

# Matlab-Realized Visual A\* Path Planning Teaching Platform

<https://doi.org/10.3991/ijet.v13i10.9463>

Mingfeng Luo

Sichuan Technology & Business College, Dujiangyan, China.

277132121@qq.com

**Abstract**—Path planning in the global known environment is one of the main problems in the field of mobile robots. Based on the characteristics of A\* search method, this paper designs a simulation platform which is visible during the operation process to achieve graphicalization of the A\* algorithm search process. The simulation platform is implemented by Matlab GUI method, which provides multi-parameter setting function, and outputs the specific traversal process of the planning method in the search space into the RGB space. The results of applying simulation platform to the teaching environment shows that this platform can provide an intuitive description of the path planning method. By extracting sample data from the actual audience, the simulation platform proposed in this paper is compared with the platformless dissemination method. The experimental results show that the simulation platform given in this paper can effectively improve the dissemination effect of the conventional multimedia teaching method.

**Keywords**—path planning; a\* algorithm; teaching simulation; mobile robot

## 1 Introduction

In the field of mobile robots, mobility is the primary indicator of robot autonomy. Generally, in the state space of the robot, “moving” refers to the process in which the robot updates from the current state to the target state [1] when the constraints such as distance, time, and collision-free are satisfied. A path that satisfies the above constraints is generally referred to as a "collision-free" path [2].

In the current research, the path planning problem can be divided into two situations: static environment and dynamic environment according to the environmental model; it can also be divided into two problems: global planning and local planning according to the scale of the environment. Among them, static and global environments usually have a known environment model, namely a map model. In the current method, the modelling method of the robot environment generally includes topology map, grid map, view modelling, Voronoi diagram, etc. [3-4]. Among them, the grid map is a relatively typical modelling method in the global planning methods, and the grid data is relatively easy to process in the planning process and is easy to update. At the same time, the grid map model is also widely used for topological analysis of environment models.

Aiming at the above-mentioned environment modelling methods, the path planning methods well known in the current research generally include genetic algorithm (GA), ant colony optimization (ACO), Dijkstra search, heuristic-featured search algorithm A\* and its improved methods [5]. Among them, A\* is a search method that combines efficiency with accuracy on the basis of Dijkstra [6, 7], it is one of the representative methods of the best priority strategies and is widely used in the teaching of mobile robot courses. The general teaching process tends to be theoretical and analytical, the experimental content focuses on the realization of the method, and lacks of intuitively described teaching methods, which constraints the effect of dissemination. Aiming at this problem, this paper implements a visual simulation teaching platform based on the search characteristics of A\* algorithm, compares the influence of parameter changes on planning problems through GUI method, uses the weight diversity to describe the core idea of planning method in the RGB space, and turns the theory of planning methods into practice more intuitively. At the same time, the experimental strategy proposed in this paper can also be promoted as an objective evaluation method.

The rest pages of this paper are arranged as follows: Part 2 introduces relevant work background, including the basic principles of the A\* method and the grid modelling method; Part 3 details the design and implementation of the simulation platform; Part 4 gives the experimental results of the simulation platform; Part 5 is the conclusion.

## 2 Working background

The path planning problem of mobile robots in globally or locally known environment under static constraints has always been an important issue in the field of mobile robots or virtual reality. Dijkstra algorithm, A\* algorithm and GA are relatively mature planning methods in current research. Among them, in the 2D environment model, the A\* algorithm is more efficient in obtaining the optimal or relative optimal aspects under the action of the heuristic function, and the local optimal solution can be obtained [5]. GA has a comparative advantage in time consumption, but its accuracy depends largely on its coding strategy and adaptation function, that is, its performance depends on the better "tuning" effect, while in complex environment it is very likely that only local optimal solution can be obtained. Such problems are generally used as the main indicators to measure the characteristics of path planning methods. They are also the main purpose of the simulation teaching platform given in this paper, that is, to better understand the algorithm through experimental data.

