

## Research Article

# Probabilistic Membrane Computing-Based SLAM for Patrol UAVs in Coal Mines

Jiachang Xu <sup>1,2</sup>, Yourui Huang <sup>3</sup>, Ruijuan Zhao,<sup>4</sup> Yu Liu <sup>1</sup> and Hongjin Li <sup>5</sup>

<sup>1</sup>State Key Laboratory of Mining Response and Disaster Prevention and Control in Deep Coal Mines, Anhui University of Science and Technology, Huainan City, Anhui Province 232001, China

<sup>2</sup>School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan City, Anhui Province 232001, China

<sup>3</sup>School of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan City, Anhui Province 232001, China

<sup>4</sup>Department of Information Engineering, PLA Army Academy of Artillery and Air Defense, Hefei City, Anhui Province 230031, China

<sup>5</sup>School of Biomedical Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian City, Liaoning Province 116024, China

Correspondence should be addressed to Yourui Huang; [hyr628@163.com](mailto:hyr628@163.com)

Received 20 September 2021; Revised 26 October 2021; Accepted 3 November 2021; Published 19 November 2021

Academic Editor: Carlos Marques

Copyright © 2021 Jiachang Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Patrol unmanned aerial vehicles (UAVs) in coal mines have high requirements for environmental perception. Because there are no GPS signals in a mine, it is necessary to use simultaneous localization and mapping (SLAM) to realize environmental perception for UAVs. Combined with complex coal mine environments, an integrated navigation algorithm for unmanned helicopter inertial measurement units (IMUs), light detection and ranging (LiDAR) systems, and depth cameras based on probabilistic membrane computing-based SLAM (PMC-SLAM) is proposed. First, based on an analysis of the working principle of each sensor, the mathematical models for the corresponding sensors are given. Second, an algorithm is designed for the membrane, and a probabilistic membrane system is constructed. The probabilistic SLAM map is constructed by sparse filtering. The experimental results show that PMC can further improve the accuracy of map construction. While adapting to the trend of intelligent precision mining in coal mines, this approach provides theoretical support and application practice for coal mine disaster prevention and control.

## 1. Introduction

UAV application scenarios have high requirements for navigation performance, especially in complex environments, and a single navigation mode has difficulty meeting these requirements. At present, inertial navigation sensors, lasers, vision sensors, and other sensors are mainly used for multi-sensor data fusion with an effective SLAM control algorithm. Navigation with a single sensor is gradually being replaced by integrated navigation due to environments with increasing complexity. Reference [1] improves the accuracy of data association by using the complementary advantages of point and line features. This method is only tested in a simulation

environment and lacks training on actual scenes. In reference [2], when the laser cannot work normally, a vision-based path planning method for UAVs is implemented by using a fully convolutional neural network; this approach achieves good results, but its generalization ability is limited for high-dimensional data or large scenes. In reference [3], UAV SLAM is realized by neural cell population coding, but the simulation can only be completed under specific constraints. In reference [4], a neural network is used to predict the error before and after particle filtering, which reduces the positioning error caused by unreliable laser data. In reference [5], based on adaptive filtering, the state noise variance matrix and noise variance matrix of the Kalman

filter are modified in real time by means of the residual mean and variance, thereby improving the adaptability and robustness of the model. Reference [6] proposes a control strategy to avoid collisions. The detection model uses distance and image width information to optimize the barrier-free space. In reference [7], based on the inertial navigation method of scene recognition, a drift error model between nodes is established, and the purpose of correcting the drift error is achieved. In reference [8], an integrated navigation control system based on biomimetic polarized light navigation sensors, metal-insulator-metal sensors (MIMs), and GPS is designed. The system error does not accumulate with time and meets the accuracy and reliability requirements of autonomous navigation. In reference [9], a visual pose estimation method for underground UAVs based on a depth neural network is proposed. This method can significantly improve the positioning accuracy of roadways in complex environments, but there are some problems, such as large computing resource demands and relatively long computing times. In reference [10], a UAV rotor system that can sense an unknown outdoor environment autonomously and plan a path automatically in real time is designed and implemented. The system basically realizes the real-time autonomous perception and path planning of UAVs in unknown outdoor scenes. In reference [11], a monocular vision semidirect visual odometry/integrated navigation system (SVO/INS) algorithm is proposed to estimate the position, attitude, and velocity of a UAV. In terms of vision and inertial measurement unit (IMU) combinations, whether from a single or dual combination mode and combined with traditional or intelligent algorithms, the autonomous navigation of UAVs can be realized under certain conditions. Because the accuracy of pose acquisition depends mostly on the extraction and matching of environmental features, it is a challenging task to carry out effective SLAM in combination with the characteristics of sensors in the face of a complex coal mine environment.

Membrane computing (a p-system) is derived from natural computing and was developed by Păun in 1998 [12]. As a framework of computational model research inspired by the mechanism of cell organization, membrane computing has made rich research achievements in theoretical explorations and application studies and has gradually extended its advantages to the field of optimal control [13–15]. Based on the membrane computing model, an effective membrane algorithm has been designed to solve optimization control problems in various fields of application. An optimization algorithm, which integrates a membrane structure and computing methods, has successively solved SAT [16–18], satisfiability [19–22], HPP [23, 24], and function optimization problems [25, 26]. In recent years, the author's team has been engaged in attitude control and path planning for UAVs in coal mines [27]. At the same time, membrane computing is gradually achieving breakthroughs in the field of optimal control, and it has become a hot research topic.

In this paper, a navigation algorithm for UAVs, IMUs, LiDAR systems, and depth cameras based on probabilistic membrane computing-based SLAM (PMC-SLAM) is proposed, and the main contributions are as follows:

- (1) Based on the sensor model, a corresponding mathematical model is given
- (2) A probabilistic membrane system-based calculation model is further constructed
- (3) An algorithm is designed to realize the map construction process in the membrane

## 2. Sensor Model and Information Fusion

**2.1. Inertial Measurement Unit.** An IMU is composed of accelerometers and gyroscopes. Gyroscopes measure angular velocity and accelerometers measure linear acceleration; the former is the principle of inertia and the latter is the principle of force balance. The accelerometer is correct in a long time; it has errors in a short time due to signal noise. The gyroscope is more accurate in a short time and has errors in a long time due to drift. Therefore, the two need to adjust each other to ensure the correct heading. The IMU is used to measure the object's uniaxial, biaxial, or triaxial attitude angle (or angular rate) and acceleration device. For inertial applications in attitude, it consists of a combination of a three-axis gyroscope and a three-axis accelerometer. The depth camera used in this paper, Intel RealSense D435i, integrates an IMU unit, Bosch BMI055, which can synchronize IMU data and depth in real time, suitable for UAV system position awareness. The BMI055 is a 6-axis inertial sensor consisting of a digital 3-axis 12-position acceleration sensor and a 3-axis digital 16-position gyroscope with  $\pm 2000^\circ/\text{SEC}$ . It has a total of 6 degrees of freedom. It allows the measurement of the angular velocity and acceleration of the three vertical axes with very low noise, thus detecting the UAV posture. The measurement value of an inertial measurement is based on the coordinate system of the sensor itself and the coordinate system of a non-UAV body. According to the UAV mechanical model constructed by the author of this paper [27], the IMU is modeled; the body state vector pose is assumed to be  $\{c_{vi}, r^{vi}\}$ , the angular velocity is  $w^{vi}$ , and the acceleration is  $\dot{w}^{vi}$ . Furthermore, it is assumed that the position relationship between the sensor and the body coordinate system is determined, where  $r^{sv}$  represents rotation and  $c_{sv}$  represents translation. The relationship between the angular velocity  $w$  and the speed of the body in the sensor coordinate system is shown as follows [27, 28].

