}\times\begin{bmatrix}x_{t-1}\\y_{t-1}\\v_{t-1}^x\\v_{t-1}^y\end{bmatrix}+\omega_t \tag{12}$$

The measurement transformation matrix is given by Equations (13)-(14):

$$H=\begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix} \tag{13}$$

$$Z_t=\begin{bmatrix}Z_t^x\\Z_t^y\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix}\times\begin{bmatrix}x_{t-1}\\y_{t-1}\\v_{t-1}^x\\v_{t-1}^y\end{bmatrix}+v_t \tag{14}$$

Within this framework, Z_t^x,Z_t^y represent the observed values of the measurements; v_t denotes the measurement noise, which follows $v_t\sim N(0,R)$. This paper sets $R=0.01E_2$, where E_2 is the second-order unit matrix; $\omega_t t$ represents state noise, satisfying the Gaussian distribution $\omega_t\sim N(0,Q)$. In this paper, $Q=0.001E_4$ is adopted. A larger Q value yields prediction results closer to the actual values, while a larger R value yields prediction results closer to the observed data. E_4 is the fourth-order unit matrix. T is the system operating cycle, set to $T=100$ ms.

The predicted state variables and predicted minimum mean square error matrix equations are given by Equations (15)-(16):

$$X_t^- = FP_{t-1}^+ F^T \tag{15}$$

$$P_t^- = FP_{t-1}^+ F^T + Q \tag{16}$$

The computational error gain, optimal state estimate equation, and mean square error update matrix are given by Equations (17)–(19), respectively:

$$K_t = \frac{P_t^- H^T}{R + H_k P_t^- H_k^T} \tag{17}$$

$$X_t^+ = X_t^- + K_t(Z_t - HX_t^-) \tag{18}$$

$$P_t^+ = (I - K_t H)P_t^- \tag{19}$$

In the equation, X_t^- and X_t^+ denote the predicted state vector at time (t-1) for time t and the optimal estimate at time t, respectively, where t is the current time and (t-1) is the previous time; P_t^- and P_t^+ denote the predicted mean square error (MSE) matrix at time (t-1) for time t and the corrected MSE matrix at time t, respectively; P_{t-1}^+ is the corrected MSE matrix at time (t-1); K_t is the error gain matrix. The motion prediction is obtained through the mathematical model of uniformly accelerated motion as expressed in Equations (20)-(21):

$$x = x_0 + v_0 t + \frac{at^2}{2} \tag{20}$$

$$y = y_0 + vy_0 t + \frac{ayt^2}{2} \tag{21}$$

Here, x and y represent horizontal and vertical displacements, respectively, while v_0 and vy_0 denote the corresponding accelerations, with t being time.

For obstacles in dynamic states, their movement speed and pattern should be assessed to determine whether they will impact the AGV's trajectory. When their path overlaps with the AGV's trajectory, they should be updated in the map for real-time path planning.

3.2 Local Path Planning Based on the Q-Learning Algorithm

3.2.1 Global Path Guidance for Local Path Planning

Mark obstacles on the map, dividing path planning into global planning for the entire route and local planning for obstacle avoidance routes, with the global path guiding the local obstacle avoidance route planning. When the AGV implements local path planning guided by the global path, the overall route should follow the predefined global path as closely as possible. Upon detecting an obstacle, the AGV must select an appropriate local obstacle avoidance point from the global path and use this to guide subsequent local path planning.

Local obstacle avoidance path planning can be viewed as a sequential decision problem at fixed time intervals. The AGV uses various sensors to acquire surrounding environmental information. When an obstacle enters the planned global route area, the AGV initiates the

obstacle avoidance procedure and ultimately returns to the global path planning route after completing the local path planning. During obstacle avoidance, the AGV continuously receives real-time data from each sensor. It generates corresponding commands by assessing its own position and orientation, movement speed, and the coordinates of the target point. Obstacle avoidance is successful if the AGV reaches the target point of the local path planning without collision; otherwise, it fails. The goal of path planning is to select the optimal route while ensuring no collisions occur.

3.2.2 Basic Principles of the Q-learning Algorithm

Q-learning is a model-free reinforcement learning algorithm that enables agents to learn action values for specific states within an environment. Exploration in Q-learning is balanced by a trade-off managed by the $\varepsilon-$greedy policy, where the probability of selecting random actions decreases as the AGV gains confidence in its learned values. This strategy ensures the AGV avoids getting stuck in local optima and continues exploring the environment to discover potentially superior paths.

When exploring the environment on a constructed map, the Q-learning algorithm receives rewards or penalties based on the consequences of movement. For example, getting closer to the target state is rewarded by adding positive reinforcement, whereas hitting any obstacles results in a punishment for the same action. The Q values are continuously updated through this approach to create a policy that directs the AGV towards an optimal path to the target state.

In conclusion, the Q-learning algorithm is a viable option for path planning of autonomous guided vehicles operating in complicated environments. Q-learning algorithms do not require predetermined models of the environment like the conventional approaches to path planning. The algorithms are adaptable to changes in the environment and help AGVs find optimized paths within unfamiliar environments. Moreover, Q-learning algorithms can be applied to multi-AGV systems.

