ISBN 978-81-933894-1-6
5th International Conference on Future Computational Technologies (ICFCT'2017)
Kyoto (Japan) April 18-19, 2017

Path Planning in Structured Environment using Harmonic Potential Fields via Block Iterative Method

Azali Saudi[1] and Jumat Sulaiman[2] and Siti Hasanah Tanalol[2]

[1]Faculty of Computing & Informatics
[2]Faculty of Science and Natural Resources
Universiti Malaysia Sabah Kota Kinabalu, Malaysia
Email: azali@ums.edu.my, jumat@ums.edu.my, hasnah@ums.edu.my

***Abstract*:** *Harmonic potential fields have been known very useful to solve path planning problems globally. This study presents the use of harmonic potential fields (HPFs) for robot navigation in a known indoor environment. The harmonic potential fields were obtained using an iteration procedure based on block methods and subsequently used in the path planning simulation. The simulation was designed in a structured environment which contains various forms of obstacles such as narrow corridors, corners and small rooms, where several different start and goal positions were tested. For performance comparison purposes, the standard point Gauss-Seidel (GS) and Successive Overrelaxation (SOR) methods were also considered. The simulation results showed that the HPFs were capable of providing smooth path for robot navigation in a structured environment. It was also shown that the block iterative methods were more efficient compared to the standard point iterative methods in computing the HPFs.*

Keywords: *Path planning, harmonic potential fields, block iterative method, robot navigation.*

1. Introduction

The idea of utilizing HPFs in robotics was first initiated by Khatib [1] for controlling robotic manipulators and mobile robots using a goal with an attractive potential and obstacles with repulsive potentials. A main problem with this potential field approach, however, was that the robot being attracted to local minima instead of the goal. A solution to this problem is to utilize potential fields that are solutions to the Laplace's equation [2-4]. These Laplace's potentials have the advantage over simple potential field based approach, as they have no local minima. Moreover, they offer a complete path planning algorithm, and paths derived from them are generally smooth. In the past, an exact method based on the HPFs were applied in many area of researches including ship navigation [5], trajectory control [6], space robot path planning [7], navigation of snake-like robot [8], UAV motion planning [9], marine vessel path planning [10], 3D motion planning for UAV [11,12], space exploration [13], etc. In robot navigation problem, the HPFs were successfully applied in various types of environments [14-20]. As an exact method, the approach based on HPFs posses the advantage of complete algorithm property to guarantee that the solutions can be found if they exist, or otherwise the algorithm can detect when no solution exists. In other words, there are no approximations or sampling errors that becomes the main problem in heuristics methods. The HPFs are computed in a global manner over the entire region, thus they do no exhibit spurious local minima.

In this study, we focus on the problem of path planning for a robot operating in a structured indoor environment. The path planning algorithm utilizes the surface gradient of the computed HPFs to generate path from arbitrary start position to the specified goal position. The HPFs are computed using block iteration procedures that are superior to the standard point iterative methods.

2. Physical Analogy

Assuming that a real robot vehicle can be reduced to a point moving in a known environment, path planning problem of the robot can be formulated as a steady-state heat transfer problem. In the heat transfer analogy, the goal is treated as a sink pulling heat in. The obstacles are indicated by zero (or very low) thermal conductivity. According to the tasks and environment, the sources are either assigned to the robots or to discrete nodes in the free-space. As the result of a heat conduction process, a temperature distribution develops and the heat flux lines that are flowing to the sink fill the workspace. Such a field can be seen as a communication medium among the goal, robots and robots. The path can be easily found by following the heat flux. This study follows the above analogy, where the robot is represented by a point in a static known structured environment. The path planning problem is then posed as an obstacle avoidance problem for the point robot from the start point to the goal point. The environment has either square or rectangular outer boundaries, inner walls and varying shapes of obstacles. The environment is discretized into grid form and function values associated with each node are computed. The highest function value is assigned to all points at the boundaries, whereas the goal point is assigned the lowest. This setting amounts to Dirichlet boundary conditions.

The potential field is computed in a global manner over the entire region, and the harmonic solutions to Laplace's equation are used to find the path lines for a robot to move from the start point to the goal point. Obstacles are considered as current sources and the goal is considered to be the sink, with the lowest assigned potential value. This amounts to using Dirichlet boundary conditions. Then, following the current lines, a succession of points with lower potential values leading to the point with least potential (goal) is found out. It is observed by Connolly et al. [2] that this process guarantees a path to the goal without encountering local minima and successfully avoiding any obstacle.

