Increased Visibility Sampling for Probabilistic Roadmaps

Rahul Kala

Citation: R. Kala, "Increased visibility sampling for probabilistic roadmaps," 2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), Brisbane, QLD, 2018, pp. 87-92.

Full Text Available At: https://ieeexplore.ieee.org/document/8376276

***Abstract*— Sampling based planning algorithms solve the problem of Robot Motion Planning by sampling a number of vertices to make a roadmap or a tree, which is then searched for a solution. The sampling strategy denotes the mechanism to generate samples used to construct the tree or the roadmap. In this paper new sampling strategies are proposed for the Probabilistic Roadmap technique that generate samples aiming at maximizing the sample visibility. The increased visibility makes it easier to construct edges with the neighboring samples and thus contribute to get a solution early. Based on this principle three new samplers are pro-posed. The first sampler generates samples inside corridors and promotes them exactly to the corridor centres. The second sampler uses a distance threshold bi-nary search to approximately place the samples in the corridor centre. The last sampler attempts to bias the sampling towards narrow corridors, while still placing the samples approximately at the corridor centres. The increased visibility pays off for the increased computation effort incurred therein. The approach is tested for narrow corridor scenarios and is experimentally found to surpass all state-of-the-art sampling techniques of Probabilistic Roadmap.**

I. Introduction

Motion Planning [1] deals with computing a trajectory of the robot τ: $[0,1]\rightarrow C^{free}$ from a given source state ($\tau(0)=S\in C^{free}$) to a given goal state ($\tau(1)=G\in C^{free}$). In order to carry out the search, the problem is seen at the configuration space (C), which is a collection of all possible robot configurations. Using the collision checking algorithms, the configuration space may be divided into free configuration space (C^{free}) and obstacle configuration space (C^{obs}). C^{free}, denotes the free configuration space, which is a space of configurations at which the robot is neither in collision with the obstacles, nor is it in the state of self-collision. Cobs is the complement of C^{free} over C ($C^{obs}=C\backslash C^{free}$), and denotes the configurations at which some collision occurs.

One of the popular variants of the problem is the multi-query methodology of solving the problem, wherein C^{free} is first converted into a roadmap in an offline manner. The roadmap is then used to answer a number of queries consisting of source and goal pairs. Sampling based approaches are based on the working methodology of sampling out a number of configurations from the continuous and high dimensional configuration space, which are used for the search. Probabilistic Roadmap (PRM) [2] technique accepts the sampled out configurations as vertices, while an attempt is made to connect every configuration to the neighboring configurations by using a local planner. Typically the nearest k vertices or the vertices at a radius of k are considered [3, 4]. The PRM* [5] algorithm adapts the same parameter for asymptotic optimality. If the local planner succeeds in connecting the two vertices, the vertices are connected by an edge as the local trajectory used by the local planner. A common technique is to use the straight line connection as the local planner.

The performance of the sampling based motion planning techniques is largely reliant upon the quality of the sampler used. A popular sampling strategy is the uniform sampling strategy that samples uniformly in C^{free}. The areas near the obstacles are more important which denote a mechanism of avoiding the obstacles, and thus the obstacle-based sampling strategy [6], aims to generate more samples at the obstacle boundaries. A typical way to do so is to first generate samples inside C^{obs}, and to then move them randomly (or towards a configuration in C^{free}) till the motion produces a sample in C^{free}, normally very close to the obstacle boundary. Similarly Gaussian sampling [7] is used to generate samples whose distance from the obstacle boundary is roughly given by a Gaussian distribution.

Narrow corridors are marked by a small volume of C^{free} sandwiched between C^{obs}. Since the volume of C^{free} inside the narrow corridor is very small in proportion to the size of complete C^{free}, the probability of generation of a sample inside the narrow corridor is small. A bridge test sampler [8] generates two samples in close vicinity inside C^{obs}, and a sample in-between. If the in-between sample is in C^{free}, it is known to be inside narrow corridor, being in-between two samples in C^{obs}. A combination of uniform and bridge-test sampling [9] is also a good technique, wherein the uniform samples enable an enhanced connectivity of the samples to connect the roadmap. The maximum clearance sampler [10] solves the problem by attempting to generate samples of the maximum possible clearance. The typical way to do so is to keep generating samples for a few iterations, and to accept the one with the largest clearance. It can never solve the problems with narrow corridors, as the sampler would prefer a sample outside corridor with a larger clearance. The other problem is a naïve way of maximizing clearance by repeated sample generation, which is computationally heavy.

