

Path Planning for Mobile Robot using 4EGSOR via Nine-Point Laplacian (4EGSOR9L) Iterative Method

Azali Saudi

School of Engineering and Information Technology
University of Sabah, Kota Kinabalu, Malaysia

Jumat Sulaiman

School of Science and Technology
University of Sabah, Kota Kinabalu, Malaysia

ABSTRACT

This paper presents an attempt to solve path planning problem for a mobile robot operating in indoor environment model using iterative numerical technique. It is based on the use of Laplace's Equation to compute the potential functions in the environment grid model of the robot. The proposed block iterative method, better known as Four Point-Explicit Group via Nine-Point Laplacian (4EGSOR9L), employs a finite-difference scheme to compute the potential functions to be used in generating smooth path between start and goal points. The simulation results demonstrate that the proposed 4EGSOR9L method performs faster than the previous methods in computing the potential functions of the environment model.

Keywords

Robot path planning; Four-Point Explicit Group SOR via Nine-Point Laplacian (4EG9LSOR); Laplace's equation.

1. INTRODUCTION

Path planning problem for a mobile robot occurs in various industrial application scenarios. In industrial automation, the robot is often required capable of moving quickly to its home position from a given initial point while avoiding possible collisions with the environment. It is therefore not surprising that the research activities in this field have been steadily increasing over the last two decades.

In this paper, we present our work on implementing mobile robot path planning via numerical potential function in configuration space based on the theory of heat transfer. The heat transfer problem is modeled with Laplace's equation. Solutions of Laplace's equation are called harmonic functions, which consequently represent temperature values in the configuration space to be used for simulation of path generation. Various approaches had been used to obtain harmonic functions, but the most common method is via numerical techniques due to the availability of fast processing machine and their elegant and efficiency in solving the problem. In this work, several experiments were conducted to study the performance of using 4EG9SORL iterative method for generating mobile robot path in several sizes of environment with varying number of obstacles.

2. RELATED WORK

Pioneer work by Khatib [1] utilized the use of potential functions for robot path planning, in which every obstacle produces a repelling force and the goal exerts an attractive force. Meanwhile, Koditschek [2] concludes that geometrically at least in certain types of domains, the potential functions can be used to guide the effectors from almost any point to a given point. These potential field

methods, however, suffer from the generation of local minima.

Connolly *et al.* [3] and Akishita *et al.* [4], both of them developed independently a global method that generates smooth path by using solutions to Laplace's equations, where the potential fields are computed in a global manner over the entire region. In [5], Connolly & Gruppen show that harmonic functions have a number of properties useful in robotics. Then, Sasaki [6] computed harmonic functions using numerical technique to be used for mobile robot navigation. Waydo & Murray [7] used stream functions that are similar to harmonic functions to generate motion planning for a vehicle. Daily & Bevely [8] used harmonic potential field for path planning of high speed vehicles. Garrido *et al.* [9] used finite elements to compute harmonic functions for robotic motion. More recently, Szulczyński *et al.* [10] employed harmonic potential functions for real-time obstacle avoidance.

3. HEAT TRANSFER MODEL

Mobile robot path planning problem can be modeled as a well-known steady-state heat transfer problem, where heat sources come from the boundaries and the heat sink will pull the heat in. This heat conduction process produces a temperature distribution and the heat flux lines that are flowing to the sink fill the workspace. In a mobile robot environment model setup, the goal point is treated as a heat sink whilst the boundary walls and obstacles are considered as heat sources that are fixed with constant temperature values. Once the temperature distribution in the field is obtained, it will be used as a guide to generate path for mobile robot to move from the start point to the goal point. The idea is to follow the heat flux that will flow from high temperature sources to the lowest temperature point in the environment. The temperature distribution of the configuration space is computed by employing harmonic function to model the environment setup.