The Q-learning method is dependent on the formation of a Q-table that helps in exploring paths to reach the optimal path. In the first place, an action space needs to be identified, which can be done through a four-neighborhood configuration or eight-neighborhood configuration. Examples of action sets have been provided in Figure 2. It is the choice of the particular problem under consideration that determines which action space is selected. Four-neighborhood action spaces consist of four actions, including moving up, moving down, moving left, and moving right. For each action performed, the mobile robot gets either a reward or penalty according to the reward function. Eight-neighborhood action spaces comprise eight states or actions, four more than those of four-neighborhood action spaces. This increases computational complexity but results in smoother path generation. However, using the octant neighborhood expands the state space, potentially slowing convergence speed. Since the AGV's movement mechanism is not well-suited for turning at small angles, all Q-learning algorithms in this paper utilize the four-neighborhood state set.

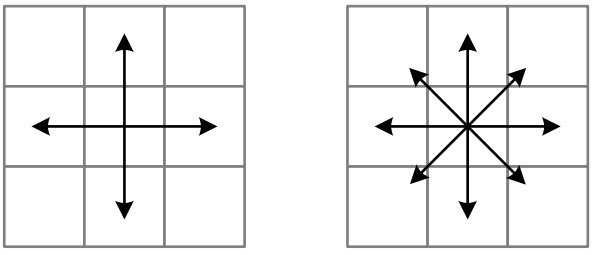


Figure 2: Settings of Actions

Create a Q-table to store the Q-values corresponding to each state and action. Initially, the values in the Q-table are typically set to zero or random values. The Q-value for the current state is updated based on the observed reward and the Q-value of the next state. The specific update is given by Equation (22):

$$Q(s,a) = Q(s,a) + \alpha * (R + \gamma * \max Q(s+1, a+1) - Q(s,a)) \tag{22}$$

Here, $Q(s,a)$ denotes the Q-value of taking action a in state s. α represents the learning rate, R is the observed reward, γ is the discount factor used to balance the importance of current rewards versus future rewards, $s+1$ is the next state, and $a+1$ is the optimal action in the next state. By continuously interacting with the environment and updating the Q-table, the agent gradually learns the optimal action strategy.

In AGV path planning, rewards are typically based on distance to the destination or avoiding obstacles. The reward function influences the final convergence speed and exploration cycle. In the Q-learning algorithm, rewards are commonly set as in Equation (23):

$$R_0(S_t, a_t) = \begin{cases} 1, & S_t = S_{goal}; \\ 0, & S_t = S_{freedom}; \\ -1, & S_t = S_{obstacle}; \end{cases} \tag{23}$$

In this case, $R_0(S_t, a_t)$ indicates that when a_t moves to position S_t, S_{goal} is the target position, $S_{freedom}$ is a movable position, and $S_{obstacle}$ is an obstacle position.

By repeatedly executing actions and updating Q-values until reaching the destination, exceeding the maximum number of steps, or achieving Q-value convergence, the values in the Q-table will stabilize after sufficient iterations. At this point, the optimal policy can be extracted from the Q-table: for each state, select the action with the highest Q-value.

4 Effectiveness of Obstacle Detection and Real-Time Path Adjustment Mechanism

4.1 Evaluation of Obstacle Recognition and Detection Systems

4.1.1 Obstacle Recognition

Considering the practical application scenarios of AGVs and the depth-of-field range commonly used by navigation system cameras, objects positioned 10mm above the ground and within 1000mm of the AGV cart are defined as obstacles. Detection threshold values are specified as -190 for height distance and 1000 for horizontal distance. This section will discuss the performance of the obstacle detection system design in recognizing both static and dynamic obstacles in a sequential manner.

(1) Static Obstacles

A real height of 165mm, placed at a distance of 700mm from the optical centre of the camera of the AGV Obstacle Detection and Recognition system is known as the Static obstacle. By extracting its columnar pixels, setting the number of white pixels to 10, and establishing an anomaly threshold of 50, the three-dimensional coordinates, horizontal distance, and height distance detection data obtained from the cardboard box obstacle's parallax information are presented in Table 1.

Table 1: Carton obstacle detection data

Unit (mm)	Three dimensional coordinates (x, y, z)	Horizontal distance z	Height distance x
1	(-26.6, -1.9, 724.8)	724.8	-26.6
2	(-62.6, 14.1, 721.2)	721.2	-62.6
3	(-20.3, 7.9, 720.6)	720.6	-20.3
4	(-25.8, 47.6, 722.4)	722.4	-25.8
5	(-69.4, 40, 728.4)	728.4	-69.4
6	(-64.2, 132. 6,729)	729	-64.2
7	(-52.3, 133.5, 724.1)	724.1	-52.3
8	(369, 354.4, -54877.7)	-54877.7	369
9	(-61.4, 41.7, 727.2)	727.2	-61.4
10	(-32.9, -7, 725.5)	725.5	-32.9

Based on the data in Table 1, the horizontal distance of the cardboard box object, after removing outliers, yielded an average value of 724.8 < 1000. The height distance, after removing outliers, had a maximum value of -20.3 > -190. Thus, the designed obstacle recognition and detection system identified the experimental cardboard box object as an obstacle. Additionally, the depth distance calculation error for the carton object in the designed obstacle recognition and detection system is 724.8 - 700 = 24.8 mm, with an error rate of only 3.54%, which remains within the practical application range.

