Proc. of The Third Intl. Conf. on Advances in Mechanical and Robotics Engineering - AMRE 2015

ISBN: 978-1-63248-066-8 doi: 10.15224/ 978-1-63248-066-8-105

A Fast Optimal Sampling-based Motion Planning Algorithm based on the Poisson-Disk Sampling Distribution

W. Khaksar, K.S.M. Sahari, and S.H. Tang

***Abstract*— Sampling-based motion planning algorithms have been proven to work well with difficult planning tasks in a variety of problems. Recently, asymptotic optimal algorithms have been proposed to overcome the non-optimality inefficiency of these planners but with extra computational costs associated with the additional processing requirements. In this paper, new extensions of optimal sampling-based motion planning algorithms are presented which overcome this drawback by utilizing the Poisson-disk sampling distribution. The proposed planners replace the original uniform sampling with the Poisson-disk sampling by defining a sampling radius along with the neighborhood radius in the original optimal planners. The main advantage of the proposed planners is their ability to reach different levels of optimality with fewer sampling attempts which reduces the running time of the planner significantly. The proposed algorithms have shown to solve different motion planning tasks with considerably smaller set of samples. The simulation studies have been conducted and support the superiority claim of the proposed algorithms.**

***Keywords*—robotics, motion planning, sampling, poisson-disk, optimal planning**

I. Introduction

Motion planning is a key problem in robotics that has motivated research since the past three decades. The goal of robot motion planning is to decide automatically what motions a robot should execute in order to achieve a task specified by initial and goal spatial arrangements of physical objects [1]. Due to widespread applications of motion planning in different fields such as computer games, animation, assembly planning and computational biology [2], there have been a great number of studies on motion planning algorithms in recent years. The simplest form of the motion planning problem is planning a collision-free path for a robot made of an arbitrary number of polyhedral bodies among an arbitrary number of polyhedral obstacles, between two collision free positions of the robot. Complexity analysis has shown this problem to be PSPACE-complete [3]. By taking into account the physical properties and actuator limitations in a real robot, it is not known if the problem is even decidable except for some particular cases [4]. Some of the well-known complete motion planning algorithms are cell decomposition and visibility roadmaps [5]. For practical purposes, complete algorithms turn out to be computationally expensive and hard to implement.

Weria. Khaksar, Khairul Salleh Mohamed Sahari

Center of Mechatronics and Advanced Robotics, College of Engineering, University Tenaga Nasional (UNITEN), Malaysia.

Tang Sai Hong

Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, University Putra Malaysia.

Adding various restrictions to the problem made the use of complete algorithms [6] possible. For the general case of the problem, a breakthrough was achieved with the development of sampling-based motion planners [5]. Sampling-based motion planning algorithms have been used for solving planning problems with high degrees of freedom with proven capability and advantages. These algorithms do not require a complete representation of the configuration space; instead, they solely rely on a simple procedure which can decide whether a given configuration is collision-free or not. These planners are unique in the fact that planning occurs by sampling the configuration space of the robot. Generally, sampling-based algorithms can be divided into two main groups namely multi-query and single-query. In the multi-query class, the configuration space is sampled in a learning phase and then the resulted graph will be used to solve any given query. The basic and most important multi-query planner is the Probabilistic Roadmaps (PRM) algorithm [7, 8].

One of the fundamental inefficiencies of sampling-based algorithms is that they normally produce suboptimal solutions that peregrinate inside the configuration space. It has been recently shown that RRT almost surely does not find an optimal solution [9]. Therefore, the optimal versions of PRM and RRT algorithms i.e. PRM* and RRT* have been designed and shown to be asymptotically optimal, with the probability of finding the optimal solution approaching unity as the number of iterations approaches infinity. The RRT* planner possesses a unique ability which is capability of finding the optimal path from the start position to every configuration inside the free space [9].

The basic issue with these optimal sampling-based algorithms is the high computational time requirement. The PRM* and RRT* planners require additional computational time for selection of neighbor nodes. Furthermore, the RRT* algorithm performs a rewiring procedure to eliminate the non-optimal paths which increases the runtime of the planner. This issue has been pointed out as a drawback for the optimal sampling-based planning [10, 11]. Recently, few attempts have been made to improve the performance of the RRT* algorithm in terms of computational time. The informed RRT* algorithm [12] has been proposed which grows the RRT* tree only in an ellipsoidal subset of the configuration space. Although the resulted planner is faster than the RRT*, but it loses the ability of reaching every configuration in the space. Another extension of RRT* has been introduced which is a bi-directional version of optimal RRT [13] and grows two trees from start and goal configurations in order to speed up the convergence of the planner. This algorithm also sacrifices the multi-goal property of the RRT*. Combining the asymptotic optimality of the PRM*/RRT* planners with low computational requirement property seems to be a very challenging task.