3. Harmonic Functions

A harmonic function on a domain $\Omega \subset \Re^n$ is a function which satisfies Laplace's equation

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0 \tag{1}$$

In the case of robot path construction, the boundary of Ω consists of the outer boundary of the workspace and the boundaries of inner walls and all the obstacles as well as the goal point, in a configuration space representation. The spontaneous creation of a false local minimum inside the region is avoided if Laplace's equation is imposed as a constraint on the functions used, as the harmonic functions satisfy the min-max principle. Laplace's equation can be solved in one of two ways, numerically or analytically. Numeric solutions rely on setting boundary conditions on the potentials at the outer boundaries, inner walls and obstacle boundaries. Typically the potentials would be relatively high at the start position and lowest at the goal position, so that the virtual gradient of the potentials point descendingly toward the goal position. In the path planning literature, Gauss-Seidel [2] and Successive Overrelaxation (SOR) [19] had been used for computing the solutions of Laplace's equation (1). It was shown that SOR performed considerably faster than Gauss-Seidel. Alternatively, Daily and Bevly [14] use analytical solution for arbitrarily shaped obstacles. More recently, Saudi and Sulaiman employed block iteration procedure for behaviour-based robot in relatively very simple environment [16, 20]. Block iterative methods had been extensively studied by Evans [21, 22], Akhir et al. [23], Muthuvalu et al. [24], and Kew and Ali [25]. They pointed out that block iterative methods are superior to the traditional point iterative methods. In this study, we apply the combination of block iterative methods with SOR for computing the HPFs in a static indoor structured environment. Unlike simple environment in our previous studies [16, 20], in structured environment the path planner has to consider the inner and outer boundary walls and avoid various shapes of obstacles, as well as difficult corners, narrow paths and small rooms. For performance comparison, both point Gauss-Seidel and SOR iterative methods were also considered.

4. Point Iterative Method

The application of equation (1) to model the harmonic potential values in the path planning problem often results in large linear system with sparse coefficient matrix. Therefore, numerical solution of (1) using iterative method is often used to efficiently solve such linear system which is oftenly large and sparse. The standard five-point finite difference formula to approximate equation (1) is given as

$$\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1} - 4\phi_{i,j} = 0. \tag{2}$$

Here, we assume that a rectangular grid in the (x, y) plane with grid spacing h in both directions with $x_i=ih$, $y_j=jh$ is used and $\phi_{i,j}=\phi(x_i,y_j)$ with $i,j=0,1,2,...,n$. The corresponding Gauss-Seidel [26] and SOR [27] iterative schemes for the standard five-point difference formula can be written as

$$\phi_{i,j}^{(k+1)} = \frac{1}{4}(\phi_{i-1,j}^{(k+1)} + \phi_{i+1,j}^{(k)} + \phi_{i,j-1}^{(k+1)} + \phi_{i,j+1}^{(k)}), \tag{3}$$

$$\phi_{i,j}^{(k+1)} = \frac{\omega}{4}\left(\phi_{i-1,j}^{(k+1)} + \phi_{i+1,j}^{(k)} + \phi_{i,j-1}^{(k+1)} + \phi_{i,j+1}^{(k)}\right) + (1-\omega)\phi_{i,j}. \tag{4}$$

Applying these finite difference approximations to equation (1) will result in systems of algebraic equations that can be stated as

$$Au = b \tag{5}$$

where A and b are known and u is unknown. For simplicity, let matrix A be decomposed into

$$A = D - L = U \tag{6}$$

where D is a block diagonal matrix, L is a lower triangular matrix and U is an upper triangular matrix. Thus, the corresponding Gauss-Seidel and SOR iterative schemes in matrix form can be written as

$$Du^{(k+1)} = Lu^{(k+1)} + Uu^{(k)} + \omega b, \tag{7}$$

$$Du^{(k+1)} = \omega Lu^{(k+1)} + \omega Uu^{(k)} + (1-\omega)Du^{(k)} + \omega b. \tag{8}$$

Note that, if ω, equation (8) simplifies to the standard Gauss-Seidel scheme (7). The iteration process of equations (7) and (8) continues until the convergence criterion is satisfied, i.e. $|| \phi^{(k+1)} - \phi^{(k)} || < \varepsilon$. The convergence error tolerance, ε must be set to a very small value to avoid the occurrence of saddle points in the resulting Laplace's potential. These saddle points could prevent the path tracing process from reaching the goal point. The weighted parameter, ω is the range $1 \leq \omega < 2$. The optimal value of ω can be obtained by conducting several runs of test until it gives the least number of iterations, k. During computation, only non-occupied nodes are considered. All other occupied nodes are ignored, since their values are held fixed.