$$w = w^{vi} c_{sv}. \quad (1)$$

According to the principle of acceleration measurement, the measured value of acceleration is expressed as

$$a = (\bar{r} - g) c_{si}. \quad (2)$$

$\bar{r}$  is the acceleration of the origin  $s$  of the sensor coordinate, and  $g$  is the acceleration of gravity. Since the value expressed in Formula (2) is not the value in the body coordinate system, it is necessary to convert the coordinates of the sensor and the body.

$$r^{si} = c_{vi}^T r^{sv} + r^{vi}. \quad (3)$$

By using Poisson's formula to derive Formula (3) twice, it can be concluded that

$$\bar{r}^{si} = c_{vi}^T r^{sv} \dot{w}^{vi^\wedge} + c_{vi}^T r^{sv} w^{vi^\wedge} w^{vi^\wedge} + \bar{r}^{vi}. \quad (4)$$

The right side of Formula (4) consists of the initial parameters. Substituting Formula (4) into Formula (2) yields

$$a = c_{sv} \left( (\bar{r} - g) c_{si} + r^{sv} w^{vi^\wedge} w^{vi^\wedge} + r^{sv} \dot{w}^{vi^\wedge} \right). \quad (5)$$

Formulas (1) and (5) are further expressed in matrix form:

$$\begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} w^{vi} c_{sv} \\ c_{sv} \left( (\bar{r} - g) c_{si} + r^{sv} w^{vi^\wedge} w^{vi^\wedge} + r^{sv} \dot{w}^{vi^\wedge} \right) \end{bmatrix}. \quad (6)$$

**2.2. LiDAR.** LiDAR uses a laser pulse to realize the position measurement of the sensor and describes the azimuth and elevation angle by controlling the reflection angle of the laser beam; additionally, the distance is measured by the flight time. In LiDAR ranging methods, both direct and oblique laser triangulations can achieve high precision and noncontact measurement of the measured object, but the resolution of the direct laser is not as high as that of the oblique laser. The RPLIDAR LiDAR of Silan Technology used in this paper adopts the oblique laser triangle ranging method with high resolution. Its unique RPLIDAR 3.0 laser ranging engine can conduct up to 16,000 ranging actions per second with a range radius of 25 meters and an angular resolution of up to 0.225°. Driven by the motor, the ranging core will rotate clockwise, so as to realize the 360° omnidirectional scanning ranging detection of the surrounding environment. In each ranging process, the RPLIDAR series laser radar will emit a modulated infrared laser signal; the laser signal after irradiating the reflective target object will be received by the visual acquisition system and then through the embedded RPLIDAR DSP (Digital Signal Processor) real-time solution. The distance between the irradiated target object and the LiDAR including the current included angle information is outputted through the communication interface; the schematic diagram is shown in Figure 1 [29].

In Figure 1, the coordinates of point P are expressed as [30]

$$p = r^{ps} = [s, y, z]^T. \quad (7)$$

In the form of a matrix, the above equation can be rewritten as

$$p = c^T(\alpha) c^T(\varepsilon) [r, 0, 0]^T. \quad (8)$$

In Formula (8),  $r$  is the distance,  $\varepsilon$  is the pitch angle,  $\alpha$  is the direction angle, and  $c$  is the rotation matrix along

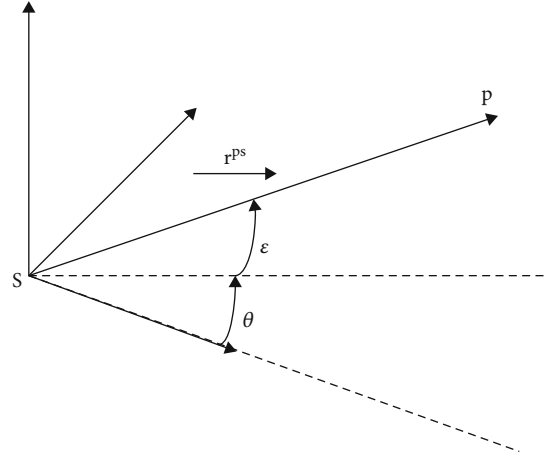


FIGURE 1: LiDAR schematic diagram.

the coordinate axis. To further express the relationship between Formulas (7) and (8), the rotation is inverted.

$$\begin{bmatrix} \varepsilon \\ \alpha \\ r \end{bmatrix} = \begin{bmatrix} \sin \left( z / \sqrt{x^2 + y^2 + z^2} \right)^{-1} \\ \tan (y/x)^{-1} \\ \sqrt{x^2 + y^2 + z^2} \end{bmatrix}. \quad (9)$$

**2.3. Depth Camera.** Compared with a traditional camera, a depth camera can measure the depths of pixels. The Intel RealSense D435i depth camera is taken as an example and is shown in Figure 2.

In Figure 2, the coordinates of point P ( $x, y, z$ ) and the models of the left and right cameras are defined as [31]

$$\begin{bmatrix} u_{l,r} \\ v_{l,r} \end{bmatrix} = p \underbrace{\begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_K \frac{1}{z} \begin{bmatrix} x \pm \frac{b}{2} \\ y \\ z \end{bmatrix}. \quad (10)$$

In Formula (10),  $K$  is the camera parameter,  $f_u$  is the horizontal pixel focal length, and  $f_v$  is the vertical focal length. Since the parameter matrices of the two cameras are the same, when the left camera obtains an observation value, the corresponding observation value (vertical) of the right camera can be obtained along the real line in Figure 2. Therefore, Formula (10) can be further developed to obtain the model of the whole camera as

$$\begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix} = \begin{bmatrix} f_u & 0 & c_u & f_u \frac{b}{2} \\ 0 & f_v & c_v & 0 \\ f_u & 0 & c_u & f_u \frac{b}{2} \\ 0 & f_v & c_v & 0 \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \quad (11)$$

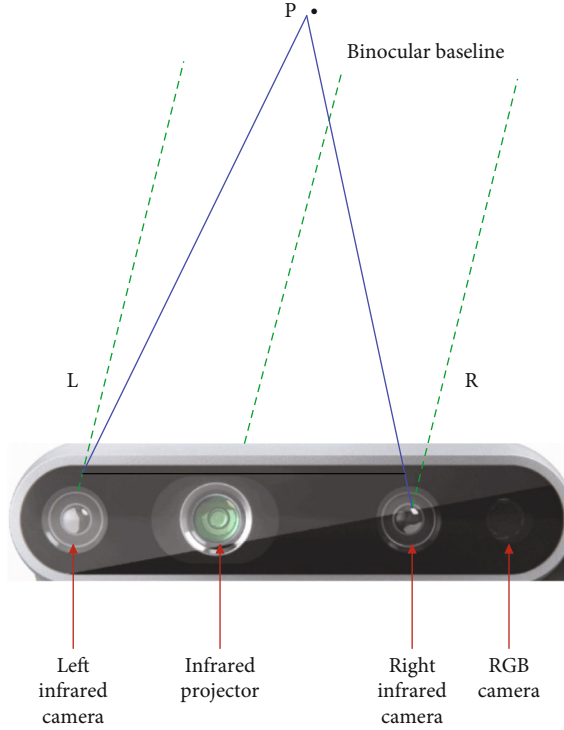


FIGURE 2: D435i depth camera model.