Proc. of The Third Intl. Conf. on Advances in Mechanical and Robotics Engineering - AMRE 2015

ISBN: 978-1-63248-066-8 doi: 10.15224/ 978-1-63248-066-8-105

In this paper, a new sampling strategy for PRM* and RRT* algorithms is proposed which utilizes the Poisson-Disk sampling strategy [14] to reduce the size of the required graph/tree. The proposed planner is able to capture the connectivity of the obstacle-free configuration space with significantly less number of samples. The proposed sampling strategy can be stated as follows: each generated sample should satisfy an additional condition i.e., it should be far enough from the other samples. This sample acceptance condition makes the resulted graph/tree to be more sparse and capable of learning the whole free-space with less number of samples and accordingly requires less computational time. The performance of the proposed sampling strategy is presented in Figure 1 where normal PRM* and the proposed algorithm have been used to construct a graph with 200 nodes in a simple 2D environment with three polygonal obstacles.

II. Background

A. Problem Formulation

The problem is defined similar to [9]. Let Q_free be the obstacle-free subset of the configuration space Q with dimension d. A feasible path can be defined as ω:[0,1]→ Q_free such that ω(0)=q_start and ω(1)=q_goal if such path exists. A feasible path ω^* is optimal if for a cost function c, c(ω^*)=min{c(ω),if ω is feasible} if such path exists. A common cost function can be defined as the sum of the Euclidean distance between each consecutive segment of ω. Let ω_ibe the i^th segment of the path ω, then the cost function can be defined as:

$$c = \sum_{i=0}^{n-1} \parallel \omega_{i+1} - \omega_i \parallel \qquad (1)$$

Where $\parallel a - b \parallel$ denotes the Euclidean distance between a and b. In the context of sampling-based motion planning, these path segments can be defined as the distance between two neighbor nodes in the graph/tree which belong to the set of the nodes in the final solution.

B. PRM*/RRT*

The optimal PRM algorithm is similar to the standard PRM with only one difference, i.e. connections are attempted between roadmap vertices that are within a radius r as a function of n.

$$r^*(n) = \gamma[\log(n)/n]^{1/d} \qquad (2)$$

$$\gamma > 2(1 + 1/d)^{1/d}\,[\mu(Q_{free})/\sigma_d]^{1/d} \qquad (3)$$

Where, n is the cardinality of the set of samples, d is the dimension of the space, $\mu(Q_{free})$ is the Lebesgue measure and σ_d is the volume of the unit ball in the d-dimensional Euclidean space. This connection radius decreases with the number of samples and the decay rate is such that the average number of connections attempted from a roadmap vertex is proportional to log(n). In the RRT* algorithm, initially performs similar to the classic RRT planner. It adds new points q_{new} to the vertex set but also considers connections from q_{new} to its close neighbors within a radius $r^*(n)$ which can be formulized as follows.

$$r^*(n) = \min\{\gamma[\log(n)/n]^{1/d}, step\} \qquad (4)$$

$$\gamma > 2(1 + 1/d)^{1/d}\,[\mu(Q_{free})/\sigma_d]^{1/d} \qquad (5)$$

Where, step is the constant appears in the local steer function. However, an edge is created from $q_{neighbor}$ to q_{new} if they can be connected along a path with minimum cost, i.e. $c(\omega^*)$ and new edges are created from q_{new} to $q_{neighbor}$ if the path through q_{new} has lower cost than the path through the current parent. In this case, the edge linking $q_{neighbor}$ to its current parents is deleted to keep the tree structure. The main improvement in PRM* is in line 4 of Algorithm 1 where the connection between neighbor nodes are attempted only if they are closer than $r^*(n)$. In the RRT* algorithm, the improvements are in lines 12-16 and 17-20 in Algorithm 2 where first, an edge is connected only if it can reach the current point through a path with minimum cost and second, if there is any other nodes in the neighborhood of the current node with lower cost, this node will replace the current parent of the current node. These improvements utilize the PRM/RRT planners with the ability to find the shortest path. However, there will be substantial computational cost associated with the improved planners. This additional cost appears in PRM* when the neighbor nodes are checked to be within a fixed distance and it happens in RRT* when the corresponding paths of the neighbor are checked for cost improvement and also in the rewiring procedure.

