



Article

An Inverse Pheromone Approach in a Chaotic Mobile Robot's Path Planning Based on a Modified Logistic Map

Eleftherios K. Petavratzis ^{1,*}, Christos K. Volos ^{1,†}, Lazaros Moysis ¹,
Ioannis N. Stouboulos ^{1,†}, Hector E. Nistazakis ^{2,†}, George S. Tombras ² and
Kimon P. Valavanis ^{3,†}

¹ Laboratory of Nonlinear Systems—Circuits and Complexity, Physics Department, Aristotle University of Thessaloniki, 541 24 Thessaloniki, Greece; volos@physics.auth.gr (C.K.V.); moysis.lazaros@hotmail.com (L.M.); stouboulos@physics.auth.gr (I.N.S.)

² Department of Electronics, Computers, Telecommunications and Control, Faculty of Physics, National and Kapodistrian University, 157 72 Athens, Greece; enistaz@phys.uoa.gr (H.E.N.); gtombras@phys.uoa.gr (G.S.T.)

³ Daniels Felix Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO 80208, USA; kimon.valavanis@du.edu

* Correspondence: elpetavr@physics.auth.gr

† This paper is an extended version of our paper published in Proceedings of the 8th International Conference on Modern Circuits and Systems Technologies (MOCASST), Thessaloniki, Greece, 13–15 May 2019.

Received: 20 November 2019; Accepted: 4 December 2019; Published: 6 December 2019



Abstract: One major topic in the research of path planning of autonomous mobile robots is the fast and efficient coverage of a given terrain. For this purpose, an efficient method for covering a given workspace is proposed, based on chaotic path planning. The method is based on a chaotic pseudo random bit generator that is generated using a modified logistic map, which is used to generate a chaotic motion pattern. This is then combined with an inverse pheromone approach in order to reduce the number of revisits in each cell. The simulated robot under study has the capability to move in four or eight directions. From extensive simulations performed in Matlab, it is derived that motion in eight directions gives superior results. Especially, with the inclusion of pheromone, the coverage percentage can significantly be increased, leading to better performance.

Keywords: autonomous mobile robot; chaos; path planning; inverse pheromone method; random bit generator

1. Introduction

Nowadays, autonomous robotic systems [1] integrate numerous sections of society, ranging from simple domestic applications [2] to complex tasks like rescue missions [3–5], military missions [6–12], or industrial and space applications [13–15]. In most of the above tasks, the design of an efficient motion strategy is required so that the robot can cover a given area fast and efficiently [16,17]. More importantly, in many military, exploratory, and surveillance applications, the performing robot needs to explore a given area moving unpredictably on the grid. This option is chosen so that it is harder for an opponent to place obstacles or surveillance cameras based on predicted motion, or so as not to arise suspicion (in case of a camouflaged robot).

In order to design a seemingly random motion that will also guarantee efficient grid exploration, many researchers have considered chaotic path planning [2,18–33]. In this approach, a chaotic dynamical system is utilized to generate a chaotic motion trajectory, that is then performed by the

robot to achieve efficient coverage. Chaotic systems are dynamical systems that are highly sensitive to initial conditions and parameter changes [34–36]. This means that a small deviation in the system's initial value or in a parameter will lead to a completely different solution. This is why chaotic systems are very hard to predict and control, a fact that makes them highly applicable to a plethora of fields, such as engineering, cryptography, communication, and many others [37–44].

In the problem of path planning, many well-known continuous and discrete time dynamical systems, such as the logistic map, Lorenz, Arnold, Taylor-Chirikov, and many others have been applied for designing robot path planning controllers, giving very interesting results [2,7,22,23].

In this work, a modification of the well-known logistic map is used to generate a chaotic pseudo random bit generator (CPRBG), thus generating a chaotic bitstream from the values of the system, using a simple rule. The generated bit sequence is then used to produce a motion command for the robot that moves either in four or in eight directions. Furthermore, to improve the grid coverage and also to reduce the number of times that the robot revisits previous cells, the idea of an inverse pheromone method is used [24]. The use of a pheromone is inspired from ant colonies, where ants use pheromone to track their path and attract other ants to it [45–47]. Here, the inverse of this technique is used. So as the robot moves, it saves in its memory the last F places it visits, where F is the chosen pheromone level, which decreases with each new step. So, when a movement to a new cell is about to be performed, the robot checks if the new point has been visited before, based on its memory. So, if the new cell has been visited in the last F moves, i.e., its pheromone level is non-zero, then the robot remains in its place and awaits the next movement command. This method is very satisfying in reducing the number of revisits in previous cells, while also giving faster grid coverage.

For the evaluation of the coverage performance of the proposed method, extensive simulation results are performed in Matlab, with varying simulation parameters, like the number of movements, the amount of pheromone, and the initial position of the robot. It is seen that as the number of steps increases, the grid coverage converges to 100%. In addition, movement in eight directions is more efficient than movement in four. In both cases, the inclusion of pheromone leads to higher coverage and a lower number of revisits in visited cells.

The structure of the work is as follows. In Section 2, the proposed modified logistic map is presented. In Section 3, the design of the proposed PRBG is examined. In Section 4, the robotic motion controller, which uses the inverse pheromone method, is explained and the simulation results for different scenarios are presented and evaluated. Finally, Section 5 concludes this work with a discussion on future aspects.

2. The Modified Logistic Map

The introduction of the notion of chaos in the 1960s led the researchers to find the theoretical tools of nonlinear systems' theory, in order to study continuous or discrete time chaotic dynamical systems. As mentioned in the previous section, the uniqueness of chaotic systems lies in the fact that they are extremely sensitive to initial conditions and parameter changes.

Discrete-time dynamical systems can be described as an iterative map $f : R^n \rightarrow R^n$, given by the state equation

$$x_{k+1} = f(x_k) \quad , \quad k = 0, 1, 2, \dots \quad (1)$$

where n is the dimension of the state-space, k denotes the discrete time of the system, $x_k \in R^n$ is the state of the system at time k , and x_{k+1} is the state at the next time instance.

In 1844 Pierre Francois Verhulst introduced a growth model for a closed population facing a living environment with limited resources for the subsistence of its members [48]. With the contribution of May [49], the logistic map is now one of the most commonly used discrete time maps in various engineering applications, such as noise generator in electronics, bit generators, cryptosystems, and is also applied in the modeling of economic or population systems [50–55].

The logistic map is described by the equation

$$x_{k+1} = r \cdot x_k \cdot (1 - x_k), \quad 0 \leq x \leq 1, \quad (2)$$

where the parameter r varies in the interval $[0, 4]$.

According to the value of the parameter r , the logistic map can have the following three different dynamics.

- For $r < 1$, x decays to a fixed point ($x \rightarrow 0$).
- For $1 \leq r < 3$, the previous point loses its stability and a new fixed point appears ($x = 1/r$).
- For $3 \leq r \leq 4$, the system goes from a periodic trajectory into chaos.

Especially for $r = 4$, the values of x_k cover the complete interval $[0, 1]$. The bifurcation diagram of the logistic map is shown in Figure 1.