Visibility Roadmap [11, 12] is another variant of the PRM which aims at producing a roadmap of the smallest

*Research supported by the Science and Engineering Research Board, Department of Science and Technology, Government of India through Research Grant ECR/2015/000406.

R. Kala is with the Robotics and Artificial Intelligence Laboratory, Indian Institute of Information Technology, Allahabad, India (phone: +91-532-292-2117; +91-7521050744; e-mail: rkala001@gmail.com).

size, still producing a solution which is complete to handle all queries. The roadmap accepts new samples only if they are not under the visibility of any sample already in the roadmap, or samples which connect two disconnected roadmaps. The roadmap produces a minimally connected tree structure, which is used to answer queries. The problem with the approach is the optimality. The Sparse Roadmap Spanners for Asymptotically Near-Optimal Motion Planning (SPARS), and its extension SPARS2 [13] address the problem of optimality of Visibility Roadmaps, however still face the problem of connectivity in narrow corridor situations.

The individual samplers have their own pros and cons, and therefore many researchers have proposed Hybrid Sampling [8, 14] techniques, wherein multiple samplers are used simultaneously. Heuristics or a learnable function can be used to select the selection of sampler based on the current context and performance indicator. In a related work Kala [15] used hybrid sampling techniques for initial sample generation followed by hybrid edge detection techniques for the construction of the roadmap. The roadmap construction was made biased towards discovery of all homotopic groups. Again the performance largely relies upon the quality of the individual samplers.

In this paper the proposal is to produce samples which have as high visibility as possible. First a sample is generated and then the same sample is moved so as to maximize its visibility. In order to produce the sample, first two samples in C^{obs} are generated (q_1^{obs} and q_2^{obs}). Then a sample is generated in-between q_1^{obs} and q_2^{obs}, as centrally placed as possible between the two obstacles sampled by q_1^{obs} and q_2^{obs} (if q_1^{obs} and q_2^{obs} are from different obstacles in C^{obs}, separated by a region of C^{free}). Based on the same idea, three samplers are proposed. The first sampler finds the boundaries of the obstacle sampled by q_1^{obs} and the immediately first obstacle in the direction of q_2^{obs} (if they represent different obstacles) and then places a new sample exactly in the middle of the corridor, ensuring that there is no obstacle anywhere in-between. So it is a search guaranteed to generate a sample in the middle of the corridor. The second sampler does the same thing, except for using binary search which is under some threshold to search for the middle of the corridor made by the two obstacles. It does not check for presence of another obstacle in between to speed up computation. The sampler is thus approximate in the sense of placement of sample and selection of adjacent obstacles. The third sampler does the same thing, however, generates the samples q_1^{obs} and q_2^{obs} by a Gaussian random distribution so as to have more samples in narrow corridors and lesser in open areas.

II. Algorithm Design

A. Increased Visibility Sampling

This sub-section presents the general principle of increased visibility sampling, based on which the other samplers are derived. Throughout the paper, maximization of visibility is caused by maximization of clearance. We leave it for the readers to verify that maximization of clearance results in increased visibility of the sample. Let $q^{free} \in C^{free}$ be a sample. The clearance is defined by (1).

$$Cl\left(q^{free}\right) = \min_{o \in \partial C^{obs}} d\left(q^{free}, o\right). \quad (1)$$

Here $Cl()$ is the clearance function, o is iterated over all obstacle boundaries and $d()$ is the distance function. The aim is to maximize the clearance and therefore the sample q^{free} must be moved in C^{free} in the direction of the derivative of the clearance function. Since the obstacle is not explicitly modelled in the configuration space, and especially considering that the configuration space is high dimensional, it is not possible to get the direction of traversal. Hence the clearance is approximated to be computed from a sampled set of directions. Let q_1^{obs}, be a sample in C^{obs}. The clearance of q^{free} in a direction towards q_1^{obs} is given by (2). The sampled clearance is hence taken by sampling a few samples q_1^{obs}, given by (3). Here and throughout the paper the addition operation is only an abuse of notation and refers to the generic interpolation equation which is possible in non-Euclidean spaces.

$$\begin{aligned} &Cl\left(q^{free}, q_1^{obs}\right) = \\ &\min l : \left(\frac{l}{d(q^{free}, q_1^{obs})}\right) q_1^{obs} + \left(1 - \frac{l}{d(q^{free}, q_1^{obs})}\right) q^{free} \in C^{obs} \end{aligned} . \quad (2)$$

$$Cl^s\left(q^{free}\right) = \min_{q_1^{obs}} C\left(q^{free}, q_1^{obs}\right). \quad (3)$$

Consider that the minimum clearance is recorded in a sampled direction q_1^{obs}, which can hence be increased by moving in a direction opposite to q_1^{obs}. Let the direction opposite to q_1^{obs} be given by a sample q_2^{obs}. q_2^{obs} can be computed by the fact that q^{free} is in-between q_1^{obs} and q_2^{obs}, given by (4).