In Formula (11), there is a pose relationship ( $c_{l,r} = 1$ ) between the left and right cameras, and  $v_l = v_r$  can be obtained.

**2.4. Sensor Information Fusion.** The UAV used in this paper is equipped with an IMU, LiDAR (an SLR A2), and a depth camera (an Intel D435i). During the process of mapping, multisensor information needs to be fused. The multisensor information fusion technology makes the multilevel and multispace information complementary and optimal combination processing of various sensors and finally produces the consistent interpretation of the observation environment. In this process, multisource data should be fully used for rational control and use, and the ultimate goal of information fusion is to extract more useful information through multi-level and multispect combination of information based on the separated observation information obtained by each sensor. This process not only takes advantage of the cooperative operation of multiple sensors but also comprehensively processes the data of other information sources to improve the intelligence of the whole sensor system. Hybrid fusion framework has strong adaptability, including the advantages of centralized fusion and distributed, and its stability is strong. The structure of the hybrid fusion method is more complex than that of the first two fusion methods. Although it increases the cost of communication and computation, the airborne computer in this paper is sufficient to bear the load of computation force. Gaussian convolution is used for image projection. The convolution is a filtering operation, weighted by adding the center point to its neighborhood to get a new value for the center point. After filtering, the pixel

value of the central point is replaced by the weighted average of the pixel value of its surrounding points, making the boundary more blurred (low-pass filtering). The function image of the Gaussian kernel is a normally distributed bell-shaped line. The closer the coordinates are to the center point, the larger the value is, and vice versa. The closer you are to the center, the more weight you have; the farther away you are from the center, the less weight you have. According to the point coordinates of the filter, the value calculated by the Gaussian kernel is the value of the filter, which is the corresponding weight of each point on the image. The final processing result is obtained by rolling multiplication of the filter and the original image. If obstacles are not identified by simultaneous interpretations under different sensors, this situation may lead to serious differences or even errors in information, and thus, the dependence on map generation cannot be coupled.

In this paper, simultaneous interpretation maps are built under different sensors because the data between different sensors are not interdependent. Suppose that the maps generated by the three sensors are represented as  $\text{map}^3$ , which can be decomposed by de Morgan's law:

$$p(\text{map}) = 1 - \prod (1 - p(\text{map}^3)). \quad (12)$$

### 3. Extended SLAM Based on Probabilistic Sparsity

**3.1. Probabilistic SLAM.** When the UAV does not know its position, it cannot obtain a map of its environment, and all the data are concentrated on the measurement data and control data. From the perspective of probability, the SLAM problem is divided into full SLAM and online SLAM. Online SLAM only includes the variable estimation problem at time  $t$ , and its algorithm is incremental; the corresponding control quantity and measurement value are discarded after processing. In full SLAM, in addition to the pose of the UAV, the path and the posterior map need to be calculated. The probability at time  $t$  can be expressed as

$$p(x_{1-t}, m | y_{1-t}, u_{1-t}). \quad (13)$$

In online SLAM, the real-time pose and map posterior are considered, and the temporal probability is expressed as

$$p(x_t, m | y_{1-t}, u_{1-t}). \quad (14)$$

In Formulas (13) and (14),  $x_t$  represents the position and attitude of the UAV at time  $t$ , respectively,  $m$  is a map, and  $y_{1-t}, u_{1-t}$  denotes the measurement and control data. The respective models of these two types of SLAM are shown in Figure 3.

The difference between online SLAM and full SLAM mainly lies in the different branches of the algorithm, especially in the application of practical scenarios. The online SLAM problem is the integration of the past positions of the full SLAM problem; the process of online SLAM can be realized by integrating the past state of the full SLAM; this

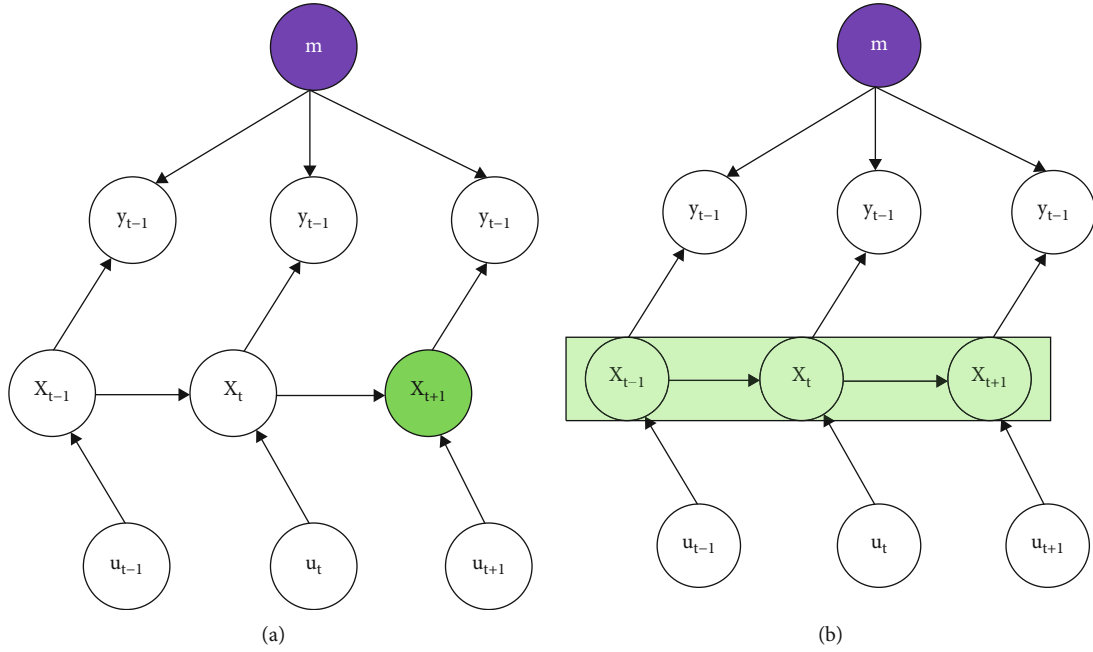


FIGURE 3: (a) Online SLAM and (b) full SLAM.

integration presents a continuous state, which is carried out in turn, as shown in the following formula:

$$p(x_t, m | y_{1-t}, u_{1-t}) = \iint \dots \int p(x_{1-t}, m | y_{1-t}, u_{1-t}) dx_1 dx_2 \dots dx_{t-1}. \quad (15)$$

Among continuous and discrete problems, a continuous problem includes the UAV pose and location in its map, and the study object is represented by a beacon during the feature representation process. When an object is detected, the algorithm calculates the correlation between the detected object and the UAV, exhibiting a discrete feature, namely, the “0” or “1” state. Therefore, it is necessary to define a consistency variable. Combining Formulas (13) and (14), the online and full SLAM models with consistency variables are as follows:

$$\begin{cases} p(x_{1-t}, m, f_{1-t} | y_{1-t}, u_{1-t}), \\ p(x_t, m, f_t | y_{1-t}, u_{1-t}). \end{cases} \quad (16)$$

In Formula (16),  $f_t$  is the vector corresponding to the consistency variable.