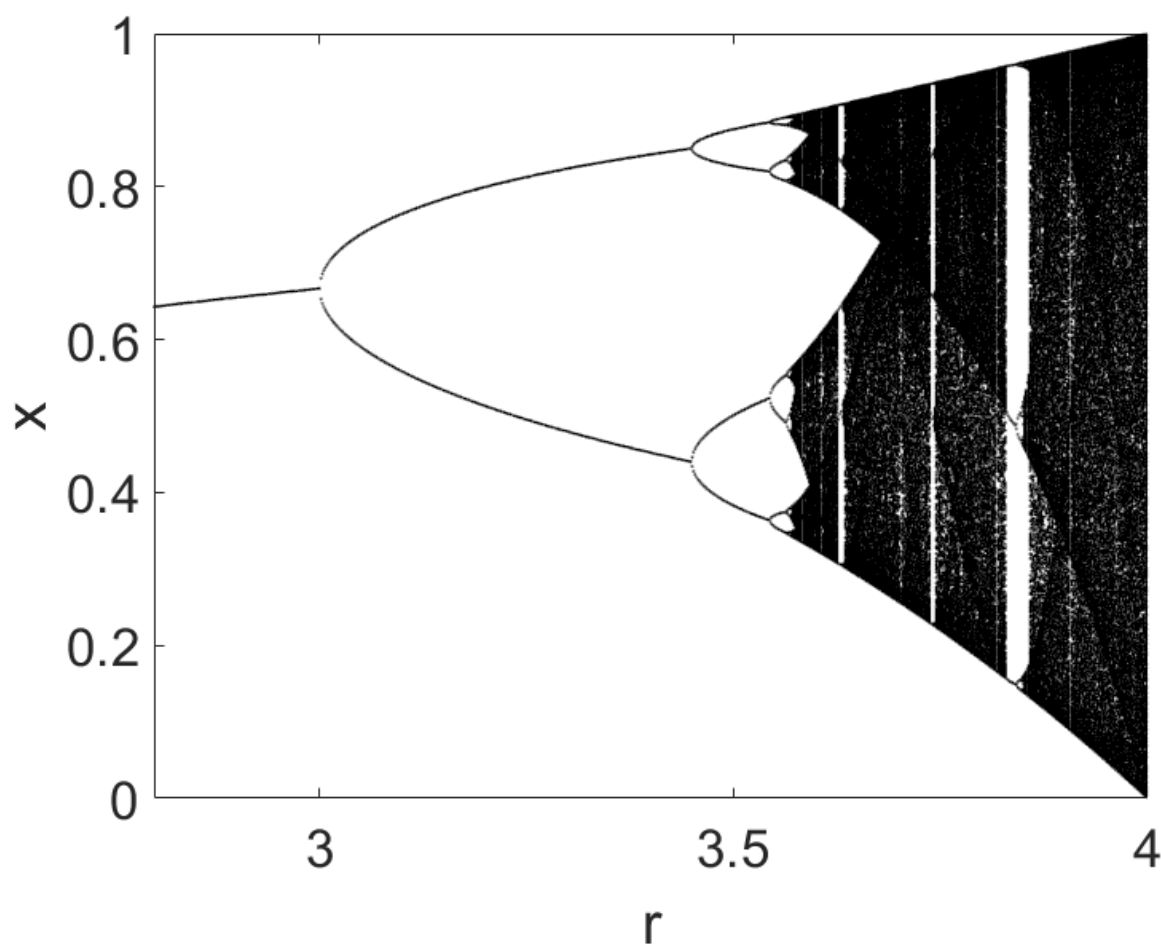


Figure 1. Bifurcation diagram of the logistic map.

Here, a modified logistic map is selected because of its rich dynamical behavior [24]. This modified map is described by:

$$x_{k+1} = r \cdot x_k \cdot [\alpha - b \cdot x_k^2 + c \cdot x_k^4]. \quad (3)$$

Its bifurcation diagram is shown in Figure 2. Looking at this Figure, the set of $(\alpha, b, c) = (1.25, 5.001, 4)$ gives an interesting dynamical behavior in regard to the value of r . According to Figure 2, for $1.5 \leq r \leq 2.1$ the system shows a periodic behavior. After that, it enters a chaotic region with some small windows of periodic behavior. This region ends for $r = 3.25$ where it shows a periodic-2 state and finally for $3.5 \leq r \leq 4$ it enters chaos.

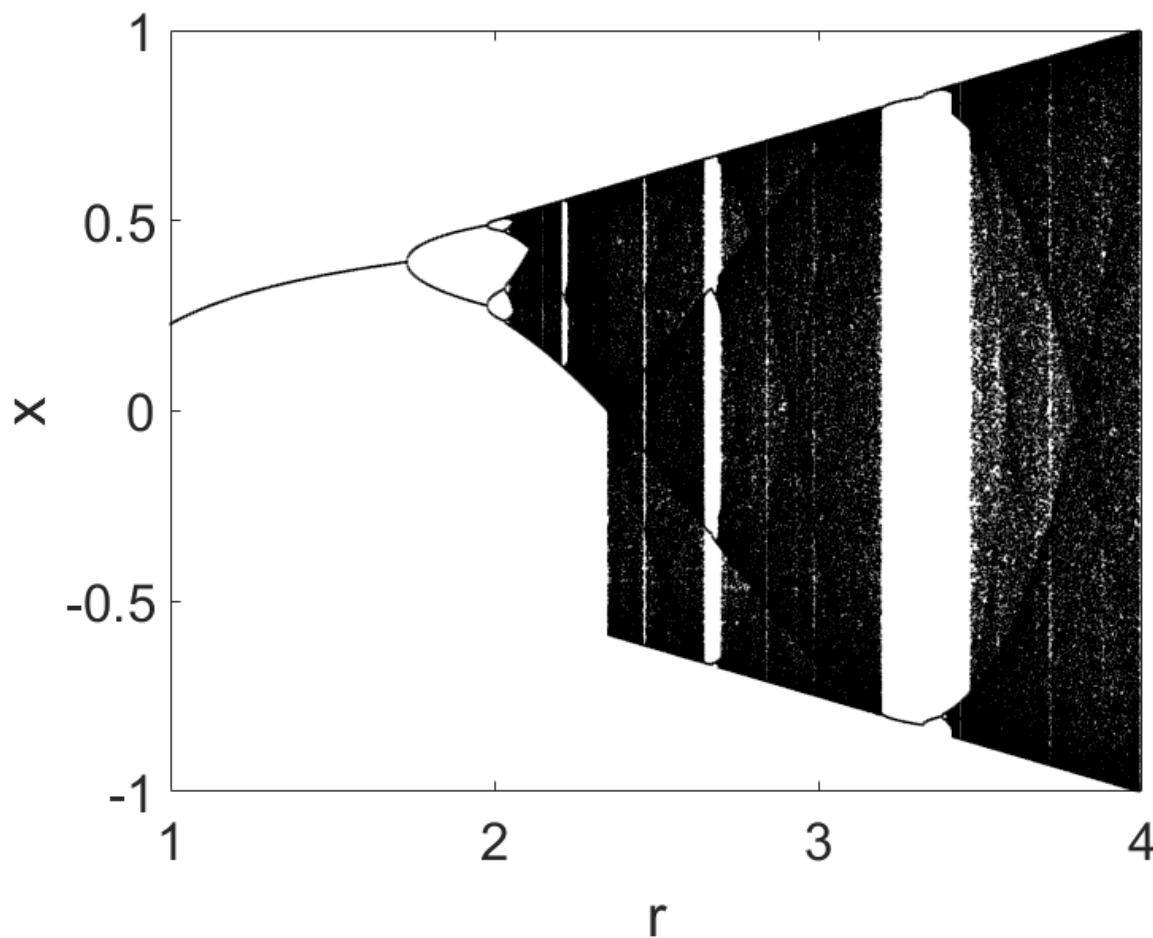


Figure 2. Bifurcation diagram of the modified logistic map for parameters $(a, b, c) = (1.25, 5.001, 4)$.

3. Pseudo Random Number Generator

From the values of the modified logistic map of (3), a pseudo random bits sequence b_i is produced, using the following rule

$$b_i = \begin{cases} 0 & x_i < 0 \\ 1 & x_i \geq 0. \end{cases} \quad (4)$$

As the bit sequence is produced, it is noticed that many pairs of “00” and “11” appear. By applying the de-skewing technique [56], these pairs are discarded and also the “01” is converted into an output “0” and “10” into “1”. As a result of this technique, we have a reduction of the correlation of the bitstream. The next step is to check if the sequence is indeed random. The FIPS-140-2 (Federal Information Processing Standards) tests of National Institute of Standards and Technology (NIST) [57] ensures the “randomness” of the bitstream. As it can be seen from Table 1, for a 1,000,000 bits sequence the tests ensure that the sequence is random.