Recently, some extensions of PRM*/RRT* have been proposed to improve the computational cost of the planner. An intelligent method has been proposed [14] that deactivates unpromising parts of the configuration space to improve the execution time of the PRM planner. A connectivity-based approach has been introduced [15] that enhances the sampling procedure by biasing the sampling towards difficult parts of the space. Also, a speeding up method has been reported [16] based on the locality-sensitive hashing technique to approximate the neighbor search procedure of the sampling-based algorithms. Although these methods have shown pleasant performances, their performances degrade as the dimension of the configuration space increases. Moreover, they require more accurate information about the configuration space which is a costly requirement. For single-query optimal planners, an informed RRT* algorithm has been proposed [12] which makes the sampling to focus on an ellipsoidal subset of the configuration space which includes the two inputs of the query. A bidirectional RRT* has been introduced [13] that grows two optimal trees rooted at the query inputs in order to find the optimal solution as soon as possible. Despite the considerable performance of these planners, they suffer from a basic drawback, i.e. the powerful ability of the tree-based planners in finding a path to different states of the space is sacrificed in order to lower the runtime of the planner.

C. Poisson-Disk Sampling

The algorithm presented in this paper utilizes the Poisson-Disk sampling strategy for generating the collision-free samples for optimal sampling-based planning. Poisson-disk distributions are known to have excellent blue noise characteristics and desirable in sampling patterns. The basic idea is to assign a circular area around each sample and forbid any further sampling in the corresponding regions of the current set of samples [14]. There are two basic Poisson-disk sampling strategies namely normal sampling and maximal sampling. In normal sampling, the whole area is not going to be covered and the sampling terminates after certain amount of iterations, i.e. desirable size of the graph.

Proc. of The Third Intl. Conf. on Advances in Mechanical and Robotics Engineering - AMRE 2015

ISBN: 978-1-63248-066-8 doi: 10.15224/ 978-1-63248-066-8-105

On the other hand, for generating a maximal Poisson-disk distribution, the whole sampling area should be covered by samples. Later, we will show that the generated set of samples for the proposed method does not need to be maximal.

Assume a current set of samples denoted by $V = \{q_i = (x_i, y_i), i = 1, \ldots, n\}$. A sampling radius is associated with each member of V which determines the minimum space between any two samples. Let r^s be the sampling radius. A set of samples is considered to follow a Poisson-disk distribution if they satisfy the following condition for any pair of samples. This simplest form of the problem is sometimes referred to as the "dart throwing problem".

$$\| q_i - q_j \| \leq r^s, i = 1, \ldots, n \quad (6)$$

Although the Poisson-disk samples are known to have excellent performances, they are generally regarded as too computationally expensive to generate in real time [14]. To overcome this difficulty, several techniques have been proposed in the past each of which, utilize a heuristic idea to reduce the computational complexity of this sampling distribution. A spatial data structure has been introduced for fast Poisson-disk sampling [17] based on representing the sampling area by a set of disjoint scalloped sectors. This method has been proved to run in $O(N)$ time and space. A fast Poisson-disk sampling method has been proposed which permits generation of samples in $O(N)$ time by initializing an n-dimensional background grid over the sampling domain [18]. A simple algorithm has been introduced for generating maximal Poisson-disk samples in $O(N\log N)$ time using the concept of Voronoi diagram [19]. Another fast approach for generating Poisson-disk samples has been proposed which intelligently excludes the areas which are known to be covered by current samples [20]. In [21], an efficient sampling strategy has been introduced for generating maximal and unbiased Poisson-disk samples over bounded non-convex domains. This method initially uses a background grid of square cells for sampling and completes the maximal covering by calculating the connected components of the remaining uncovered sectors. Generating uniform base grids have been proposed in an algorithm for fast maximal Poisson-disk sampling [22]. The size of the grids will be further reduced to maintain the maximally of the samples. Recently, an effective algorithm has been proposed which takes a non-maximal set of Poisson-disks and improves it to reach the maximal property [23]. The key idea is to convert the complicated problem of plain or space searching into a simple searching on circles or spheres.