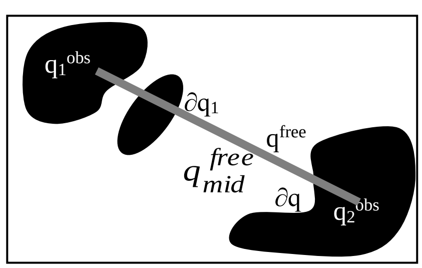
$$q_2^{obs} \in \left\{ q \in C^{obs} : \exists \lambda, \lambda q_1^{obs} + (1-\lambda) q = q^{free}, 0 < \lambda < 1 \right\}. \quad (4)$$

Out of all possible q that satisfy (4), any one value may be selected. Let us calculate the point around q^{free} that maximizes clearance in the line from q_1^{obs} to q_2^{obs}. Let ∂q_1 be the point at the boundary of the nearest obstacle towards q_1^{obs} and similarly let ∂q_2 be the point at the boundary of the nearest obstacle towards q_2^{obs}. Note that ∂q_1 and q_1 may not necessarily represent the same obstacle, due to the presence of an obstacle in-between the two in the same direction. The notations are illustrated in Fig. 1. The two boundary points are given by (5) and (6).

$$\begin{aligned} &\partial q_1 = \lambda_1 q_1^{obs} + (1-\lambda_1) q^{free}, \\ &\lambda_1 = \min \lambda - \varepsilon : \lambda q_1^{obs} + (1-\lambda) q^{free} \in C^{obs}, 0 \le \lambda \le 1 \end{aligned} . \quad (5)$$

$$\begin{aligned} &\partial q_2^{obs} = \lambda_2 q_2^{obs} + (1-\lambda_2) q^{free}, \\ &\lambda_2 = \min \lambda - \varepsilon : \lambda q_2^{obs} + (1-\lambda) q^{free} \in C^{obs}, 0 \le \lambda \le 1 \end{aligned} . \quad (6)$$

Figure 1. Notations Used in Sampling



Here ε is a small number to force the existence of the two boundary samples in C^{free}. Accordingly, the configuration with the best clearance on the line from q_1 to q_2 is given by (7).

$$q_{mid}^{free} = 0.5\partial q_1 + 0.5\partial q_2 \,. \quad (7)$$

If the sample q^{free} is promoted to q_{mid}^{free}, the clearance will increase to a value of $d(\partial q_1, \partial q_2)/2$. Correspondingly the visibility will also be increased. By repeatedly sampling the direction vectors and moving the sample so as to maximize the clearance, the sample clearance will increase, till the sample is found at the configuration wherein the clearance is locally maximum, or visibility is locally maximized.

B. Exact Mid-Corridor Sampler

The process of maximizing clearance as indicated by the generic approach has problems which are addressed in the design of the first sampler. The first problem is that computing the opposite direction from equation (4) is not possible for all kinds of configuration space. The second problem is that the PRM cannot invest too much time in the generation of a single sample, as even the sparsest roadmap will have a large number of samples for high dimensional spaces. The third problem is that the samples are promoted to the configuration which is locally maximum in terms of clearance (or visibility), meaning multiple samples will converge to similar points, while roadmaps may require points at non-locally maximum clearance configurations as well for connectivity. As an example a point inside the narrow corridor cannot be promoted to a maximum clearance point outside the corridor.

The problems are addressed by sampling q_1^{obs} and q_2^{obs}, instead of first sampling q^{free} and maximizing its clearance. Further, since the intention is not to ultimately promote the sample to the locally maximum clearance, the sampled direction of measuring and maximizing clearance is restricted to one pair only. Investing time in generating good samples is a good return in PRM type approach, since only a few number of good samples need to be ultimately generated. Investing less time and generating large number of samples is also a good technique, provided a subset of the samples are good enough for connectivity. By selecting only one pair of sampled direction tradeoffs between the computational expense and the quality of the sample (measured in terms of its clearance). By intuition, one direction mostly increases the clearance to a good enough value.

