

An Approach To Safe Path Planning For Mobile Robot In The Dynamic Environment Based On Compact Maps

LanFeng Zhou

Shanghai Institute of Technology, 201418, Shanghai,China

Email: lfzhouhit@126.com

Jian Jiang

Shanghai Institute of Technology, 201418, Shanghai,China

Email: jiangjianhit@sina.com

Abstract—For path planning of the mobile robot in dynamic environment, this paper proposed an approach to safe path planning based on compact maps. This algorithm utilizes an efficient stream decimation algorithm of massive mesh to build compact maps of the environment models. And it uses a knowledge based genetic algorithm to find the global optimal safe path. The fitness of the algorithm is linear with gap, and has less computation demand. The proposed knowledge based genetic algorithm incorporates the domain knowledge into its initial population and adjusts adaptively the parameter, which increases greatly the efficiency of the algorithm, in order to plan safe path of the mobile robot in the dynamic environment. Various simulation results show the effectiveness and efficiency of the presented method.

Index Terms—compact maps, mobile robot, safe path planning, genetic algorithm, dynamic environment

I. INTRODUCTION

Path planning in dynamic environment is an important research subject of mobile robot. Artificial potential field(APF), Artificial neural network(ANN) and fuzzy logic are commonly used [1-3]. APF, which was proposed by Khatib, is quite simple in theory, but it's easy to bring local minimum problem and failing to find path in the narrow channel, as well as other problems, for example, the result may vary anomaly when robot is near obstacles^[1]. ANN is based on guided learning, so it's learning speed is very slow in unstructured environment^[2]. The main limitation of fuzzy logic method is in a complex environment, making fuzzy rule is difficult, in the mean time, adjustment of fuzzy rules relying on experience of human is very time-consuming^[3]. The path of security is a dynamic environment of mobile robot navigation is another important issue [4-5]. Another important subject in mobile robot navigation in dynamic environment is safety[4-5]. Whether it is safe can be measured by the gap between path and obstacles. When the gap between path line and all the obstacles are greater than a certain limit (space we want), then we consider the

path is safe. Most of the methods tried to expand the size of obstacles to ensure the safety of mobile robot path planning, the consequences is some of good path is cut off. APF can carry out safe path planning, but it did not conduct a global search, thus the final path is neither the shortest nor the safest. A genetic algorithm based safe path planning method in a static environment is stated in literature [4], and a genetic algorithm based safe path planning method in a dynamic environment is stated in literature [5], but it is a more complicated calculation. Modeling terrain techniques have been more and more compactly integrated with robotics. Environment models are useful in mobile robot mapping, location and navigation. But, the amount of data contained in these models is too huge to deal with. Some models are even too large to fit in main memory, and can not fulfill the requirement of real-time computing. These models would also cause inconvenience in rendering and transmitting. Therefore, building compact maps of the environment models has become an indispensable research topic^[6]. This paper presents a strategy of safe path planning for mobile robot in dynamic environments based on compact maps. Knowledge-based genetic algorithm is employed to find global optimization of safety path; in the algorithm, the relationship between adapt values and gap is linear, the calculation is relatively simple at the same time. Incorporating of domain knowledge into the initial population and the adaptive adjustment of control parameters greatly improved the efficiency of genetic algorithm, a safe path planning in dynamic environments is achieved. The simulation results show the correctness and validity of this method.

II. QUESTION REPRESENTATIONS

In an environment with obstacles, we are required to make the mobile robot move from a starting point to a target without any collision. Robot's work environment is identified by the number of raster, each number represents some environment point in one location.

Robot's working environment and path are shown in Figure 1, in figure 1, S represents the starting point, T is the target point. A solid line from S to T is on of the feasible path and the dashed lines are not a feasible path. Feasible path is not necessarily a safe path, but a safe path must be a feasible path.

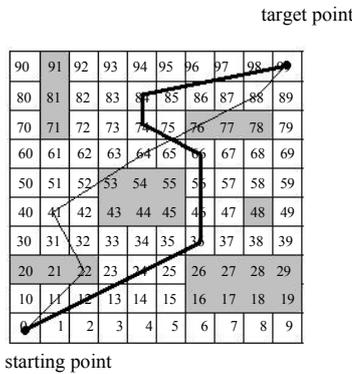


Figure 1 Robot's work environment and path

III. MODELING COMPACT MAPS

3D modeling techniques have been more and more compactly integrated with robotics. 3-D environment models are useful in mobile robot mapping, location and navigation. They could help to locate a mobile robot fast and accurately, and to navigate it while it travels in the environment. The accuracy of models acquired with range finder is typically higher than that of models acquired with stereo vision. And vision sensors usually can not process vision data in real-time. Therefore, laser range finders, mounted on a mobile robot, can be chosen for acquiring 3D information-referred to as range image-about unknown and rugged models of the environment.