**3.2. Sparse Extended Filtering.** The Kalman filter in the linear problem was proven to be the optimal estimation; it is a very big limitation; only processing the linear model and measuring model for precise estimates of the nonlinear scenario does not achieve optimal estimates of the effect; in order to be able to set up a linear environment, a false process model for the constant velocity model is required, but in the actual application, this is not the case. Both process models and measurement models are nonlinear. Compared with the

extended Kalman filter, for the purpose of online operation and high computational efficiency, the sparse extended filter represents information with high efficiency, inherits UAV poses and map posteriori, and maintains the sparse matrix through nonzero elements. The calculation process of the sparse extended filter includes measurement updates, motion updates, sparsification, and estimation.

The information matrix  $\Omega$  and vector  $\zeta$  are updated to complete the processing of the control information according to the Kalman filter [32].

$$\dot{\mu}_t = F_x^T \delta + \dot{\mu}_{t-1}, \quad (17)$$

$$\Sigma_t = F_x^T R_t F_x + G_t \Sigma_{t-1} G_t^T. \quad (18)$$

In Formulas (17) and (18),  $\Sigma$  is the covariance matrix,  $F_x$  is the UAV state vector matrix,  $G_t$  is the derivative of the Jacobian matrix with respect to time  $t$ , and  $\dot{\mu}_t$  represents the estimated mean value at time  $t$ . The expressions of  $G_t$ ,  $F_x$ , and  $\delta$  are as follows:

$$G_t = F_x^T \Delta F_x + I, \quad (19)$$

$$F_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (20)$$

$$\delta = \begin{pmatrix} \omega_t \Delta t \\ \frac{v_t}{\omega_t} \sin(\mu_{t-1, \theta} + \omega_t \Delta t) - \frac{v_t}{\omega_t} \sin(\mu_{t-1, \theta}) \\ \frac{v_t}{\omega_t} \cos(\mu_{t-1, \theta} + \omega_t \Delta t) - \frac{v_t}{\omega_t} \cos(\mu_{t-1, \theta}) \end{pmatrix}. \quad (21)$$

The following can be deduced from Formulas (18) and (19).

$$\dot{\Omega} = \left[ F_x^T R_t F_x + (F_x^T \Delta F_x + I)^T \Omega_{t-1} (F_x^T \Delta F_x + I) \right]^{-1}. \quad (22)$$

In Formula (22), the dimension of  $\Omega$  is random and implemented in finite time. Assuming that  $\Omega$  is sparse, the update efficiency is enhanced, and the following can be defined:

$$\Phi_t = G_t^{-1} \Omega_t [G_t^T]^{-1}. \quad (23)$$

According to Formula (22), it can be concluded that

$$\dot{\Omega} = [F_x^T R_t F_x + \Phi_t^{-1}]^{-1}. \quad (24)$$

From the inverse of the matrix, it is further obtained that

$$\dot{\Omega} = [F_x^T R_t F_x + \Phi_t^{-1}]^{-1} = \kappa_t \Phi_t. \quad (25)$$

It is assumed that  $\Phi_t$  is calculated based on  $\Omega$  in a limited time frame; then, the calculation is feasible under the condition of finite time. Using the matrix elements (nonzero) of the UAV poses and map features, the calculation does not depend on the size of  $\Omega$ . Considering the inverse of  $G_t$ ,  $G_t^{-1}$  can be calculated as follows:

$$G_t^{-1} = (I + F_x^T (I + \Delta) F_x + F_x^T \Delta F_x) = \psi_t + I. \quad (26)$$

In Formula (26), the corresponding map feature element is nonzero.

The measurement update considers the filter update in the flight process of the UAV, which is realized by an extended Kalman filter.

$$\begin{cases} \Omega = H_t^T Q_t^{-1} H_t + \dot{\Omega}_t, \\ \xi_t = H_t^T Q_t^{-1} \dot{\xi}_t. \end{cases} \quad (27)$$

In Formula (27),  $Q_t$  is the noise covariance matrix.

**3.3. Sparseness.** Sparse extended filtering is necessary for sparse  $\Omega$  information matrices. Through sparse representation, the posterior distribution is in a sparse state. Based on this, the relationship between the UAV positions and the map features is eliminated, and the number of features

is further limited. To realize the above idea, two new connections are introduced. First, a feature is activated by an inactive connection, and a new connection is introduced between the UAV pose and the corresponding feature. Second, UAV motion introduces two new connections between active features to limit the number of active features and to avoid having two nonsparse boundaries. At this time, the sparsity is obtained by less active features.

In the process of sparse definition, the feature set is divided into three (disjoint) subsets.

$$m = m^0 + m^1 + m^2. \quad (28)$$

In Formula (28),  $m^1$  is the feature set of activity continuation,  $m^0$  is the characteristic of the activity to be stimulated, and  $m^2$  is inactive. The inactive state is continued in the sparsification step; at the same time, the connection between the UAV pose and  $m^0$  is deleted. Sparseness is introduced into the posterior because  $m^1$  and  $m^0$  contain all the current features, and the posterior can be characterized as follows:

$$\begin{aligned} p(y_t | z_{1-t}, u_{1-t}, f_{1-t}) \\ &= p(x_t, m^0, m^1, m^2 | z_{1-t}, u_{1-t}, f_{1-t}) \\ &= p(m^0, m^1, m^2 | z_{1-t}, u_{1-t}, f_{1-t}) p(x_t, m^0, m^1, m^2 | z_{1-t}, u_{1-t}, f_{1-t}) \\ &= p(m^0, m^1, m^2 = 0 | z_{1-t}, u_{1-t}, f_{1-t}) p(x_t, m^0, m^1, m^2 | z_{1-t}, u_{1-t}, f_{1-t}). \end{aligned} \quad (29)$$

In Formula (29), if  $m^0$  and  $m^1$  are known,  $x_t$  does not depend on the inactive feature  $m^2$ , so  $m^2$  can be taken as a random value. Using the general term sparsification protocol, reducing the dependence on  $m^0$ , and taking  $m^2 = 0$ ,

$$\begin{aligned} \dot{p}(x_t, m^0, m^1, m^2 | z_{1-t}, u_{1-t}, f_{1-t}) \\ &= p(m^1, m^2 = 0 | z_{1-t}, u_{1-t}, f_{1-t}) p(x_t, m^0, m^1, m^2 | z_{1-t}, u_{1-t}, f_{1-t}) \\ &= \frac{p(x_t, m^1, |m^2 = 0, z_{1-t}, u_{1-t}, f_{1-t})}{p(m^1 | m^2 = 0, z_{1-t}, u_{1-t}, f_{1-t})} p(m^0, m^1, m^2 | z_{1-t}, u_{1-t}, f_{1-t}). \end{aligned} \quad (30)$$

## 4. Probability Membrane System-Based SLAM

**4.1. Probabilistic Membrane System.** In a probabilistic membrane system, rules are executed in a probabilistic mode, so the system can process data quickly. Combined with the powerful distributed and parallel characteristics of membrane computing, an independent and collaborative membrane system for data processing is constructed. The modeling principle and process for UAVs in coal mines are shown in Figure 4 by using a probabilistic membrane system.