However once a single pair of direction is selected, the exact mid-corridor sampler places the samples at the exact middle of the corridor, ensuring selection of two adjacent obstacles in the configuration space, with no obstacle in-between. Since the configuration space is highly dimensional and large, it is possible that the obstacles represented by q_1^{obs} and q_2^{obs} will have multiple obstacles in between in the line from q_1^{obs} to q_2^{obs}. Once q_1^{obs} and q_2^{obs} are sampled, a traversal is made in the line from q_1^{obs} to q_2^{obs}. The first transition from C^{obs} to C^{free} is on encountering ∂q_1, while on further traversal from ∂q_1 towards q_2^{obs} another transition from C^{free} to C^{obs} is on encountering q_2. The two samples are given by (8) and (9). The new sample is given by (10), which is added in the roadmap.

$$\begin{aligned} &\partial q_1 = \lambda_1 q_2^{obs} + (1-\lambda_1) q_1^{obs}, \\ &\lambda_1 = \min \lambda : \lambda q_2^{obs} + (1-\lambda) q_1^{obs} \in C^{free}, 0 \le \lambda \le 1 \end{aligned} \,. \quad (8)$$

$$\begin{aligned} &\partial q_2 = \lambda_2 q_2^{obs} + (1-\lambda_2) \partial q_1, \\ &\lambda_2 = \min \lambda - \varepsilon : \lambda q_2^{obs} + (1-\lambda) \partial q_1 \in C^{obs}, 0 \le \lambda \le 1 \end{aligned} \,. \quad (9)$$

$$q_{mid}^{free} = 0.5\partial q_1 + 0.5\partial q_2 \,. \quad (10)$$

C. Approximate Mid-Corridor Sampler

The problem with the exact mid-corridor sampler is that it undertakes a long walk from q_1^{obs} to ∂q_2 in order to compute the exact centre of the corridor. The computational expense should normally not be a problem, since computational time of similar nature will anyways be additionally used to connect the sample to the neighboring samples. However if the configuration space is very large and if a very fine resolution of collision checking is used as per the preference of the algorithm, the samples q_1^{obs} to ∂q_2 may be very far and traversing them may take a very long time. On the contrary the neighboring samplers for edge connectivity may be reasonably near, thus not computationally expensive to connect.

Hence the exact mid-corridor sampler is extended to an approximate one that aims to compute the corridor centre by using some approximations. The sampler attempts to find a free sample q^{free} between q_1^{obs} and q_2^{obs}. Henceforth a rather strong assumption is made, that is there is neither an obstacle between q^{free} and q_1^{obs} (apart from the obstacle represented by q_1^{obs}), nor is there any obstacle between q^{free} and q_2^{obs} (apart from the obstacle represented by q_2^{obs}). The assumption will obviously not hold a good number of times. If the assumption holds, the algorithm nearly finds the corridor centre. However, if the assumption does not hold, the sampler may either place the new sample anywhere in the corridor centre (depending upon the unknown location of the obstacle), or may instead not generate any sample. The wrongly placed samples are also of general value. Hence the time saved, even if that results in insertion of some samples not in the middle of the corridor, can be of value, if the saved time span can result in insertion of some good samples. Correspondingly, for some configuration spaces, wherein the assumption does not hold well, the time saved may as well not be of value.

In this technique first the samples q_1^{obs} and q_2^{obs} in C^{obs} are generated. First a queue based search is used to search for any sample q^{free} in C^{free} that lies in between q_1^{obs} and q_2^{obs}. The requirement is given by (11).

$$q^{free} \in \left\{ q \in C^{free} : \lambda q_1^{obs} + (1-\lambda) q_2^{obs} = q, 0 < \lambda < 1 \right\}. \quad (11)$$

First the mid-point of q_1^{obs} and q_2^{obs} is checked. If the same is collision prone, then the mid-point in-between q_1^{obs} and q_2^{obs} and the earlier computed mid-point is checked. In this manner, at every level the mid-points of the previously checked points are checked for the possibility of a collision. The first computed collision-free point is returned. The search practically performs better than a linear search due to the size of the obstacle.

Now the algorithm assumes that there is no obstacle between q_1^{obs} and q^{free} and uses a binary search to compute the sample ∂q_1. Similarly the algorithm assumes no obstacle between q_2^{obs} and q^{free} and uses a binary search to compute the sample ∂q_2. Increasing visibility over a certain amount may not really be worth the computational time. Similarly the mid-placement is more important for narrow corridor like hard scenarios, rather than scenarios with wide open spaces. The binary search is hence distance limited, and if the estimated distance between the valid samples increases more than a threshold (η), no further attempt is made to compute the precise middle of the corridor. The pseudo-code is given by Algorithm 1. Here Δ is the resolution of collision checking.