(2) Dynamic Obstacles

Pedestrian leg images were selected as dynamic obstacles, with an actual height of 525 mm and a distance of 850 mm from the optical center of the AGV obstacle recognition and detection system's camera. Columnar pixels were extracted, with 10 white pixels and an anomaly threshold of 50 set. Based on the parallax information of the pedestrian legs, the detection data for their three-dimensional coordinates, horizontal distance, and height distance are summarized in Table 2.

Table 2: Pedestrian obstacle detection data

Unit (mm)	Three dimensional coordinates (x, y, z)	Horizontal distance z	Height distance x
1	(-14.2, 6.5, 876.6)	876.6	-14.2
2	(21.6, 16.1, 871.4)	871.4	21.6
3	(40.9, 55.9, 862.7)	862.7	40.9
4	(78.8, 86.7, 869)	869	78.8
5	(70, 79.9, 881.6)	881.6	70
6	(94.4, 132.6, 880.1)	880.1	94.4
7	(54878.3, 354.4, -54877.7)	-54877.7	54878.3
8	(-54877.7, 354.4, -54877.7)	-54877.7	-54877.7
9	(71.9, 114.8, 863.9)	863.9	71.9
10	(83.8, 127.9, 876.2)	876.2	83.8

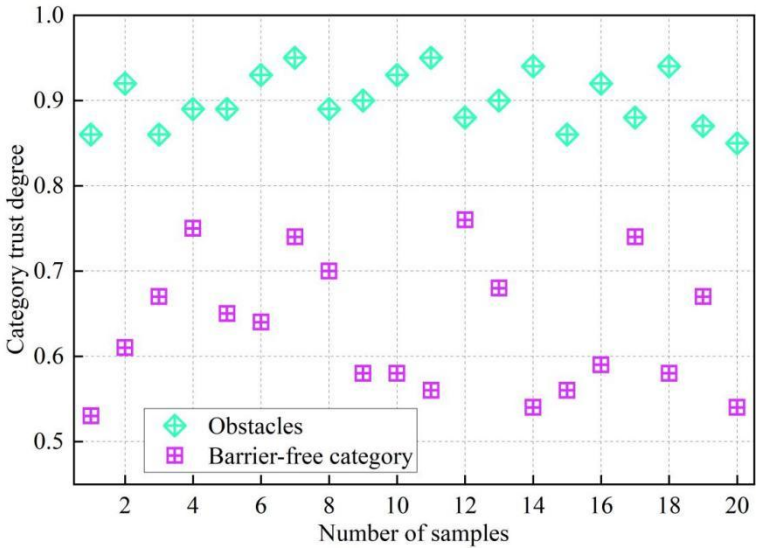
Based on the data in Table 2, after removing outliers, the mean value of the pedestrian leg image information is 872.69 < 1000. The maximum height distance after outlier removal is 94.4 > -190. The designed obstacle recognition and detection system identifies the experimental pedestrian's legs as obstacles. The error in the depth distance measurement for the pedestrian’s leg is 872.69 – 850 = 22.69mm, with a percentage error of 2.60% that falls within the acceptable

range.

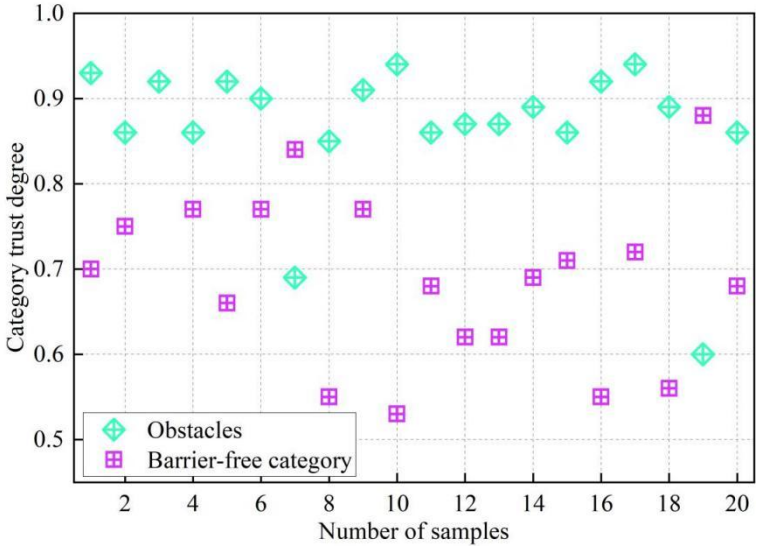
The obstacle detection system using parallax information shows excellent precision in recognizing both stationary and moving obstacles.