The presented algorithm in this paper utilizes the excellent sampling properties of the Poisson-disk distribution to improve the computational requirements of the optimal sampling-based algorithms. Two new extensions of PRM* and RRT* algorithms are proposed with the ability to solve the planning queries with significant reduction in the size of the graph/tree while maintaining the asymptotic optimal property of the planner.

III. proposed algorithm

The proposed algorithm works similar to the original PRM*/RRT* planners. The only difference is in the sampling procedure. While the original planners let any collision-free samples to be added to the structure, the proposed extensions contain an additional checking stage which ensures that the samples possess the Poisson-disk distribution property. Among all efficient techniques for generating fast Poisson-disk samples, in this research, the boundary sampling technique [17] is being used for generating the Poisson-disk samples because of its simplicity and ease of implementation. In this method, a sample's available neighborhood collapses to a collection of circular arcs centered at the sample and with the radius of r^s. By directly implementing this method, the available neighborhood is presented as a set of per-point angular ranges at which a point can be placed on the boundary. Figure 1 illustrates the sampling procedure as proposed in [17] where the gray circles represent the Poisson-disks and the green regions represent the forbidden sampling areas. After the first rejection, the boundary of the union of the forbidden regions will be selected and a random point is generated accordingly as shown by the orange circle. The sampling radius is defined as follow.

$$r^s(n) \leq \tau \sqrt{Q_{free}/n} \quad (7)$$

Where τ is a scaling constant ranging within (0,1]. The idea behind the sampling radius is that for filling an obstacle free square space with n disks with radius r^s, the volume of the space should be approximately equal to $n(r^s)^2$. Forcing the samples to follow the Poisson-disk distribution usually reduces the number of samples. In other words, it is almost impossible to find n samples with $\tau\sqrt{Q_{free}/n}$ distance from one another with a randomized sample generator.

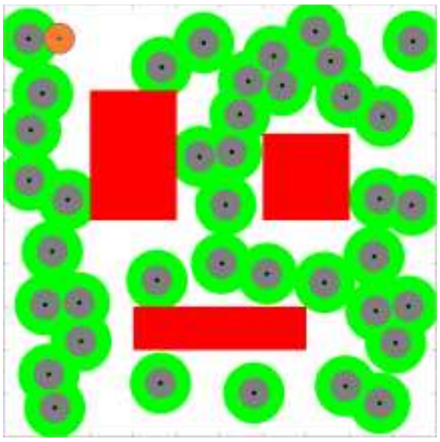


Figure 1. The implementation of the boundary sampling in a simple 2D environments with three obstacles. The new sample (orange circle) will be generates on the boundary of the union of forbidden region after the occurrence of the first rejection.

After generating the Poisson-disk samples, the algorithm will proceed exactly as the original PRM*/RRT* does. One important fact about the performance of the proposed algorithm is the relation between the neighborhood and sampling radiuses. It is vital for $r^s(n)$ to be smaller than $r^*(n)$. Otherwise, no connection can be made between different nodes and the resulted graph/tree will not be able to be constructed. Figure 2 shows the relation between these two radiuses along with the ratio of $r^s(n)$ over $r^*(n)$.

According to the definitions of neighborhood and sampling radiuses, the cardinality of the graph/tree should follow the following rule:

$$n \geq 10^{\pi\tau^2/6} \quad (8)$$

Proc. of The Third Intl. Conf. on Advances in Mechanical and Robotics Engineering - AMRE 2015

ISBN: 978-1-63248-066-8 doi: 10.15224/ 978-1-63248-066-8-105

This criterion ensures that there will be at least one sample within the neighborhood region of the current sample and with eligibility to be connected. Recalling the range of τ, i.e. $(0,1]$, one can conclude that in a 2D space, the sampling radius is smaller than the neighborhood radius if $n \geq 4$.

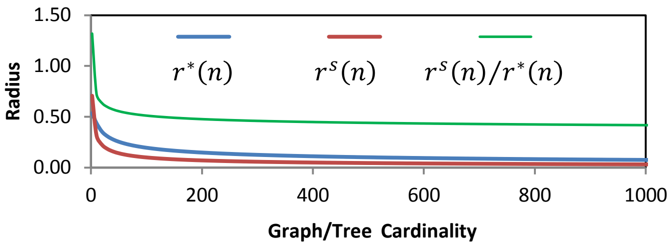


Figure 2. Different values for naighborhood and sampling radiuses and the corresponding ration for $\mu(Q_{free}) = 1$ and $\tau = 1$. For $n \geq 4$, the neighborhood radius is always greater than the sampling radius.

