

A Region-specific Hybrid Sampling Method for Optimal Path Planning

Regular Paper

Chengcheng Zhong¹ and Hong Liu^{1*}

¹ Key Laboratory of Machine Perception (Ministry of Education), Shenzhen Graduate School, Peking University, China

*Corresponding author(s) E-mail: hongliu@pku.edu.cn

Received 23 August 2015; Accepted 11 March 2016

DOI: 10.5772/63031

© 2016 Author(s). Licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

Finding high quality paths within a limited time in configuration space is a challenging issue for path planning. Recently, an asymptotically optimal method called fast marching tree (FMT*) has been proposed. This method converges significantly faster than its state-of-the-art counterparts when addressing a wide range of problems. However, FMT* appears unable to solve the narrow passage problem in optimal path planning, since it is based on uniform sampling. Aiming at solving this problem, a novel region-based sampling method integrating global scenario information and local region information is proposed in this paper. First, global information related to configuration space is extracted from an initial sample set obtained via hybrid sampling. Then, local regions are constructed and local region information is captured to make intelligent decisions regarding regions that are difficult and need to be boosted. Finally, the initial sample set is sent to FMT* using a modified locally optimal one-step connection strategy in order to find an initial and feasible solution. If no solution is found and time permits, the guided hybrid sampling will be adopted in order to add more useful samples to the sample set until a solution is found or the time for doing so runs out. Simulation results for six benchmark scenarios validate that our method can achieve significantly better results than other state-of-the-art methods when applied in challenging scenarios with narrow passages.

Keywords Optimal Path Planning, Narrow Passage Problem, Non-uniform Sampling, Region-based

1. Introduction

Over the past number of decades, path planning in static scenarios has been extensively studied. In terms of configuration space, the basic problem can be generally described as computing a collision-free path for an object to move in, from a given start configuration to a goal configuration. Sampling-based methods such as probabilistic roadmaps (PRM) [1], rapidly exploring random trees (RRT) [2] and the many variants of these have shown outstanding performance, especially in high-dimensional configuration space. However, traditional sampling-based methods do not pay much attention to path quality (path cost), which can be measured in terms of length, clearance, smoothness, energy consumption, etc. Hence, when considering path quality, the majority of traditional methods produce suboptimal solutions. This phenomenon has promoted the study of optimal path planning, which targets improved path quality.

Concerning path quality, recent asymptotically optimal sampling-based methods have shown promising results. Karaman and Frazzoli provide a definition of asymptotic

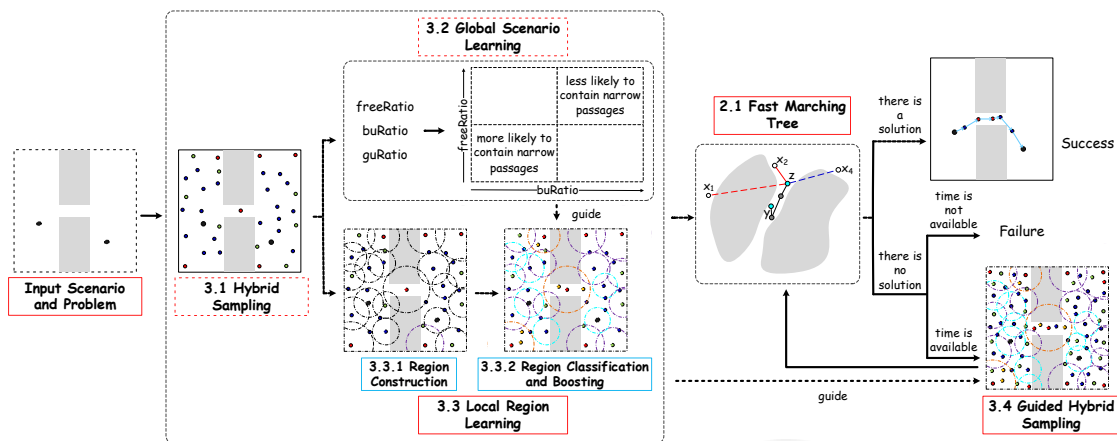


Figure 1. Flowchart. Our method is primarily described in section 2 and section 3, and corresponds to the pipeline shown in this figure.

optimality and prove that PRMstar (PRM*), rapidly-exploring random graph (RRG) and RRTstar (RRT*) can converge to an optimal solution, as the number of samples approaches infinity, with probability one [3]. Although these methods have solutions that can be improved over time, they do not guarantee a reasonable rate of convergence. To compensate for this problem, many variants of PRM* and RRT* have been proposed for improving efficiency. Building on PRM*, some works [4, 5] that provide guaranteed sub-optimality are presented, where optimality is traded for faster computation. Salzman and Halperin present the lower bound tree RRT (LBTRRT) method [6] which is also asymptotically-near optimal. In order to mitigate the greediness of RRT*, RRTsharp (RRT[#]) [7] and RRTx (RRT*) [8] are developed using the idea of lifelong planning A* (LPA) [9] in a rewiring cascade. A lazy strategy is also adopted in lazy PRM* [10] and lazy RRG [11], where the core idea is to delay the collision check until it absolutely necessary. Additionally, many different heuristics for speeding up convergence are presented in [12- 17].