As the techniques of 3D scanning develop, however, the amount of data contained in these models is too huge to deal with. Some models are even too large to fit in main memory, and can not fulfill the requirement of real-time computing. These models would also cause inconvenience in rendering and transmitting. Therefore, building compact 3D maps of the environment models has become an important research topic.

This paper uses an algorithm for generating compact models of outdoor environments in real-time from laser range-finder. However, Due to the advances in 3D scanning technology and numerical simulation, the complexity of geometric models is increasing much faster than the graphics and computing performance of current PC hardware. This is why mesh decimation has been an active research area over the past ten years.

While the first generation of mesh decimation algorithms are based on the assumption that the complete geometry data fits into a global in-core data structure, the second generation algorithms relax this restriction by reducing the size of the active working set and storing the major part of the data externally on disk. For such algorithms, the actual in-core memory requirements are

independent from the size of the input or output mesh but the data in external memory usually has to be accessed several times which slows down the process significantly. Therefore, Our algorithm is mainly used to simplify meshes exceeding the size of main memory. The decimation algorithms do not load the whole model geometry into the actual in-core memory, but temporarily store the large parts on disk, known as the out-of-core simplification algorithm. Therefore the memory requirement of the methods is independent of the complexity of the input as well as output models.

A. Quadric error metric

In order to select a contraction to perform during a given iteration, we need some notion of the cost of a contraction. To define this cost, we attempt to characterize the error at each vertex. To do this, we associate a symmetric 3×3 matrix \mathbf{Q} with each vertex, and we define the error at vertex $\mathbf{v}=[x, y, z]^T$ to be the quadratic form

$$\mathbf{Q}(\mathbf{v}) = \mathbf{v}^T \mathbf{A} \mathbf{v} + 2\mathbf{b}^T \mathbf{v} + c \tag{1}$$

where $\mathbf{A} = \mathbf{nn}^T = \begin{bmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{bmatrix}$,

$\mathbf{n}=[a \ b \ c]^T$ is a unit normal (i.e., $a^2+b^2+c^2=1$) and $\mathbf{b}=\mathbf{dn}$ is a unit vector, and d and c are two scalar constants. For a given quadric \mathbf{Q} , we can try to find the point \mathbf{v}_0 such that $\mathbf{Q}(\mathbf{v}_0)$ is minimal. Since $\mathbf{Q}(\mathbf{v}_0)$ is quadratic, finding its minimum is a linear problem; the minimum occurs where

$$\frac{\partial \mathbf{Q}(\mathbf{v})}{\partial x} = \frac{\partial \mathbf{Q}(\mathbf{v})}{\partial y} = \frac{\partial \mathbf{Q}(\mathbf{v})}{\partial z} = 0$$

Taking partial derivatives, we see that the gradient of \mathbf{Q} is $\nabla \mathbf{Q}(\mathbf{v}) = 2\mathbf{A}\mathbf{v} + 2\mathbf{b}$ (2)

Solving for $\nabla \mathbf{Q}(\mathbf{v})=0$, we find that the optimal position is $\mathbf{v}_0 = -\mathbf{A}^{-1}\mathbf{b}$ (3)

and its error is

$$\mathbf{Q}(\mathbf{v}_0) = \mathbf{b}^T \mathbf{v}_0 + c = -\mathbf{b}^T \mathbf{A}^{-1} \mathbf{b} + c \tag{4}$$

If the matrix \mathbf{A} is not invertible, we fall back on choosing \mathbf{v}_0 from amongst the endpoints and the midpoint.

B. Modeling compact map algorithm

The stream algorithm for mesh decimation does not require to permanently store the data at all (not even on disk). The concept of stream processing is that input data are taken in sequentially without backtracking. The output data is written sequentially as well and no feedback to the input stream is possible. The amount of in-core memory that is allocated by a stream algorithm does not depend on the amount of data to be processed. Since stream algorithms process the data in one single pass, we can implement a software system where the geometry generating pre-process feeds its output directly into the decimation post-process without storing it to disk.

