Information Technology and Management Science

doi: 10.2478/v10313-012-0007-6

2012 /15

Motion Planning of an Autonomous Robot in Closed Space with Obstacles

Edvards Valbahs[1], Peter Grabusts[2], [1-2]Rezekne Higher Education Institution

***Abstract* – The paper deals with path planning software for a mobile robotic platform. The aim of the research paper is to analyse path planning algorithms that comprise the design of simulation software. The software is necessary as an environment model to obtain the simulation data. The simulation application is based on the Rapidly-Exploring Random Tree (RRT) algorithm and Simulated Annealing (SA). The results of the thorough analysis have been used to achieve optimal path planning algorithms.**

***Keywords* – robotic, robot, RRT, Simulated Annealing, path planning**

I. INTRODUCTION

It is necessary to provide iterative motions among points of the goals in order to reach the wide range of the robotic application. For instance, in industry a mobile robotic platform can replace any components between a storehouse and an assembly department. Ammunition replacement is widely used in military forces. It can be used in ports, airports, recycling sites and etc. Mobile robots can be used in monitoring if it is necessary to observe control points in a secret place. There are a lot of scenarios where the iterative motion is applied [2].

The environment used for such mobile robots is complicated, various, non-structural and dynamic by nature. Robots must get rid of the obstacles which are different in size, form, location and can appear or disappear at any time. When avoiding the obstacles, the possible collision risk, sensor information and movement planning imprecision, error location and uneven surface are always foreseen. Robots can be damaged, trapped or embedded in any construction, etc. Environmental imprecision is always very hazardous for mobile robots [2].

At the same time, the effective utilization is required from robots. Robots should work as much as possible and fast and safely [2], [4], [5], [10], [12], [13]. However, the safety of people and the place of their living are primary, and the same can be said about the safety of robots.

Algorithms for motion planning have proved themselves as optimal methods in this planning. The best conditions are normally measured with a distance. However, it is possible to measure the conditions mentioned differently. For instance, in order to find safe ways, space robots take into consideration surface, roughness and slope. However, the efficiency of complex, dynamic and partly unknown spaces has not been investigated for a long time. Nowadays, there has been little research conducted on the choice of motion planning in dynamic conditions. Approaches [3], [9], [11], [14], [15] admit that the environmental structure is known a priori. Admittedly, the unknown environment is static and unchangeable during some period of time in a source [15], [11]. Sources [6] - [8], [16] reflect uncertainty except for the two previous statements.

Commonly, in order to work out any motion planning system, which is used in a mobile robotic platform, it is necessary to perform the following tasks:

- to familiarize with the essence of motion planning task, i.e., to observe the algorithm of classical and modern planning;
- to identify the advantages and disadvantages of the algorithm;
- to select the most relevant algorithm;
- to work out the system design.

The representation of working area for the robot can vary from continuous geometrical behaviour to the approved decomposition of topological maps. The first step for any motion planning system is permanent environmental model transformation in the map relevant to the motion planning algorithm chosen if it is possible. The motion planning is distinguished due to the influence on the discrete decomposition. The three basic decomposition strategies can be mentioned [2]:

- motion maps: to show the package of motion in free space;
- unit decomposition: to distinguish empty from occupied units;
- potential field: the robot reaches the target unit if it follows the least resistance direction. This function is sometimes called a navigation function if additional conditions are added.

Then some motion planning algorithms can be briefly envisaged, i.e., Rapidly-Exploring Random Tree (RRT) and Simulated Annealing (SA).