Another noteworthy property of the proposed algorithm appears in the RRT* planner where the steering parameters in the original RRT* which is *step* will be spontaneously replaced by the sampling radius. According to the previous discussion about the Poisson-disk sampling procedure and the method that have been used, the samples are generated randomly on the boundary of the current Poisson disks. Therefore, the maximum distance between samples will exactly be equal to the sampling radius and the edge between any two neighbor samples is less or equal to $r^s(n)$. As stated in section II, the cost of the optimal path can be defined by summing up the Euclidean distance of all segments of the optimum path. The cost of the optimal path in the proposed algorithms can be calculated as follows.

$$c(\omega^*) = \sum_{i=1}^{n^*} ||\omega_{i+1}^* - \omega_i^*|| \quad (9)$$

Where n^* is the number of nodes in the optimal path. Considering the fact that $\|\omega_{i+1}^* - \omega_i^*\| < r^s(n)$, now it is possible to find an upper bound for the path of the optimal solution.

$$c(\omega^*) \leq n^* r^*(n) \quad (10)$$

This upper bound solely depends on the number of nodes in the final solution. On the other hand, reducing the total number of samples (n), will decrease the number of samples in any solution. Therefore, one may conclude that selecting the samples from a Poisson-disk distribution will reduce the cost of the final solution or at least keeps the asymptotic optimality property of the planner. Figure 3 illustrates the graph construction phase for the PRM* planner.

IV. Simulation Studies

The proposed algorithms have been simulated in MatLab R2013a on a computer with Intel Core i5 CPU @ 3479GHz and 8 GB of RAM. Five 2D environments have been created and used for testing the performance of the proposed algorithms including one plain environment without any obstacle, two simple workspaces with three obstacles, one maze environment and finally an environment with a narrow passage with the tenuity factor of $w_{nar}/w_{env} = 0.02$. The performances of the proposed algorithms are shown in Figure 4 where the final path is shown by blue lines and the start and goal configurations are illustrated by yellow and green squares respectively.

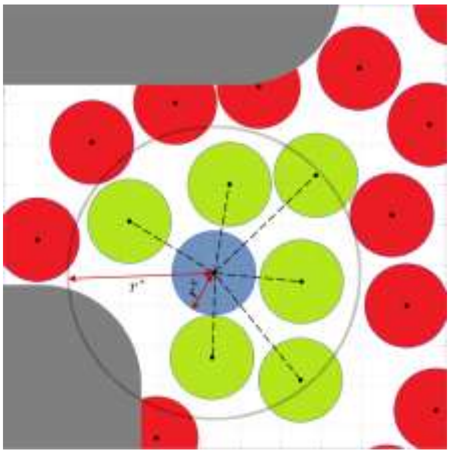


Figure 3. The performance of the proposed algorithm (PRM*). Sampling and neighborhood radiuses are illustrated in a simple 2D environment. Current sample (blue circle) is being connected only to the neighbors within the neighborhood radius *r*(n)* while the samples follow the sampling radius $r^s(n)$ associated with the Poisson-disk sampling. Eligible neighbor nodes are depicted by green circles.

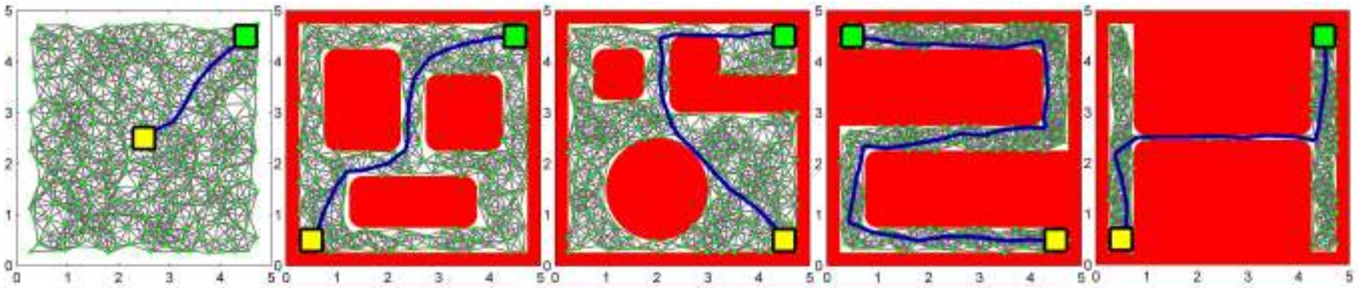
The proposed extensions of the original PRM*/RRT* planners efficiently solve the planning queries with a significant reduction in the number of required samples. The generated paths are almost optimal and the entire configuration space has been covered by the samples by means of the Poisson-disk property of the graph/tree. The most difficult problem is the last workspace in Figure 5 where the environment includes a narrow passage which is a classic dilemma for sampling-based algorithms. The proposed algorithms efficiently find the entrance of the narrow passage and successfully connect other parts of the workspace through this narrow region. The classic PRM* planner requires around 400 samples for solving this problem while the proposed algorithm solves this narrow passage query with approximately 250 nodes.