### 2.1 Path planning problem

In a known environment  $M = \{m_i \mid i = 1, 2, \dots, n\}$ , where  $N$  represents a natural number set and  $M \subseteq N$  is a subset. Define a set of attributes corresponding to  $M$ ,  $P = \{p_i \mid i=1, 2, \dots, n\}$ , where  $p_i=0$  or  $1$ ,  $0$  represents a non-free area, and  $1$  represents a free area. It is stipulated that  $x(t)$  represents the state of the robot at time  $t$ , where "state" refers to the position of the robot in the environment, define the starting position as the

state of the robot at  $t = 0$ , when  $t = k$  the robot's state is the target position, order  $x(k) = S$ , then the path planning problem under environment  $M$  can be defined as:

1. Obtain information of  $x(t)|_{t=0}$  and  $x(t)|_{t=k}$ ;
2. Traverse  $P$  to find two  $p_i$  corresponding to  $x(t)|_{t=k=0}$  above;
3.  $f(x(t))$  is the planning function, and  $f$  outputs the  $m_i$  corresponding to the time  $k+1$ ;
4. If  $p_i = 1$  and  $m_i$  is not the end point, turn to (3), otherwise output the planning sequence.

## 2.2 Grid modelling method

According to the description of the first part, the grid method is one of the main methods used in the current path planning research for environment modelling. The simulation platform given in this paper will use the grid model to represent the robot workspace in 2D. In addition to the above features, the grid model is also limited by the grid scale. The number of grids in the map is proportional to the computational cost. In addition, the grid has a certain impact on the local smoothing of the planning results in practical applications. However, it is generally required to be processed in parallel with local motion planning. The grid maps in this paper are defined as follows:

Definition: In the grid map, any two adjacent grids (including the diagonal direction) are connected areas, that is, the "point" robot can move freely in the two non-obstacle grid areas. As described in Part 2.1,  $M$  represents a grid array of size  $u \times w$  ( $n = u \times w$ ) obtained by abstracting the robot workspace.

In general:

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,w} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,w} \\ \vdots & \vdots & \vdots & \vdots \\ m_{u,1} & m_{u,2} & \cdots & m_{u,w} \end{bmatrix}_{u \times w}$$

Where,  $m_{i,j}$  is an element in the subset of the natural number set above, and the map grid is sequentially numbered from 1 to  $u \times w$ .

Similarly, the grid attribute array  $P$  is a binary array of  $u \times w$ , which is represented as obstacle areas and non-obstacle areas on the grid, as shown in Figure 1.

In Figure 1, the black areas represent obstacle areas that robot cannot pass through, the obstacle grid attribute is represented by 0; the white areas are movable areas, and the attribute value is 1. The ultimate goal of the planning task is to find an optimal or near-optimal movable grid sequence between the starting and ending grid by distance, which is a collision-free path.

1	0	1	1	1
1	1	1	0	1
0	1	0	1	1
1	0	0	1	1
1	1	0	1	0

Fig. 1. Grid model

### 2.3 A\* planning method

In the current research, A\* algorithm is one of the representative method in heuristic search methods [8,9], and heuristic function is the core idea of this kind of algorithm. There are many improved versions of this method, such as the expansion of the neighborhood range, the combination of artificial potential fields, the ACO and other multiple methods<sup>[10,11]</sup>.

In general, in the planning process of the A\* algorithm, the search space is traversed one by one in the search, and the cost estimate of the node is calculated by a heuristic function for a current node, which is conceptually similar to the adaptive value of the genetic algorithm. Determine whether to continue the search by judging the current node attribute and the cost estimate according to the search rule. Its cost evaluation function is generally defined as:

$$f(\mu) = g(\mu) + h(\mu), \tag{1}$$

Where, the cost evaluation function  $f$  is the sum function of the actual cost function  $g$  of the node  $\mu$  and its heuristic function  $h$ ;  $g$  is generally the actual consumption from the current node to the starting node; and  $h$  is generally the minimum estimated consumption from the current node to the target position. In special cases, when  $h=0$ , the A\* algorithm is restored to the Dijkstra algorithm.

S				
	1 $\mu_4$			
		O		
			E	

Fig. 2. Evaluation function calculation

In the design process, heuristic function plays a decisive role in the rule strategy. For the path planning problem, if the actual optimal path cost for the current node to the target node is  $h'(\mu)$ , then  $h(\mu) \leq h'(\mu)$ . When solving the planning problem on the grid model, the calculation of the cost between nodes is relatively clear. Generally, it can be defined as the number of grids or the Euclidean distance between grids. The calculation method is shown in Figure 2.