II. GOALS

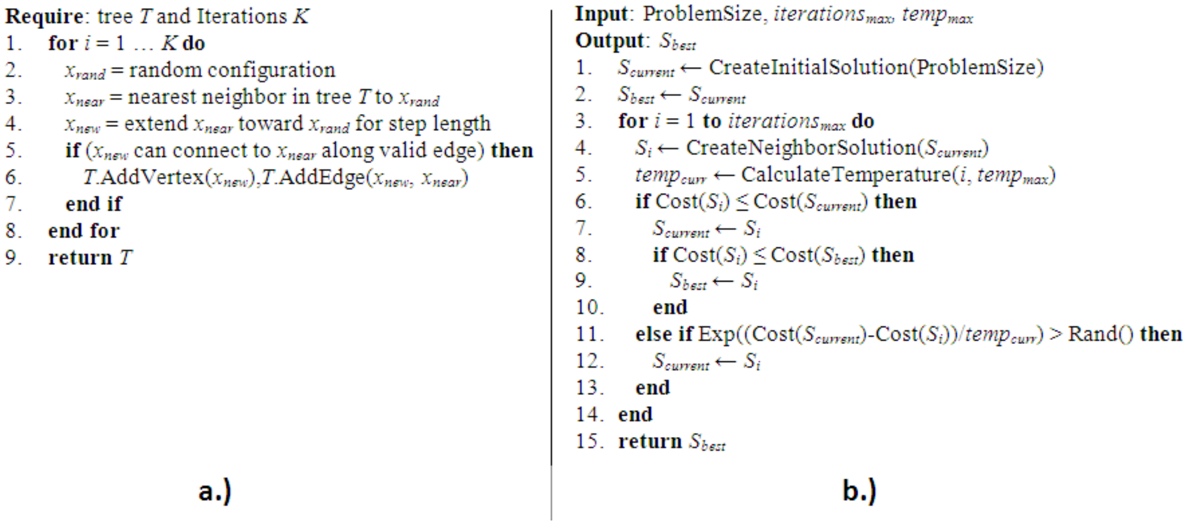
According to the above-mentioned statement, it is becoming rather interesting to compare RRT and SA algorithms by means of simulation.

The aim of the research paper is to analyse motion planning algorithms that contain the design of simulation software. The software is necessary as an environment model to obtain the simulation data. The software is based on RRT algorithm. The simulation data provide the opportunity to conduct thorough analyses for a selected algorithm. The analysis involves the simulation data interpretation and comparison with other data obtained using the SA algorithms for motion planning.

RRT – in the last decade the motion planning algorithms were proved to be efficient ones on the basis of step sample and they were worked out theoretically as probable definiteness (see Fig. 1). The theoretical limits of these algorithms have not been solved so far. It is proved that RRT algorithms always converge but do not insure the optimal magnitude. The storage of the search graph starts to become impractical at high dimensions, and the need to use probabilistically complete algorithms such as RRT increases.

The SA method [17], [18] is widely used in applied science (Fig. 1). The well-known traveling salesman problem has effectively been solved by means of this method. For instance, the arrangement of many circuit elements on a silicon substrate is considerably improved to reduce interference among the elements [1], [19].

The use in practice is related to autonomous robots that move in the space and are able to plan a route on their own. One of such robots exiting in our everyday life is autonomous vacuum cleaner. Autonomous vacuum cleaners do not usually use the motion planning algorithm. They are based on different simple algorithms, for example spiral cleaning, crossing the premises by avoiding the walls, and their moving is casual after touching the walls. The philosophy of this design was proposed by the scientists of Massachusetts Institute of Technology. Robots must be like insects having primitive controlling devices aimed at the environment. As a result, an autonomous vacuum cleaner is very effective in cleaning premises, but much more time is required for them as compared with work made by a person. There is a drawback, the autonomous vacuum cleaning robot cleans one and the same place many times, but other places are cleaned only once. The use of motion planning algorithms can raise the efficiency of an autonomous vacuum cleaner.



```
Require: tree $T$ and Iterations $K$
for $i = 1 \ldots K$ do
  $x_{rand}$ = random configuration
  $x_{near}$ = nearest neighbor in tree $T$ to $x_{rand}$
  $x_{new}$ = extend $x_{near}$ toward $x_{rand}$ for step length
  if ($x_{new}$ can connect to $x_{near}$ along valid edge) then
    $T$.AddVertex($x_{new}$),$T$.AddEdge($x_{new}$, $x_{near}$)
  end if
end for
return $T$
```