5. Block Iterative Method

5.1. Block of Two Points Method

By applying equation (2) to a block of two node points as depicted in Figure 1, we obtain

$$\phi_{i,j} = \frac{1}{4}\left(\phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1}\right),$$

$$\phi_{i+1,j} = \frac{1}{4}\left(\phi_{i,j} + \phi_{i+2,j} + \phi_{i+1,j-1} + \phi_{i+1,j+1}\right). \tag{9}$$

In linear form (5), these equations can be rewritten as

$$\begin{bmatrix} 4 & -1 \\ -1 & 4 \end{bmatrix}\begin{bmatrix} \phi_{i,j} \\ \phi_{i+1,j} \end{bmatrix} = \begin{bmatrix} \phi_{i-1,j} + \phi_{i+1,j} + \phi_{i,j-1} + \phi_{i,j+1} \\ \phi_{i,j} + \phi_{i+2,j} + \phi_{i+1,j-1} + \phi_{i+1,j+1} \end{bmatrix}. \tag{10}$$

By determining the inverse matrix of the coefficient matrix in equation (10), the general scheme of two point-block iterative method can be rewritten as (Evans [21, 22])

$$\begin{bmatrix} \phi_{i,j} \\ \phi_{i+1,j} \end{bmatrix} = \frac{1}{15}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} \phi_{i-1,j} + \phi_{i,j-1} + \phi_{i,j+1} \\ \phi_{i+2,j} + \phi_{i+1,j-1} + \phi_{i+1,j+1} \end{bmatrix}. \tag{11}$$

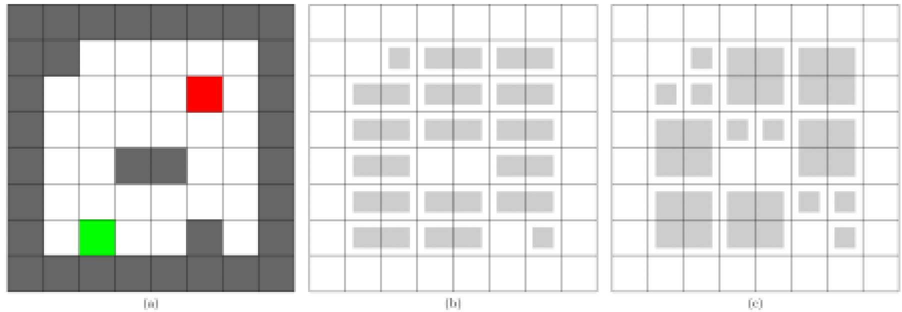

Fig. 1: (a) The GREEN and RED cells denote the start and goal points, respectively. GREY cells are boundary and obstacles. The cells are computed in group of either (b) two or (c) four node points.

Based on equation (11), the iterative scheme for the two point-block iterative method can be defined as

$$\phi_{i,j}^{(k+1)} = \frac{1}{15}(3b_1 + c_1), \quad \phi_{i+1,j}^{(k+1)} = \frac{1}{15}(c_1 + 3b_2). \tag{12}$$

where

$$b_1 = \phi_{i,j-1}^{(k+1)} + \phi_{i,j+1}^{(k)} + \phi_{i-1,j}^{(k+1)}, \quad b_2 = \phi_{i+1,j-1}^{(k+1)} + \phi_{i+1,j+1}^{(k)} + \phi_{i+2,j}^{(k)}, \quad c_1 = b_1 + b_2.$$