Another asymptotically optimal method, proposed by Janson and Pavone, is fast marching trees (FMT*) [18]. To alleviate the inherently unordered search of sampling-based methods, FMT* extends graph-search methods to sampling-based methods and concurrently performs graph construction and graph search within cost-to-come space. It is faster than PRM* and RRT* in terms of converging to an optimal solution within many different scenarios. However, FMT* selects a predetermined number of samples and is not an any time method. Generally, an any time method can quickly find an initial feasible solution and then, given more computation time, improve the solution towards an optimal solution [19]. As FMT* selects a predetermined number of samples, if a lower cost path is needed, it must increase the number of samples and replan. Moreover, it is difficult to select an appropriate number of samples, which is problem-dependent. To solve these problems, an anytime version FMT (aFMT*) is adopted in

[20]. It chooses a small number of samples and applies FMT*, then doubles this number and repeats the process if time permits. In addition, to speed up convergence, motion planning using lower bounds (MPLB) [20] extends aFMT* by including the lower bounds on cost, while batch informed trees (BIT*) [21] introduces an ellipsoidal heuristic. However, as a sampling-based method, FMT* with a uniform sampling strategy suffers from the notorious narrow passage problem [22], where sampling in small volumes is difficult. If a robot has to pass a narrow passage to arrive at the goal position in a scenario, it may cause significant challenges for FMT* and the above noted extended methods.

To cope with the narrow passage problem, many non-uniform sampling strategies have been proposed to increase the proportion of free configurations in some areas. These include obstacle-based PRM (OBPRM) [23] and Gaussian PRM [24] sample nodes near the surfaces of obstacles; a bridge test sampler [25] generates nodes inside narrow passages. Medial axis PRM (MAPRM) [26] retracts nodes to the medial axis of the free space. Although there exist many sampling strategies, no one outperforms all others for all problem instances. To achieve a better distribution of samples, different sampling strategies are combined in [27, 28]. In addition, region-based methods [29,30] divide the configuration space into some local regions and use local region information to decide where to boost sampling intelligently, or which sampler to apply. However, some region-based methods use only the local region information and no other global information for guiding the region classification. Concerning global information, Rantanen used the connectivity of the entire PRM roadmap to decide which regions were useful [31].

In this paper, we propose a region-based sampling method integrating global scenario information and local region information to solve the narrow passage problem in optimal path planning. The flowchart for doing so is shown in Fig.1. First, hybrid sampling is adopted to sample a small predetermined number of samples. These samples can be

classified into three types: uniform (blue), Gaussian (green) and bridge (red). Second, the global scenario is characterized based on the proportion of these samples. Third, the configuration space is partitioned into regions and local region information is exploited to determine the difficulties of regions. According to the difficulties of regions, we classify the regions into three categories: easy (cyan), normal (purple) and difficult (orange). Then, more samples (yellow) are added to the difficult regions. Finally, the initial sample set is sent to FMT* with a modified locally optimal one-step connection strategy to solve the optimal path planning problem. If our method cannot find a solution on the current sample set and if extra time is available, the following process is repeated: double the number of samples using the guided hybrid sampling and apply FMT*. Since our goal is to find a low-cost solution in challenging scenarios within the shortest space of time, a novel measure of cost called harmonic mean cost is also utilized for evaluation to better describe the performance of asymptotically optimal methods.

The remainder of this paper is organized as follows. Section 2 describes some preliminaries, including FMT*, aFMT*, a modified locally optimal one-step connection strategy of FMT* and quartiles. Section 3 gives the details of our method. Experiments and discussions pertaining to six scenarios are shown in section 4. Finally, we come to conclusions and discuss possible extensions in section 5.

2. Preliminaries

Let $C \in \mathbb{R}^d$ be a configuration space (C-space) with d dimension, while C_{free} and $C_{obstacle}$ denote free space and obstacle space, respectively. The configuration of a robot is described as $x \in C$. The start configuration is $x_{init} \in C_{free}$ and the goal region $C_{goal} \subseteq C_{free}$ is a set of goal configurations. A function $\sigma: [0,1] \rightarrow \mathbb{R}^d$ is called a path if it is continuous. Therefore, a path planning problem can be defined as $(C_{free}, x_{init}, C_{goal})$ to find a feasible path σ , which lies in C_{free} such that $\sigma(0) = x_{init}$ and $\sigma(1) \in C_{goal}$. In addition, the cost function $c(\sigma)$ calculates the arc length of σ with respect to the Euclidean metric. Therefore, an optimal path planning problem is to find a feasible path σ^* such that $c(\sigma^*) = \min\{c(\sigma) : \sigma \text{ is feasible}\}$.