Table 1. Results of NIST 140-2 tests for the generator.

Monobit Test	Poker Test	Runs Test	Long Run Test
$n_0 = 9935$ (49.67%)	32.02% (for $n = 4$)	$B_1 = 2337$ $B_2 = 1258$ $B_3 = 627$ $B_4 = 335$ $B_5 = 154$ $B_6 = 177$	No
Passed	Passed	Passed	Passed

One sufficient randomness index is the measure-theoretic entropy of the generator [58]. The measure-theoretic entropy of the proposed generator is given by the equation:

$$H_n = \lim_{n \rightarrow \infty} \left(\frac{-\sum_{B^n} P(B^n) \ln P(B^n)}{n} \right), \quad (5)$$

where $P(B^n)$ is the probability of occurrence of a binary subsequence B of length n . From Figure 2 the appropriate value of r is selected. For $r = 3.7$ the system has a chaotic behavior and the measured-entropy for $n = 3$ and $n = 4$ is calculated as $H_3 = 0.6928$ and $H_4 = 0.6923$, respectively. In order to create our random bit generator, we chose to use the $(r, a, b, c) = (3.7, 1.25, 5.001, 4)$ set of parameters.

4. Robot's Motion Controller

After obtaining the bitstream of required length, a sequence of motion commands must be generated from it. In this paper, the robot is tested for motion in four or eight directions. In the case of four directions the pairs of "0" and "1" are used to generate a motion command as shown in Table 2. In the case of eight motions, the commands which are used are shown in Table 3. For both cases, when the robot reaches the boundaries of the terrain or an obstacle, it stops and waits for the next motion command.

Table 2. Bits pair transformation into motion commands in four directions.

Bits Pair	Motion Command
00	up
01	right
10	down
11	left

Table 3. Bits transformation into motion commands in eight directions.

Bits Pair	Motion Command
000	up
001	up-right
011	right
101	down-right
110	down
111	down-left
100	left
010	up-left

The above method could be used for different kinds of terrain, both in size and shape. In the simulations of this work, a terrain with dimension 100×100 unit cells is used as a platform.

The coverage rate (C), which represents the effectiveness of the proposed method, as the amount of the surface that was visited by the robot, is calculated by:

$$C = \frac{1}{M} \cdot \sum_{i=1}^M I(i), \quad (6)$$

where $I(i)$ is the coverage situation for each cell and takes the value of “1” if the i cell is covered and “0” if it is not. According to the length of bitstream ($n = 100,000$ for four motions and $n = 150,000$ for eight motions), the controller produced 50,000 motion commands.

4.1. Inverse Pheromone Method

It has been noticed that in many cases the robot is showing unsatisfactory behavior. For example, sometimes the motion commands force it to move into cells that have previously been visited. This affects both the coverage rate and also the number of visits in each cell. For the improvement of the motion of the robot, the inverse pheromone method has been proposed. According to this method, the robot places an amount of this substance in each cell that it visits. Then, a virtual trail is created which fades over time. When the robot receives a motion command that leads it to a cell with pheromone, it stops and waits for the next motion command which could lead it to an uncovered cell. The result of this behavior is the improvement of the performance of the robot. It reduces the number of times that a cell is visited and as a consequence the coverage rate rises drastically.

4.2. Motion in Four Directions

The initial position of the robot was changed and tested for four different cases in order to study its effectiveness regarding the coverage rate. As it can be seen in Figure 3, the starting point plays a crucial role in the final result. When the robot starts from the (1,1) position, the 25.08% of the given space is covered. That rate grows steadily when the start is moved close to the middle of the terrain. This behavior is expected because in the middle of the workspace there is a high chance the motion command could lead the robot to an unvisited cell.

In addition, a different scenario was studied. A terrain with three obstacles was given and Figure 4 shows the results of the coverage in this case. As it can be noticed, especially for (99,99) initial point, the coverage is significantly higher than the previous and the regions that the robot visits multiple times are reduced.

As seen before, the results from the use of the bit generator are not very satisfactory yet. So, an inverse pheromone method is used and it was noticed that the improvement of the coverage is significant. From the starting point at (1,1), the coverage rate rises up to 81.91% (Figure 5). The amount of pheromone is selected to be 39 units and the coverage result is studied. As a result, the number of visits of same cells is reduced drastically which helps the robot to visit a great part of the terrain.

The same behavior also appears for a terrain with obstacles. When a pheromone is inserted, the number of visits of same cells is reduced to 1/10, in relation to motion in the same workspace without the use of the substance. The insertion of obstacles does not seem to reflect the coverage rate (Figure 6).

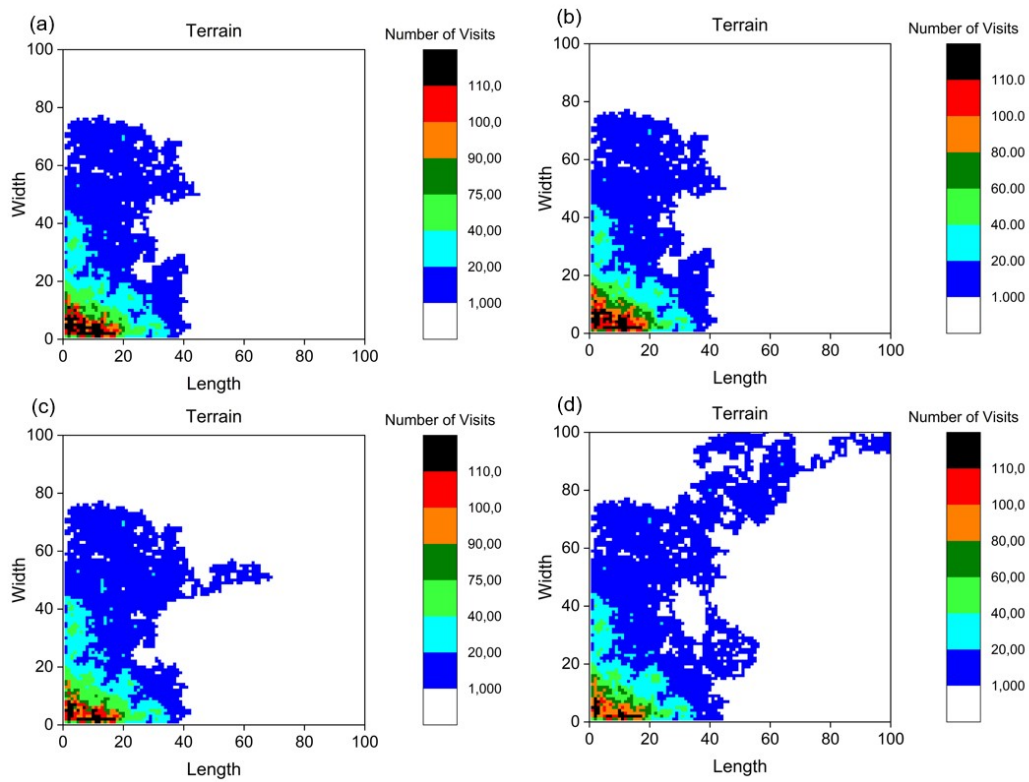


Figure 3. Terrain coverage with no pheromones and starting point at position (a) (1, 1), (b) (10, 10), (c) (50, 50) and (d) (99, 99).