In Figure 2, the starting, obstacle, target and current points are indicated by S, O, E and  $\mu$ , respectively, and the quantity is used as the metric, then  $g(\mu)=1$  and  $h(\mu)=4$ , and the evaluation of the point is obtained as 5. The search strategy of the A\* algorithm is in a neighborhood of the previous state (the default is 8 neighborhoods), traversing its neighbor nodes, and calculating the evaluation values of each node, and finally determining the search direction according to the node attributes and the minimum evaluation value. The entire algorithm is described in detail as follows:

- (1) Set  $h(\mu)$  as the Euclidean distance:

$$h = |\mu + R|, \quad (2)$$

Here,  $\mu$  and R respectively represent the current node and the reference node. When the current node is the start point S, the reference node is the end point E; according to the coordinates of the two in the grid model, the evaluation value of the start point S is calculated by the evaluation function;

- (2) Define the Table Open and Table Close, and put the start point into Table Close;
- (3) Perform a null operation on Table Open, if the table is empty, jump to (8);
- (4) Search the Table Open, find the minimum evaluation value node and load into Table Close;
- (5) Traverse the non-obstacle neighborhood of  $\mu$ , take a non-obstacle grid in the neighborhood as  $\mu'$ , if  $\mu'$  is E, jump to (8), else, calculate the evaluation value of  $\mu'$ ;
- (6) Load the above  $\mu'$  into Table Open;
- (7) Update  $\mu = \mu'$ , and turn to (3) to continue the searching;
- (8) End the planning, if the planning fails, output -1; else, the planning path is output.

### 3 Simulation platform

The goal of the simulation platform given in this paper is to provide an intuitive path planning process, and to provide preparation data for the analysis planning method through parameter changes at the same time. The GUI interface of the simulation platform is shown in Figure 3.

The GUI interface is realized by Matlab. In the design process, it offers simulation map scale selection function to analyze the impact of map scale changes on the planning. At the same time, the platform has the ability to demonstrate at different time slots, that is, planning speed customization. When the default normal planning is set, the platform directly outputs the planning result; when the planning speed is specified, the platform can intuitively output the entire planning process, which has a stronger visual effect.

Other contents are implementation of functions, including two major parts: the map start and end location designation, and the obstacle location selection and planning time output. Figure 4 depicts the use flow of the simulation platform:

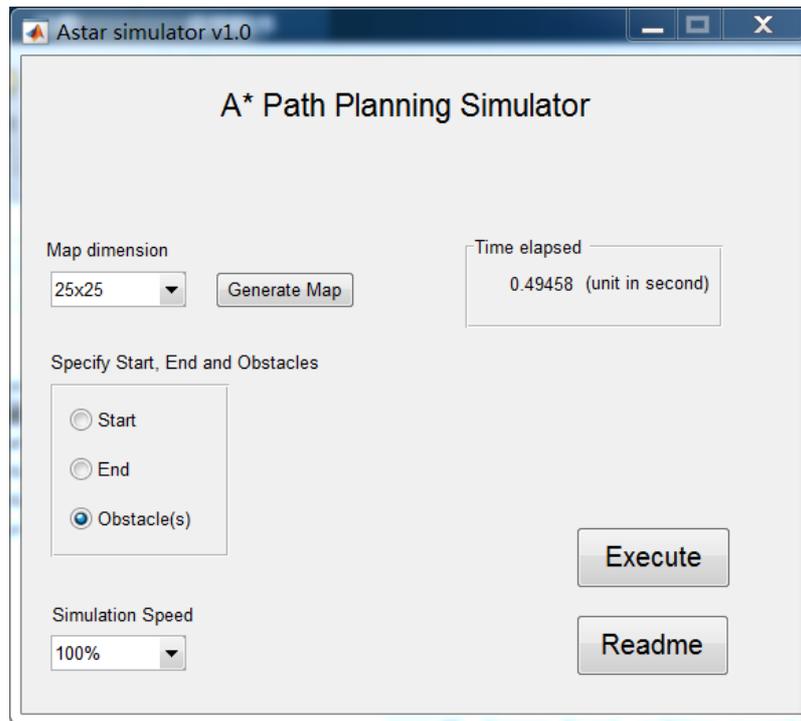


Fig. 3. Simulation platform interface

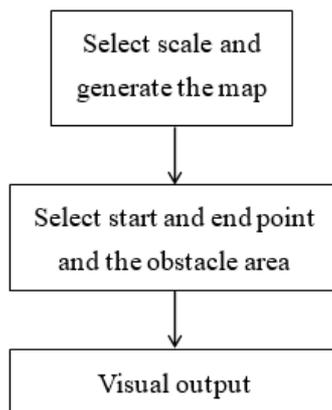
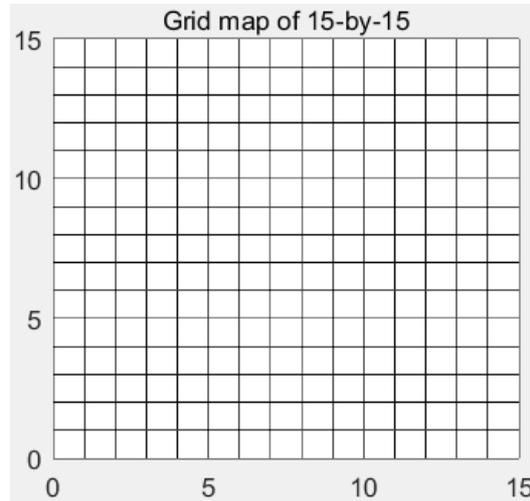