a.)

```
Input: ProblemSize, $iterations_{max}$, $temp_{max}$
Output: $S_{best}$
$S_{current} \leftarrow$ CreateInitialSolution(ProblemSize)
$S_{best} \leftarrow S_{current}$
for $i$ = 1 to $iterations_{max}$ do
  $S_i \leftarrow$ CreateNeighborSolution($S_{current}$)
  $temp_{curr} \leftarrow$ CalculateTemperature($i$, $temp_{max}$)
  if Cost($S_i$) $\leq$ Cost($S_{current}$) then
    $S_{current} \leftarrow S_i$
    if Cost($S_i$) $\leq$ Cost($S_{best}$) then
      $S_{best} \leftarrow S_i$
    end
  else if Exp((Cost($S_{current}$)-Cost($S_i$))/$temp_{curr}$) > Rand() then
    $S_{current} \leftarrow S_i$
  end
end
return $S_{best}$
```

b.)

Fig. 1. Pseudocodes: a.) – for RRT; b.) – for SA

III. ASSUMPTIONS

In order to fulfil the aim of the research paper, the following conditions are introduced:

- the premises, where an object moves;
- the robot (or object) moves around the premises;
- the path, along which the robot moves in the premises.

The premises are presented as a two-dimensional plane. The plane of premises is equally divided into cells. The cell dimensions are equal to the size of the robot that moves in the premises. The robot moves only one cell forward and back. During one motion, the object can move to one cell out of eight empty ones (eight cells around one cell) taking into consideration that a cell is not occupied by the obstacle.

RRT algorithm is introduced to the software as the only one, which calculates the motion planning task (fully covers all empty space). The results have been compared with the SA algorithm data.

In this research paper, both algorithms have been compared practically using and combining different locations of obstacles in the unchangeable two-dimensional space. All the results have been obtained using one and the same computer (2.66 GHz processor and 2GB RAM), operating systems (Ubuntu 12.04.1 LST Linux) have been used. The following information has been collected:

- the coverage density for each cell;
- the time, which was necessary for both algorithms to plan the route.

The given illustrations show coverage density (see Fig. 3). The density scale (see Fig. 2) is the same for all coverage densities. Coverage density shows how often the robot covers each cell.

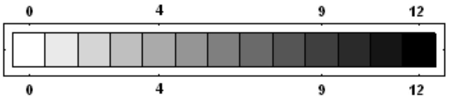
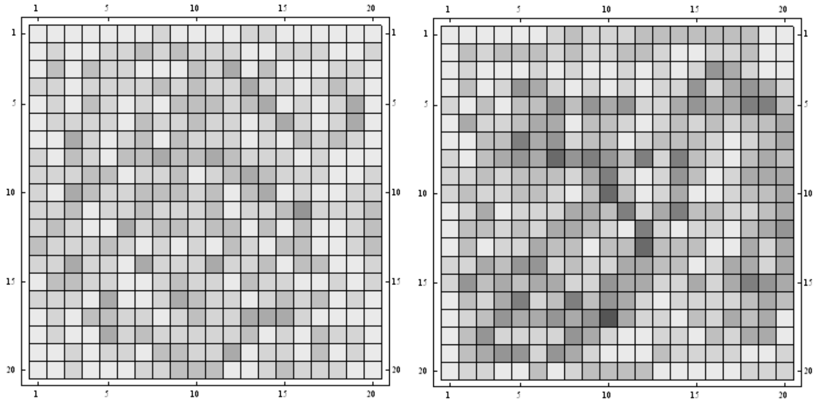
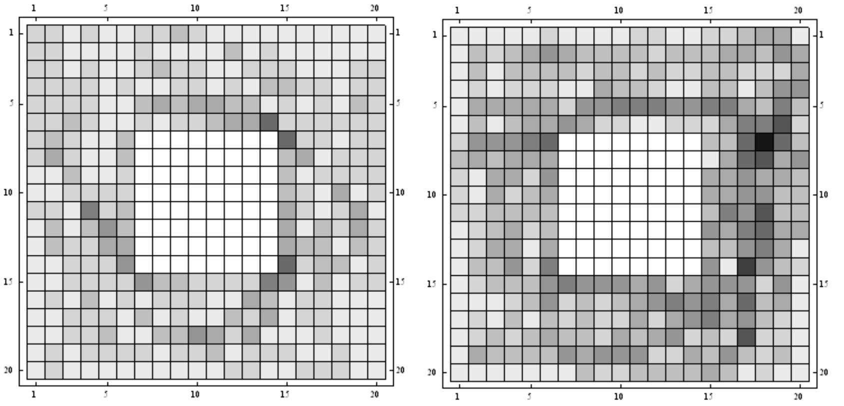