4.1.2 Detection of Obstacle Object Images

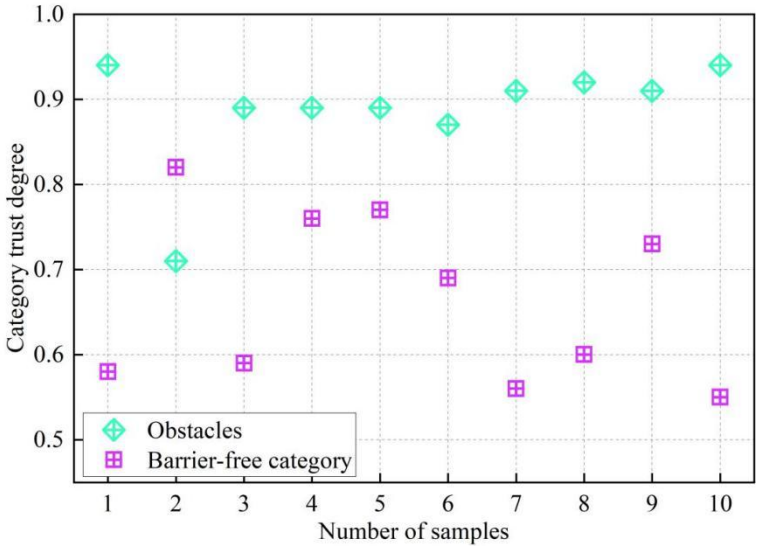
A total of 50 potential obstacle pictures were randomly picked to conduct obstacle detection confidence experiment as experimental subjects, which included 20 still obstacle pictures, 20 moving obstacle pictures, and 10 obstacle free pictures. The experiment was carried out based on the multi-layer pixel histograms generated from the picture samples above. The confidence detection results for static obstacle images are shown in Figure 3(a), those for dynamic obstacle images in Figure 3(b), and those for obstacle-free images in Figure 3(c).



(a) Static obstacle object image detection



(b) Dynamic obstacle object image detection



(c) Barrier-free object image detection

Figure 3: Obstacle detection result

Following the principle of maximum confidence (where the category with the highest confidence is assumed as the obstacle detection result), the overall obstacle recognition and detection system demonstrated high accuracy in obstacle detection. For the detection of 20 static obstacle images, the confidence levels assigned to obstacle categories (>0.80) were consistently higher than those for non-obstacle categories (0.53–0.76), indicating consistent agreement with actual labels. For the detection of 20 dynamic obstacle images, one sample differed from its actual category label. Among the 10 obstacle-free image detections, two samples exhibited discrepancies from their actual category labels.

4.2 Feasibility of Real-Time Path Adjustment Systems

This section simulates a warehouse operation scenario involving AGV carts (covering an area of 600m²) to conduct simulation experiments on the designed real-time path adjustment system. Various types of obstacles are set up, with the specific simulated obstacle scenarios shown in Figure 4. The yellow areas represent moving warehouse personnel (dynamic obstacles), while the green areas indicate randomly stacked goods (static obstacles).

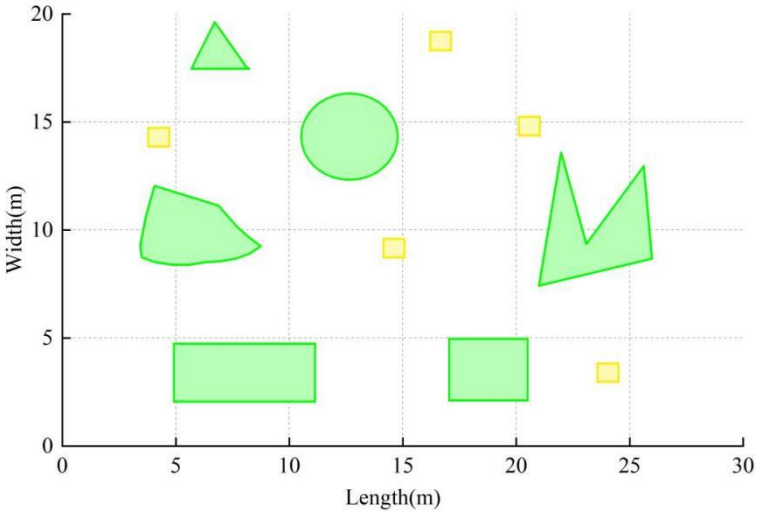


Figure 4: Obstacle simulation scene

In this simulated scenario, the dynamic obstacle-avoidance path planning results for the AGV equipped with the real-time path adjustment system described in this paper are shown in Figure 5. The diamond-shaped point indicates the starting point of the AGV's movement, the circular point denotes the endpoint, and the red curve represents the planned dynamic obstacle-avoidance path.