2.1 Fast Marching Tree

Fast marching tree is an asymptotically optimal method. Its anytime version is outlined in Alg.1 and its notations and functions are described in Tab.1. First, aFMT* samples a small number of free samples; then, these samples, x_{init} and at least one sample in C_{goal} are placed into the sample set V (Alg.1, line 3). The set V is partitioned into three subsets, $V_{unvisited}$, V_{open} and V_{closed} (Alg.1, line 4). The algorithm repeats the following process: the lowest-cost node z is chosen from V_{open} (Alg.1, line 17). For each neighbour x of z in $V_{unvisited}$,

aFMT* finds its neighbours X_{near} in V_{open} and a locally optimal one-step connection from y_{min} to x (Alg.1, lines 8-10). If the connection from y_{min} to x is collision-free, add this edge to the tree and move x from $V_{unvisited}$ to V_{open} (Alg.1, lines 11-13). After each neighbour x of z has been traversed, z is moved from V_{open} to V_{closed} (Alg.1, line 14). This process repeats until V_{open} is empty (Alg.1, lines 15-16). If time is available, double n and repeat the algorithm (Alg.1, line 18).

As sampling-based methods, FMT* and extended methods [20, 21] employing a uniform sampling strategy meet the challenge posed by the narrow passage problem. To cope with this situation, they have to increase the number of samples until there are enough samples in narrow passages. Obviously, this is a time-consuming process. Non-uniform sampling strategies are widely used to solve the narrow passage problem and the asymptotic optimality of FMT* with a non-uniform sampling strategy has already been demonstrated in [18]. Therefore, in this paper, a novel non-uniform sampling method will be applied in FMT* to solve the narrow passage problem.

| function/notation | description |
|-----------------------|---|
| $SampleFree(n)$ | return n random free configurations |
| $Near(x, k, V)$ | return k nearest neighbours of x in V |
| $CollisionFree(x, y)$ | return <i>false</i> if the straight line from x to y collides with obstacles |
| $Cost(x, y)$ | cost of the straight line from x to y |
| $Cost_T(x)$ | cost of the path from x_{init} to x on the tree T |
| $CostToGo(x)$ | cost of the straight line from x to C_{goal} |
| $V_{unvisited}$ | samples that have not been considered for addition to the tree |
| V_{open} | samples that have been added to the tree and are considered for further connections |
| V_{closed} | samples that have been added to the tree and are no longer considered for further connections |

Table 1. The Definitions of Notations and Functions in FMT*

Another problem, i.e., the locally optimal one-step connection strategy of FMT* (Alg.1, line 10), makes it difficult to add a new node to the tree when the y_{min} lies in a narrow passage. An illustration of this problem is given in Fig.2(a)-(c). In Fig.2(a), FMT* selects the lowest-cost node z in V_{open} (Alg.1, line 17) and finds its neighbours X_{near} in $V_{unvisited}$ (Alg.1, line 7). For each x in X_{near} , FMT* searches for a locally optimal one-step connection (Alg.1, line 10). Here, x_1 and x_2 select y as y_{min} , and x_3 and x_4 select z as y_{min} . Only the connection from z to x_3 is collision-free; thus, x_3 and this connection are added to the tree (Alg.1, lines 11-13). Following this step, z is moved to V_{closed} and is no longer considered for further connections (Alg.1, line 14). Then, x_3 is moved to V_{open} (Alg.1, line 14) and selected as the next z (Alg.1, line 17). In Fig.2(b), in the second iteration, x_1 , x_2 still select y as y_{min} and x_4 selects z as y_{min} (Alg.1, line 10). None

Algorithm 1: Anytime fast marching tree (aFMT*)**Input:** x_{init} , C_{goal} , initial number of samples n_0 **Output:** T

```

1  $V \leftarrow \{x_{init}\}; n \leftarrow n_0;$ 
2 while time permits do
3    $V \leftarrow V \cup \text{SampleFree}(n); E \leftarrow \emptyset; T \leftarrow (V, E);$ 
4    $V_{unvisited} \leftarrow V \setminus \{x_{init}\}; V_{open} \leftarrow \{x_{init}\}; V_{closed} \leftarrow \emptyset;$ 
5    $z \leftarrow \{x_{init}\};$ 
6   while  $z \notin C_{goal}$  do
7      $V_{new} \leftarrow \emptyset; X_{near} \leftarrow V_{unvisited} \cap \text{Near}(z, k, V);$ 
8     for  $x \in X_{near}$  do
9        $Y_{near} \leftarrow V_{open} \cap \text{Near}(x, k, V);$ 
10       $y_{min} \leftarrow \text{argmin}_{y \in Y_{near}} \{\text{Cost}_T(y) + \text{Cost}(y, x)\};$ 
11      if  $\text{CollisionFree}(y_{min}, x)$  then
12         $E \leftarrow E \cup (y_{min}, x);$ 
13         $V_{new} \leftarrow V_{new} \cup x; V_{unvisited} \leftarrow V_{unvisited} \setminus \{x\};$ 
14       $V_{open} \leftarrow (V_{open} \cup V_{new}) \setminus \{z\}; V_{closed} \leftarrow V_{closed} \cup \{z\};$ 
15      if  $V_{open} = \emptyset$  then
16        break;
17       $z \leftarrow \text{argmin}_{y \in V_{open}} \{\text{Cost}_T(y) + \text{CostToGo}(y)\};$ 
18       $n = 2n;$ 
19 return  $T = (V_{open} \cup V_{closed}, E)$ 

```

of these connections are collision-free. Then, z is moved to V_{closed} (Alg.1, line 14) and y is selected as the next z (Alg.1, line 17). In Fig.2(c), in the final iteration, x_1, x_2 and x_4 select z as y_{min} (Alg.1, line 10). All of these connections are invalid. Then, z is removed from V_{open} and V_{open} is empty. FMT* returns a FAILURE outcome.