Algorithm 1: Approximate Mid-Corridor Sampler

Sample q_1^{obs} and q_2^{obs} in C^{obs}
Search for q^{free} between q_1^{obs} and q_2^{obs}
$\partial q_1 \leftarrow q^{free}$, $\partial q_2 \leftarrow q^{free}$
if q^{free} does not exist, return null
while true
    if $d(q_1^{obs}, \partial q_1) \geq \Delta$
        $q_{mid} \leftarrow 0.5\ q_1^{obs} + 0.5\ \partial q_1$
        if $q_{mid} \in C^{free}$, $\partial q_1 \leftarrow q_{mid}$
        else $q_1^{obs} \leftarrow q_{mid}$
    if $d(q_2^{obs}, \partial q_2) \geq \Delta$
        $q_{mid} \leftarrow 0.5\ q_2^{obs} + 0.5\ \partial q_2$
        if $q_{mid} \in C^{free}$, $\partial q_2 \leftarrow q_{mid}$
        else $q_2^{obs} \leftarrow q_{mid}$
    if $d(q_1^{obs}, \partial q_1) < \Delta$ and $d(q_2^{obs}, \partial q_2) < \Delta$, break
    if $d(\partial q_1, \partial q_2) > \eta$, break
$q_{mid} \leftarrow 0.5\ \partial q_1 + 0.5\ \partial q_2$
if $q_{mid} \in C^{free}$, return q_{mid}
else return null

D. Narrowness Biased Mid-Corridor Sampler

The approach, unlike the uniform sampling technique, samples the obstacles and is hence naturally in a position to sample out the narrow corridors as they are discovered. However since the two obstacle samples are randomly chosen, it is possible that a large number of samples are generated in wide open spaces and a smaller number of samples are generated in the narrow corridors. Generation of samples inside the narrow corridor is not the only challenge, to redundantly connect them with the rest of the roadmap is a harder challenge. This requires additional samples inside the narrow corridor.

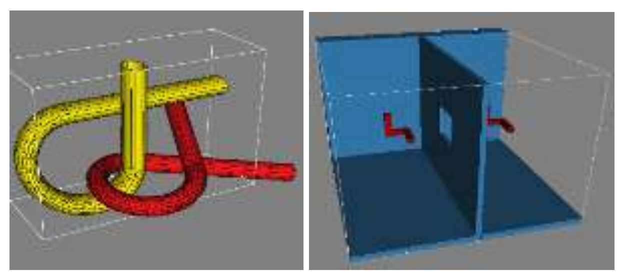
Hence a strategy proposed is to have larger number of samples inside the narrow corridor and a smaller number of samples in the wide open areas. In order for a sample to be in a narrow corridor a valid sample q^{free}, must exist in-between two invalid samples q_1^{obs} and q_2^{obs} in C^{obs}. The maximum width of the corridor is given by $d(q_1^{obs}, q_2^{obs})$. The actual width may be much smaller since q_1^{obs} and q_2^{obs} may not be boundary points and the smallest width of the pair of obstacle may not be in the direction of the line from q_1^{obs} to q_2^{obs}.

This sampler is similar to the approximate mid-corridor sampler with the only difference that the initial samples q1obs and q2obs are chosen such that the distance between them is taken from a Gaussian Distribution, that is, $q_1^{obs} \sim U(C^{obs})$, $q_2^{obs} \sim N(q_1^{obs}, \sigma)$: $q_2^{obs} \in C^{obs}$. Here σ is the standard deviation, which is an algorithm parameter. $U()$ is the uniform distribution function, while $N()$ is the normal distribution function. The algorithm samples the width of the corridor. As per the generic working of the Gaussian distribution, the smaller width corridors get sampled more while the larger width corridors get sampled less. This creates a biased distribution of samples in favor of the samples inside the narrow corridor. The additional samples in the narrow corridor enable better visibility of the narrow corridor and a better connectivity with the rest of the roadmap.