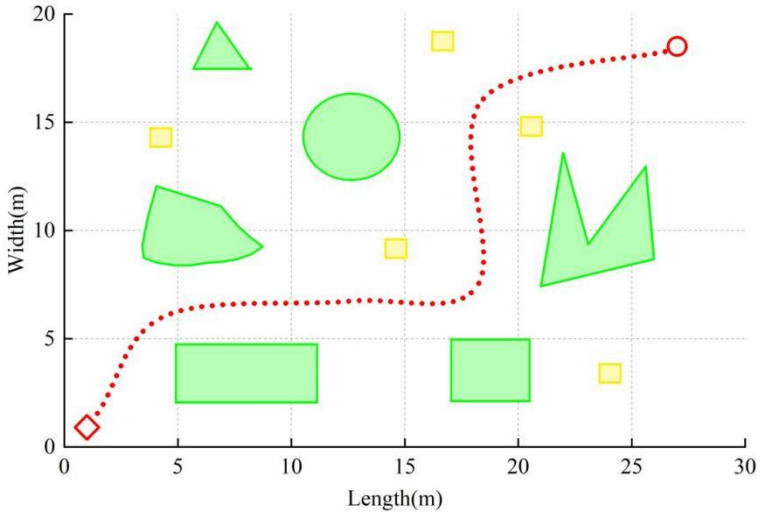


Figure 5: Dynamic obstacle avoidance paths for AGV carts

It can be observed that under the obstacle avoidance path planning method described in this paper, the AGV cart can flexibly adjust its travel path when encountering various pre-set obstacles. This strategy not only manages to avoid obstacles on its way to the destination but is also the most efficient path that it can take. This method allows for safe navigation and the reduction of travel time to its minimum possible extent.

4.3 Verification of Path Adjustment Mechanisms for Vision-Based Navigation Systems

This part examines the efficiency of the path correction technique in real time within the visual navigation system, as shown by the following two aspects: (1) Evaluation of the advantages of the suggested local path planning technique using the Q-learning technique; (2) Assessment of the performance of the mathematical model for obstacle objects' movement.

4.3.1 Comparison of Path Planning Algorithms

In order to verify the validity, reliability, and practicability of the path planning algorithm mentioned above, the work environment of the warehouse is chosen again as the test site. In contrast to the obstacle simulation test in the last part, the new environment includes more obstacles of different kinds. A simple obstacle grid map is generated to establish an environmental model. The traditional algorithms, which include the A* algorithm, Dijkstra algorithm, and BFS algorithm, are often used as control algorithms for obstacle avoidance path planning of AGVs. The path planning results of these four algorithms are presented in Figure 6.

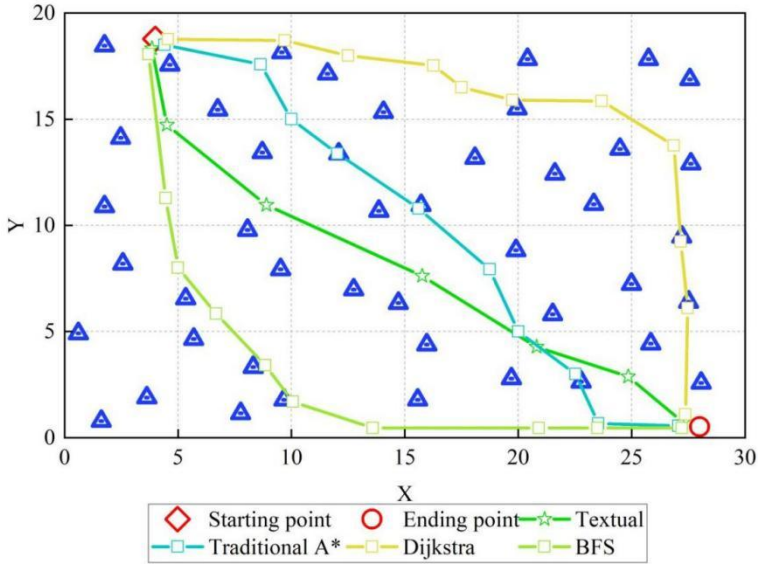


Figure 6: Path planning based on different algorithms

While the conventional A* path planning algorithm creates paths of medium length with less number of nodes than other methods, its regulation of safety distance around obstacles is not effective enough, causing about 3 collisions with obstacles. The path created by the Dijkstra algorithm is quite long and it suffers from the same problem as the traditional A* algorithm concerning safety distance management from obstacles. In addition, it involves the highest number of nodes in the four algorithms considered, and this makes it inefficient due to the lengthy process. On the other hand, the BFS algorithm gives convergent node paths with smooth trajectory; however, it is expected to create two collisions with obstacles and gives the longest path length. Path planning by the proposed algorithm avoids collisions with obstacles and gives the shortest path length with fewest path nodes.