To overcome the above problem, we made a modification to the locally optimal one-step connection strategy of FMT*. Namely, if x has selected y as y_{min} in previous iterations, this y cannot be y_{min} of the x again. The modified strategy is illustrated in Fig.2(a) and Fig.2(d). In Fig.2(a), the invalid connections are marked by x_1 and x_2 . In Fig.2(d), namely the second iteration, x_1 and x_2 select z rather than y as y_{min} . The connection between x_2 and z is collision-free. Then, x_2 and this connection will be added to the tree, and the tree successfully grows out of this narrow passage.

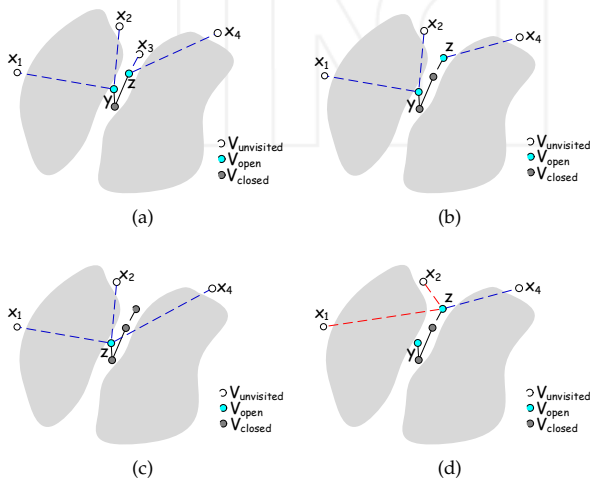


Figure 2. The locally optimal one-step connection strategy of FMT* is illustrated in (a)-(c). The modified locally optimal one-step connection strategy is illustrated in (a) and (d). Obstacles are coloured grey. The edges of the tree are depicted by solid black lines. The valid and invalid connections are depicted by solid and dotted lines, respectively.

2.2 Quartiles

In descriptive statistics, the quartiles of a ranked set of data values are the three points that divide the data set into four equal groups, each group comprising a quarter of the data [32]. As shown in Fig.3, the lower quartile Q_1 is defined as the middle value between the smallest value and the median of the data set. The median Q_2 is the median of the data set. The upper quartile Q_3 is the middle value between the median and the highest value of the data set. The interquartile range IQR is the difference between the upper and lower quartiles. The IQR is a relatively robust statistic compared to the standard deviation and the range, and can be used to characterize the data outliers lying outside the defined fences. The fences are defined as follows:

$$\begin{aligned}
 IQR &= Q_3 - Q_1 \\
 \text{lower inner fence} &= Q_1 - 1.5 * IQR \\
 \text{upper inner fence} &= Q_3 + 1.5 * IQR \\
 \text{lower outer fence} &= Q_1 - 3 * IQR \\
 \text{upper outer fence} &= Q_3 + 3 * IQR
 \end{aligned} \tag{1}$$

A point beyond an inner fence on either side is considered as a mild outlier, while a point beyond an outer fence is considered as an extreme outlier. The quartiles and the fences will be used in the region classification.

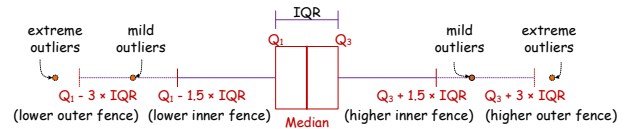


Figure 3. Box plot with quartiles and an interquartile range

3. Region-specific Hybrid Sampling Method

3.1 Hybrid Sampling

The hybrid sampling fully combines a uniform sampler, a Gaussian sampler [24] and a bridge test sampler [25], and is described in Alg.2 and Fig.4. An initial sample set V is generated by the hybrid sampling and these samples can be classified into three types: uniform (Alg.2, line 4), Gaussian (Alg.2, line 9) and bridge (Alg.2, line 13). Different from [27], our hybrid sampling does not filter out any type of free samples. Therefore, our hybrid sampling is a uniform sampling strategy and works almost as fast as a uniform sampler. Different from a uniform sampler, our hybrid sampling provides more information about global scenarios for learning.

3.2 Global Scenario Learning

Based on these different types of samples, we present three measures for characterizing the global scenario. The first is

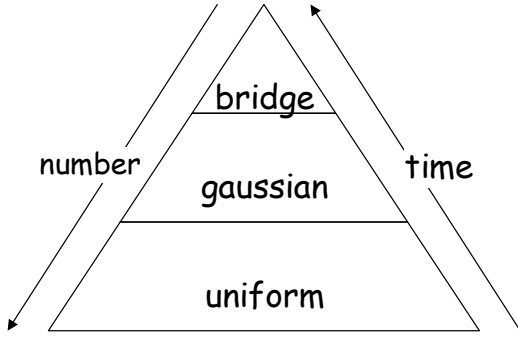


Figure 4. Hybrid Sampling. From top to bottom, the number of each type of samples is on the increase, and the time cost on them is on the decrease.