Fig. 2. Density scale (white – uncovered; black – covered the most)



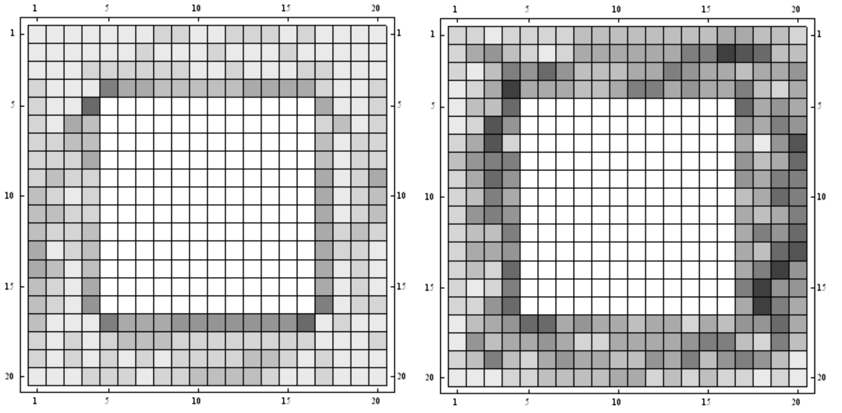
a.) **b.)**

Fig. 3. Coverage density for the space without obstacles: a.) – for SA; b.) – for RRT



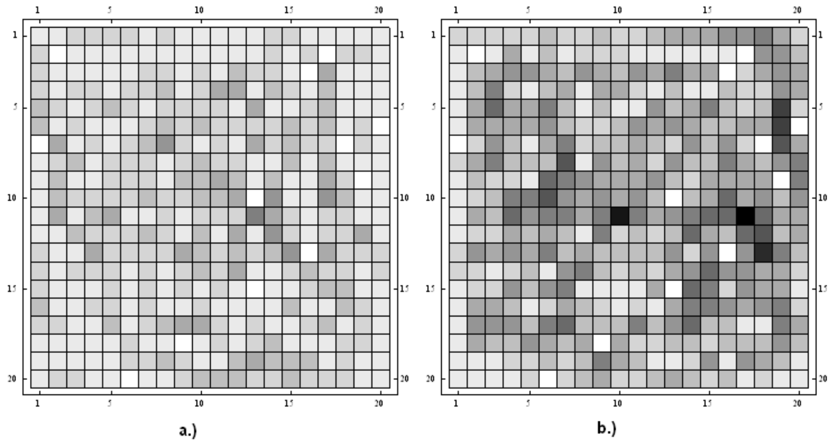
a.) **b.)**

Fig. 4. Coverage density with the obstacle consisting of 64 cells (the obstacle is in the middle of the premises): a.) – for SA; b.) – for RRT