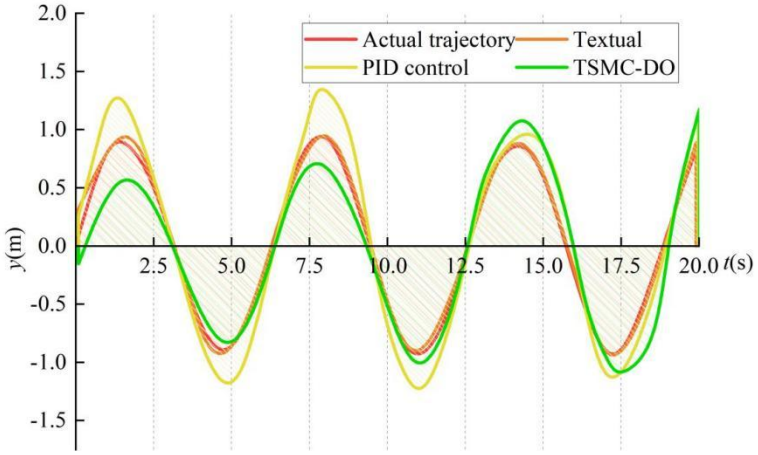
In order to provide a better understanding of the performance capability of the path planning algorithms through a numerical comparison, the path length, path nodes, and average path risk value (in terms of obstacle collisions) are compared on a simplified grid map in Table 3 below. According to the average path risk value, the rank from the best to the worst is given as: Proposed Algorithm (0.15) > Traditional A* Algorithm (0.24) > BFS Algorithm (0.26) > Dijkstra Algorithm (0.37).

Table 3: Comparison of path planning performance of different algorithms

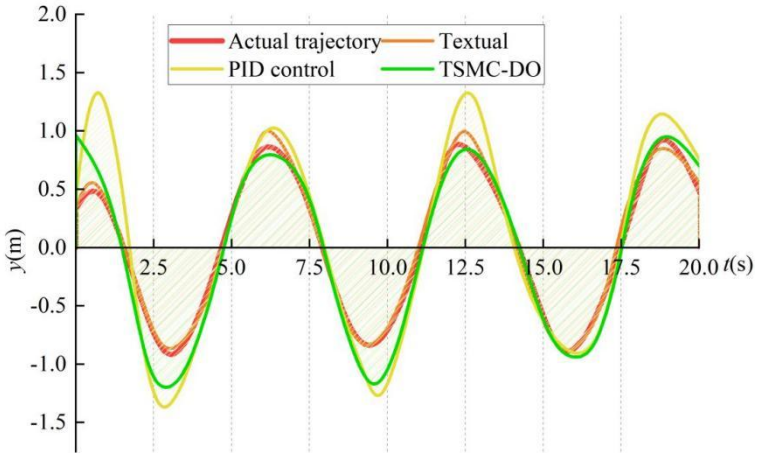
Algorithm	Path length (m)	Path node	Average risk value of the path
Traditional A*	21	9	0.24
Dijkstra	28	11	0.37
BFS	25	9	0.26
Textual	23	6	0.15

4.3.2 Evaluation of Obstacle Motion Models

In this section, an analysis of the performance of the motion model is conducted regarding tracking and prediction of trajectory for the moving obstacles. Two methods namely TSMC-DO approach and PID control tracking approach are used as benchmarks. X Tracking Error Curves have been depicted in Figure 7(a) while Y Tracking Error Curves have been depicted in Figure 7(b).



(a) x-axis tracking error curve



(b) y-axis tracking error curve

Figure 7: Tracking error curve

Upon observing Figures 7(a)-(b) as a whole, it can be noticed that the PID control tracking approach has a large error in comparison to the actual trajectory of the moving obstacle. Although the TSMC-DO approach has smaller error in comparison to the actual trajectory, there is an element of error in the prediction. As for the prediction of motion trajectory by using the motion model of this paper, it is highly similar to the actual trajectory to the extent that they are barely distinguishable because of the presence of small errors only. It is due to the use of the Kalman filter in this approach that helps in updating the data at each stage of uniform motion.

5 Conclusion

The research in this paper classifies the AGV obstacle avoidance into three main categories including object detection and recognition, trajectory estimation, and path planning in order to build up a dynamic real-time AGV object detection and path adjustment technology. It ensures an obstacle recognition error rate of 3.54% in dynamic condition and 2.60% in static condition. In comparative simulation experiment on path planning, the path designed showed minimum travel distance (23m), minimum number of nodes (6), and minimum average risk (0.15).

With the help of navigation vision technology, the AGV object detection and real-time path

adjustment technique helps identify and classify objects accurately. With the aid of motion models, it predicts the trajectories of objects. It then designs an optimal path to avoid obstacles.

Funding

This work was supported in part by the Key Natural Science Research Project of Anhui Province (Grant 2023AH052396, 2025AHGXZK30721); in part by the Educational Teaching Research Planning Project of Anhui Vocational and Adult Education Society (Grant AZCJ2024287); in part by the Key Natural Science Research Project of Wuhu Vocational and Technical University (Grant wzyzrzd202312); in part by the "Talent Project" Outstanding Young Top-Notch Talent Project of Wuhu Vocational and Technical University (Grant rc2023qnbj03).

About the Author

Houhuang Bi was born in Anqing, China, in 1988. He received the M.S. degree in mechanical engineering from Kunming University of Science and Technology, Kunming, China, in 2017. Since 2017, he has been with Wuhu Vocational and Technical University, Anhui, China. He has presided over and participated in eight teaching and research projects at or above the university level and published more than 10 academic papers. His research interests include digital design and manufacturing, as well as unmanned driving technology.