III. RESULTS

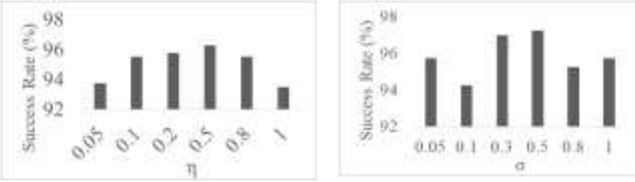
The proposed samplers were developed in the Open Motion Planning Library (OMPL) [16]. Two scenarios are used for testing, the Alpha-1.5 problem and the Twistycool problem. Both the scenarios have complex narrow corridor in the configuration space. The problems are given in Fig. 2. All experiments are done on a standalone system with 4 GB RAM and Intel i7 2.0 GHz 2.0 GHz processor.

Figure 2. (a) Alpha-1.5 Problem (b) Twistycool Problem



The first task is parameter optimization. The exact sampler uses no parameter. The approximate sampler has the parameter η which is used to threshold the binary search in wide open corridors. Corridors larger than η are wide enough and the sample is directly placed in the middle without necessitating a search. Since the binary search is much faster than the time invested in collision checking, spending a few extra iterations has small effect in terms of computational time. So the parameter has very little effect on the algorithm performance. The results confirm the same and are shown in Fig. 3(a). The parameter value is taken as a ratio of the maximum permissible distance. Since the scenarios are difficult, the only metric used is the percentage of times a solution is found. The testing is done for a class of very difficult problems, for which finding a single feasible solution is very difficult and not for problems where feasible solutions are easily available. For experiments, no appreciable difference in solution lengths between methods was observed and hence the metric. The results reported are an average of 400 runs. Each run is restricted to 20 seconds.

Figure 3. Parameter Tuning for (a) η (b) σ



Similarly the narrowness biased sampler uses a Gaussian distribution for sample generation, which has a parameter as the standard deviation (σ). Using the best parameter value of η, further experiments are done to select the best value of the parameter σ under the same settings. The results are shown in Fig. 3(b). Again the differences are not very large since the basic sampling technique itself generates samples inside narrow corridors due to promotion of samples from both ends.

The algorithm is compared against all popular sampling based multi-query planners. These include the uniform sampling PRM, obstacle based PRM, bridge-test sampling based PRM, Gaussian sampling PRM, maximum clearance sampling based PRM, SPARS and SPARS2. As it can be clearly seen from Fig. 4(a) that the proposed sampler performed the best as compared to all approaches. Moreover the performance was significantly better than all other approaches. Amongst the three samplers, the exact mid-corridor and approximate mid-corridor placement samplers had the same performance. The performance of the narrowness biased sampler was a percent less. To test the algorithms, another scenario, Twistycool was used. The results reported are an average of 400 runs. Each run is restricted to 20 seconds. The results are shown in Fig. 4(b). Again the proposed samplers performed significantly better than all the other samplers. The exact mid-corridor sampler and approximate mid-corridor sampler had the same best performance, while the narrowness biased sampler had a percent less performance.

Figure 4. Comparative Analysis on (a) Alpha-1.5 Problem (b) Twistycool Problem for 20 secs

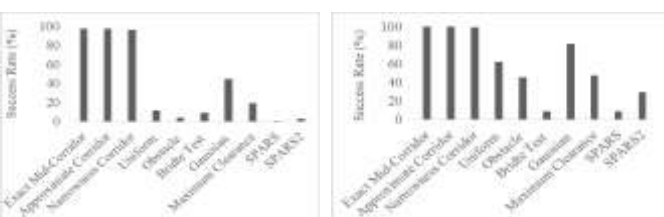
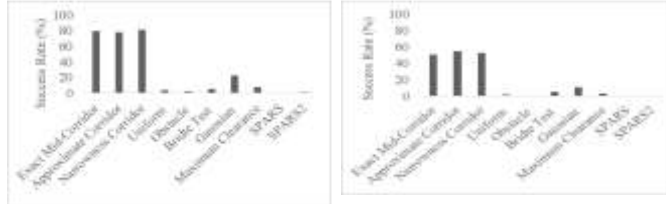


Figure 5. Comparative Analysis on the (a) Alpha-1.5 Problem ((b) Twistycool Problem for 10 secs

Figure 6. Comparative Analysis on the (a) Alpha-1.5 Problem ((b) Twistycool Problem for 5 secs.



The experiments were repeated to better study the performance in lower time limits and the results are shown in Fig. 5 for a time limit of 10 seconds and Fig. 6 for a time limit of 5 seconds. The results repeat exactly the same trends with respect to the difference with the other samplers, while the three proposed samplers had a similar performance.