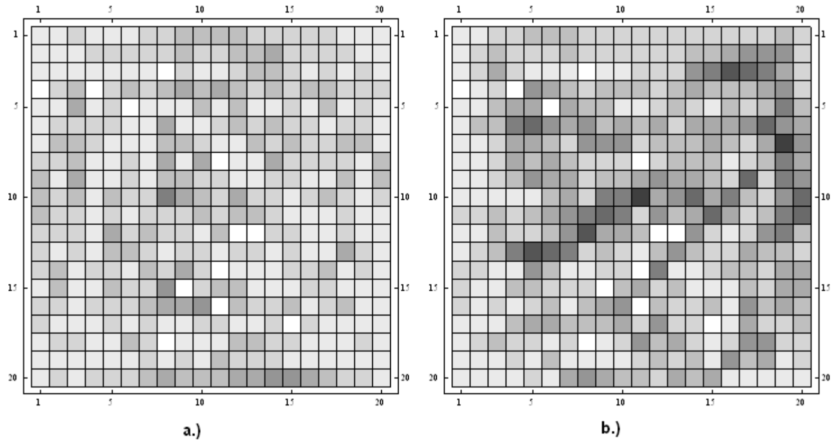
a.) **b.)**

Fig. 5. Coverage density with the obstacle consisting of 144 cells (the obstacle is in the middle of the premises): a.) – for SA; b.) – for RRT



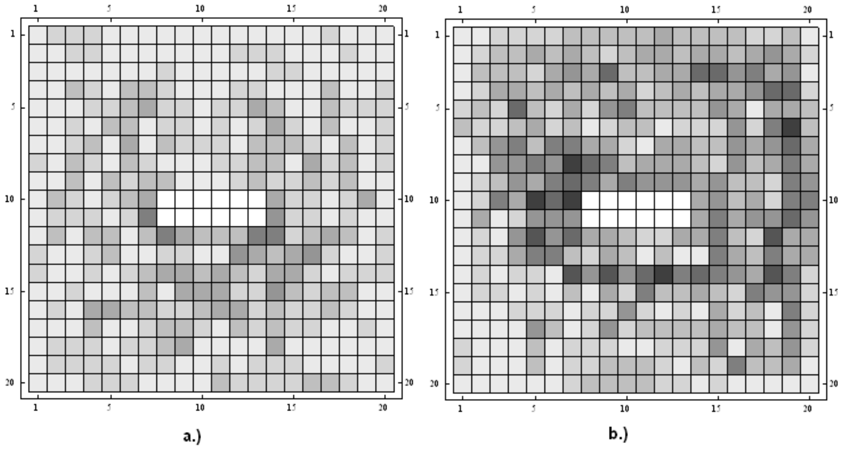
a.) b.)

Fig. 6. Coverage density with the 12 random obstacles: a.) – for SA; b.) – for RRT



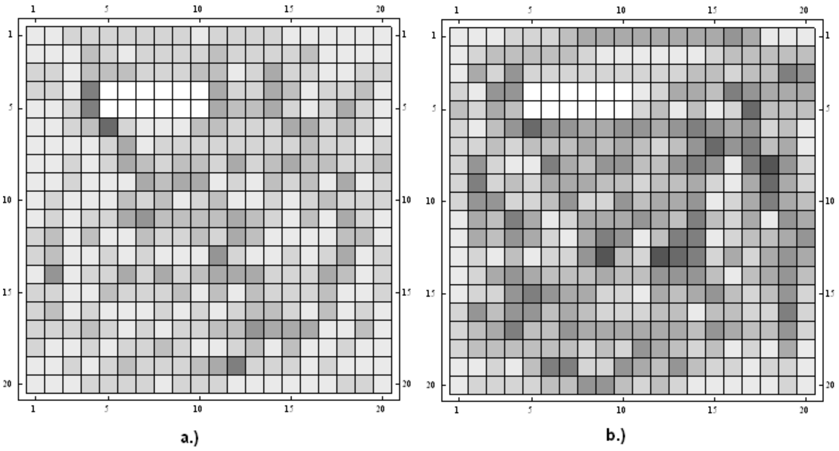
a.) b.)