Algorithm 2: Hybrid Sampling

Input: number of samples n_0

Output: sample set V

```

1 while the number of free samples is less than  $n_0$  do
2    $x_1 \leftarrow$  a random configuration;
3   if  $x_1 \in C_{free}$  then
4      $x_1.type = uniform$ ; add  $x_1$  to the sample set  $V$ ;
5     continue;
6    $d \leftarrow$  a distance chosen according to a Gaussian distribution;
7    $x_2 \leftarrow$  a random configuration at distance  $d$  from  $x_1$ ;
8   if  $x_2 \in C_{free}$  then
9      $x_2.type = gaussian$ ; add  $x_2$  to the sample set  $V$ ;
10    continue;
11   $x_3 \leftarrow (x_1 + x_2)/2$ ;
12  if  $x_3 \in C_{free}$  then
13     $x_3.type = bridge$ ; add  $x_3$  to the sample set  $V$ ;
14    continue;
```

the ratio of free samples to all samples. It can be calculated by:

$$\begin{aligned}
 \#free &= \#uniform + \#gaussian + \#bridge \\
 \#failed &= \#attempts - \#free \\
 \#collision &= \#gaussian + \#bridge * 2 + \#failed * 3 \\
 freeRatio &= \frac{\#free}{\#free + \#collision} \in [0,1]
 \end{aligned} \tag{2}$$

Here, $\#x$ represents the number of x . This measure can estimate the volume of C_{free} and $C_{obstacle}$. The bigger the volume of $C_{obstacle}$, the bigger the $\#collision$ and the closer the $freeRatio$ approaches zero. When the $freeRatio$ is close to zero, it is difficult to obtain a free sample through a uniform sampler. Moreover, if collision samples are also added to sample set V to construct local regions as in [30], more time has to be spent on calculating the nearest neighbours, while nearest neighbour queries may serve as the computational bottleneck of asymptotically-optimal methods [33]. Therefore, hybrid sampling is adopted in this paper and only free samples are added to the sample set V .

The second and third measures are the ratio of bridge samples to uniform samples and the ratio of Gaussian samples to uniform samples, which are named $buRatio$ and $guRatio$, respectively, as follows:

$$\begin{aligned}
 buRatio &= \frac{\#bridge}{\#uniform} \in [0,1] \\
 guRatio &= \frac{\#gaussian}{\#uniform} \in [0,1]
 \end{aligned} \tag{3}$$

The $buRatio$ can characterize whether configuration space obstacles are evenly distributed and can be used to guide sampling towards more important areas. When the $buRatio$ is close to zero, it is likely that there will be large open areas in the scenario, because the configuration space obstacles are not evenly distributed. On the contrary, when the $buRatio$ is close to one, it is likely that the scenario will be cluttered with obstacles. Two corresponding instances are illustrated in Fig.5(a) and Fig.5(b), respectively. In Fig. 5(a), there are large open areas in the scenario and as such, the $buRatio$ is small. In Fig.5(b), the scenario is cluttered with obstacles and these obstacles are almost evenly distributed; as such, the $buRatio$ is large. Since it is unnecessary to keep a large number of samples in large open areas, we can use the $buRatio$ and the $guRatio$ to adjust the proportion of three types of samples in the guided hybrid sampling. This will be discussed in following subsections.

In addition, as shown in Fig.6, combining the $freeRatio$ and the $buRatio$ can provide additional information about narrow passages. When the $freeRatio$ and the $buRatio$ are both small, it is likely that the scenario contains narrow passages, and more attention should be given to this scenario.

3.3 Local Region Learning

3.3.1 Region Construction

There are many ways in which to construct a set of regions [28-31]. Our region construction method is related to [30] but differs in two aspects. First, we only consider free samples. Second, according to the sample type, start, goal, bridge and Gaussian samples have priority in terms of deciding whether to be selected as a region centre. The region construction method is depicted in Alg.3. Each new region centre is randomly selected from the samples that are not already in other regions in the subsets of V (Alg.3, line 3). Neighbouring samples are selected from V (Alg.3, line 4). Each new region radius is the maximum distance from the region centre to the samples in this region (Alg.3, line 7), while each new region's average radius is the median distance from the region centre to the samples in this region (Alg.3, line 8). Then, the $buRatio$ and the $guRatio$ of this region are calculated (Alg.3, line 10). Different regions may overlap in some areas. When all samples are included in some regions, the region construction is completed. A region construction case is illustrated in Fig.7.

3.3.2 Region Classification and Boosting

In [30], the entropy of the regions is used to classify the regions into four classes: free, surface, narrow and blocked.