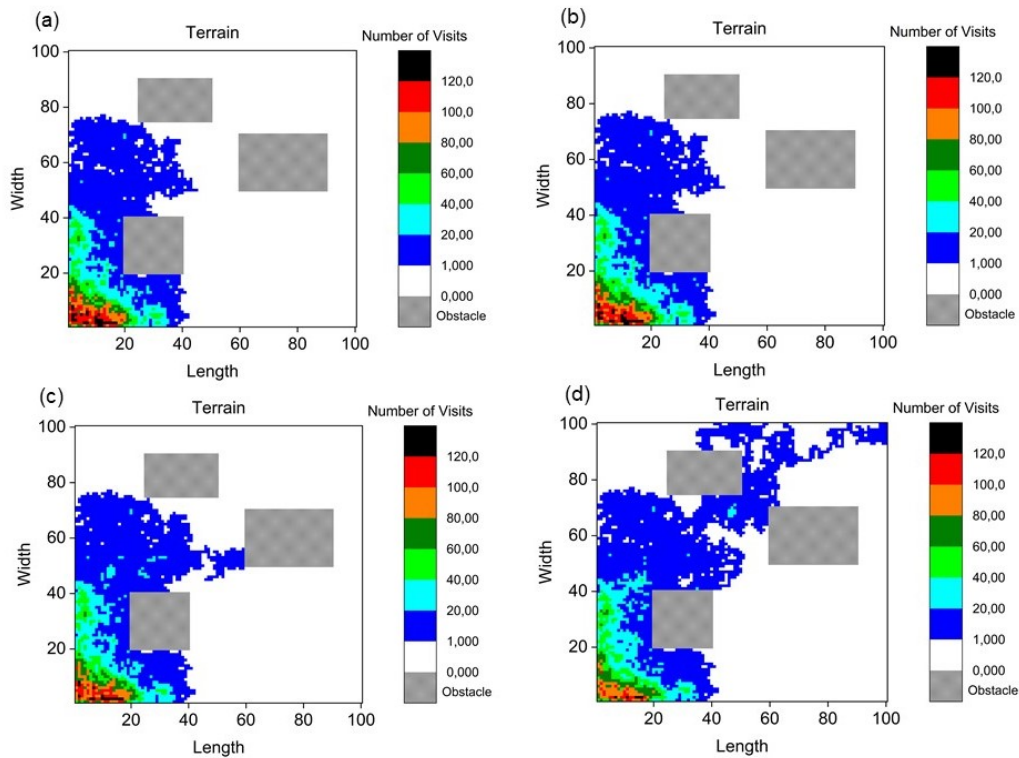


Figure 4. Terrain coverage with no pheromones and obstacles with starting point at position (a) (1, 1), (b) (10, 10), (c) (50, 50) and (d) (99, 99).

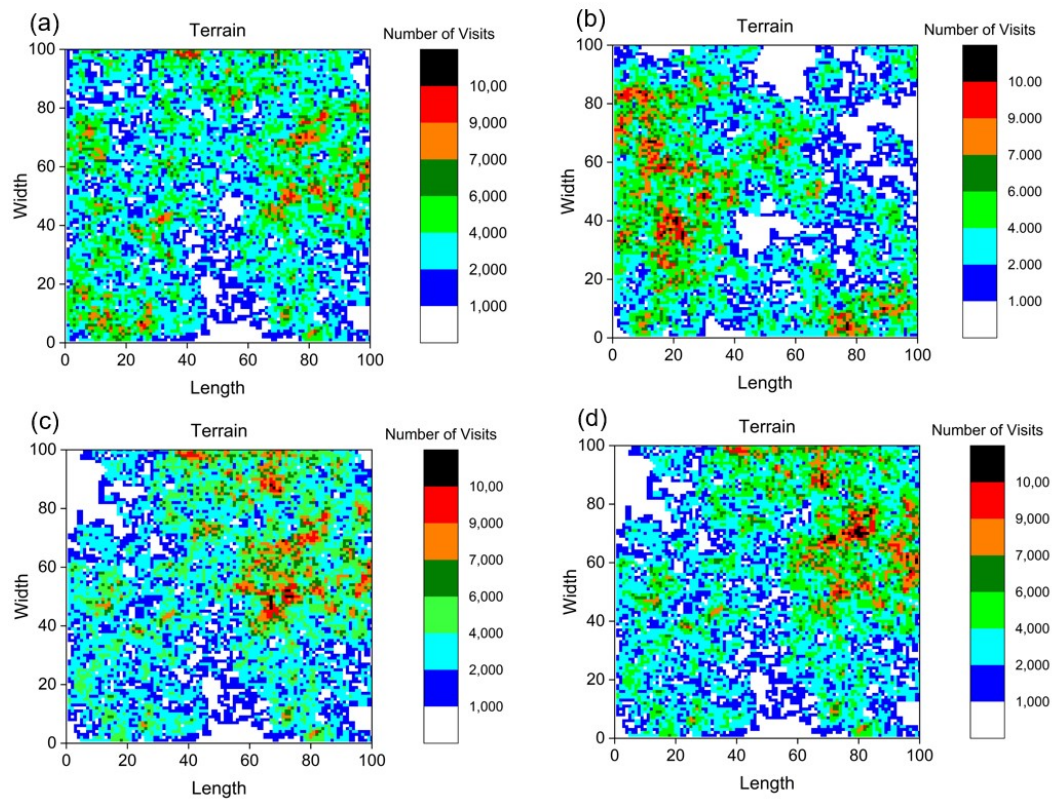


Figure 5. Terrain coverage with 39 pheromones and starting point at position (a) (1, 1), (b) (10, 10), (c) (50, 50) and (d) (99, 99).

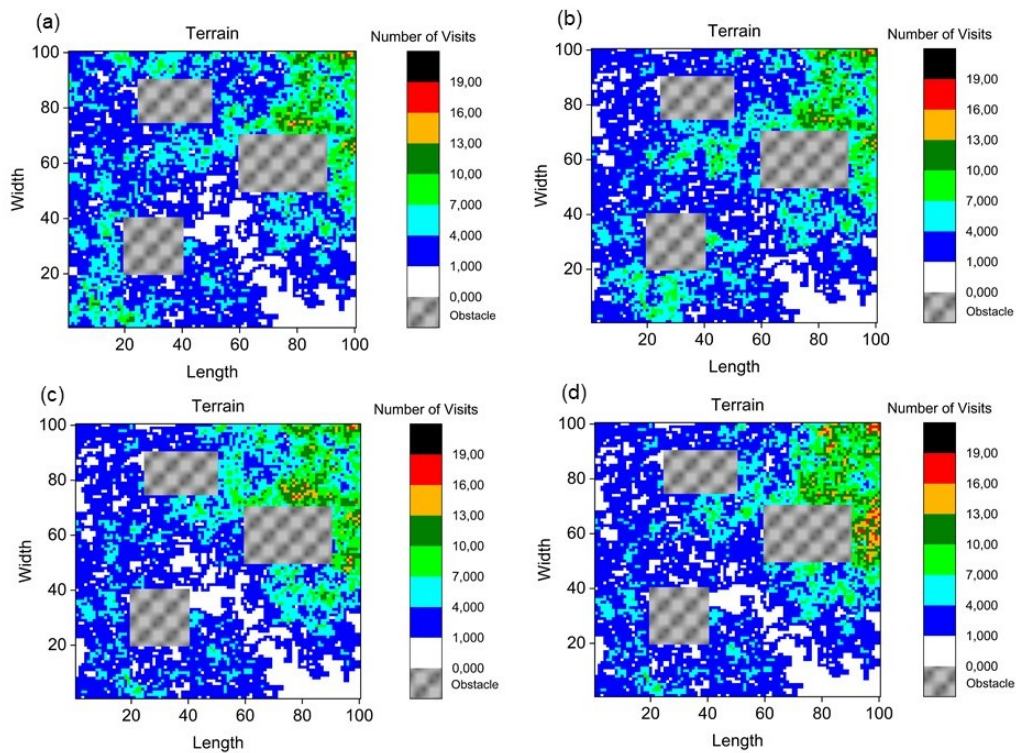


Figure 6. Terrain coverage with 39 pheromones and obstacles with starting point at position (a) (1, 1), (b) (10, 10), (c) (50, 50) and (d) (99, 99).