It can be clearly observed in equation (12) that the calculations of $\phi_{(k+1)}$ and $\phi_{(k+1)}$ are completely independent. Thus, block iterative method is very suitable for parallel implementation. Figure 1(a) illustrates the configuration space of the environment, where the occupied and non-occupied cells are drawn in GREY and WHITE colours, respectively (the start (GREEN) and goal (RED) points are considered as non-occupied cells). For the two point-block iterative method, each group consists of two node points, as shown in Figure 1(b). All ungroup nodes next to the boundary or obstacle cells are computed using direct method [21]. Furthermore, by adding a weighted parameter, ω, equation (12) can be rewritten as

$$\phi_{i,j}^{(k+1)} = \frac{\omega}{15}(3b_1 + c_1) + (1-\omega)\phi_{i,j}^{(k)}, \quad \phi_{i+1,j}^{(k+1)} = \frac{\omega}{15}(c_1 + 3b_2) + (1-\omega)\phi_{+1i,j}^{(k)}. \tag{13}$$

Equation (13) represents the iterative scheme for Two Point-Block Successive Overrelaxation (2-BLSOR) method.

5.2. Block of Four Points Method

Consider a group of four points (4 × 4) as depicted in Figure 1 (c). By applying the standard five-point finite difference approximation (2) to this group of points, the block iterative method based on four points can be expressed as

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ -1 & -1 & 0 & 4 \end{bmatrix} \begin{bmatrix} \phi_{i,j} \\ \phi_{i+1,j} \\ \phi_{i,j+1} \\ \phi_{i+1,j+1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}. \tag{14}$$

where $f_1 = \phi_{i-1,j} + \phi_{i,j-1}$, $f_2 = \phi_{i+2,j} + \phi_{i+1,j-1}$, $f_3 = \phi_{i-1,j+1} + \phi_{i,j+2}$, $f_4 = \phi_{i+2,j+1} + \phi_{i+1,j+2}$.

Figure 2 depicts the computational molecules of the four points. Now, the explicit solution of equation (14) can be defined as

$$\begin{bmatrix} \phi_{i,j} \\ \phi_{i+1,j} \\ \phi_{i,j+1} \\ \phi_{i+1,j+1} \end{bmatrix} = \frac{1}{24} \begin{bmatrix} 7 & 2 & 2 & 1 \\ 2 & 7 & 1 & 2 \\ 2 & 1 & 7 & 2 \\ 2 & 2 & 1 & 7 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}. \tag{15}$$

Based on equation (15), we obtain the iterative scheme of four point-block iterative method that can be expressed as

$$\phi_{i,j}^{(k+1)} = \frac{1}{24}(6b_1 + c_1), \quad \phi_{i+1,j}^{(k+1)} = \frac{1}{24}(6b_2 + c_2), \quad \phi_{i,j+1}^{(k+1)} = \frac{1}{24}(6b_3 + c_2), \quad \phi_{i+1,j+1}^{(k+1)} = \frac{1}{24}(6b_4 + c_1), \tag{16}$$

where

$$b_1 = \phi_{i-1,j}^{(k+1)} + \phi_{i,j-1}^{(k+1)}, \quad b_2 = \phi_{i+2,j}^{(k)} + \phi_{i+1,j-1}^{(k+1)}, \quad b_3 = \phi_{i-1,j+1}^{(k+1)} + \phi_{i,j+2}^{(k)}, \quad b_4 = \phi_{i+2,j+1}^{(k)} + \phi_{i+1,j+2}^{(k+1)},$$

$$c_1 = 2(b_2 + b_3) + b_1 + b_4, \quad c_2 = 2(b_1 + b_4) + b_2 + b_3.$$

By adding a weighted parameter, ω, equation (16) can be rewritten as

$$\phi_{i,j}^{(k+1)} = \frac{\omega}{24}(6b_1 + c_1) + (1-\omega)\phi_{i,j}^{(k)}, \quad \phi_{i+1,j}^{(k+1)} = \frac{\omega}{24}(6b_2 + c_2) + (1-\omega)\phi_{i+1,j}^{(k)},$$

$$\phi_{i,j+1}^{(k+1)} = \frac{\omega}{24}(6b_3 + c_2) + (1-\omega)\phi_{i,j+1}^{(k)}, \quad \phi_{i+1,j+1}^{(k+1)} = \frac{\omega}{24}(6b_4 + c_1) + (1-\omega)\phi_{i,j+1}^{(k)}. \tag{17}$$