Total simulation results are presented in Table 1. Note that for asymptotic optimal planners, i.e. PRM*, RRT*, PRM*_Poisson_Disk and RRT*_Poisson _Disk, the table only shows the results for 95% of optimality to present the minimum number of samples required to solve the query successfully. For the original PRM and RRT planners, maximum obtained optimality is given. These results are averaged over 1000 iterations for each planner to provide a more realistic conclusion about the performances. According to these results, the proposed planners successfully solve the given planning queries with smaller graphs/trees. For instance, consider the second environment (Simple 1) where a point robot is moving in a workspace with three simple obstacles. The classic PRM and RRT planners require at least 300 samples to reach the optimality rate around 75% and the original PRM* and RRT* algorithms reached the 95% optimality with 200 nodes while the proposed planners the same level of optimality with 100 and 150 nodes for multi and single-query cases.

Proc. of The Third Intl. Conf. on Advances in Mechanical and Robotics Engineering - AMRE 2015

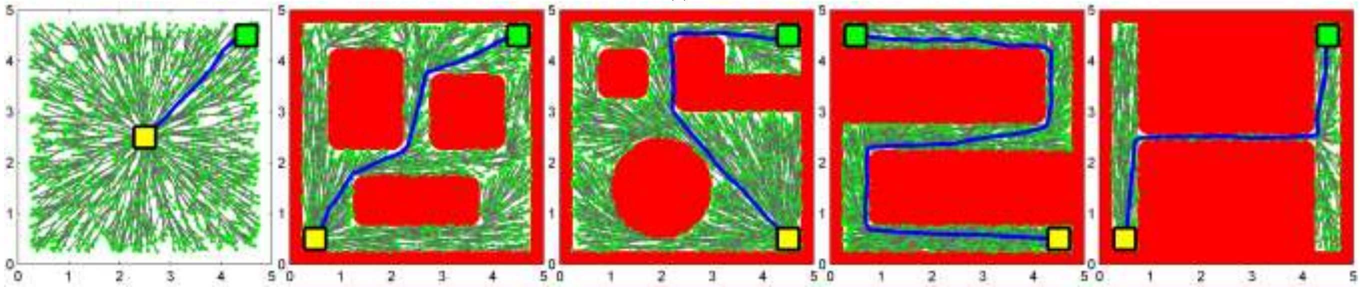
ISBN: 978-1-63248-066-8 doi: 10.15224/ 978-1-63248-066-8-105

Figure 5 illustrates different levels of optimality based on the number of samples for multi and single-query planners and it shows that for obtaining any level of optimality the proposed planners require smaller graphs/trees. For example, in order to reach 80% of optimality, the PRM* and RRT* require more than 600 samples while the proposed algorithms are able to reach this level of optimality with less than 400 samples. Even for a fixed number of samples, the proposed planners outperform the original PRM* and RRT*. Consider the value of optimality with 500 nodes in Figure 5. The original optimal planners at most can achieve 75% optimality rate using 500 nodes while the proposed algorithms can reach up to 90% of optimality with the same number of samples.



Plain Simple 1 Simple 2 Maze Narrow

(a)



Plain Simple 1 Simple 2 Maze Narrow

(b)

Figure 4. The performance of the proposed algorithms in five different test environments. (a) The PRM* planner with Poisson-disk sampling distribution which used around 200 nodes in each environment. (b) the performance of the proposed extension of RRT* algorithm in test problems.each problem has been solved using approximately 1000 samples. Final paths are illustrated by blue lines and the query pair are shown by yellow and green squares. For all problems $\tau = 1$.