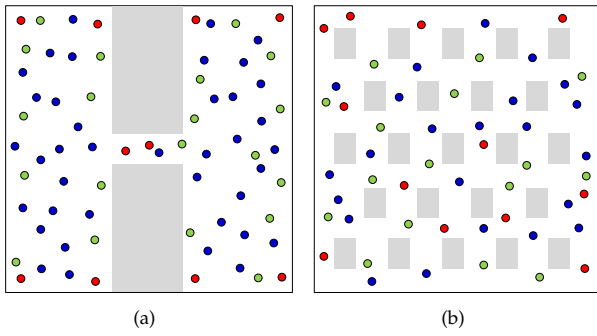


Figure 5. The *buRatio* can characterize whether configuration space obstacles are evenly distributed. Uniform, Gaussian and bridge samples are coloured blue, green and red, respectively. Obstacles are coloured grey.

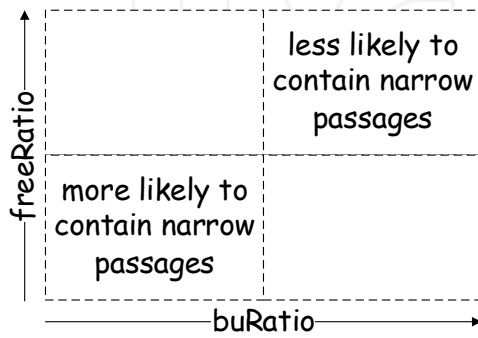


Figure 6. Combining the *freeRatio* and the *buRatio* can provide additional information about narrow passages

Algorithm 3: Region Construction.

Input: initial sample set V , which contains n_0 samples and initial number of samples in each region k'

Output: region set R

```

1 partition  $V$  into five subset:  $V_{start}, V_{goal}, V_{bridge}, V_{gaussian}, V_{uniform}$ 
2 traverse each subset according to the above sequence
3 while there exists an unmarked sample  $x$  in the current subset do
4    $N \leftarrow \text{Near}(x, k', V)$ ;
5   construct a new region  $r$ ;
6    $r.center \leftarrow x$ ;  $r.v \leftarrow \{x \cup N\}$ ;
7    $r.radius \leftarrow$  maximum distance from  $x$  to each neighbour in  $N$ ;
8    $r.avgradius \leftarrow$  median distance from  $x$  to each neighbour in  $N$ ;
9   count the number of bridge, Gaussian and uniform samples in this region:  $r.bridge, r.gaussian$  and  $r.uniform$ ;
10   $r.buRatio = \frac{r.bridge}{r.uniform}$ ;  $r.guRatio = \frac{r.gaussian}{r.uniform}$ ;
11  add  $r$  to  $R$ ;
12  mark  $x$  and  $N$ ;
```

However, to achieve relatively good performance in terms of classification, several thresholds should be fine-tuned in different scenarios. Therefore, it is crucial to develop a method that is scenario-independent and requires as little as possible user intervention. In this paper, a novel method based on statistics is introduced to classify the regions.

Since only free samples are stored, the scale of the radius for different regions varies dramatically. As shown in Fig. 7, the region radius is small in open areas, while large in

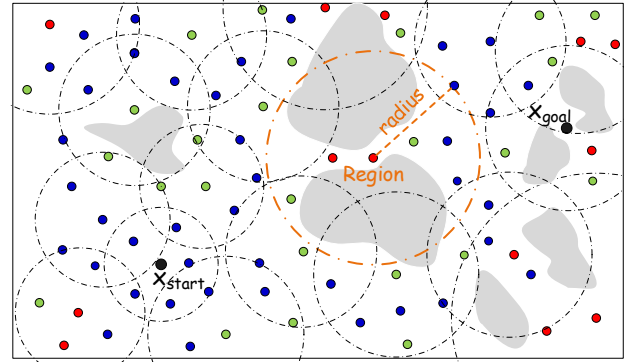


Figure 7. Region construction. Uniform, Gaussian and bridge samples are coloured blue, green and red, respectively. Obstacles are coloured grey. The initial number of samples in each region k' is set to 6 in this figure.

cluttered areas. Intuitively, we can use the region radius as a measure to determine the difficulty of a region. However, maximum and mean are not robust statistics, and they may be susceptible to outliers. Therefore, we chose the average radius rather than the radius to characterize a region. Similarly, the region average radius is small in open areas and large in cluttered areas. In addition, the quartiles were adopted to define the thresholds of the region's average radius, which was in turn utilized to classify the regions. According to the difficulties of regions, we classified them into the following categories: easy, normal and difficult.