Mathematically, a harmonic function on a domain $\Omega \subset R^n$ is a function which satisfies Laplace's equation, in which x_i is the i -th Cartesian coordinate, and n is the dimension. In the case of robot path construction, the domain Ω consists of the outer boundary walls, all obstacles in the workspace, start points and the goal point. In this model, the robot is represented by a point in the configuration space. The configuration space is designed in grid form. The function values associated with each node are then computed iteratively via numerical technique to satisfy equation in Eq. (1). The highest temperature is assigned to the start point whereas the goal point is assigned the lowest, meanwhile different initial temperature values are assigned to the outer wall boundaries and obstacles. Initial temperature values are not required to be assigned to the start points. Then, the solutions to the

Laplace's equation are examined with Dirichlet boundary conditions.

$$\nabla^2 \phi = \sum_{i=1}^n \frac{\partial^2 \phi}{\partial x_i^2} = 0 \quad (1)$$

4. FORMULATION OF 4EGSOR9L ITERATIVE METHOD

In the literature, Jacobi, Gauss-Seidel and SOR [6] had been used for solving any linear system. Daily and Bevely [8] employed analytical solution for arbitrarily shaped obstacles. Meanwhile, the works by Evans [11], Evans & Yousif [12], Ibrahim [13] and Sulaiman *et al.* [14] on block iterative methods utilized various points of Explicit Group (EG) methods. In their works, they pointed out that block iterative methods were superior in comparison to the traditional point iterative methods. Adams *et al.* [15] studied the effect of SOR iteration for the 9-point Laplacian.

In this study, we propose faster numerical solver than our previous works [16 - 19] by employing Four-Point Explicit Group Successive-Over-Relaxation via Nine-Point Laplacian (also known as 4EGSOR9L) iterative method for computing the potential values. Our previous works [18, 19] that employed 9-point formula (also known as 9-point Laplacian) produced better performance than the standard 5-point formula [16, 17]. The finite difference approximation for 9-point formula uses 9-point stencil as illustrated in Figure 1. By having more points (9 points instead of 5 points) in the formulation, each calculation would give greater accuracy thus lead to faster convergence.

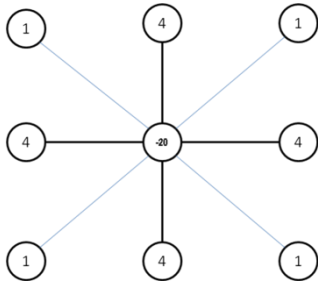


Figure 1: The stencil for 9-point formula.

Now, let us consider the two-dimensional Laplace's equation in Eq. (1) defined as

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \quad (2)$$

The standard discretization of Eq. (2) based on 5-point formula can be shown as below

$$U_{i,j} = \frac{1}{4}(U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1}) \quad (3)$$

Whereas for the 9-point Laplacian, the approximation equation can be defined as below

$$4(U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1}) + U_{i-1,j-1} + U_{i+1,j-1} + U_{i-1,j+1} + U_{i+1,j+1} - 20U_{i,j} = 0 \quad (4)$$

In the implementation of 4EGSOR9L block iterative method, the values of four points will be obtained in each calculation. This can be achieved by having a block of four points calculated simultaneously as illustrated in Figure 2.

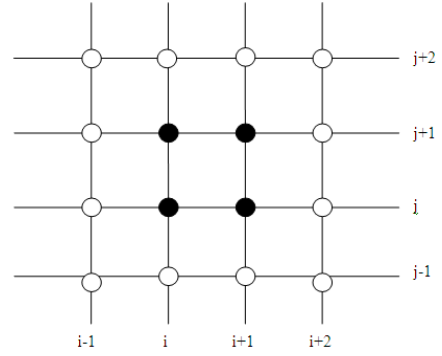


Figure 2: Illustration of a block of four node points to be computed iteratively.

The matrix for the block of four node points can be defined as

$$\begin{bmatrix} 20 & -4 & -4 & -1 \\ -4 & 20 & -1 & -4 \\ -4 & -1 & 20 & -4 \\ -1 & -4 & -4 & 20 \end{bmatrix} \begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \quad (5)$$

where

$$\begin{aligned} S_1 &= 4U_{i-1,j} + 4U_{i,j-1} + U_{i-1,j} + U_{i+1,j} + U_{i-1,j+1}, \\ S_2 &= 4U_{i+2,j} + 4U_{i+1,j-1} + U_{i,j-1} + U_{i+2,j-1} + U_{i+2,j+1}, \\ S_3 &= 4U_{i-1,j+1} + 4U_{i,j+2} + U_{i,j+1} + U_{i-1,j+2} + U_{i+1,j+2}, \\ S_4 &= 4U_{i+2,j+1} + 4U_{i+1,j+2} + U_{i+2,j} + U_{i,j+2} + U_{i+2,j+2}. \end{aligned}$$