TABLE 1: SIMULATION RESULTS FOR THE MULTI-QUERY PLANNERS

Problem	PRM					PRM*					PRM*_Poisson_Disk				
	Optimality		Runtime		n_{min}	Optimality		Runtime		n_{min}	Optimality		Runtime		n_{min}
	Mean	S.t.d	Mean	S.t.d		Mean	S.t.d	Mean	S.t.d		Mean	S.t.d	Mean	S.t.d	
Plain	81.2	3.7	2.4	1.12	100	95	-	2.9	0.5	50	95	-	2.8	0.5	30
Simple 1	72.7	4.8	5.7	1.18	300	95	-	5.9	0.6	200	95	-	6.1	0.5	100
Simple 2	68.5	3.7	6.1	1.85	400	95	-	7.5	0.5	250	95	-	7.3	0.6	150
Maze	72.2	3.4	8.1	2.1	400	95	-	9.5	0.8	200	95	-	8.8	0.8	180
Narrow	70.4	6.3	12.6	3.5	600	95	-	15.4	1.1	400	95	-	16.4	0.9	250

TABLE 2: SIMULATION RESULTS FOR THE SINGLE-QUERY PLANNERS

Problem	RRT					RRT*					RRT*_Poisson_Disk				
	Optimality		Runtime		n_{min}	Optimality		Runtime		n_{min}	Optimality		Runtime		n_{min}
	Mean	S.t.d	Mean	S.t.d		Mean	S.t.d	Mean	S.t.d		Mean	S.t.d	Mean	S.t.d	
Plain	86.7	2.9	2.1	1	100	95	-	3.6	0.4	100	95	-	3.5	0.5	100
Simple 1	78.1	3.3	4.3	1.2	300	95	-	6.5	0.3	200	95	-	6.6	0.4	150
Simple 2	74	3.3	5.7	1.5	400	95	-	8.6	0.5	300	95	-	8.9	0.3	200
Maze	77.7	3.1	6.9	1.6	400	95	-	9.9	0.8	300	95	-	9.7	0.2	200
Narrow	75.8	3.4	11.6	1.9	600	95	-	12.8	1.1	350	95	-	12.4	1.1	250

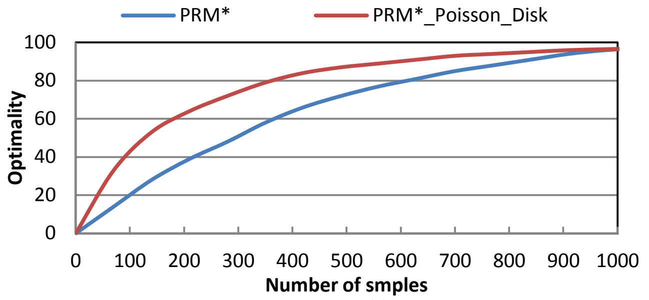
V. Conclusion

In this paper, new extensions of the original optimal sampling based motion planning algorithms, i.e. PRM* and RRT* have been proposed to reduce the computational requirements of these planners. These extensions utilize the Poisson-disk sampling distribution to spread the samples more equally inside the configuration space. A sampling radius $r^s(n)$ is defined and instead of sampling points, random disks will be generated which is not able to overlap other disks. This property is added to PRM* and RRT* planners and the new algorithms contain two important concepts. First, the neighborhood radius which was proposed to improve the length of the generated paths, and second, the sampling radius which is proposed to reduce the running time of the process. The proposed algorithms have been simulated and tested in different environments against the classic PRM and RRT as well as the original PRM* and

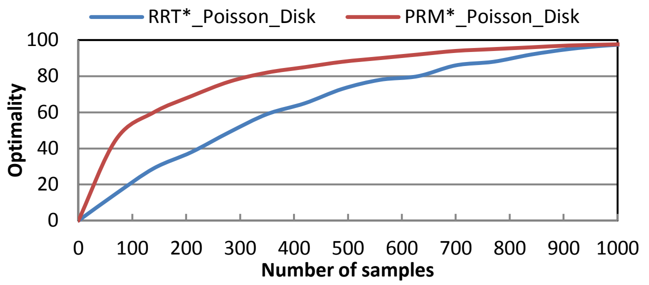
Proc. of The Third Intl. Conf. on Advances in Mechanical and Robotics Engineering - AMRE 2015

ISBN: 978-1-63248-066-8 doi: 10.15224/ 978-1-63248-066-8-105

RRT* planners. Simulation studies show the superiority of proposed planners in all tested problems. In terms of minimum required number of samples, the proposed multi-query planner has improved the classic PRM with 77% and the PRM* with 45%. The proposed single-query has reduced the minimum number of nodes to 63% and 25% for RRT and RRT* respectively. These results have been averaged over all testing problems.