Each region average radius is pushed into a queue. For this queue, we compute the lower quartile Q_1 , the median Q_2 , the upper quartile Q_3 , the interquartile range IQR , the lower fences and the upper fences. First, the region where average radius is beyond the lower fences or the upper fences can be considered as an "outlier"; thus, this region can be directly classified as easy or difficult. The only exception is the region where the average radius is higher than the upper outer fence and where the region centre is not a start or a goal. This case is illustrated in the purple region in the lower right of Fig.8. As the centre of this region lies in a sharp corner where it is easy to obtain a bridge sample, the average radius of this region is abnormally large. However, this region is not very difficult and is useless. Therefore, this region is classified as normal rather than difficult. Second, combining the *guRatio*, the *buRatio* and the average radius, more regions can be classified as easy or difficult. The region where average radius is beyond Q_3 has a relatively large average radius and as such, is likely to be a difficult region. The *guRatio* and the *buRatio* of this region are compared with the *guRatio* and the *buRatio* of the global scenario. The region where the *guRatio* and the *buRatio* are higher than the global *guRatio* and the global *buRatio* can be classified as difficult. Similarly, the region where the average radius is beyond Q_1 and where the *guRatio* and the *buRatio* are lower than the global *guRatio* and the global *buRatio* can be classified as easy. The rest of the regions are simply classified as normal. The above conditions and results are summarized in Tab.2. According to Tab.2, regions can be classified as easy, normal or difficult. A region classification case is illustrated in Fig.8.

| | condition | type |
|---|---|-----------|
| 1 | $r.avgraius \in [0, Q_1 - 1.5 * IQR)$ | easy |
| | $r.avgraius \in [Q_1 - 1.5 * IQR, Q_1)$ | |
| 2 | $r.buRatio < buRatio$ $r.guRatio < guRatio$ | |
| 3 | $r.avgraius \in [Q_1 - 1.5 * IQR, Q_1)$ $r.buRatio \geq buRatio$ $r.guRatio \geq guRatio$ | normal |
| 4 | $r.avgraius \in [Q_1, Q_3)$ | |
| | $r.avgraius \in [Q_3, Q_3 + 1.5 * IQR)$ | |
| 5 | $r.buRatio \leq buRatio$ $r.guRatio \leq guRatio$ | difficult |
| | $r.avgraius \in [Q_3, Q_3 + 1.5 * IQR)$ | |
| 6 | $r.buRatio > buRatio$ $r.guRatio > guRatio$ | |
| 7 | $r.avgraius \in [Q_3 + 1.5 * IQR, Q_3 + 3 * IQR)$ | normal |
| 8 | $r.avgraius \in [Q_3 + 3 * IQR, +\infty)$ $r.center.type = start$ or $r.center.type = goal$ | |
| | $r.avgraius \in [Q_3 + 3 * IQR, +\infty)$ | |
| 9 | $r.center.type \neq start$ $r.center.type \neq goal$ | normal |

Table 2. Region Classification

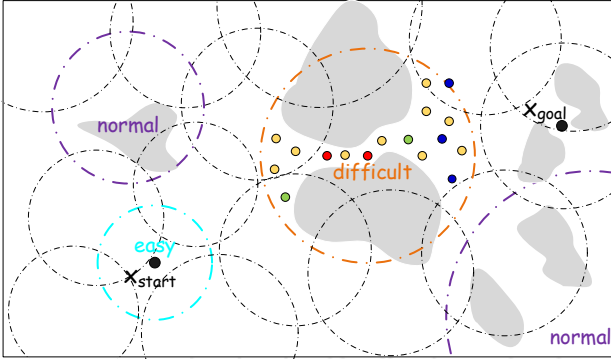


Figure 8. Region classification and boosting. Uniform, Gaussian and bridge samples are coloured blue, green and red, respectively. Boosted samples are coloured yellow. Obstacles are coloured grey.

Algorithm 4: Region Boosting

Input: region r , sample set V

Output: sample set V

```

1 if  $r$  is difficult then
2   calculate the distances from  $r.center$  to each sample in  $r$  and
   add these distances to the queue  $q$ ;
3   repeat
4     add a sample  $x$  in  $r$  using our hybrid sampling; add  $x$ 
     to the sample set  $V$ ;
5     update  $r.bridge$ ,  $r.gaussian$ ,  $r.uniform$ ,  $r.buRatio$ ,
      $r.guRatio$ ;
6     calculate the distance from  $r.center$  to  $x$  and add this
     distance to the  $q$ ;
7     choose  $k'$  shortest distances from  $q$  and calculate the
     median  $m$  of  $k'$  shortest distances;
8   until  $m \leq Q_3$ ;

```

In regions that have been classified as difficult, samples need to be increased. As shown in Alg.4, we do not set a fixed threshold for the number of increased samples. Instead, the difficult region is boosted by adding samples until the average radius of k' nearest samples of the region centre is lower than Q_3 (Alg.4, lines 3-8). The orange region in Fig.8 is a difficult region that contains a narrow passage. The number of samples in the narrow passage is dramatically increased through region boosting.

When region boosting is finished, the sample set V is sent to FMT* to find an initial feasible path. If FMT* cannot find a feasible solution for the sample set V and extra time is available, more useful samples can be included in the sample set V . This will be presented in the following subsection.