References

[1] Hyla, P., & Szpytko, J. (2017). Automated guided vehicles: the survey. Journal of KONES, 24(3), 101-110.

[2] Kabir, Q. S., & Suzuki, Y. (2018). Increasing manufacturing flexibility through battery management of automated guided vehicles. Computers & Industrial Engineering, 117, 225-236.

[3] Kim, D. H., Hai, N. T., & Joe, W. Y. (2017, November). A guide to selecting path planning algorithm for automated guided vehicle (AGV). In International Conference on Advanced Engineering Theory and Applications (pp. 587-596). Cham: Springer International Publishing.

[4] Vancea, A. P., & Orha, I. (2019). A survey in the design and control of automated guided vehicle systems. Carpathian Journal of Electrical Engineering, 12(2).

[5] Yang, X., Hu, H., & Jin, J. (2023). Battery-powered automated guided vehicles scheduling problem in automated container terminals for minimizing energy consumption. Ocean & Coastal Management, 246, 106873.

[6] Han, W., Xu, J., Sun, Z., Liu, B., Zhang, K., Zhang, Z., & Mei, X. (2022). Digital twin-based automated guided vehicle scheduling: A solution for its charging problems. Applied Sciences, 12(7), 3354.

[7] Liu, Y., Hou, Z., Tan, Y., Liu, H., & Song, C. (2020). Research on multi-AGVs path

planning and coordination mechanism. IEEe Access, 8, 213345-213356.

[8] Xia, W., Goh, J., Cortes, C. A., Lu, Y., & Xu, X. (2020, August). Decentralized coordination of autonomous AGVs for flexible factory automation in the context of Industry 4.0. In 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE) (pp. 488-493). IEEE.

[9] Kumbhar, S. G., Thombare, R. B., & Salunkhe, A. B. (2018). Automated guided vehicles for small manufacturing enterprises: A review. SAE International journal of materials and manufacturing, 11(3), 253-258.

[10] Chawla, V. K., Chanda, A. K., & Angra, S. (2019). Automatic guided vehicle systems in flexible manufacturing system–A review. International Journal of Industrial Engineering: Theory, Applications and Practice, 26(5).

[11] Aizat, M., Azmin, A., & Rahiman, W. (2023). A survey on navigation approaches for automated guided vehicle robots in dynamic surrounding. IEEE Access, 11, 33934-33955.

[12] Moshayedi, A. J., Jinsong, L., & Liao, L. (2019). AGV (automated guided vehicle) robot: Mission and obstacles in design and performance. Journal of Simulation and Analysis of Novel Technologies in Mechanical Engineering, 12(4), 5-18.

[13] Tang, Y., & Wu, Q. (2023, October). Design and realization of magnetic and laser hybrid navigation in AGV system. In Fourth International Conference on Artificial Intelligence and Electromechanical Automation (AIEA 2023) (Vol. 12709, pp. 1126-1136). SPIE.

[14] Vale, A., Ventura, R., Lopes, P., & Ribeiro, I. (2017). Assessment of navigation technologies for automated guided vehicle in nuclear fusion facilities. Robotics and Autonomous Systems, 97, 153-170.

[15] Zhang, Y., Li, B., Sun, S., Liu, Y., Liang, W., Xia, X., & Pang, Z. (2023). GCMVF-AGV: Globally consistent multiview visual–inertial fusion for AGV navigation in digital workshops. IEEE Transactions on Instrumentation and Measurement, 72, 1-16.

[16] Benzidia, S., Ageron, B., Bentahar, O., & Husson, J. (2019). Investigating automation and AGV in healthcare logistics: a case study based approach. International Journal of Logistics Research and Applications, 22(3), 273-293.

[17] Zhang, Z., Chen, J., & Guo, Q. (2023). Application of Automated Guided Vehicles in Smart Automated Warehouse Systems: A Survey. CMES-Computer Modeling in Engineering & Sciences, 134(3).

[18] Jang, J. Y., Yoon, S. J., & Lin, C. H. (2023). Automated guided vehicle (AGV) driving system using vision sensor and color code. Electronics, 12(6), 1415.

[19] Yingbo, Z., Shichao, X., Yuan, H., & Xinyu, B. (2025). Study on automated guided vehicle navigation method with external computer vision. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 239(5), 635-647.

[20] Munadi, M., Radityo, B., Ariyanto, M., & Taniai, Y. (2024). Automated guided vehicle (AGV) lane-keeping assist based on computer vision, and fuzzy logic control under

varying light intensity. Results in Engineering, 21, 101678.

[21] Chun-Fu, W., Xiao-Long, W., Qing-Xie, C., Xiao-Wei, C., & Guo-Dong, L. (2017, October). Research on visual navigation algorithm of AGV used in the small agile warehouse. In 2017 Chinese Automation Congress (CAC) (pp. 217-222). IEEE.