(a)



(b)

Figure 5. The minimum number of samples needed to reach different levels of optimality for (a) PRM* and PRM*_Poisson_Disk and (b) RRT* and RRT*_Poisson_Disk. The results are averaged over all seven cases presented in Figures 5 and 6 for 1000 iterations.

REFERENCES

[1] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*, Cambridge University Press, New York, 2010.

[2] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*, MIT Press, 2005.

[3] J. Canny, *The Complexity of Robot Motion Planning*, MIT Press, Cambridge: MA, 1988.

[4] P. Cheng, G. Pappas, and V. Kumar, "Decidability of motion planning with differential constraints," in: *IEEE International Conference on Robotics and Automation*, 2007, pp. 1826–1831.

[5] K. I. Tsianos, I. A. Sucan, and L. E. Kavraki, "Sampling-based robot motion planning: Towards realistic applications," *Comput. Sci. Rev.*, vol. 1, pp. 2–11, 2007.

[6] S. Hirsch, D. Halperin, Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane, in: WAFR, Nice, 2002, pp. 225–241.

[7] L.E. Kavraki, P. Svestka, J.C. Latombe, M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Trans. Robot. Autom*. vol. 12, no. 4, pp. 566–580, 1996.

[8] S.M. LaValle and J.J. Kuffner, "Randomized kinodynamic planning, *Int. J. Rob. Res*., vol. 20, no. 5, pp. 378–400, 2001.

[9] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. J. Rob. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[10] M. Elbanhawi and M. Simic, "Sampling-Based Robot Motion Planning: A Review," *Access, IEEE* , vol. 2, pp. 56,77, 2014.

[11] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime Motion Planning using the RRT*," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1478-1483.

[12] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *IROS*, 2997–3004, 2014.

[13] Starek, E. Schmerling, L. Janson, and M. Pavone. Bidirectional Fast Marching Trees: An optimal sampling-based algorithm for bidirectional motion planning. Technical report, 2014. Submitted to Proc. IEEE Conf. on Robotics and Automation,

[14] M. Rantanen, "A connectivity-based method for enhancing sampling in probabilistic roadmap planners," *J. Intell. Robot. Syst*. vol. **64,no.** 2, pp. 161–178, 2011.

[15] M. T. Rantanen, M. Juhola. A configuration deactivation algorithm for boosting probabilistic roadmap planning of robots. *Inte. J. Autom. Comput.*, vol. 9, no. 2, pp. 155–164, 2012.

[16] T.R. Mika and M. Juhola, "Speeding up probabilistic roadmap planners with locality-sensitive hashing," *Robotica*: 1-16.

[17] D. Dunbar and G. Humphreys, "A spatial data structure for fast poisson-disk sample generation," *ACM Trans. Graph*., vol. 25, no. 3, pp. 503–508, 2006.

[18] R. Bridson, "Fast Poisson disk sampling in arbitrary dimensions," in: *SIGGRAPH '07: ACM SIGGRAPH 2007 sketches*, ACM, New York, NY, USA, 22.

[19] R.J. Thouis, "Efficient generation of Poisson-disk sampling patterns," *Journal of Graphics Tools*, vol. 11, no. 2, pp:27–36, 2006.

[20] W.B. Kenric, D. Cline, and P.K. Egbert, "Poisson disk point sets by hierarchical dart throwing." in: *IEEE Symposium on Interactive Ray Tracing*, pp. 129-132, 2007.

[21] M.S. Ebeida, , A.A. Davidson, A. Patney, P.M. Knupp, S.A. Mitchell, and J.D. Owens, "Efficient maximal poisson-disk sampling." *ACM Trans. Graph.*, vol. 30, no. 4, p. 49, 2011.

[22] M.S. Ebeida, S.A. Mitchell, A. Patney, A.A. Davidson, and J.D. Owens. "A Simple Algorithm for Maximal Poisson-Disk Sampling in High Dimensions." in *Computer Graphics Forum*, vol. 31, no. 2pt4, pp. 785-794, 2012.

[23] Y. Xiang, Z. Li, and Y. He. "A parallel algorithm for improving the maximal property of Poisson disk sampling." *Computer-Aided Design* vol. 46, pp. 37-44, 2014.