4.3. Motion in Eight Directions

One basic improvement into the motion trajectory of the robot can be achieved with the insertion of diagonal movements. This assumption could force the robot to move more naturally in the terrain. As a result from starting point at (1,1), 64.64% of the given space is covered which is by far better than the coverage with four motions (Figure 7). The next task is to reduce the regions with the biggest number of multiple visits. In order to achieve that, 37 pheromones are used and the results are very satisfactory as it can be seen in Figure 8. The black and red regions (with large number of visits) are reduced significantly.

Similar behavior appears for a terrain with three obstacles as it can be seen in Figure 9. At first from the starting point at (1,1), 84.52% of the given space is covered but many regions are visited multiple times. When 37 pheromones are inserted into the robot the rate rises to 88.68% for the same starting point (Figure 10) and the regions with multiple visits of same cells have disappeared.

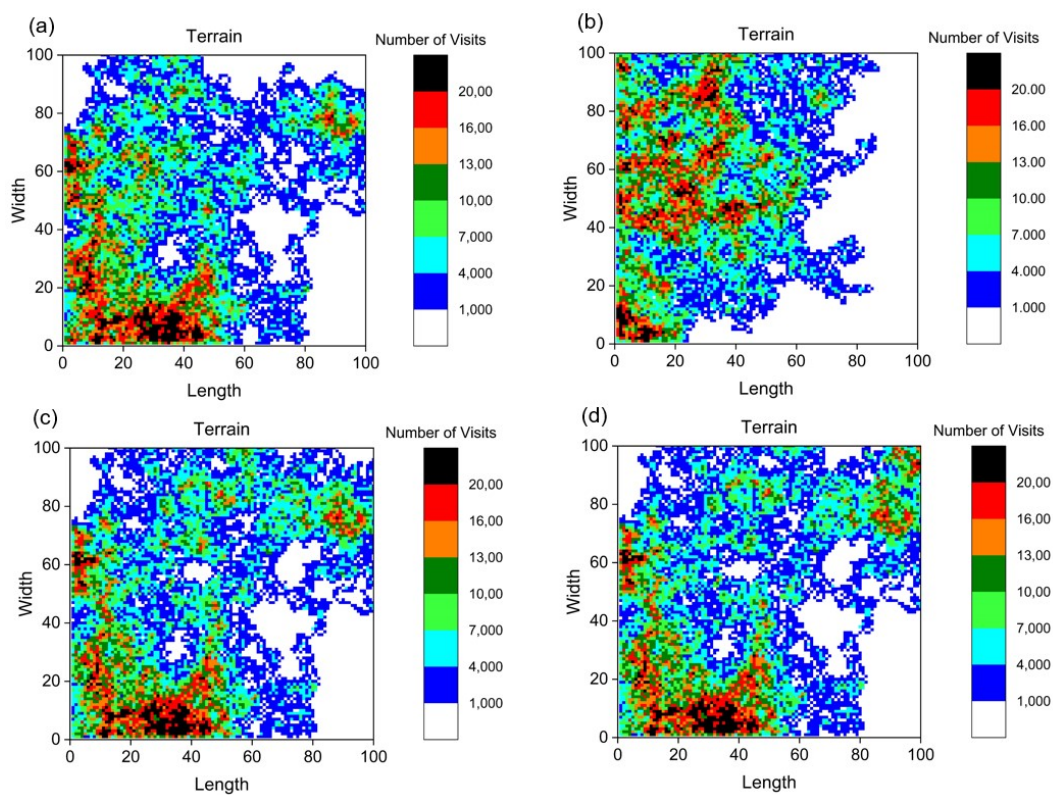


Figure 7. Terrain coverage with no pheromones and starting point at position (a) (1, 1), (b) (10, 10), (c) (50, 50) and (d) (99, 99).

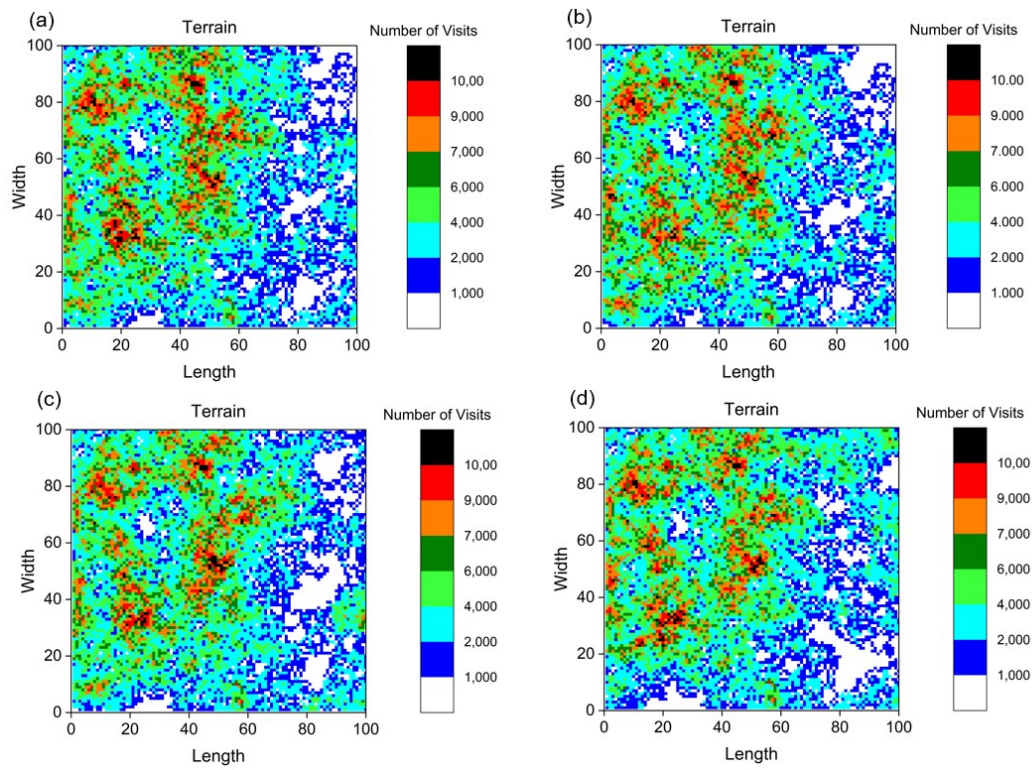


Figure 8. Terrain coverage with 37 pheromones and starting point at position (a) (1, 1), (b) (10, 10), (c) (50, 50) and (d) (99, 99).