Equation (17) represents the Four Point-Block SOR (4-BLSOR) iterative method. The algorithm to implement 4-BLSOR method is described below:

t_1 := startclock, k := 0
repeat
    for all groups of two node points **do**
        Use equation (17) to compute $\phi_{i,j}^{(k+1)}$, $\phi_{i+1,j}^{(k+1)}$, $\phi_{i,j+1}^{(k+1)}$ and $\phi_{i+1,j+1}^{(k+1)}$
    for all ungroup node points **do**
        Compute all nodes using equation (4)
    $k := k + 1$
until $\| \phi^{(k+1)} - \phi^{(k)} \| < \varepsilon$
t_2 := stopclock, $t_{elapsed} := t_2 - t_1$

6. Experiment and Simulation Results

The path planning algorithm begins by loading the map of the structured environment. The position of goal point is then indicated on the map. After that, the initial potential values of the environment are stored in a 2D matrix. All node points occupied by inner walls, outer boundaries and obstacles are assigned with relatively high potential values. Meanwhile, the goal point is assigned with the lowest potential value. The potential values of the occupied node points and goal points are fixed and do not change during the iteration process. Initial values that are equal to the potential values of the occupied nodes are assigned to all other free spaces. Since high precision is required, the convergence criterion is set to a very small error tolerance, $\varepsilon = 10^{-16}$. The harmonic potentials of the environment are then computed using the considered iterative methods, i.e. GS, SOR, 2-BLSOR and 4-BLSOR. When the convergence criterion is satisfied, the iteration process is terminated. By using the computed harmonic potential fields, the required path can be traced using the standard Gradient Descent Search (GDS) procedure. From initial position, the GDS picks the lowest point from its four neighbourhood points. This process continues until the goal point with the lowest potential value is found out. A stack is used to store points along the path line. The simulation was conducted by using a static known indoor structured environment that consists of inner wall, outer boundaries and various shapes of obstacles. Narrow paths, difficult corners and small rooms were also present. Two examples, i.e. Room 1 and Room 2 are considered covering an area of approximately 330 × 270 and 290 × 290 units, respectively. To examine the robustness of the path planning algorithm, several different start and goal points were tested. Figures 2 and 3 depict the corresponding generated paths for Room 1 and Room 2.

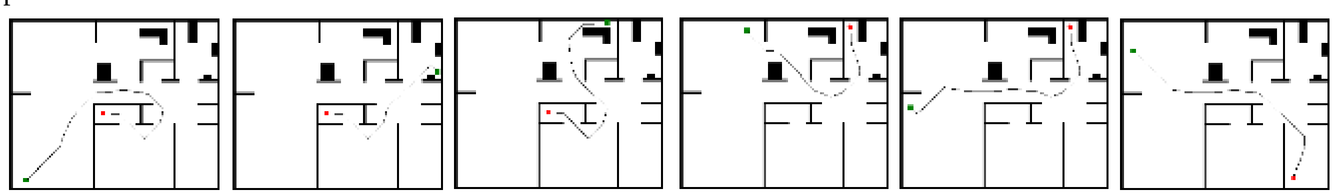

Fig. 2: The generated paths for Room 1 covering an area of 330 × 270.

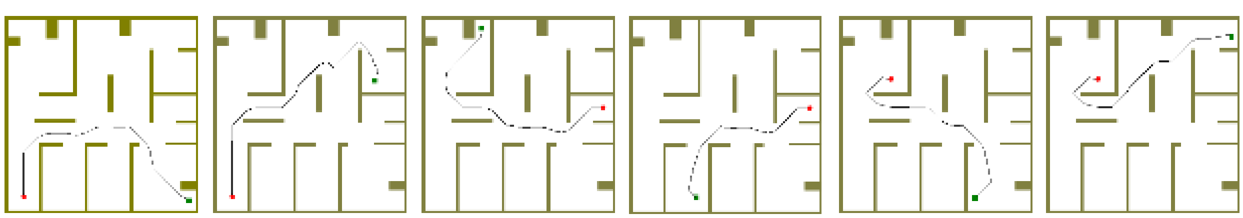

Fig. 3: The generated paths for Room 2 covering an area of 290 × 290.

The solid green box red circle denotes the start and goal point, respectively. The algorithm was capable of generating paths even with the occurrence of difficult regions. The path generation process was also very fast, since it only involved simple evaluation of finding the next lower potential value. The number of iterations and CPU time of the considered methods are shown in Table 1. The value of ω was set to 1.85 that gave the optimum performance in terms of number of iterations and execution time. Against the standard GS method, the SOR drastically reduced the iterations approximately by 90%. The block methods method gave the least number of iterations, where the iterations for 2-BLSOR and 4-BLSOR were approximately 25% and 50% less than the point SOR iterative method. In terms of CPU time, SOR method was 10 times faster than the standard GS method. The block methods gave the best performance, where the respective CPU time for 2-BLSOR and 4-BLSOR were 20% less and 55% than the point SOR iterative method.