When decimating a stream of geometry data, the mesh is logically split into three parts: the unread postfix of the input stream representing the unprocessed part of the input mesh, the in-core portion of the mesh that is

currently processed by the decimation algorithm, and the written prefix of the output stream representing the part of the output mesh that has already been decimated to the target resolution(see Fig. 2).

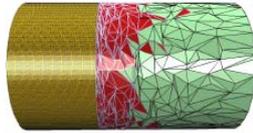


Figure 2The snapshot of the stream decimation to a cylinder mesh shows the yet unprocessed part of the input data(left), the current in-core portion(middle) and the already decimated output(right).The data in the original file happened to be pre-sorted from right to left.

Let N_{max} be the maximum number of triangles that fit into the in-core triangle buffer. The stream algorithm is to perform three different operations that affect the filling level $N_{current} \leq N_{max}$ of that buffer.

The modeling compact maps 's basic step is as follows:

Step 1: PRE-PROCESSING Tag all the boundary triangles of the input mesh.

Step 2: READ(k) takes the next k triangles from the input stream and inserts them into the current in-core portion (which maintains both geometry and topology) of the mesh. $N_{current} \leftarrow N_{current} + k$

Step 3: DECIMATE(k) performs k edge collapse operation on the in-core portion of the mesh according to the multiple choice optimization strategy. Each edge collapse removes two triangles from the mesh.

$N_{current} \leftarrow N_{current} - 2k$

Step 4: WRITE(k) removes k triangles from the in-core portion and writes them into the output stream. $N_{current} \leftarrow N_{current} - k$

While the algorithm is running, these four operations are applied in arbitrary order. The only hard restriction is that the buffer must not overflow which implies that if $N_{current} = N_{max}$, we have to apply DECIMATE or WRITE before reading the next input triangle. Some additional weak restrictions are that the filling level should be kept as high as possible to provide a sufficiently large set of candidates for the multiple choice optimization and that the number of DECIMATE and WRITE operations should be balanced such that the output mesh has the prescribed target resolution.

For a real implementation of this ideal algorithm, we have to make a few modifications in order to make the algorithm run effectively. In the introduction we explained that while a stream algorithm is running, the data is split into three disjoint parts (A: postfix of the input stream, B: in-core portion, and C: prefix of the output stream).In the case of mesh decimation these three parts are submeshes of the original data at different resolution levels.

In order to maintain the global mesh consistency, the two boundary polygons P_{ab} and P_{bc} which represent the interfaces between the parts A/B and B/C respectively, must not be modified by the DECIMATE operations. For

P_{ab} this is necessary since no vertex should be removed before all its neighbors are read from the input stream. For P_{bc} this restriction guarantees that the output mesh has a consistent triangle neighborhood relation, i.e. the neighborhood of a triangle is not further decimated after it has been written to the output stream. In our implementation the operators DECIMATE and WRITE take care of this consistency checking. Notice that, depending on the geometric shape of the input mesh, the boundary polygons P_{ab} and P_{bc} usually consist of many connected components.

Another issue is that with a pure random decision whether a DECIMATE or a WRITE operation is performed next, it can happen that a very small (high resolution) triangle is accidentally written to the output stream. This typically leads to strong variations in the vertex densities of the output mesh. In a real implementation we therefore have to make sure that the WRITE operation is only executed when a sufficient number of coarse triangles is present in the in-core buffer, i.e. when a sufficient number of decimation steps has been performed before.

Finally in a real operating system the file and stream access is usually much faster if we read and write the data in larger blocks. Hence we do not want to apply the READ and WRITE operation one by one but rather in larger chunks.

The major difference between the ideal and the real stream decimation algorithm is that the decision whether to DECIMATE or to WRITE is no longer random. Only the decision which candidate to DECIMATE or to WRITE still remains a random process. The blockwise READ and WRITE is achieved by introducing a "hysteresis" which means that we are not reading any data before the filling level $N_{current}$ is down to 50%. However, whenever we actually perform a READ operation we read in the maximum number of triangles $N_{max} - N_{current}$ to raise the filling level to 100%.

The detailed comments refer to the literature[6].

VI. KNOWLEDGE BASED GENETIC ALGORITHM

A. Coding

Take the sequence number as the of path coding identity. Provide the identification serial number in each path can not be duplicated. In the case of Figure 1, a path can be expressed as 0-24-36-66-74-84-99, where 0 is the starting point, 99 as the target point, and 36,66,74 and 84 for the intermediate nodes.