The results are obviously surprisingly good, exceeding all the available multi-query planners available in OMPL by a significantly large margin. The goodness of the results is proved on prolonged number of runs and for different scenarios. This is considering the fact that the proposed samplers are extremely simple in nature and can be coded in any programming platform in just a very few lines of code. The interestingness of the results provokes further research in the same area.

IV. DISCUSSIONS

The major criterion of assessment is the early completeness of the algorithm; that is the algorithm should be able to compute a path as soon as possible. For the purpose of discussions, let us assume that the algorithm checks connections between all pairs of vertices rather than the nearest *k* vertices in the roadmap. Let $V(q)$ be the visibility of C^{free} from q, wherein any point in $V(q)$ can be directly connected to q. Any roadmap based planning algorithm under the settings is complete if (12).

$$\bigcup_{q_i \in R} V(q_i) = C^{free} . \qquad (12)$$

Here R is the set of vertices in the roadmap. Let us start with a PRM approach. The aim is to continuously sample out and grow the graph based on some sampling strategy (S) and hence the strategy is given by repeated calls to $q_i \sim S(C^{free})$, with a visibility $V(q_i)$. Let $\Delta: C^{free} \rightarrow C^{free}$ be the algorithm to promote the sample from q_i to the increased visibility sample $\Delta(q_i)$, that is $Volume(V(\Delta(q_i))) > Volume(V(q_i))$. The excess contribution to the criterion (11) of roadmap is given by $Volume(V(\Delta(q_i)) - V(q_i))$, with a loss of $Volume(V(q_i) - V(\Delta(q_i)))$ in visibility metrics. Since, $Volume(V(\Delta(q_i))) > Volume(V(q_i))$ there is definitely an added advantage. Another assumption made here is that increased visibility is not in the region of C^{free} which is already visible by some other sample q_j. That is, more correctly the improvement is given by (13) and the loss by (14).

$$V_1 = Volume\left(V(\Delta(q_i)) - V(q_i) - \bigcup_{q_j \in R, j \neq i} V(q_j)\right). \qquad (13)$$

$$V_2 = Volume\left(V(q_i) - V(\Delta(q_i)) - \bigcup_{q_j \in R, j \neq i} V(q_j)\right). \qquad (14)$$

In the design of the sampler it was stated how the sampler avoids getting too many samples in the same place by under-sampling of directions, and therefore the bias towards over-sampling in similar areas by the new scheme considering large and complex space is minimal. Hence the improvements are more than the losses, and the sampler is expected to perform well.

However the improvement in performance came with an additional cost of $T(\Delta(q_i))$, where $T()$ is the computation time of the algorithm, which could have been used to inject more useful examples that also aided in increasing visibility. On insertion of the sample, the collision checking algorithm works. Since the metric is completeness, the sample q_i will be connected to at least one sample q_j by the collision checking algorithm which will return a feasible connection, in which case taking time proportional to $d(q_i,q_j)$. The mid-placement algorithm in the worst case performs an exact search and takes a computation time of $d(q_1^{obs},q_2^{obs})$. The complexity of the two algorithms is the same, while the constant is larger for the algorithm Δ since q_1^{obs} and q_2^{obs} are large distance apart. The approximate algorithm uses a complexity of $\log(d(q_1^{obs},q_2^{obs}))$, which is smaller than the one used by the collision checking algorithm, while the Gaussian sampler uses the same complexity with a smaller value of $d(q_1^{obs},q_2^{obs})$ which came from a Gaussian distribution. The excess computation time is hence an issue primarily with the exact sampler, wherein the excess computation is clearly small.

The little added computational time is not very limiting, since the number of samples due to increased visibility sampling will be smaller and hence the general complexity of the PRM approach $O(|R| \log |R|)$ will lead to computational benefit. A little cheat in the discussions is that in the working of the algorithm q_i was not promoted to $\Delta(q_i)$, rather a q_i maximizing visibility was directly computed. This negates the possibility to use existing customized samplers for further improvement, which is not a loss since nearly all samplers are made up from the basic samplers which can all be easily formulated in the generic framework used in the paper. Overall, the proposed approach is able to give very good results for very hard problems involving narrow corridors. This is proven experimentally, has a strong intuitive background and is further formally assessed in this section. For comparisons on an experimental level, all available samplers in OMPL were tried. From a theoretical basis more samplers were critiqued.