To determine the real-time position of a UAV, the membrane controller receives the airborne sensor data  $(x, y, \theta)^T$  and the output position update data  $(x', y', \theta')^T$  at each

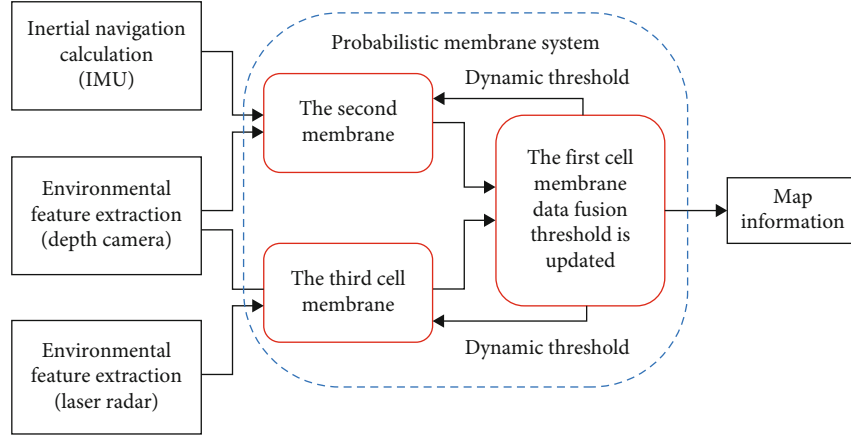


FIGURE 4: Modeling process of the probabilistic membrane system.

cycle during the beginning of the execution process, and a probabilistic membrane system with a degree of 3 is established.

$$\prod = (M, \mu, w_1, w_2, w_3, R, \{c_r\}_{r \in R}). \quad (31)$$

Formula (31) has the following characteristics:

- (1)  $M = \{x_{ij}, y_{ij}, \theta_{ij}, \text{Err} : i, j \in [1, 2]\}$ , where  $M$  is the character set element contained in the probabilistic membrane system
- (2)  $\mu = [[ ]_2]_3$ , where  $\mu$  is the membrane mechanism of the probabilistic membrane system of the UAV
- (3)  $w_1 = p(x_t | \mu_t, x_{t-1})$ , where  $w_1$  represents the solution probability density
- (4)  $w_2 = p(x'_t | \mu'_t, x_{t-1})$ , where  $w_2$  represents the probability density after noise interference
- (5)  $w_3 = p(x_t | \mu_t, x_{t-1})$ , where  $w_3$  represents the ideal probability density
- (6)  $R$  is the rule set, and energy conservation is maintained during the transfer process
- (7)  $c_r$  is the probability of the evolution of objects based on  $R$  rules

**4.2. Probabilistic Membrane Calculation Model.** According to Section 3.2 of this paper, the state variable submatrix is extracted by calculating the matrix for all excluded variables that obey the distribution  $p(x_t, m^0, m^1 | m^2 = 0)$ .

$$\Omega'_t = F_{x,m^1,m^0} F_{x,m^1,m^0}^T \Omega_t F_{x,m^1,m^0} F_{x,m^1,m^0}^T. \quad (32)$$

According to the matrix lemma, the matrix information of  $p(m^1 | m^2, z_{1-t}, u_{1-t}, f_{1-t})$  and  $p(x_t, m^0, m^1, m^2 | z_{1-t}, u_{1-t}, f_{1-t})$  are defined as follows:

$$\begin{cases} \Omega'_t = \Omega'_t - \Omega'_t F_{m^0} (F_{m^0} \Omega'_t F_{m^0}^T)^{-1} \Omega'_t F_{m^0}^T, \\ \Omega'_t = \Omega'_t - \Omega'_t F_{x,m^0} (F_{x,m^0} \Omega'_t F_{x,m^0}^T)^{-1} \Omega'_t F_{x,m^0}^T. \end{cases} \quad (33)$$

In Formula (33), all states are projected into a projection matrix containing variable quantum set states. Similarly,  $p(m^0, m^1, m^2 | z_{1-t}, u_{1-t}, f_{1-t})$  can be transformed into a matrix.

$$\Omega'_t = \Omega'_t - \Omega'_t F_x (F_x \Omega'_t F_x^T)^{-1} \Omega'_t F_x^T. \quad (34)$$

The combination of Formulas (32) and (34) can be obtained:

$$\begin{aligned} \dot{\Omega}_t &= \Omega'_t + \Omega'_t + \Omega'_t = \Omega'_t - \Omega'_t F_{m^0} (F_{m^0} \Omega'_t F_{m^0}^T)^{-1} \Omega'_t F_{m^0}^T + \Omega'_t \\ &\quad - \Omega'_t F_{x,m^0} (F_{x,m^0} \Omega'_t F_{x,m^0}^T)^{-1} \Omega'_t F_{x,m^0}^T + \Omega'_t \\ &\quad - \Omega'_t F_x (F_x \Omega'_t F_x^T)^{-1} \Omega'_t F_x^T, \end{aligned} \quad (35)$$

$$\dot{\xi}_t = \mu_t \dot{\Omega}_t = \mu_t (\Omega'_t + \Omega'_t + \Omega'_t). \quad (36)$$

Due to the limited cycle time of the probabilistic membrane algorithm, it does not depend on the scale of the map itself. At the same time, control is added to the calculation estimation. Through the efficient calculation of features, the UAV position and feature vectors are represented during the process of building the map and updating the elements in the matrix (vector). The P-Lingua file framework for the probabilistic membrane calculation model is shown in Figure 5.