Determining the inverse matrix of the coefficient matrix in Eq. (4), the general scheme of this block iterative method can be written as

$$\begin{bmatrix} U_{i,j} \\ U_{i+1,j} \\ U_{i,j+1} \\ U_{i+1,j+1} \end{bmatrix} = \frac{1}{2079} \begin{bmatrix} 116 & 28 & 28 & 17 \\ 28 & 116 & 17 & 28 \\ 28 & 17 & 116 & 28 \\ 17 & 28 & 28 & 116 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \quad (6)$$

In order to solve Eq. (6) computationally, it can be rewritten as

$$\begin{aligned} U_{i,j} &= \frac{1}{2079}(116S_1 + 28S_2 + 28S_3 + 17S_4) \\ U_{i+1,j} &= \frac{1}{2079}(28S_1 + 116S_2 + 17S_3 + 28S_4) \\ U_{i,j+1} &= \frac{1}{2079}(28S_1 + 17S_2 + 116S_3 + 28S_4) \\ U_{i+1,j+1} &= \frac{1}{2079}(17S_1 + 28S_2 + 28S_3 + 116S_4) \end{aligned} \quad (6)$$

Finally, the implementation of SOR into Eq. (6) would further speed up the computation. With SOR, the formulation can be shown as follows (Young [16, 17, 18]):

$$\begin{aligned} U_{i,j}^{k+1} &= \frac{\omega}{2079}(116S_1 + 28S_2 + 28S_3 + 17S_4) + (1-\omega)U_{i,j}^k \\ U_{i+1,j}^{k+1} &= \frac{\omega}{2079}(28S_1 + 116S_2 + 17S_3 + 28S_4) + (1-\omega)U_{i+1,j}^k \\ U_{i,j+1}^{k+1} &= \frac{\omega}{2079}(28S_1 + 17S_2 + 116S_3 + 28S_4) + (1-\omega)U_{i,j+1}^k \\ U_{i+1,j+1}^{k+1} &= \frac{\omega}{2079}(17S_1 + 28S_2 + 28S_3 + 116S_4) + (1-\omega)U_{i+1,j+1}^k \end{aligned} \quad (7)$$

The optimal value of ω is defined in the range $1 \leq \omega < 2$. In order to find the optimal value, several runs of computer program implementation of Eq. (7) are carried out with different value of ω . The value of ω is considered optimal when the program converges with the less number of iterations.

5. EXPERIMENTS AND RESULTS

The experiment was carried out with various size of static environment, i.e. 128x128, 256x256 and 512x512, that consists of a goal point, three starting points and varying number of obstacles (L, T and Boxes objects). Initially, the outer boundary walls, inner walls and obstacles were fixed with high temperature values, whereas the goal point was set to very low temperature, and all other free spaces were set to zero temperature value. The experiments run on Intel Core 2 Duo CPU running at 1.83GHz speed with 1GB of RAM. The software code, now known as RobotPath Simulator (see Figure 3), was implemented in Delphi for very fast computation. Previously [16 - 19] the code was written in MatLab. The computation speed up was increased 5 folds with this Delphi implementation.

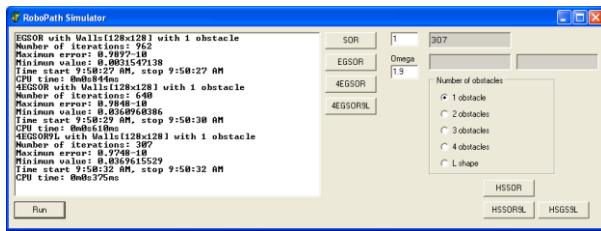


Figure 3: RoboPath Simulator was written in Delphi for very fast computation.

The iteration process stops when the computation converges to a specified very small value, i.e. 1.0^{-10} , where there are no more significant changes in temperature values. The number of iterations, maximum error and elapsed time for various numerical techniques are shown in Table 1. It is clearly shown in Figure 4 that 4EGSOR9L iteration performs faster than the various previous methods. Note that the speed of computation gets faster as the number of obstacles increases, since nodes occupied by obstacles are ignored during computation, see Table 2. The performance differences, however, is very minimal.