B. The generation of initial population

The initial production process of the robot path is as follows: from a starting point, and by using local search technology, randomly selected a non-obstacle point adjacent to the starting point as the next path point, so repeatedly, until you find the end point. Variable length chromosomes is used.

C. Determination of fitness

This article improve the adaptation value method literature [6] presented, so as to fit in calculating the security pass for mobile robot. Set individual adaptation value of p_n in initial population is $f(p_n)$. The fitness $f(p_n)$

can be expressed as $f(p_n) = \sum_{i=1}^N [d_i + w_c c_i - w_s c_s]$, where d_i denotes distance between points i and j , c_i denotes the shortest distance between i -lines segments and all the obstacles, c_s denotes the gap we need, N denotes the number of Lines in path p_n , w_c denotes the gap weight coefficient, and w_s denotes coefficient of the of depth of the intersected obstacles, which is defined as follows:

$$w_s = \begin{cases} 1 & \text{if } c_i \geq c_s \\ c & \text{if } c_i < c_s \\ c + \sum_{j=1}^M a_j & \text{if segment intersect with obstacles} \end{cases}$$

where M denotes the number of line segments intersect with obstacles, a_j denotes the shortest distance between line segments and the obstacle, c denotes the maximum value of the constant higher than c_i .

D. Genetic operators

Take reference [7] as an example, this paper designs the six genetic operators, they are not only merged into domain knowledge, but also combined with local search technology, hence greatly enhanced the efficiency of genetic operator, as shown in figure 3.

crossover operator: Randomly select two nodes from parent individuals, then exchange them. as shown in figure3 (a).

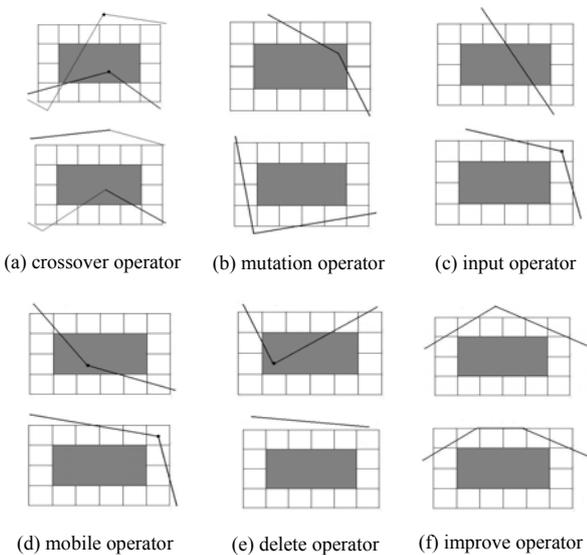


Figure 3 6 genetic operators

mutation operator: Randomly select one node, replace it with a non-path node. Mutation operator plays a key role to increase population diversity, therefore, It is not necessary that individual point after the variation is better than before, as shown in figure 3 (b).

input operator: Using local search technology to choose the best node adjacent to the barrier, then insert it into the infeasible line segment between two nodes, as shown in figure 3 (c).

mobile operator: The operator can be applied to a feasible path and non-feasible path. Randomly select a node using local search technology, move it to the best grid in the adjacent areas, as shown in figure 3 (d).

delete operator: The operator can be applied to a feasible path and non-feasible path. Randomly select a node, check its two neighboring nodes and connect them, if deleting the selected node is useful, then remove it, as shown in figure3 (e).

improve operator: The operator can be applied to a feasible path. Select a node from the path, insert a new node, respectively, from the two line segments connected with the node, connect the two new nodes, if this path is feasible, then delete the selected node, as shown in figure 3 (f).

E. Adaptive parameter adjustment

The relationship between the average population fitness value and the best individual fitness value reflect the stability of the average fitness[8]. Let f_{min} denotes the optimal fitness value in the population, f_{mean} denotes the average fitness value in the population, then $d = f_{mean} - f_{min}$. Smaller the value of d , smaller the difference between the individuals' fitness value in the population, indicating that this population has a greater chance to achieve local optimum, and a greater chance of premature convergence; On the contrary, bigger the value of d , bigger the difference between individual dispersion characteristics and its fitness value in the population, indicating that this population has a greater fitness differences. We can use the difference between the average fitness value and the best fitness value to reflect the degree of convergence in the population. In this way, This crossover probability p_c and mutation probability p_m are determined by $f_{mean} - f_{min}$. Indeed the value of p_m, p_c is inversely proportional with $f_{mean} - f_{min}$, they can be expressed as follows:

$$p_c = k_1 / (f_{mean} - f_{min}) \quad (5)$$

$$p_m = k_2 / (f_{mean} - f_{min}) \quad (6)$$

where k_1 and k_2 denotes the scale factor. In the meanwhile, in order to protect the best individual in the same generation, p_m and p_c should be different accordingly. To protect those have lower fitness value, p_m and p_c should be reduced. And for those who have higher fitness value, p_m and p_c should be increased. So we can see p_m and p_c is not only related to $f_{mean} - f_{min}$, but also related to $f_{min} - f'$ and $f' - f_{min}$, where f denotes the fitness value of individual variation, f' denotes the smaller one in those two cross-fitness individuals. They can be expressed as follows:

$$p_c = k_1 (f' - f_{min}) / (f_{mean} - f_{min}), f' \leq f_{mean} \quad (7)$$

$$p_c = k_3, f' > f_{mean} \quad (8)$$

$$p_m = k_2 (f - f_{min}) / (f_{mean} - f_{min}), f \leq f_{mean} \quad (9)$$

$$p_m = k_4, f > f_{mean} \quad (10)$$

where k_1, k_2, k_3, k_4 are all constants, and their values are all satisfy equation $0 \leq k_1, k_2, k_3, k_4 \leq 1.0$. In this algorithm, $k_3 = 0.75$ and $k_4 = 0.5$.

In addition, you can choose the larger probability of 0.9 in the early age of evolution for the other proposed operator, because there is a bigger chance of average

fitness value then, in the medium-term of the evolution, you can choose a moderate probability of 0.5, in the late term of the evolution, you can choose a smaller probability of 0.1 for the population is apt to be more convergent, in this way, you may achieve the design of operator.

F. The proposed path planning method (here in after referred to KBGA Law)

The following is specific steps of path planning based on genetic algorithm presented in this paper:

Step1. Environment modeling. Do environmental modeling using grid method proposed by Simon, and Yang.

Step2. Coding and initialization. To encode possible paths, set the counter of evolution algebra $t \leftarrow 0$; set the maximum evolution algebra T ; generated n individuals randomly as initial population $P(0)$.

Step3. Individual evaluation. Using the equation

$$f(p_n) = \sum_{i=1}^N [d_i + w_c \cdot c_i - w_s \cdot c_s]$$

to calculate each individual's fitness value in population $P(t)$.

Step4. Determine whether the environment changes. If the environment changes, then run step 5. Otherwise, go to step 6.

Step5. Re-evaluation for population $P(t)$.

Step6. Tournament selection and elitist selection. To retain the best individual in parent population, run tournament selection in parent groups.

Step7. Crossover operation. Use the single-point adaptive crossover operator on the group.

Step8. Mutation operation. Use the adaptive mutation operator on the group.

Step9. Other operations. Use other operator on group $P(t)$. After these sessions of selections, we finally get the next generation $P(t+1)$.

Step10. To determine whether it is in the dynamic environment. If it is, then go to step 4. Otherwise, run step 11.

Step11. To check if the final condition is meet If $t \leq T$, then $t \leftarrow t+1$ is right, go to step 3; if $t > T$, we can consider the minimum fitness value of the individual as the optimal solution, output it and stop computing.

V. SIMULATION EXPERIMENT

To demonstrate the effectiveness of the algorithm, we have done the simulation test on PC, the main experimental parameter are set as follows: the environment is a 32×32 grid, population size $M = 40$, $k1 = 0.8$, $k2 = 0.05$, the maximum evolution algebra $T = 500$, $cs = 1$, $c = 10$.

Figure 4 gives simulation results of safe path planning using our method in static environment.

From figure 4, we can know that the planning path using our method in static environment is not only globally optimized, but also safe.

From figure 5, we can conclude that safe path planning using our method can be achieved in dynamic environment.

Knowledge-based genetic algorithm is employed to find global optimization of path based on compact maps, which not only makes the calculation relatively simple, but also decreases greatly memory consumption. Therefore, the efficiency of our algorithm is obviously improved.