TABLE I: ITERATIONS (k) AND CPU TIME (t) OF THE CONSIDERED METHODS

	Methods	Number of iteration (k)	CPU time (t) in seconds
Room 1	**GS**	50490	173.59
	SOR	4609	16.39
	2-BLSOR	3519	13.06
	4-BLSOR	2361	7.48
Room 2	**GS**	19194	59.40
	SOR	1696	5.69
	2-BLSOR	1287	4.54
	4-BLSOR	854	2.44

7. Conclusion and Future Work

Harmonic potential fields provide smooth gradient surface that are very useful for navigation of a robot. This study has proven its effectiveness to assist navigation in unstructured environments containing obstacles of various shapes. Simulations conducted show that the proposed block iterative methods provide better performance than the point iterative methods. Furthermore, the block iterative method is particularly suitable for parallel implementation. In the future, the study will consider dynamic and unstructured environments. The application of block iteration procedure on a higher dimensional space will be considered. The implementation of iterative methods on the real physical robot and integration in the cloud environment will also be considered.

8. Acknowledgment

This work is supported by the Ministry of Higher Education, Malaysia (Fund: FRGS/1/2016/ICT01/UMS/02/1).

9. References

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in Proceedings of the IEEE International Conference on Robotics and Automation, vol. 2, 1985, pp. 500–505.
https://doi.org/10.1109/robot.1985.1087247

[2] C. I. Connolly, J. B. Burns, and R. Weiss, "Path planning using Laplace's equation," in Proceedings of the IEEE International Conference on Robotics and Automation, 1990, pp. 2102–2106.
https://doi.org/10.1109/ROBOT.1990.126315

[3] S. Akishita, S. Kawamura, and K.-i. Hayashi, "New navigation function utilizing hydrodynamic potential for mobile robot," in Proceedings of the IEEE International Workshop on Intelligent Motion Control, vol. 2, 1990, pp. 413–417.
https://doi.org/10.1109/IMC.1990.687354

[4] S. Waydo and R. M. Murray, "Vehicle motion planning using stream functions," in Proceedings of the IEEE International Conference on Robotics and Automation, vol. 2, 2003, pp. 2484–2491.
https://doi.org/10.1109/robot.2003.1241966

[5] C. Shi, M. Zhang, and J. Peng, "Harmonic potential field method for autonomous ship navigation," in Telecommunications, 2007. ITST'07. 7th International Conference on ITS, June 2007, pp. 1–6.
https://doi.org/10.1109/itst.2007.4295916

[6] A. A. Masoud, "Dynamic trajectory generation for spatially constrained mechanical systems using harmonic potential fields," in Proceedings 2007 IEEE International Conference on Robotics and Automation, April 2007, pp. 1980–1985.
https://doi.org/10.1109/ROBOT.2007.363612

[7] Y. Yanoshita and S. Tsuda, "Space robot path planning for collision avoidance," in Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 2, 2009.

[8] S. Charifa and M. Bikdash, "Motion planning of a snake-like robot using an optimized harmonic potential field," Paladyn, vol. 1, no. 3, pp. 187–197, 2011.

[9] A. A. Masoud, "A harmonic potential approach for simultaneous planning and control of a generic UAV platform," Journal of Intelligent and Robotic Systems, vol. 65, no. 1-4, pp. 153–173, 2012.
https://doi.org/10.1007/s10846-011-9570-8

[10] M. D. Pedersen and T. I. Fossen, "Marine vessel path planning and guidance using potential flow," in Proceedings of the 9th IFAC Conference on Manoeuvring and Control of Marine Craft, 2012, pp. 188–193.
https://doi.org/10.3182/20120919-3-it-2046.00032

[11] K. Motonaka, K. Watanabe, and S. Maeyama, "3-dimensional kinodynamic motion planning for an x4-flyer using 2-dimensional harmonic potential fields," in Control, Automation and Systems (ICCAS), 2014 14th International Conference on, Oct 2014, pp. 1181–1184.
https://doi.org/10.1109/iccas.2014.6987738