Table 1. Performance comparison of four numerical techniques with 1 obstacle.

	Iterative Method	Size of environment		
		128x128	256x256	512x512
Number of iterations	SOR	1314	5059	18699
	EGSOR	962	3768	13982
	4EGSOR	640	2609	9783
	4EGSOR9L	307	461	787
Maximum error	SOR	0.9952^{-10}	0.9945^{-10}	0.9998^{-10}
	EGSOR	0.9897^{-10}	0.9983^{-10}	0.9998^{-10}
	4EGSOR	0.9848^{-10}	0.9983^{-10}	0.9986^{-10}
	4EGSOR9L	0.9748^{-10}	0.9993^{-10}	0.9710^{-10}
Elapsed time m:s.ms	SOR	0:1:344	0:24:641	7:11:118
	EGSOR	0:1:109	0:19:828	5:39:250
	4EGSOR	0:0:765	0:13:485	3:46:328
	4EGSOR9L	0:0:516	0:02:922	0:21:470
SOR: Gauss-Seidel with SOR; EGSOR: Explicit Group (EG) with SOR; 4EGSOR: 4-Point EGSOR; 4EGSOR9L: 4EGSOR via 9-Point Laplacian.				

Table 2. Performance of 4EGSOR9L with varying number of obstacles.

	Number of obstacles	Size of environment		
		128x128	256x256	512x512
Number of iterations	1	307	461	787
	2	306	460	786
	3	306	455	784
	4	306	444	779
Maximum error	1	0.9748^{-10}	0.9993^{-10}	0.9710^{-10}
	2	0.9792^{-10}	0.9403^{-10}	0.9976^{-10}
	3	0.9709^{-10}	0.9411^{-10}	0.9563^{-10}
	4	0.9261^{-10}	0.9503^{-10}	0.9672^{-10}
Elapsed time m:s.ms	1	0:0:516	0:2:922	0:21:470
	2	0:0:453	0:2:719	0:20:703
	3	0:0:445	0:2:672	0:20:547
	4	0:0:437	0:2:703	0:20:406

Once the temperature distributions were obtained, the path was generated by performing steepest descent search from the start points to the goal point. The process of generating the paths was very fast. From the current point, the algorithm simply picked the lowest temperature value from its eight neighbouring points. This process continues, until the generated path reached the goal point. Figure 5 (a) - (d) shows the generated paths in varying number of obstacles. Note that SOR, EGSOR, and 4EGSOR (refer to our previous works in [16 - 19]) produced very similar visual.

6. CONCLUSIONS

This study shows that solving robot path planning problem using numerical techniques are indeed very attractive and feasible due to the recent advanced and new found techniques, as well as the availability of fast machine nowadays. The proposed 4EGSOR9L iterative method performs faster than the previous methods described in our earlier works [16 - 19]. In the future work, we would consider faster numerical computation using half-sweep (Akhir et al. [23]) and quarter-sweep (Muthuvalu & Sulaiman [24, 25]) iterative methods.

7. REFERENCES

- [1] Khatib, O. 1985. Real time obstacle avoidance for manipulators and mobile robots. IEEE Transactions on Robotics and Automation 1:500–505.
- [2] Koditschek, D. E. 1987. Exact robot navigation by means of potential functions: Some topological considerations. Proceedings of the IEEE International Conference on Robotics and Automation: 1-6.
- [3] Connolly, C. I., Burns, J. B., & Weiss, R. 1990. Path planning using Laplace's equation. Proceedings of the IEEE International Conference on Robotics and Automation: 2102–2106.
- [4] Akishita, S., Kawamura, S., & Hayashi, K. 1990. Laplace potential for moving obstacle avoidance and approach of a mobile robot. Japan-USA Symposium on flexible automation, A Pacific rim conference: 139–142.
- [5] Connolly, C. I., & Gruppen, R. 1993. On the applications of harmonic functions to robotics. Journal of Robotic Systems, 10(7): 931–946.
- [6] Sasaki, S. 1998. A Practical Computational Technique for Mobile Robot Navigation. Proceedings of the IEEE