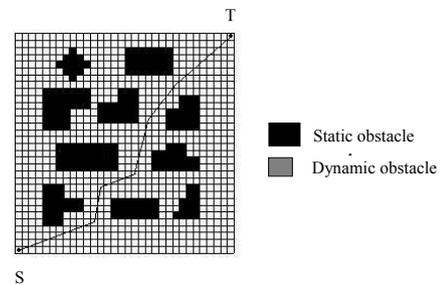


Figure4 simulation results under static environment

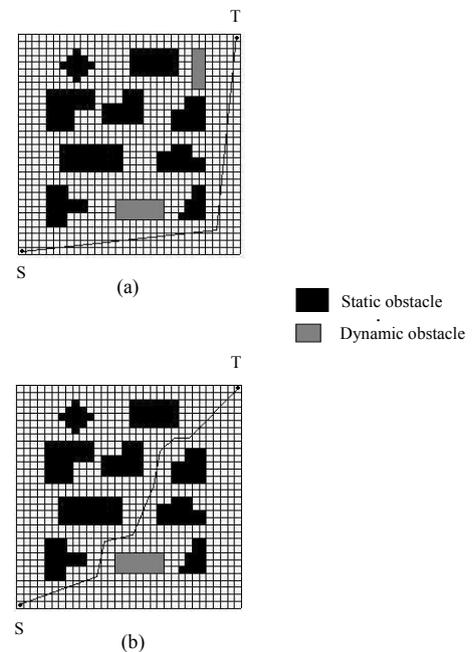


Figure5 simulation results under dynamic environment

VI. CONCLUSION

The safe navigation of mobile robot in dynamic environment is a very important issue. In this paper, a safe path planning method in dynamic environment based on compact maps is achieved. It use an efficient stream decimation algorithm of massive mesh to build compact maps of the environment models. And it employs knowledge-based genetic algorithm to find the global optimization of safety path; the fitness value of the algorithm is linear with the gap to obstacles, the process of calculating is relatively simple and efficiency is high, and it is feasible in both static environment and dynamic environment.

Although stated here in the context of finding rectangular flat surfaces, the algorithm is more general in

that it can easily handle a much richer variety of geometric shapes. The extension of our approach to richer classes of objects is subject to future research. Another topic of future research is to augment the algorithm to estimate the robot's location during mapping.

ACKNOWLEDGMENT

This work was supported by Project ZR2009GM020 of the National Science Foundation for Shandong Province and the Shanghai Institute of Technology Foundation of China(YJ2009-37).

REFERENCES

- [1] O.Khatib, J.Burdick. Motion and force control of robot manipulators. Proc. IEEE Robotics and Automation, 1986, 1381-1386.
- [2] P. Pal, A.Kar. Mobile robot navigation using a neural network[A]. Proceedings of the IEEE International Conference on Robotics and Automation, 1995, 1503--1508.
- [3] P. G. Zavrangas, S. G. Tzafestas. Industrial Robot Navigation and Obstacle Avoidance Employing Fuzzy Logic. Journal of Intelligent and Robotic Systems. 2000,27(1):23-30
- [4] C. Mingwu. Safety Considerations in the Optimisation of Paths for Mobile Robots Using Genetic Algorithms. Genetic Algorithms in Engineering Systems: Innovations and Applications 12-14 September 1995, Conference Publication No. 414, O IEE, 1995
- [5] Elshamli, Ahmed. Genetic algorithm for dynamic path planning. Canadian Conference on Electrical and Computer Engineering, v 2, Canadian Conference on Electrical and Computer Engineering; Technology Driving Innovation, 2004, 0677-0680
- [6] L.F. Zhou, Hong Bingrong. Real-Time Acquisition of

Compact Volumetric Maps. HIGH TECHNOLOGY LETTERS. 2006,12(1): 55~57

- [7] Y. Hu, S. X. Yang et al. A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot. Proc. of the 2004 IEEE Intern. Conference on Robotics & Automation, New Orleans, 2004: 4350-4355.
- [8] Y. H. Zhou, Y.C. Lu, C.Y. Shi. Adaptive and parallel genetic algorithm based on solving premature convergence. JOURNAL OF TSINGHUA UNIVERSITY SCIENCE AND TECHNOLOGY, 1998, 38(3): 93-95. (in Chinese)



LanFeng Zhou JiangSu Province, China. Birthdate: January, 1966. is a teacher in Deptment of Computer Science and Information Engineering in Shanghai Institute of Technology, graduated from Dept. Computer Application Technology of Harbin Institute of Technology. And research interests on intelligent robot, autonomous navigation and virtual

reality.



Jian Jiang Zhejiang Province, China. Birthdate: July, 1976. is a teacher in Deptment of Mechanical Engineering in Shanghai Institute of Technology, graduated from Dept. Mechanical and Electronic Engineering of Harbin Institute of Technology. And research interests on multirobot coordination and intelligent control.