**4.3. Map Consistency Fusion.**  $\Omega_t$  and  $\xi_t$  are inputted according to the UAV posterior. The displacement  $d$  and the angle vector  $\alpha$  determine the current position in the coordinate

```

@ model<probabilistic>
Def init_membrane_structure()
{
  @Mu=[[[]' 2[]' 3[]' 4[]' 5] 1;
}
Def init_multisets()
{
  @ ms(1) +=x, y, z, 0 ;
  @ ms(2)=Measure update;
  @ ms(3)=Active update;
  @ ms(4)=Map update;
}
Def init_rules()
{
  Evolutionary rule-sets;
}
Def main()
{
  Call init_membrane_structure();
  Call init_multisets();
  Call init_rules();
}

```

FIGURE 5: P-Lingua file frame for the membrane computing model.

```

1:  $\Delta = [d_x \ d_y \ \alpha]^T$ .
2:  $\mathbb{R} = \begin{bmatrix} 0 & \sin \alpha & \cos \alpha \\ 0 & \cos \alpha & \sin \alpha \\ 1 & 0 & 1 \end{bmatrix}$ .
3:  $\Omega = \mathbb{R}^T \Omega^i \mathbb{R}$ .
4:  $\xi = (\xi^i - \Delta \Omega^i) \mathbb{R}$ .
5:  $\Omega = \begin{pmatrix} 0 & \Omega^1 \\ \Omega^3 & 0 \end{pmatrix}$ .
6:  $\xi = \begin{pmatrix} \xi^1 \\ \xi^3 \end{pmatrix}$ .
7: C: for any eigenvector in the map
8:  $\Omega = FF^T + \Omega$ .
9: Return  $\xi, \Omega$ 

```

ALGORITHM 1: Map fusion.

TABLE 1: Main hardware information parameters of the experimental platform.

Platform	Parameter information
Software 1	P-Lingua virtual machine
Software 2	MATLAB simulation
Helicopter stand	JCZK450L
Development board	NVIDIA Jetson TX2
Inertial measurement unit	MTi-8ATG6
Depth camera	Intel RealSense D435i
Motor	HF500



FIGURE 6: The experimental scene of coal mine roadway.

system. At the same time, the displacement and vector are mapped to the body coordinate system. Based on the rotation and translation, the pose and map characteristics of the UAV are expressed as follows [33, 34]:

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{pmatrix} x_t^i \\ y_t^i \\ \theta_t^i \end{pmatrix} \begin{pmatrix} 0 & \sin \alpha & \cos \alpha \\ 0 & \cos \alpha & \sin \alpha \\ 1 & 0 & 1 \end{pmatrix} + \begin{pmatrix} d_t^i \\ d_t^i \\ \alpha_t^i \end{pmatrix}, \quad (37)$$

$$m_t = \begin{pmatrix} m_t^i \\ m_t^i \\ m_t^i \end{pmatrix} \begin{pmatrix} 0 & \sin \alpha & \cos \alpha \\ 0 & \cos \alpha & \sin \alpha \\ 1 & 0 & 1 \end{pmatrix} + \begin{pmatrix} d_t^i \\ d_t^i \\ 0 \end{pmatrix}. \quad (38)$$

Formulas (37) and (38) describe the rotation and translation of matrices and vectors, respectively. The next step gives the proofs of these two formulas.  $\mathbb{R}$  and  $\delta$  are defined as

$$\mathbb{R} = \begin{bmatrix} 0 & \sin \alpha & \cos \alpha \\ 0 & \cos \alpha & \sin \alpha \\ 1 & 0 & 1 \end{bmatrix}, \quad (39)$$

$$\delta = [d_x \ d_y \ \alpha]^T. \quad (40)$$

The state vector derivation for Formulas (39) and (40) can be obtained as follows:

$$y_t = \mathbb{R}y_t^i + \Delta. \quad (41)$$

According to the similarity principle of spatial coordinate transformation, the posterior model of the UAV with respect to time is defined by translation and rotation, the information matrix, and the vector as follows:



$$\begin{aligned}
& p(y^{1\sim 3} | z_{1\sim t}^1, u_{1\sim t}^1) \\
&= \eta \exp \left( y^{1\sim 3} \xi^{1\sim 3} - \frac{1}{2} y^{1\sim 3, T} \Omega^{1\sim 3} y^{1\sim 3} \right) \\
&= \eta \exp \left( \xi^{1\sim 3} (\mathbb{R} y_t^i + \Delta) - \frac{1}{2} (\mathbb{R} y_t^i + \Delta) \Omega^{1\sim 3} (\mathbb{R} y_t^i + \Delta) \right) \\
&= \eta \exp \left( y_t^{i, T} \Omega^{1\sim 3} \Delta + y_t^{i, T} \mathbb{R}^T \xi^{1\sim 3} - \frac{1}{2} \xi^{1\sim 3} \mathbb{R}^T y_t^{i, T} \right).
\end{aligned} \tag{42}$$

Because  $\mathbb{R}^T = \mathbb{R}^{-1}$ , the following exists for  $\mathbb{R}$ :

$$\begin{cases} \Omega^{1\sim 3} = \mathbb{R}^T \Omega^1 \mathbb{R}, \\ \xi^{1\sim 3} = \mathbb{R} (\xi - \Omega^{1\sim 3} \Delta). \end{cases} \tag{43}$$

There is a problem of data equivalence in the process of data fusion. By adding constraints to further control and map the feature penalty matrix  $C$ , the larger the value of  $C$  is, the stronger the constraint. The fusion algorithm is as follows in Algorithm 1.

The implementation process of Algorithm 1 is described. The relative pose between the UAV coordinate systems is determined by  $[d_x \ d_y \ \alpha]^T$ , which includes the local rotation and translation of the information matrix and vector while maintaining the sparsity of the algorithm. Map fusion is realized by constructing a joint posterior map, which contains the corresponding features in different maps. For the same two features, a connection between the two features in the information matrix is added.

## 5. Experimental Verification

To verify the proposed SLAM algorithm, experiments are performed, and the experimental software and hardware platform and parameters are shown in Table 1.

The coal mine roadway shown in Figure 6 is selected for the experiments. The roadway is 200 meters long, 3.5 meters wide, and 4 meters high. The UAV flies from point A along the green line to point B. The methods based on “LiDAR,” “LiDAR+IMU,” and “LiDAR+IMU+vision (depth camera)” are used for positioning and mapping experiments, respectively. The mapping effects are shown in Figures 7–10. The attitude error and trajectory error are shown in Figures 11–13.

By comparing the experimental results in Figures 7–9, it can be seen that when using only LiDAR for mapping, the mapping is more accurate in a small area, but in a long-distance, large-scale (space) area, the mapping accuracy decreases, and the precision and shape distance error are large. In the case of the “LiDAR+IMU” mapping, due to the auxiliary role of the IMU, the mapping accuracy and effect are significantly improved. However, as the UAV flies farther away, the cumulative error gradually increases. As shown in Figure 7, for the terminal B area, the corridor arc area, and the square area overlap, mapping errors occur. In the “LiDAR+IMU+vision (depth camera)” method, a depth camera is added to provide image feature information. Even

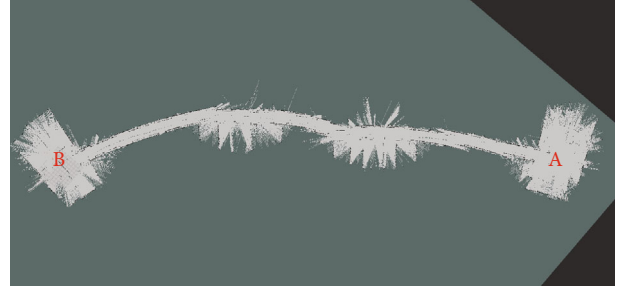


FIGURE 7: Map effect based on LiDAR.