- International Conference on Control Applications: 1323-1327.
- [7] Waydo, S. & Murray, R.M. 2003. Vehicle motion planning using stream functions. In Proc. of the Int. Conf. on Robotics and Automation (ICRA), 2003, pp.2484-2491.
- [8] Daily, R., & Bevly, D.M. 2008. Harmonic Potential Field Path Planning for High Speed Vehicles. In the proceeding of American Control Conference, Seattle, June 11-13, 4609-4614.
- [9] Garrido, S., Moreno, L., Blanco, D. & Monar, F.M. 2010. Robotic Motion Using Harmonic Functions and Finite Elements. Journal of Intelligent and Robotic Systems archive. Volume 59, Issue 1, July 2010. Pages 57 – 73.
- [10] Szulczyński, P., Pazderski, D. & Kozłowski, K. 2011. Real-Time Obstacle Avoidance Using Harmonic Potential Functions. Journal of Automation, Mobile Robotics & Intelligent Systems. Volume 5, No 3, 2011.
- [11] Evans, D. J. 1985. Group Explicit Iterative methods for solving large linear systems. Int. J. Comp. Maths., 17: 81-108.
- [12] Evans, D.J. & Yousif, W. S. 1986. Explicit Group Iterative Methods for solving elliptic partial differential equations in 3-space dimensions. Int. J. Computer Maths, 18:323-340.
- [13] Ibrahim, A. 1993. The Study of the Iterative Solution of Boundary Value Problem by the Finite Difference Methods. PhD Thesis. Universiti Kebangsaan Malaysia.
- [14] Sulaiman, J., Hasan, M.K. & Othman, M. 2007. Red-Black EDGSOR Iterative Method Using Triangle Element Approximation for 2D Poisson Equations. In. O. Gervasi & M. Gavrilova (Eds). Computational Science and Its Application 2007. Lecture Notes in Computer Science (LNCS 4707): 298-308. Berlin: Springer-Verlag.
- [15] Adams, L.M., Leveque, R.J. & Young, D.M. 1988. Analysis of the SOR Iteration for the 9-Point Laplacian. SIAM Journal of Numerical Analysis 25 (5): 1156-1180.
- [16] Saudi, A. and Sulaiman, J. 2010. Numerical technique for robot path planning using four Point-EG iterative method. In proc. of the 2010 Int. Symposium on Information Technology (ITSim), Page(s): 831 – 836. ISBN: 978-1-4244-6715-0.
- [17] Saudi, A. and Sulaiman, J. 2010. Robot path planning based on four Point-EGSOR iterative method. In proc. of the 2010 IEEE Conference on Robotics Automation and Mechatronics (RAM), Page(s): 476 – 481. ISBN: 978-1-4244-6503-3.
- [18] Saudi, A. and Sulaiman, J. 2010. Block Iterative Method using Nine-Point Laplacian for Robot Path Planning. European Journal of Scientific Research. ISSN 1450-216X Vol.43 No.2 (2010), pp.204-211.
- [19] Saudi, A. and Sulaiman, J. 2012. Robot Path Planning via EGSOR Iterative Method Using Nine-Point Laplacian. In proc. of the 2012 3rd Int. Conf. on Intelligent Systems, Modelling and Simulation (ISMS), Page(s): 61 – 66. ISBN: 978-1-4673-0886-1.
- [20] Young, D.M. 1971. Iterative solution of large linear systems. London: Academic Press.
- [21] Young, D.M. 1972. Second-degree iterative methods for the solution of large linear systems. Journal of Approximation Theory. 5:37-148.
- [22] Young, D.M. 1976. Iterative solution of linear systems arising from finite element techniques. In: Whiteman, J.R. (Eds.). The Mathematics of Finite Elements and Applications II: 439-464. London: Academic Press.
- [23] Akhir, M.K.M, Othman, M., Sulaiman, J., Majid, Z.A. & Suleiman, M. 2012. Half-Sweep Iteration for Solving Two-Dimensional Helmholtz Equations. International Journal of Applied Mathematics and Statistics, Vol. 29(5):101-109. ISSN: 0973-7545.
- [24] Muthuvalu, M.S. & Sulaiman, J. 2011. Numerical Solution of Second Kind Linear Fredholm Integral Equations Using QSGS Iterative Method with High-Order Newton-Cotes Quadrature Schemes. Malaysian Journal of Mathematical Sciences. Vol. 5(1): 85-100.
- [25] Muthuvalu, M.S. & Sulaiman, J. 2011. Numerical Solutions of First Kind Linear Fredholm Integral Equations Using Quarter-Sweep Successive Over-Relaxation (QSSOR) Iterative Method. International Journal of Open Problems in Computer Science and Mathematics. Vol. 4. No. 1, March 2011:184-196.