Fig. 7. Coverage density with another set of the 12 random obstacles: a.) – for SA; b.) – for RRT



a.) b.)

Fig. 8. Coverage density with the obstacle consisting of 12 cells (the obstacle is in the middle of the premises): a.) – for SA; b.) – for RRT



a.)

b.)

Fig. 9. Coverage density with the obstacle consisting of 12 cells: a.) – for SA; b.) – for RRT

IV. RESULTS

Different coverage areas of the premises using RRT and SA algorithms in the identical conditions are shown in the research paper. The size of the premises during simulation has not been changed (20 x 20 cells), but the dimensions, placement and number of obstacles in the premises have been changed. The simplest case is when the premises do not contain any obstacles, i.e., they are completely empty. Thus, the simulation has been performed in the empty premises using the both algorithms. This process has been repeated several times. It is very important to stress that the principle of previously mentioned process has been used for the placement of any obstacles in the premises, i.e., not only in the case of completely empty premises. The simulation data has been calculated as a mean ratio of the number of simulation iterations. During each simulation, the following characteristics have been calculated:

- the number of initial empty cells;
- the time necessary to plan the route in order to cover the whole premises;
- the total number of steps, taking into account the facts that the robot can clean one and the same cell several times.

Time periods of motion planning for the RRT and SA algorithms have been compared (i.e., the RRT periods of time has been divided by the SA intervals of time). The ratio of the RRT time periods to the SA time intervals is between 1.26 and 1.88 for different placements of obstacles. It means that motion planning for the SA algorithm is at least 1.26 times faster. It is very important to stress that periods of time of motion planning are between 1 and 3 minutes (3 minutes applies to RRT). Better route planning has been performed by means of the SA algorithm, as well as the premises have been covered more regularly that may be observed from the surface diagram. The SA algorithm shows better results than RRT in case the number of occupied cells decreases. Fewer steps are necessary in the case of SA, i.e., by 27.11% to 92.77% less. It is very important to stress that the number of steps has been changing between 542 and 1361. The total period of time for the route can be between 542 and 1361 s (9 and 22.7 minutes) if considering that a real robot performs 1 step per second (see Table I).

TABLE I

SOME TYPICAL COMPARISON VALUES BETWEEN RRT AND SA

See Fig.	Number of empty cells	t_{mp_RRT} (where t_{mp} – the time period of motion planning)	N_{RRT} (where N – the number of steps)	t_{mp_SA} (where t_{mp} – the time period of motion planning)	N_{SA} (where N – the number of steps)	t_{mp_RRT}/t_{mp_SA}	N_p (%) (where N_p - less number of steps for the SA)
3	400	188	1092	114	796	1.65	27.11
4	336	163	1078	95	655	1.72	39.24
5	256	93	1021	74	542	1.26	46.91
6	388	239	1361	107	723	1.88	92.77
7	388	164	1142	99	752	1.52	53.47
8	388	162	1293	102	747	1.73	71.38
9	388	162	1236	100	798	1.55	55.68

V. CONCLUSIONS

The period of time for the route is larger than the period of time for motion planning, and it restricts all the process. The period of time for route planning can decrease if computational power increases. That is why the efficiency of robot depends on efficiency of route planning, i.e., on the number of steps. If we admit that the robot covers a cell per second, we can say the following: the number of steps is equal to the time in seconds and we can get the sum of both numbers (a period of time of motion planning and the number of steps) added together. The use of SA can improve the efficiency of a real robot from 27.11% to 92.77%. It is very important to stress that the efficiency of the process is strongly related to energy saving and it is possible to prove that 92.77% of electric power can be saved.

Both algorithms can be successfully used in the requirements of autonomous tasks for robots, military technology and dynamic games.

Therefore, it is possible to use motion planning algorithms in other fields of research, which do not involve path planning.