Fig. 4. Simulation platform planning process

During the simulation, a grid model of the target size is generated by the "Generate map" button, which is a blank grid area, as shown in Figure 5.

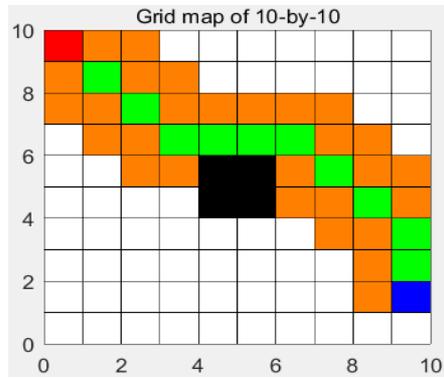


**Fig. 5.** Original grid model

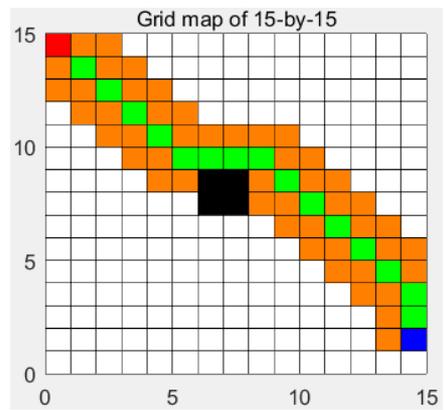
Then use the "button group" to specify the start point and target point of the planning on the grid map one by one, and select the obstacle area. If you don't need to change the simulation speed of the platform, you can directly trigger the "Execute" button to perform simulation. After the end of the planning, the time consumption is given at the top right of the simulation interface.

## 4 Experiment

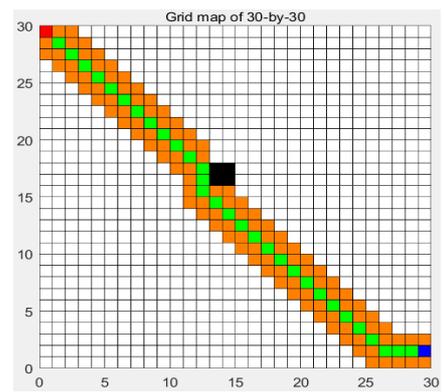
The experimental hardware environment is an Intel i5 4590 3.30GHz desktop PC with 4GB RAM; the hardware is equipped with Windows 7 64-bit operating system, and the Matlab version is R2015b 32-bit. The experimental process consists of two parts. In the first part, the planning comparison is made by constructing maps of different scales to analyze the time consumption changes brought by different environment modelling size. In this part, the grid models of  $10 \times 10$ ,  $15 \times 15$  and  $30 \times 30$  are selected successively, and obstacle areas of same numbers are deployed at the same time. The planning results are shown in Figure 6.



(a) 1010 sized grid model



(b) 1515 sized grid model



(c) 3030 sized grid model

**Fig. 6.** Grid planning results of different sizes

In the results given in Figure 6, the red, blue, and black grids respectively represent the start point, target point, and obstacle area of the planning, the orange areas represent the search area of the A\* algorithm during the planning process, and the green sequence is the final result of the planning. The three maps of this part of the experiment contain four obstacle areas located at the centre of the grid, only to maintain the consistency of the number of obstacle grids, in order to observe the changes in planning time consumption, the specific results are shown in Table 1.

**Table 1.** Planning time

Size	Number of obstacles	Planning time (unit: s)
10×10	4	0.18627
15×15	4	0.22879
30×30	4	0.32287

The planning time consumption given in Table 1 is in seconds so as to give a clearer comparison, after scale of 15, this paper directly gives a grid model of 2 times the scale. It can be seen that in the case of the same number of obstacles, the grid size is linear with the planning time. In the experiment, when the grid scale is 20 and 25, the planning time is 0.20756s and 0.29647s, respectively.