The current work using the approach involves hybridization. As the basic scheme the simplest method of using fixed contributions of different samplers has already been tried. The basic samplers have a significantly increased performance as a result of hybridization that still does not match the performance of the proposed sampler. However, when the proposed sampler is used, the performance does not significantly improve with hybridization. This adds on to the belief of the effectiveness of the proposed algorithm. There is less motivation that adaptive hybrids will improve performance, but that needs to be tested. The purpose behind the approach was to solve the pick and place problems involving very complex and narrow scenarios. The making of complex data sets and associated motion planning queries is left for the future. In the future aim is also to test the sampler on single query motion planning algorithm. Also, optimality is a factor that needs to be considered in the future version of the algorithm. The sampler is designed and tested on the scenario of narrow corridors on high dimensional search spaces. The overall aim of motion planning is to create a single algorithm for all possible simple to complex scenarios. The adaptation of the sampler for other scenarios as well needs to be studied.

V. CONCLUSIONS

The paper aimed at creation of a new sampling methodology that very quickly increases the visibility of the sample and thus helps in its connection to the rest of the roadmap. The sampled was aimed at solving complex and narrow-corridor-like scenarios in motion planning. The sampler worked by generation a free sample amidst two sampled obstacle samples, and thereafter moving the free sample to the middle of the corridor so found. An approximate version, exact version and a Gaussian sample version of the algorithm were proposed.

The samplers are very simple with a few lines of code only and surprisingly performed exceedingly well. They easily surpassed all multi-query motion planning algorithms available at the OMPL. This presents a fundamentally new look into sampling. Even though efforts on the use of visibility and clearance existed in the literature, theoretically and experimentally the proposed sampler over-performed those methods. This is due to the effective use of heuristics to get a sample quickly with very high visibility.

REFERENCES

[1] R. Tiwari, A. Shukla, R. Kala. *Intelligent Planning for Mobile Robotics: Algorithmic Approaches*. IGI Global Publishers, Hershey, PA, 2013.

[2] L.E. Kavraki, P. Svestka, J.C. Latombe, M.H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, 1996, pp. 566–580.

[3] L.E. Kavraki, J.C. Latombe, "Probabilistic Roadmaps for Robot Path Planning," In *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, John Wiley, 1998, pp. 33-53.

[4] R. Bohlin, L.E. Kavraki, "Path planning using lazy prm," In: *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE, 2000, pp. 521–528.

[5] S. Karaman, E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotic Research*, vol. 30, no. 7, 2011, pp. 846–894.

[6] N.M. Amato, O.B. Bayazit, L.K. Dale, C. Jones, D. Vallejo, "Obprm: An obstacle-based prm for 3d workspaces," In *Robotics: The Algorithmic Perspective*, A.K. Peters, 1998, pp. 155–168.

[7] V. Boor, M.H. Overmars, A.F. van der Stappen, "The Gaussian sampling strategy for probabilistic roadmap planners," In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation vol. 2*, IEEE, 1999, pp. 1018-1023.

[8] D. Hsu, T. Jiang, J. Reif , Z. Sun Z, "The bridge test for sampling narrow passages with probabilistic roadmap planners," In: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation* vol. 3, IEEE, 2003, pp. 4420-4426.

[9] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, J.H. Reif, "Narrow passage sampling for probabilistic roadmap planning", *IEEE Transactions on Robotics*, vol. 21, no. 6, 2005, pp. 1105–1115.

[10] I. Sucan, "Maximize Clearance Valid State Sampler Class Reference". *Open Motion Planning Library*, Available at: http://ompl.kavrakilab.org/classompl_1_1base_1_1MaximizeClearanceValidStateSampler.html.

[11] C. Nissoux, T. Simeon, J.P. Laumond, "Visibility based probabilistic roadmaps," In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, IEEE, 1999, pp. 1316-1321.

[12] T. Siméon, J.P. Laumond, C. Nissoux, "Visibility-based probabilistic roadmaps for motion planning," *Advanced Robotics*, vol. 14, no. 6, 2000, pp. 477-493.
[13] A. Dobson, K.E. Bekris, "Sparse Roadmap Spanners for Asymptotically Near-Optimal Motion Planning," *International Journal of Robotics Research*, vol. 33, no. 1, 2014, pp. 18-47.
[14] D. Hsu, G. Sanchez-Ante, Z. Sun, "Hybrid PRM Sampling with a Cost-Sensitive Adaptive Strategy," In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 3874–3880.
[15] R. Kala, "Homotopic Roadmap Generation for Robot Motion Planning," *Journal of Intelligent and Robotic Systems*, vol. 82, no. 3, 2016, pp. 555–575.
[16] I.A. Şucan, M. Moll, L.E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, 2012, pp. 72–82.