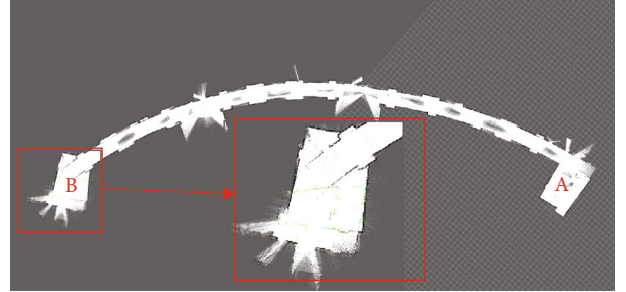


FIGURE 8: Map effect based on LiDAR+IMU.

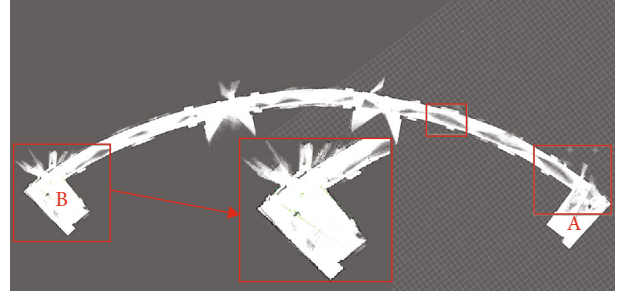


FIGURE 9: Map effect based on LiDAR+IMU+depth camera.

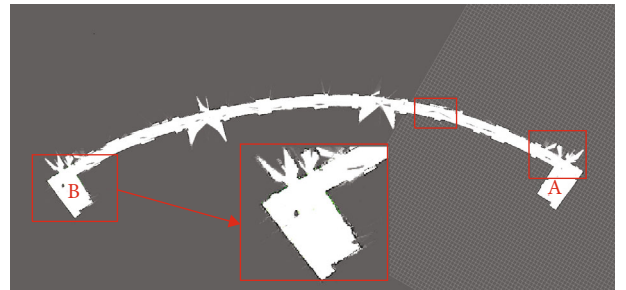


FIGURE 10: Mapping of LiDAR+IMU+camera based on PMC.

in the long-distance scene, the mapping effect is very stable; as shown in Figure 9, the end B area, arc area, and square area are more accurate. The experimental results in Figures 7–9 show that the effect sensor fusion with three kinds of sensors in the same scene is obviously better than that of a single sensor or a two-sensor combination. The

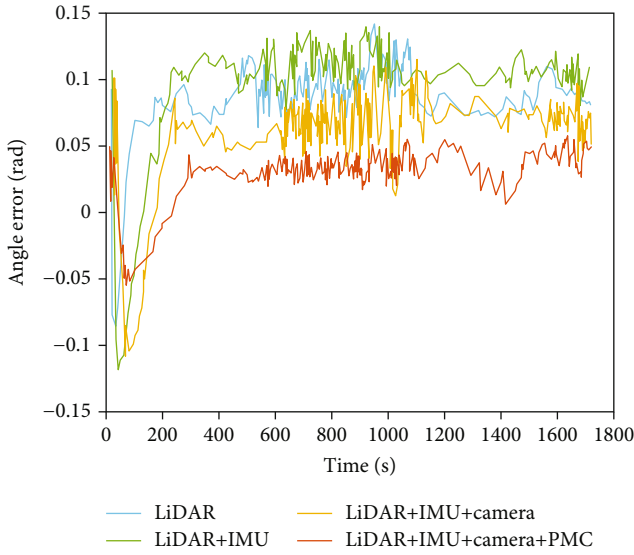


FIGURE 11: Comparison between the angle errors of different SLAM algorithms.

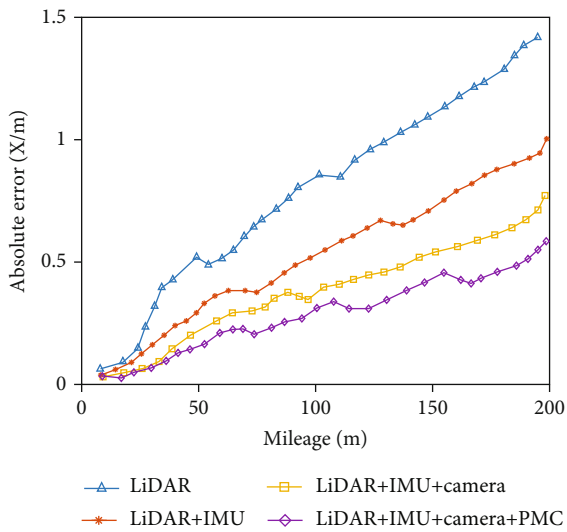


FIGURE 12: Comparison between the absolute errors of different SLAM algorithms.

mapping effect of “LiDAR+IMU+vision (depth camera)” developed based on PMC in this chapter is shown in Figure 10. Compared with that in Figure 9, the mapping effect obtained based on the PMC is obvious, especially in the red box marked in the figure. Furthermore, the simultaneous interpretation errors of the LiDAR+IMU+vision (depth camera) approach based on PMC are compared with those of Figures 11, 12, and 13. The experimental results show that the SLAM effect of the fusion approach with three sensors based on PMC is better than that of one sensor or any combination of two sensors in the long and narrow roadway that simulates the complex environment of a coal mine.

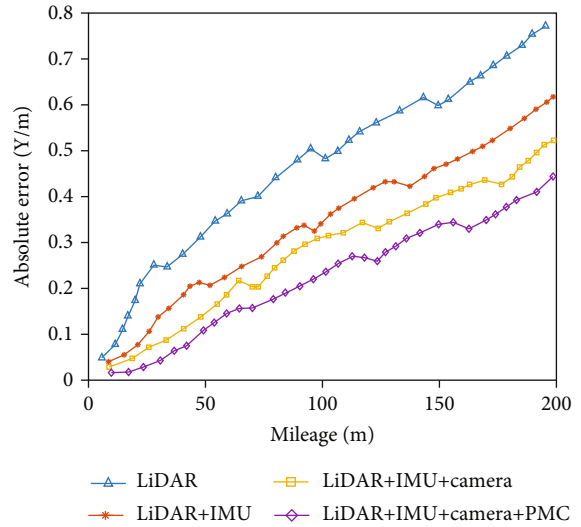


FIGURE 13: Comparison between the absolute errors of different SLAM algorithms.

## 6. Conclusion

Based on the analysis of LiDAR, IMU, and depth camera sensor-based mathematical models, this paper designs a probabilistic membrane system model and membrane algorithm. In the same tunnel application simulation scenario, a theoretical analysis and an experiment are combined. The method is compared with the single-, two-, and three-sensor fusion mapping methods, and it is verified that PMC has a good effect on the mapping performance of underground UAVs. The research results of this paper will provide good theoretical support for engineering practice with respect to disaster prevention and control during the process of precise coal mining in the future.

As an expansion of the tissue membrane system model, this paper puts forward the calculation model that each membrane has the same probability of membrane structures and evolution rules of operation, in the form of probability in data operation process of randomness and uncertainty, including the process of data fusion for collaboration. Between cells may lead to mapping of the fusion accuracy and computational efficiency decline. Next, to improve the research of this paper, we seek a better membrane computing model and algorithm for mapping effect optimization that will pave the way for the study of patrol UAV path planning in coal mines.