Furthermore, under the conditions of same map scale, the planning results after the change of the obstacle area are compared. In the experiment, a 15×15 model was taken as an example to test two maps with different obstacle numbers of 59 and 77. The results are shown in Figure 7.

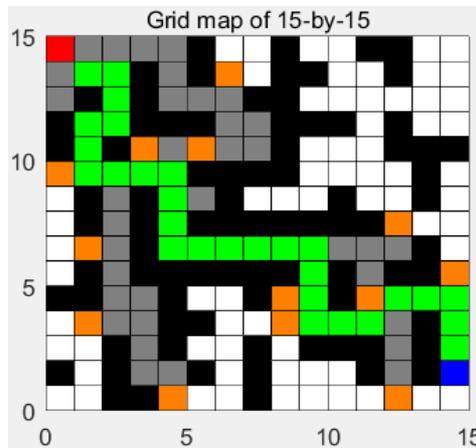
In this section, the search results are grayed out to indicate the search process, and the others are the same. The planning time consumption on the two maps is 0.26524s and 0.46427s, respectively, which intuitively gives the influence of the obstacle scale on the calculation time.

For the above platform, we selected a class with four groups of 80 people each as samples for testing, each two groups were taught on a non-simulation platform and on a simulation platform, respectively. The entire teaching process takes approximately 90 minutes, within approximately 35 minutes the testing process is focusing on the core idea of the A\* planning method. In this paper, the continuous attention of the audience [12, 13] is selected as the metric. The duration of continuous attention of college students is about 30~50 minutes. The time slot is relatively long and it is not easy to be quantified. In this case, the effect of the simulation platform is evaluated by observing the duration of the individual steady state of the samples during the experiment, details are as follows:

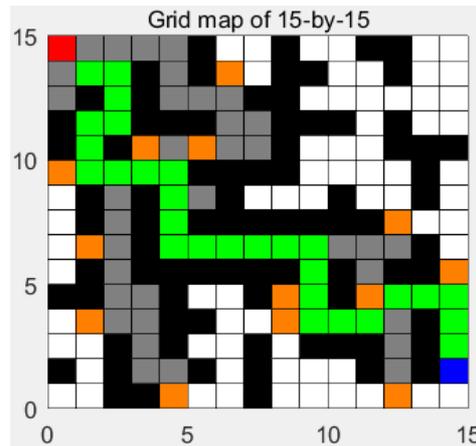
Within about 35 minutes of the dissemination of the A\* planning method, the face recognition system is used to detect the sample individuals in the field of view (the default is all individuals in each group). According to the physical conditions of the human body, only the number of individuals with no head movement for 50s and above is counted during the observation process, the sample results are shown in Figure 8.

In Figure 8, the four sets of data are labelled according to different line types. Sim A and Sim B represent two sets of data with simulation platform. The other two groups

started with “NoSim” represent that the simulation platform is not used during the dissemination process. During the 90 minutes course, extract about a 35 minutes time slot in which the planning method is taught, the change in the duration of the head movement is relatively significant. Considering that the face recognition system generates certain noise during the detection of the action, for example, it can’t determine whether a continuous action is looking down or looking up. However, two results can still be found: one is that the data with the simulation platform is significantly higher than the sample observation without the simulation platform, second, in the small time slot of the given planning result, there are more samples with stable state in the case of the simulation platform, which verifies the intuitive significance of the simulation platform.



(a) Number of obstacles: 59



(b) Number of obstacles: 77

Fig. 7. Planning results for obstacle areas with different scales

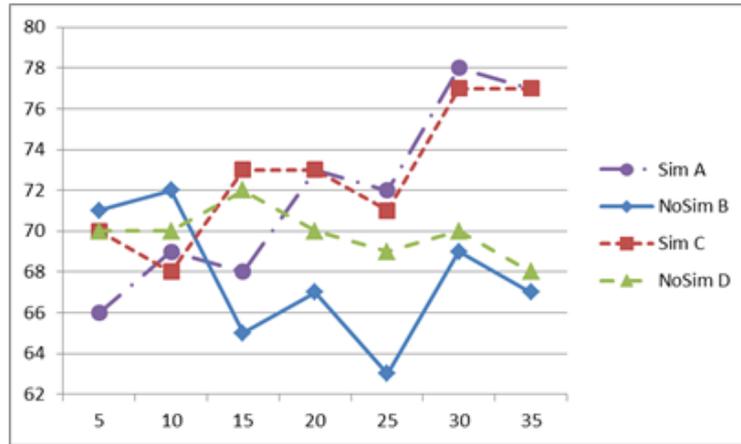


Fig. 8. Teaching test results

## 5 Conclusion

In this paper, the visual and parameter-adjustable A\* path planning simulation platform is realized by GUI method. By visualizing the planning process and planning results, the intuitive experience of the knowledge dissemination process is further promoted. In terms of performance evaluation, by putting the simulation platform into practical application, comparing the observed experimental data with the data of the non-simulation platform, it is found that the visual simulation platform given in this paper has a more significant incentive effect and has more obvious dissemination effect.