[12] H. Wang, W. Lyu, P. Yao, X. Liang, and C. Liu, "Three-dimensional path planning for unmanned aerial vehicle based on interfered fluid dynamical system," Chinese Journal of Aeronautics, vol. 28, no. 1, pp. 229 – 239, 2015.
https://doi.org/10.1016/j.cja.2014.12.031

[13] J. Vallv´e and J. Andrade-Cetto, "Potential information fields for mobile robot exploration," Robotics and Autonomous Systems, vol. 69, pp. 68–79, 2015.
https://doi.org/10.1016/j.robot.2014.08.009

[14] R. Daily and D. M. Bevly, "Harmonic potential field path planning for high speed vehicles," in Proceedings of the IEEE American Control Conference, 2008, pp. 4609–4614.
https://doi.org/10.1109/acc.2008.4587222

[15] K. Motonaka, K. Watanabe, and S. Maeyama, "Offline gain optimization in kinodynamic motion planning based on a harmonic potential field," Artificial Life and Robotics, vol. 19, no. 1, pp. 47–54, 2013.
https://doi.org/10.1007/s10015-013-0129-6

[16] A. Saudi, J. Sulaiman, and M. H. A. Hijazi, "Fast robot path planning with Laplacian Behaviour-Based Control via four-point Explicit Decoupled Group SOR," Research Journal of Applied Sciences, vol. 9, no. 6, pp. 354–360, 2014.

[17] X. Liang, H. Wang, D. Li, and C. Liu, "Three-dimensional path planning for unmanned aerial vehicles based on fluid flow," in Proceedings of the IEEE Aerospace Conference, 2014, pp. 1–13.
https://doi.org/10.1109/aero.2014.6836520

[18] A. A. Masoud and A. Al-Shaikhi, "Time-sensitive, sensor-based, joint planning and control of mobile robots in cluttered spaces: A harmonic potential approach," in 2015 54th IEEE Conference on Decision and Control (CDC), Dec 2015, pp. 2761–2766.
https://doi.org/10.1109/cdc.2015.7402634

[19] S. Sasaki, "A practical computational technique for mobile robot navigation," in Proceedings of the IEEE International Conference on Control Applications, vol. 2, 1998, pp. 1323–1327.
https://doi.org/10.1109/cca.1998.721675

[20] A. Saudi and J. Sulaiman, "Hybrid path planning for indoor robot with laplacian behaviour-based control via four point-explicit group," International Journal of Imaging and Robotics, vol. 12, no. 1, pp. 12–21, 2014.

[21] D. J. Evans, "Group explicit iterative methods for solving large linear systems," International Journal of Computer Mathematics, vol. 17, no. 1, pp. 81–108, 1985.
https://doi.org/10.1080/00207168508803452

[22] D. J. Evans and W. S. Yousif, "Explicit Group Over-Relaxation iterative methods for solving elliptic partial differential equations in 3-space dimensions," Int. Journal of Computer Mathematics, vol. 17, pp. 323–340, 1986.
https://doi.org/10.1080/00207168608803498

[23] M. K. M. Akhir, M. Othman, J. Sulaiman, Z. A. Majid, and M. Suleiman, “The solution of 2d helmholtz equations by modified explicit group iterative method,” in Proc. Int. Conf. on Advances in Computing, Control, and Telecommunication Technologies (Elsevier), ACT, 2012, pp. 70–77.

[24] M. S. Muthuvalu, E. Aruchunan, and J. Sulaiman, Regional Conference on Science, Technology and Social Sciences (RCSTSS 2014): Science and Technology. Springer Singapore, 2014, ch. Performance Analysis of 2-Point Explicit Group (2-EG) Method for Solving Second-Order Composite Closed Newton-Cotes Quadrature System, pp. 297–305.

[25] L. M. Kew and N. H. M. Ali, “New explicit group iterative methods in the solution of three dimensional hyperbolic telegraph equations,” J. Comput. Phys., vol. 294, no. C, pp. 382–404, Aug. 2015.

https://doi.org/10.1016/j.jcp.2015.03.052

[26] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003.

https://doi.org/10.1137/1.9780898718003

[27] D. M. Young, “Iterative methods for solving partial difference equations of elliptic type,” Ph.D. dissertation, Harvard University, 1950.