## Data Availability

The experimental data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This paper is supported by the Research Start-Up Fund for Introduced Talents of Anhui University of Science and Technology (No. 2021yjrc44), the Open Foundation of State Key Laboratory of Mining Response and Disaster Prevention and Control in Deep Coal Mines (No. SKLMRDPC20KF11), the National Natural Science Foundation of China (No. 61772033), and the Key Project of Natural Science Foundation in Universities of Anhui Province (No. KJ2019A0110).

## References

- [1] Y. Liu, D. Yang, J. Li, Y. Gu, J. Pi, and X. Zhang, "Stereo visual-inertial SLAM with points and lines," *IEEE Access*, vol. 6, pp. 69381–69392, 2018.
- [2] T. Gao, Y. Ma, C. Han, B. Cai, R. Weng, and L. Zhang, "Applying a micro-UAV for searching an indoor target person using laser ranging and monocular vision," *SCIENTIA SINICA Technologica*, vol. 50, no. 7, pp. 971–982, 2020.
- [3] A. Rai and R. J. Kannan, "Population coding of generative neuronal cells for collaborative decision making in UAV-based SLAM operations," *Journal of the Indian Society of Remote Sensing*, pp. 1–7, 2020.
- [4] G. Xiangzhong, C. Chenxiao, Z. Wenhua, W. Lei, and S. Peng, "Indoor integrated navigation system for unmanned aerial vehicles based on neural network predictive compensation," *ACTA AERONAUTICA ET ASTRONAUTICA SINICA*, vol. 41, no. S1, pp. 723790–723790, 2020.
- [5] Q. Guo, L. Sl, J. Jc, and Z. Mn, "Adaptive filter research on integrated navigation system of UAV," *Computer Simulation*, vol. 7, 2012.
- [6] H. Y. Lee, H. W. Ho, and Y. Zhou, "Deep learning-based monocular obstacle avoidance for unmanned aerial vehicle navigation in tree plantations," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 1, pp. 1–18, 2021.
- [7] J. Leaman and H. M. La, "A comprehensive review of smart wheelchairs: past, present, and future," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 486–499, 2017.
- [8] W. Zhi, J. Chu, J. Li, and Y. Wang, "A novel attitude determination system aided by polarization sensor," *Sensors*, vol. 18, no. 1, p. 158, 2018.
- [9] S. S. Mansouri, C. Kanellakis, D. Kominiak, and G. Nikolakopoulos, "Deploying MAVs for autonomous navigation in dark underground mine environments," *Robotics and Autonomous Systems*, vol. 126, p. 103472, 2020.
- [10] X. Liu, B. Chen, Y. He, and D. Li, "Development of an autonomous object transfer system by an unmanned aerial vehicle based on binocular vision," *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, p. 172988142090773, 2020.
- [11] J. Bian, X. Hui, X. Zhao, and M. Tan, "A monocular vision-based perception approach for unmanned aerial vehicle close proximity transmission tower inspection," *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, p. 172988141882022, 2019.
- [12] G. Păun, "Computing with membranes," *Journal of Computer & System Sciences*, vol. 61, no. 1, pp. 108–143, 2000.
- [13] G. Zhang, M. Gheorghe, L. Pan, and M. J. Pérez-Jiménez, "Evolutionary membrane computing: a comprehensive survey and new results," *Information Sciences*, vol. 279, pp. 528–551, 2014.
- [14] C. Buiu, C. Vasile, and O. Arsene, "Development of membrane controllers for mobile robots," *Information Sciences*, vol. 187, pp. 33–51, 2012.
- [15] J. Li, Y. Huang, and J. Xu, "Decoder design based on spiking neural P systems," *IEEE Transactions on Nanobioscience*, pp. 1–1, 2016.
- [16] L. Pan and T. Song, "Progress and prospect of cell computing," *Bulletin of Chinese Academy of Sciences*, vol. 29, no. 1, pp. 115–115, 2014.
- [17] C. Martin-Vide and G. Păun, "On the power of P systems with symport rules," *Journal of Universal Computer Science*, vol. 8, no. 2, pp. 317–331, 2002.
- [18] G. Păun and S. Yu, "On synchronization in P systems," *Fundamenta Informaticae*, vol. 38, no. 4, pp. 397–410, 1999.
- [19] J. Dassow and G. Păun, "On the power of membrane computing," *Journal of Universal Computer Science*, vol. 5, no. 2, pp. 33–49, 1999.
- [20] A. Alhazov, L. Pan, and G. Păun, "Trading polarizations for labels in P systems with active membranes," *Acta Informatica*, vol. 41, no. 2, pp. 111–144, 2016.
- [21] A. Alhazov and L. Pan, "Polarization less P system with active membranes," *Grammars*, vol. 7, no. 1, pp. 141–159, 2004.
- [22] G. Păun, "Computing with membrane," *Journal of Computer and System Sciences*, vol. 61, no. 1, pp. 108–143, 2010.
- [23] C. Martin-Vide, G. Păun, and J. Pazos, "Tissue P systems," *Biosystems*, vol. 296, no. 2, pp. 295–326, 2013.
- [24] M. Ionescu, G. Păun, and T. Yokomori, "Spiking neural P systems," *Fundamenta Informaticae*, vol. 71, no. 2–3, pp. 279–308, 2006.
- [25] G. Păun, "Membrane computing and economics: numerical P systems," *Fundamenta Informaticae*, vol. 73, no. 1–2, pp. 213–227, 2006.
- [26] A. B. Pavel, O. Arsene, and C. Buiu, "Enzymatic numerical P systems—a new class of membrane computing systems," in *The IEEE Fifth International Conference on Bio-inspired Computing: Theories and Applications (BIC-TA 2010) Liverpool*, pp. 1331–1336, 2010.
- [27] J. Xu, Y. Huang, and Y. Liu, "Attitude optimization control of unmanned helicopter in coal mine using membrane computing," *Mathematical Problems in Engineering*, vol. 2020, no. 1, 2020.
- [28] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, 2005.
- [29] M. Paskin and S. Thrun, "Robotic mapping with polygonal random fields," 2012, arXiv preprint arXiv: 1207.1399.
- [30] T. D. Barfoot, *State estimation for robotics*, Cambridge University Press, Cambridge, UK, 2017.
- [31] H. Xu, J. Xu, and W. Xu, "Survey of 3D modeling using depth cameras," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 5, pp. 483–499, 2019.
- [32] S. Gomez-Gonzalez, G. Neumann, B. Schölkopf, and J. Peters, "Adaptation and robust learning of probabilistic movement primitives," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 366–379, 2020.
- [33] S. Brune, S. Buschow, and P. Friederichs, "The local wavelet-based organization index—quantification, localization and classification of convective organization from radar and satellite data," *Quarterly Journal of the Royal Meteorological Society*, 2021.
- [34] S. Thrun, "Probabilistic algorithms in robotics," *AI Magazine*, vol. 21, no. 4, pp. 93–93, 2000.