REFERENCES

[1] R.H.J.M. Otten, L.P.P.P. Ginneken, *The Annealing Algorithm*. Kluwer Academic Publishers, 1989.

[2] R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, A Bradford Book The MIT Press Cambridge, Massachusetts London, England, 2011.

[3] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, S. Thrun. *Principles of Robot Motion*, The MIT Press, 2005.

[4] P. H. Batavia, I. Nourbakhsh, *Path planning for the Cye personal robot. IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, 2000.

[5] R. Biswas, B. Limketaki, S. Sanner, S. Thrun, *Towards Object Mapping in Dynamic Environments with Mobile Robots. Proceedings of the Conference on Intelligent Robots and Systems (IROS),* Lausanne, Switzerland, 2002.

[6] E. Fabrizi, A. Saffiotti, *Extracting Topology-Based Maps from Gridmaps. Proceedings of the 2000 IEEE International Conference of Robotics and Automation (ICRA 2000)*, pages 2973-2978, 2000.

[7] D. Ferguson, A. Stentz, *The Delayed D* Algorithm for Efficient Path Replanning. Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, April 2005.

[8] P. Fiorini, Z. Shiller, *Motion Planning in Dynamic Environments. The 7th International Symposium of Robotics Research*, pages 237-248, Munich, Germany, October 1995.

[9] H. Choset. Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph, PhD thesis, California Institute of Technology, 1996.

[10] H. Choset, "Coverage of Known Spaces: The Boustrophedon Cellular Decomposition," in *Autonomous Robots*, 9:247-253, Kluwer, 2000.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, in *Section 24.3: Dijkstra's algorithm*, Second Edition, MIT Press and McGraw-Hill, 2001, pp. 595-601.

[12] V. Ashkenazi, D. Park, M. Dumville, Robot Positioning and the Global Navigation Satellite System, Industrial Robots: An International Journal, 27(6), pp. 419-426, 2000.

[13] J. Buhmann, W. Burgard, A. B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, S. Thrun, The Mobile Robot Rhino, AI Magazine, 16(1), 1995.

[14] H. Choset, K. Nagatani, Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization Without Explicit Localization, IEEE Transactions on Robotics and Automation, 17(2), pp. 125-137, April 2001.

[15] C. I. Connolly, R. A. Grupen, The Application of Harmonic Functions to Robotics, Journal of Robotic Systems, 10(7), pp. 931-946, 1992.

[16] A. Elfes, Using occupancy grids for mobile robot perception and navigation, IEEE Computer, 22(6), pp. 46-57, 1989.

[17] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, Science, 220, pp. 671-680, 1983..

[18] S. Kirkpatrick, Optimization by Simulated Annealing: Quantitative Studies, Journal of Statistical Physics, 34, pp. 975-986, 1984.

[19] M.P. Vecchi, S. Kirkpatrick, Global Wiring by Simulated Annealing, IEEE Transaction on Computer Aided Design, CAD-2, pp. 215-222, 1983.

Edvards Valbahs was born in Riga, Latvia. He received his Mg.sc.ing. degree in Information Technology from Riga Technical University in 2011. Since 2009 he has been working at A/S "Exigen Services Latvia" as a Systems Analyst. He is a PhD student at Rezekne Higher Education Institution. His research interests include neural networks and path planning algorithms. His current research focuses on techniques for path planning.
E-mail: err@inbox.lv

Peter Grabusts was born in Rezekne, Latvia. He received his Dr.sc.ing. degree in Information Technology from Riga Technical University in 2006. Since 1996 he has been working at Rezekne Higher Education Institution. Since 2008 he is an Associate Professor at the Department of Computer Science. His research interests include data mining technologies, neural networks and clustering methods. His current research focuses on techniques for clustering and fuzzy clustering.
E-mail: peter@ru.lv

Edvards Valbahs, Pēteris Grabusts. Autonoma robota kustības plānošana slēgtā telpā ar šķēršļiem