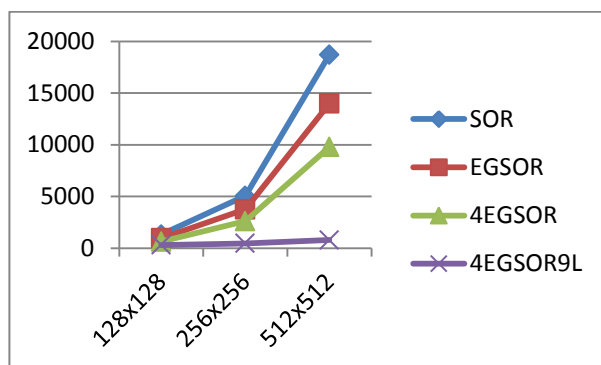
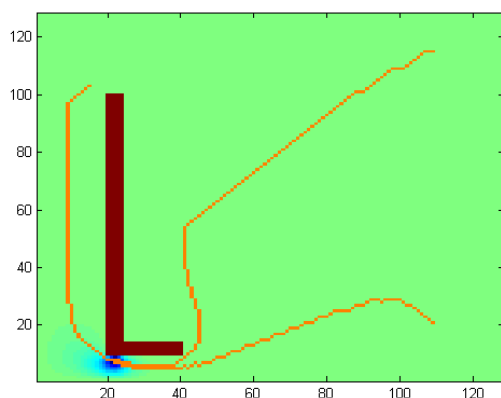
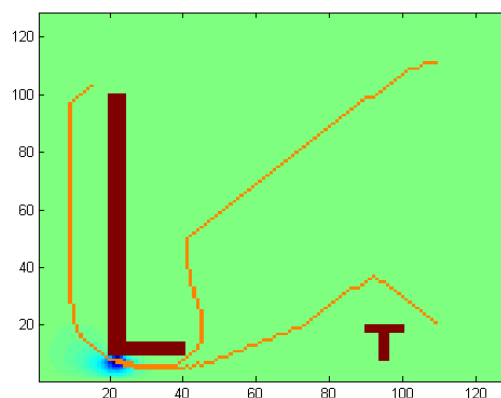


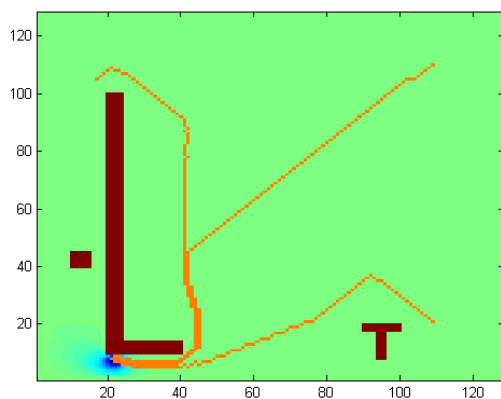
Figure 4: Graph of the number of iterations against size of environments for various iterative methods.



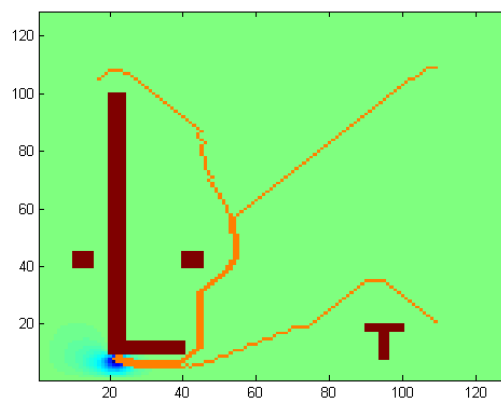
(a)



(b)



(c)



(d)

Figure 5: The paths generated with 4EG9SORL. (a) One obstacle; (b) Two obstacles; (c) Three obstacles; (d) Four obstacles.