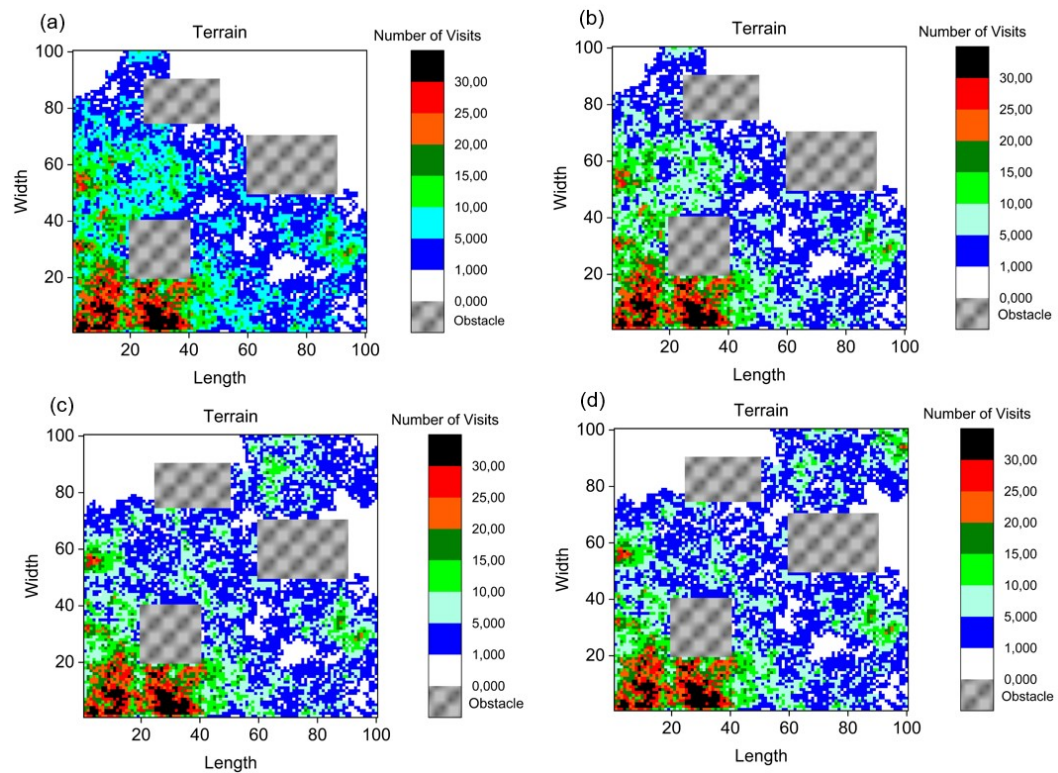


Figure 9. (Terrain coverage with zero amount of pheromone and obstacles with starting point at position (a) (1, 1), (b) (10, 10), (c) (50, 50) and (d) (99, 99).

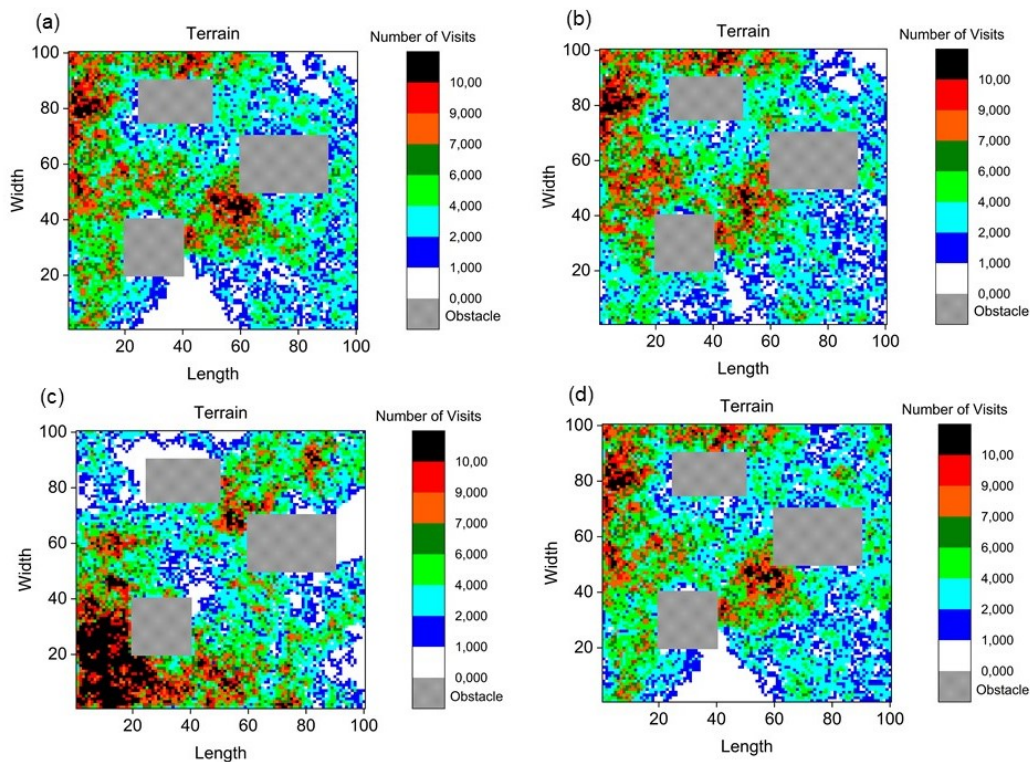


Figure 10. Terrain coverage with 37 pheromones and obstacles with starting point at the position (a) (1, 1), (b) (10, 10), (c) (50, 50) and (d) (99, 99).

From Figure 11, which shows the evolution of the coverage rate for different levels of pheromone, it can be noticed that the amount of 37 gives the best results. The difference between this amount and the others is quite impressive. This assumption is verified from Figure 12. The coverage rate is shown, according to the amount of pheromone for four and eight motion commands. It is clear that, with the entry of diagonal motions the coverage is steadily better than using only four direction commands.

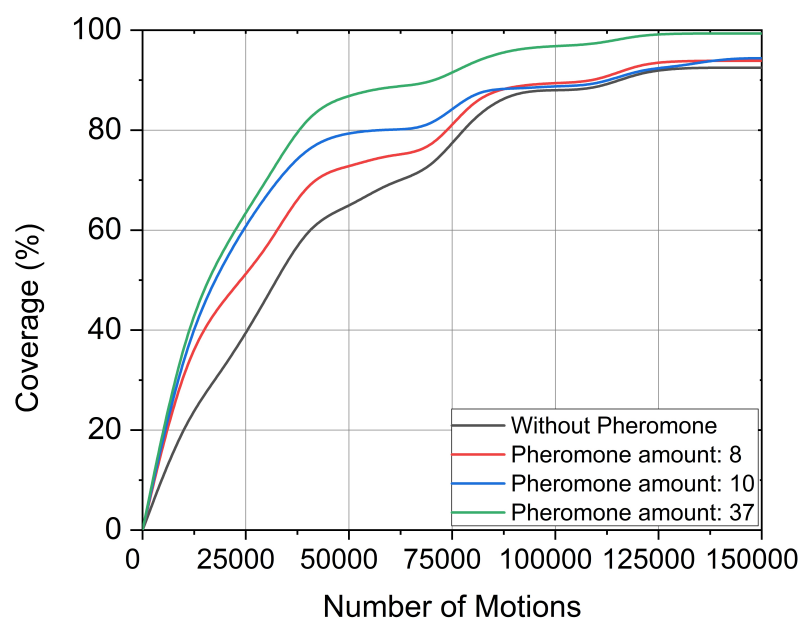


Figure 11. Coverage rate for different amounts of pheromones regarding the number of steps, in the case of eight motion commands.

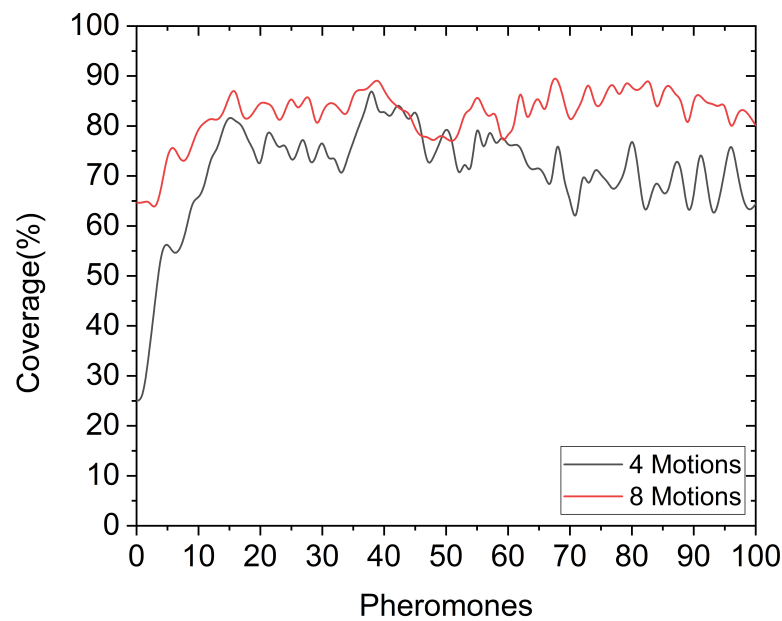


Figure 12. Coverage rate regarding the amount of pheromones, in the cases of four and eight motion commands.