Darba mērķis bija maršruta plānošanas algoritmu izpēte, kas ietver arī modelēšanas programmas sistēmas izstrādi. Programmas sistēma ir nepieciešama kā modelēšanas vide simulācijas datu iegūšanai. Programmas produkts ietver sevī SA un RRT algoritmus. Simulācijas dati dod iespēju veikt izvēlēto algoritmu daudzpusīgu analīzi. Analīze paredz simulācijas datu interpretāciju un salīdzināšanu ar citiem datiem, kas ir iegūti, pielietojot maršruta plānošanas algoritmus. Ir iegūti dažādi telpas pārklājumi, pielietojot RRT un SA algoritmus identiskos apstākļos. Telpas izmērs (dimensija) simulācijas gaitā netika mainīts (20 x 20 rūtiņas), bet mainījās fiksētu šķēršļu izvietojums telpā un to skaits. Vienkāršākajā gadījumā telpa nesatur nevienu šķērsli, t.i., ir pilnīgi tukša. Pilnīgi tukšai telpai tika veikta vairākkārtīga simulācija ar abu izvēlēto algoritmu palīdzību. Iepriekšminēto darbību princips tika pielietos arī pie fiksētu šķēršļu izvietojuma telpā, t.i., telpā ar šķēršļiem. Iegūtie simulācijas dati tika aprēķināti, kā vidējā vērtība attiecībā pret simulāciju atkārtošanas skaitu. Katrā simulācijā tika izskaitļots sākotnējais tukšo rūtiņu skaits, maršruta plānošanas laiks un nepieciešamais soļu skaits uzdotās telpas apiešanai abu algoritmu darbības laikā. Maršruta plānošanas laiki abos algoritmos tika salīdzināti, un eksperimenti parādīja, ka maršruta plānošanā uzdotajai telpai SA algoritms ir efektīvāks par RRT. Šādus algoritmus var veiksmīgi pielietot, piemēram, autonomā virszemes transporta ceļu ierobežojumu modelēšanā vai autonomo robotu kustības ar šķēršļiem plānošanā. Aplūkotos algoritmus var izmantot arī spēļu risinājumos, kad maršruta plānošana notiek dinamiskā vidē. Tādējādi šo algoritmu pielietošanas joma var būt ļoti plaša.

Эдвард Валбах, Петерис Грабуст. Планирование движения автономного робота в замкнутом пространстве с препятствиями

Цель задачи состоит в исследовании алгоритмов планирования маршрута, что включает также разработку программного обеспечения моделирования работы этих алгоритмов. Программная система необходима для получения данных симуляции и поддерживает алгоритмы SA и RRT. Моделирование позволяет провести многосторонний анализ сравнения работы выбранных алгоритмов. Анализ подразумевает интерпретацию и сравнение данных, которые были получены с различными алгоритмами. Получены различные плотности перекрытия пространства с использованием алгоритмов RRT и SA. В процессе моделирования площадь пространства была неизменной (20 x 20 клеток), но число препятствий и их размещение менялось по ходу симуляции. В простейшем случае пространство не содержит препятствий, то есть является незаполненным. Для пустого пространства проводилось многократное моделирование для обоих выбранных алгоритмов. Приведенный выше принцип также был использован для пространства с фиксированным числом и размещением препятствий. Полученные данные моделирования усреднялись по отношению к числу количества повторений симуляции. Для каждой симуляции было вычислено количество изначально пустых клеток, время планирования маршрута и количество шагов обхода пространства для обоих алгоритмов. Эксперименты показали, что для данного пространства алгоритм SA планирует маршрут эффективней, чем алгоритм RRT. Рассмотренные алгоритмы могут быть успешно применены, например, для моделирования препятствий для автономных наземных транспортных средств или в автономных роботизированных системах, а также для решения задач игровых систем, где планирование маршрута проводится в динамичной среде. Таким образом, рассмотренные алгоритмы могут быть применены не только в роботизированных системах для планирования маршрута.