In the next stage of work, this paper intends to conduct a comparative study on some improvement methods and other planning methods of A\*, further upgrade and expand the simulation platform, and improve the diversity of data and theory of the simulation platform. On the other hand, we tend to extend the simulation platform for the application scenarios and provide functions such as path planning and multi-agent planning simulation in a dynamic environment.

## 6 References

- [1] Fleckenstein, F., Dornhege C., Burgard, W. (2017). Efficient path planning for mobile robots with adjustable wheel positions, IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2454-2460. <https://doi.org/10.1109/ICRA.2017.7989286>
- [2] Hu, Y., Yang, S.X. (2004). A knowledge based genetic algorithm for path planning of a mobile robot. Robotics and Automation, Proceedings. ICRA'04. 2004 IEEE International Conference on. 5, IEEE, 2004.
- [3] Garrido, S., Moreno, L., Abderrahim, M., Martin, F. (2011). Path planning for mobile robot navigation using voronoi diagram and fast marching. Int. J. Robot. Autom 2(1), 42-64.
- [4] Hachour, O. (2008). Path planning of Autonomous Mobile robot. International journal of systems applications, engineering & development 2(4), 178-190.

- [5] Soltani, A.R., Tawfik, H., Goulermas, J.Y., Fernando, T. (2002). Path planning in construction sites: performance evaluation of the Dijkstra, A\*, and GA search algorithms. *Advanced engineering informatics* 16(4), 291-303.. [https://doi.org/10.1016/S1474-0346\(03\)00018-1](https://doi.org/10.1016/S1474-0346(03)00018-1)
- [6] Solka, J.L., Perry, J.C., Poellinger, B.R., Rogers, G.W. (1995). Fast computation of optimal paths using a parallel Dijkstra algorithm with embedded constraints. *Neuro-computing* 8(2), 195-212. [https://doi.org/10.1016/0925-2312\(94\)00018-N](https://doi.org/10.1016/0925-2312(94)00018-N)
- [7] Hart, P.E., Nils N.J., Raphael, B. (2007). A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics* 4(2), 100-107. <https://doi.org/10.1109/TSSC.1968.300136>
- [8] Edelkamp, S., Schroedl, S. (2011). *Heuristic search: theory and applications*. Elsevier.
- [9] Szer, D., Charpillat, F., Zilberstein, S. (2012). MAA\*: A heuristic search algorithm for solving decentralized POMDPs. arXiv preprint arXiv:1207.1359.
- [10] Sfeir, J., Saad, M., Saliah-Hassane, H. (2011). An improved artificial potential field approach to real-time mobile robot path planning in an unknown environment." *Robotic and Sensors Environments (ROSE), 2011 IEEE International Symposium on*. IEEE, 2011. <https://doi.org/10.1109/ROSE.2011.6058518>
- [11] Raja, P., Pugazhenth, S. (2012). Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences* 7(9), 1314-1320. <https://doi.org/10.5897/IJPS11.1745>
- [12] Wei, F.Y.F., Wang, Y. K., & Klausner, M. (2012). Rethinking college students' self-regulation and sustained attention: Does text messaging during class influence cognitive learning?. *Communication Education*, 61(3), 185-204. <https://doi.org/10.1080/03634523.2012.672755>
- [13] Chen, C.M., Wu, C.H. (2015). Effects of different video lecture types on sustained attention, emotion, cognitive load, and learning performance. *Computers & Education* 80, 108-121. <https://doi.org/10.1016/j.compedu.2014.08.015>

## 7 Author

**Mingfeng Luo** is an Associate Professor at the Mechanical and Electrical Department, Sichuan Technology & Business College, Dujiangyan 611830, China. Her current research interests include electronic technology, electrical automation and mechatronics.

Article submitted 16 July 2018. Resubmitted 23 August 2018. Final acceptance 05 August 2018. Final version published as submitted by the author.