Tables 4 and 5 show the coverage of the terrain for the same amount of pheromone. In the case of motion without the use of pheromone, compared to four motions the values of Table 5 are big enough in order to show the improvement when the diagonal movements are inserted. In addition, the highest value is different enough to show the utility of giving the robot the opportunity to move in many directions.

Table 4. Coverage rate for different numbers of pheromone traces starting from starting position (1,1) with four motion commands.

Number of Pheromone Traces	Coverage (%)
without	25.08
10	65.42
20	70.33
30	78.21
40	83.40
60	75.82
80	79.50
100	64.41

Table 5. Coverage rate for different numbers of pheromone traces starting from starting position (1,1) with eight motion commands.

Number of Pheromone Traces	Coverage (%)
without	64.64
10	79.66
20	84.91
30	83.05
40	87.01
60	78.75
80	87.57
100	80.09

4.4. Discussion

The analysis of the coverage performance for a chaotic mobile robot led us to some interesting results. First of all, the insertion of four more directions gave the opportunity of moving into new cells. This behavior was reflected in the coverage rate. In the case of eight motion commands and for the same number of steps ($n = 50,000$), more than half of the given terrain was visited by the robot. The method with the eight motion commands was by far better than the method with the four motion commands. Even with the insertion of pheromone and obstacles, the aforementioned behavior did not change. This behavior could clearly be seen from Figure 12, in which the coverage in the case of eight motion commands was better for the majority of the pheromone levels. In Figure 11, the improvement with the inverse pheromone method in the coverage in regard to the number of motions is obvious. It can be noticed that for almost 10,000 motions the coverage in the three cases of pheromone numbers is the same. After that limit and for further step commands the difference between them is pretty clear.

An important issue to consider in future works is an analytical comparative study between our chaotic path planning method using pheromone and the others that have been developed in recent years. Such a study will further explore the pros and cons of each different approach with regards to coverage, number of visits in each cell, complexity, and execution time.

5. Conclusions

In this paper, a novel path planning method for autonomous mobile robots, in order to cover a given terrain quickly and efficiently, was presented. The proposed method was based on a chaotic motion generator, which used a modified logistic map in order to design a CRBG, from which the produced bitstream was converted into motion commands in four and eight directions. For fast and efficient coverage, the method of inverse pheromone was applied to further improve the coverage rate. Extensive simulations were performed that indicate an efficient coverage for different scenarios regarding the robot's starting position with or without obstacles. Future aspects of this work include the use of different chaotic maps to generate the motion, different motion generation techniques, the application to multi-robot setups, the motion generation for a continuous grid, and also the implementation of the method in an experimental setup.

Author Contributions: Conceptualization, C.K.V.; Formal analysis, C.K.V., E.K.P., L.M.; Methodology, C.K.V., E.K.P., L.M.; Software, E.K.P., L.M.; Supervision, C.K.V., I.N.S., H.E.N., G.S.T., K.P.V.; Validation, E.K.P., L.M., C.K.V.; Visualization, E.K.P.; Writing, E.K.P., L.M., C.K.V.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Siegwart, R.; Nourbakhsh, I.R.; Scaramuzza, D. *Introduction to Autonomous Mobile Robots*; MIT Press: Cambridge, MA, USA, 2011.
2. Palacin, J.; Salse, J.A.; Valgañón, I.; Clua, X. Building a mobile robot for a floor-cleaning operation in domestic environments. *IEEE Trans. Instrum. Meas.* **2004**, *53*, 1418–1424. [\[CrossRef\]](#)
3. Tadokoro, S. (Ed.) *Rescue Robotics: DDT Project on Robots and Systems for Urban Search and Rescue*; Springer Science & Business Media: Berlin, Germany, 2009.
4. Tavera, M.J.; Dutra, M.S.; Diaz, E.Y.V.; Lengerke, O. Implementation of Chaotic Behavior on a Fire Fighting Robot. In *Mechatronics Series I—Intelligent Transportation Vehicles*; Dutra, M.S., Lengerke, O., Eds.; Bentham Science Publishers: Sharjah, UAE, 2009.
5. Murphy, R.R.; Kravitz, J.; Stover, S.L.; Shoureshi, R. Mobile robots in mine rescue and recovery. *IEEE Robot. Autom. Mag.* **2009**, *16*, 91–103. [\[CrossRef\]](#)
6. Krotkov, E.; Blitch, J. The defense advanced research projects agency (DARPA) tactical mobile robotics program. *Int. J. Robot. Res.* **1999**, *18*, 769–776. [\[CrossRef\]](#)

7. Martins-Filho, L.S.; Macau, E.E. Trajectory planning for surveillance missions of mobile robots. In *Autonomous Robots and Agents*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 109–117.
8. Martins-Filho, L.S.; Macau, E.E. Patrol mobile robots and chaotic trajectories. *Math. Probl. Eng.* **2007**, *2007*, 61543. [\[CrossRef\]](#)
9. Gohari, P.S.; Mohammadi, H.; Taghvaei, S. Using chaotic maps for 3D boundary surveillance by quadrotor robot. *Appl. Soft Comput.* **2019**, *76*, 68–77. [\[CrossRef\]](#)
10. Li, C.; Song, Y.; Wang, F.; Liang, Z.; Zhu, B. Chaotic path planner of autonomous mobile robots based on the standard map for surveillance missions. *Math. Probl. Eng.* **2015**, *2015*, 263964. [\[CrossRef\]](#)
11. Portugal, D.; Rocha, R. Msp algorithm: Multi-robot patrolling based on territory allocation using balanced graph partitioning. In Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, 22–26 March 2010; pp. 1271–1276.
12. Huang, Q.J.; Nonami, K. Humanitarian mine detecting six-legged walking robot and hybrid neuro walking control with position/force control. *Mechatronics* **2003**, *13*, 773–790. [\[CrossRef\]](#)
13. Yim, M.; Roufas, K.; Duff, D.; Zhang, Y.; Eldershaw, C.; Homans, S. Modular reconfigurable robots in space applications. *Auton. Robot.* **2003**, *14*, 225–237. [\[CrossRef\]](#)
14. Abdellilah, H.; Mohamed, B.; Abdellah, M.; Youcef, M.; Réda, A.M. Depth advanced control of an autonomous underwater robot. *Int. J. Model. Identif. Control* **2016**, *26*, 336–344. [\[CrossRef\]](#)
15. Ośmiałowski, B. On path planning for mobile robots: Introducing the mereological potential field method in the framework of mereological spatial reasoning. *J. Autom. Mob. Robot. Intell. Syst.* **2009**, *3*, 24–33.
16. Galceran, E.; Carreras, M. A survey on coverage path planning for robotics. *Robot. Auton. Syst.* **2013**, *61*, 1258–1276. [\[CrossRef\]](#)
17. Choset, H. Coverage for robotics—A survey of recent results. *Ann. Math. Artif. Intell.* **2001**, *31*, 113–126. [\[CrossRef\]](#)
18. Fahmy, A.A. Chaotic mobile robot workspace coverage enhancement. *J. Autom. Mob. Robot. Intell. Syst.* **2012**, *6*, 33–38.
19. Fahmy, A.A. Implementation of the chaotic mobile robot for the complex missions. *J. Autom. Mob. Robot. Intell. Syst.* **2012**, *6*, 8–12.
20. Low, K.H.; Dolan, J.M.; Khosla, P. Information-theoretic approach to efficient adaptive path planning for mobile robotic environmental sensing. In Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling, Thessaloniki, Greece, 19–23 September 2009.
21. Garrido, S.; Moreno, L.; Blanco, D.; Jurewicz, P. Path planning for mobile robot navigation using voronoi diagram and fast marching. *Int. J. Robot. Autom.* **2011**, *2*, 42–64.
22. Volos, C.K.; Kyprianidis, I.M.; Stouboulos, I.N.; Stavrinides, S.G.; Anagnostopoulos, A.N. An autonomous mobile robot guided by a chaotic true random bits generator. In *Chaos and Complex Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 337–343.
23. Volos, C.K.; Kyprianidis, I.M.; Stouboulos, I.N. A chaotic path planning generator for autonomous mobile robots. *Robot. Auton. Syst.* **2012**, *60*, 651–656. [\[CrossRef\]](#)
24. Petavratzis, E.K.; Volos, C.K.; Stouboulos, I.N.; Nistazakis, H.E.; Kyritsi, K.G.; Valavanis, K.P. Coverage Performance of a Chaotic Mobile Robot Using an Inverse Pheromone Model. In Proceedings of the 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, Greece, 13–15 May 2019; pp. 1–4.
25. Nasr, S.; Mekki, H.; Bouallegue, K. A multi-scroll chaotic system for a higher coverage path planning of a mobile robot using flatness controller. *Chaos Solitons Fractals* **2019**, *118*, 366–375. [\[CrossRef\]](#)
26. Goyal, J.K.; Nagla, K.S. A new approach of path planning for mobile robots. In Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, India, 24–27 September 2014; pp. 863–867.
27. Koziol, S.; Hasler, P.; Stilman, M. Robot path planning using field programmable analog arrays. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 1747–1752.
28. Kanaya, M.; Cheng, G.X.; Watanabe, K.; Tanaka, M. Shortest path searching for robot walking using an analog resistive network. In Proceedings of the IEEE International Symposium on Circuits and Systems-ISCAS'94, London, UK, 30 May–2 June 1994; Voluem 6, pp. 311–314.