[22] Wu, X., Sun, C., Zou, T., Li, L., Wang, L., & Liu, H. (2020). SVM-based image partitioning for vision recognition of AGV guide paths under complex illumination conditions. robotics and computer-integrated manufacturing, 61, 101856.

[23] Feng, T., & Jiao, B. (2017, August). The vision guidance and image processing of AGV. In Journal of Physics: Conference Series (Vol. 887, No. 1, p. 012093). IOP Publishing.

[24] Weckx, S., Vandewal, B., Rademakers, E., Janssen, K., Geebelen, K., Wan, J., ... & van Nunen, E. (2020, October). Open experimental AGV platform for dynamic obstacle avoidance in narrow corridors. In 2020 IEEE Intelligent Vehicles Symposium (IV) (pp. 844-851). IEEE.

[25] Miao, Z., Zhang, X., & Huang, G. (2021, August). Research on dynamic obstacle avoidance path planning strategy of AGV. In Journal of Physics: Conference Series (Vol. 2006, No. 1, p. 012067). IOP Publishing.

[26] Wu, B., Chi, X., Zhao, C., Zhang, W., Lu, Y., & Jiang, D. (2022). Dynamic path planning for forklift AGV based on smoothing A* and improved DWA hybrid algorithm. Sensors, 22(18), 7079.

[27] Chaudhuri, R., Acharjee, J., Deb, S., & Das, P. P. (2022, October). Depth perceived trajectory mapping using precedent obstacle avoidance (poa) for optimized agv locomotion. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon) (pp. 1-6). IEEE.

[28] Yuan, Y., Shi, Y., Yue, S., Xue, S., Yi, C., & Chen, B. (2022). A dynamic obstacle avoidance method for AGV based on improved speed barriers. Electronics, 11(24), 4175.

[29] Li, Y., Wang, D., Li, Q., Cheng, G., Li, Z., & Li, P. (2023). Advanced 3D navigation system for AGV in complex smart factory environments. Electronics, 13(1), 130.

[30] Zhang, P., Ding, Y., & Chen, Y. (2024). Obstacle Avoidance Control of the AGV With Actuator Dead Zones Based on Fixed-Time Disturbance Observers and Command Filtering. IEEE Access, 12, 66701-66716.

[31] Huang, Z., Ren, Z., Chen, T., Cai, S., & Xu, C. (2025). Autonomous Navigation and Collision Avoidance for AGV in Dynamic Environments: An Enhanced Deep Reinforcement Learning Approach With Composite Rewards and Dynamic Update Mechanisms. IET Cyber-Systems and Robotics, 7(1), e70012.

[32] Zhang, D., Huo, D., Zhou, M., & Cao, Z. (2025). MPC-DS: A Safe Path Tracking Method for AGVs in Dynamic Environments With Dense Obstacles. IEEE Transactions on Intelligent Transportation Systems.

[33] Wu, Z., & Li, T. (2025). Research on Multi-AGV Rail Transportation Path Planning and

Dynamic Obstacle Avoidance Based on Multi-Sensor Environmental Perception and Deep Reinforcement Learning. IEEE Access.

[34] Chen, Y., Chen, H., Chen, S., Guo, J., Huang, D., & Kang, C. (2025). Dynamic obstacle avoidance scheduling strategy for AGVs in dual-layer production lines considering task completion time and energy consumption. Production Engineering, 1-24.

[35] Hu, J., Niu, J., Zhang, B., Gao, X., Zhang, X., & Huang, S. (2025). Obstacle Avoidance Strategy and Path Planning of Medical Automated Guided Vehicles Based on the Bionic Characteristics of Antelope Migration. Biomimetics, 10(3), 142.

[36] Wu, X., Sun, C., Zou, T., Xiao, H., Wang, L., & Zhai, J. (2019). Intelligent path recognition against image noises for vision guidance of automated guided vehicles in a complex workspace. Applied Sciences, 9(19), 4108.

[37] Sun, M., Lu, L., Ni, H., Wang, Y., & Gao, J. (2022). Research on dynamic path planning method of moving single target based on visual AGV. SN Applied Sciences, 4(3), 86.

[38] Wang, Z. M., Li, M., Gao, C., & Jin, X. M. (2023, April). LCS-based Visual Recognition for AGV Guide Paths under Complex Illumination Conditions. In Journal of Physics: Conference Series (Vol. 2476, No. 1, p. 012069). IOP Publishing.

[39] Zhu, Q., Zheng, Z., Wang, C., & Lu, Y. (2023). Research on AGV path tracking method based on global vision and reinforcement learning. Science Progress, 106(3), 00368504231188854.

[40] Jiang, Y., Leach, M., Yu, L., & Sun, J. (2023, August). Mapping, navigation, dynamic collision avoidance and tracking with LiDAR and vision fusion for AGV systems. In 2023 28th International Conference on Automation and Computing (ICAC) (pp. 1-6). IEEE.

[41] Nie, J., Zhang, G., Lu, X., Wang, H., Sheng, C., & Sun, L. (2024). Obstacle avoidance method based on reinforcement learning dual-layer decision model for AGV with visual perception. Control Engineering Practice, 153, 106121.