29. Cheng, G.X.; Ikegami, M.; Tanaka, M. A resistive mesh analysis method for parallel path searching. In Proceedings of the 34th Midwest Symposium on Circuits and Systems, Monterey, CA, USA, 14–17 May 1992; pp. 827–830.
30. Li, C.H.; Song, Y.; Wang, F.Y.; Wang, Z.Q.; Li, Y.B. A chaotic coverage path planner for the mobile robot based on the Chebyshev map for special missions. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 1305–1319. [[CrossRef](#)]
31. Curiac, D.I.; Baniyas, O.; Volosencu, C.; Curiac, C.D. Novel Bioinspired Approach Based on Chaotic Dynamics for Robot Patrolling Missions with Adversaries. *Entropy* **2018**, *20*, 378. [[CrossRef](#)]
32. Pippin, C.; Christensen, H.; Weiss, L. Performance based task assignment in multi-robot patrolling. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; pp. 70–76.
33. Moysis, L.; Petavratzis, E.; Volos, C.; Nistazakis, H.; Stouboulos, I. A chaotic path planning generator based on logistic map and modulo tactics. *Robot. Auton. Syst.* **2019**, *124*, 103377. [[CrossRef](#)]
34. Rössler, O.E. An equation for continuous chaos. *Phys. Lett. A* **1976**, *57*, 397–398. [[CrossRef](#)]
35. Lorenz, E.N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [[CrossRef](#)]
36. Sprott, J.C. Some simple chaotic flows. *Phys. Rev. E* **1994**, *50*, R647. [[CrossRef](#)] [[PubMed](#)]
37. Banerjee, S.; Rondoni, L.; Mitra, M. *Applications of Chaos and Nonlinear Dynamics in Science and Engineering*; Springer: Berlin, Germany, 2015; Volume 4.
38. Strogatz, S.H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*; CRC Press: Boca Raton, FL, USA, 2018.
39. Kocarev, L. Chaos-based cryptography: A brief overview. *IEEE Circuits Syst. Mag.* **2001**, *1*, 6–21. [[CrossRef](#)]
40. He, H.; Wu, Y.; Zhang, B.; Zhang, D.; Tian, Y.; Wang, K.; Meng, H.; Hou, M. Discrete chaotic synchronization and secure communication design. In Proceedings of the 2009 Fifth International Conference on Natural Computation, Tianjin, China, 14–16 August 2009; Volume 5, pp. 473–476.
41. Kabi, K.K.; Pradhan, C.; Saha, B.J.; Bisoi, A.K. Comparative study of image encryption using 2D chaotic map. In Proceedings of the 2014 International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 1–2 March 2014; pp. 105–108.
42. Parida, R.R.; Singh, S.; Pradhan, C. Analysis of Color Image Encryption Using Multidimensional Bogdanov Map. In *Histopathological Image Analysis in Medical Decision Making*; IGI Global: Hershey, PA, USA, 2019; pp. 202–225.
43. Rathore, D.; Suryavanshi, A. A proficient Image Encryption using chaotic map approach. *Int. J. Comput. Appl.* **2016**, *134*, 20–24. [[CrossRef](#)]
44. Yang, T.; Wu, C.W.; Chua, L.O. Cryptography based on chaotic systems. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **1997**, *44*, 469–472. [[CrossRef](#)]
45. Cheng, C.T.; Fallahi, K.; Leung, H.; Chi, K.T. Cooperative path planner for UAVs using ACO algorithm with Gaussian distribution functions. In Proceedings of the 2009 IEEE International Symposium on Circuits and Systems, Taipei, Taiwan, 24–27 May 2009; pp. 173–176.
46. Garcia, M.P.; Montiel, O.; Castillo, O.; Sepúlveda, R.; Melin, P. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl. Soft Comput.* **2009**, *9*, 1102–1110. [[CrossRef](#)]
47. Rosalie, M.; Danoy, G.; Chaumette, S.; Bouvry, P. Chaos-enhanced mobility models for multilevel swarms of UAVs. *Swarm Evol. Comput.* **2018**, *41*, 36–48. [[CrossRef](#)]
48. Verhulst, P.F. La loi d'accroissement de la population. *Nouv. Mem. Acad. Roy. Soc. Belle-lettr. Bruxelles* **1845**, *8*, 1–58.
49. May, R.M. Simple mathematical models with very complicated dynamics. *Nature* **1976**, *261*, 459–467. [[CrossRef](#)]
50. Ausloos, M.; Dirickx, M. (Eds.) *The Logistic Map and the Route to Chaos: From the Beginnings to Modern Applications*; Springer Science & Business Media: Berlin, Germany, 2006.
51. Patidar, V.; Sud, K.K.; Pareek, N.K. A pseudo random bit generator based on chaotic logistic map and its statistical testing. *Informatica* **2009**, *33*, 441–452.
52. Lloyd, A.L. The coupled logistic map: A simple model for the effects of spatial heterogeneity on population dynamics. *J. Theor. Biol.* **1995**, *173*, 217–230. [[CrossRef](#)]
53. Li, C.; Xie, T.; Liu, Q.; Cheng, G. Cryptanalyzing image encryption using chaotic logistic map. *Nonlinear Dyn.* **2014**, *78*, 1545–1551. [[CrossRef](#)]

54. Tarasova, V.V.; Tarasov, V.E. Logistic map with memory from economic model. *Chaos Solitons Fractals* **2017**, *95*, 84–91. [[CrossRef](#)]
55. McGonigal, G.; Elmasry, M. Generation of noise by electronic iteration of the logistic map. *IEEE Trans. Circuits Syst.* **1987**, *34*, 981–983. [[CrossRef](#)]
56. Von Neumann, J. Various techniques used in connection with random digits. *Appl. Math. Ser.* **1951**, *12*, 5.
57. NIST: Security Requirements for Cryptographic Modules, FIPS PUB 140–2. Available online: <https://csrc.nist.gov/publications/detail/fips/140/2/final> (accessed on 3 December 2019).
58. Fraser, A.M. Information and entropy in strange attractors. *IEEE Trans. Inf. Theory* **1989**, *35*, 245–262. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).