3.4 Guided Hybrid Sampling

Like aFMT*, if FMT* cannot find a feasible path on the sample set V and time permits, our method doubles the number of samples. Compared to aFMT*, the novelty of our method lies in the use of guided hybrid sampling, rather than a uniform sampling strategy. The guided hybrid sampling makes good use of global scenario information and local region information in order to add more useful samples. The algorithm of our method is detailed in Alg.5. First, if the scenario contains large open areas, which is determined by the $buRatio$, the guided hybrid sampling will adjust the proportion of uniform, Gaussian and bridge samples in this sampling process according to the $guRatio$ (Alg.5, lines 4-11). Second, when new samples are added to the sample set, we will calculate which regions the new samples belong to (Alg.5, line 13). According to the region type, if a new uniform sample is only in easy regions, it will not be added to the sample set (Alg.5, lines 14-15). In addition, when new samples are added to the sample set, the corresponding regions' information will be updated. If a new sample is not inside any region, a new region based on this sample will be constructed. Then, this new region will be classified. If it is classified as difficult, it will be boosted. When the guided hybrid sampling is complete, the sample set V will again be sent to FMT* to find a solution.

Double the number of samples using the guided hybrid sampling method and apply FMT* until an initial feasible solution is found or time runs out.

4. Experiments and Discussions

The proposed method is evaluated using six scenarios: 2D bugtrap-hardest, 3D apartment, 3D abstract, 3D twistycool, 2D maze and 2D barriers, which are shown in Fig.11, respectively. The first scenario is an extension of the bugtrap in Open Motion Planning Library (OMPL 1.0.0) [34], where the area for sampling is enlarged. The final five scenarios are provided by OMPL. To investigate the advantages of our method, the first four challenging scenarios are adopted. In these scenarios, to arrive at the

goal position, the robot has to pass a narrow passage, which presents a significant challenge for FMT* and extended methods. To investigate the performance of our method under the scenarios without narrow passages, the final two prototypical scenarios are also employed.

Algorithm 5: Guided Hybrid Sampling

Input: number of samples n_0 , sample set V

Output: sample set V

```

1  $uCount, gCount, bCount = 0;$ 
2 while the number of free samples is less than  $n_0$  do
3    $x \leftarrow$  a random configuration generated by hybrid sampling;
4   if  $x.type = uniform$  then
5     if  $bCount/uCount \geq guRatio$  or  $buRatio > threshold$ 
6       then
7          $add = true;$   $uCount \leftarrow uCount + 1;$ 
8   else if  $x.type = gaussian$  then
9     if  $bCount/gCount \geq 1$  or  $buRatio > threshold$  then
10       $add = true;$   $gCount \leftarrow gCount + 1;$ 
11   else if  $x.type = bridge$  then
12      $add = true;$   $bCount \leftarrow bCount + 1;$ 
13   if  $add = true$  then
14     calculate which regions the new sample  $x$  belongs to;
15     if  $x.type = uniform$  and the regions which  $x$  belongs to are
16       all easy then
17       reject this sample  $x;$   $uCount \leftarrow uCount - 1;$ 
18     else
19       add  $x$  to sample set  $V;$ 

```

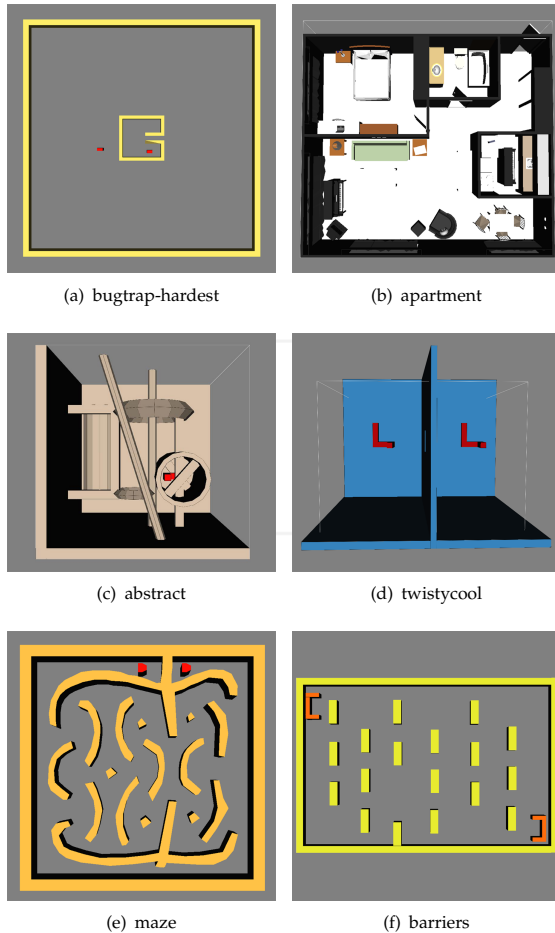


Figure 9. Benchmark scenarios. In (a) (b) (c) and (d), to arrive at the goal position, the robot has to pass a narrow passage.

We compared our method against aFMT* and RRT*, as these two methods are state-of-the-art asymptotically optimal planners. In addition, to test our modified locally optimal one-step connection strategy of FMT* and compare with other non-uniform sampling strategies, our method was also compared with aFMT* with the modified locally optimal one-step connection strategy (aFMT*+ymin), aFMT* with a bridge test sampler (aFMT*+bridge test), as well as with a combination of the two (aFMT*+bridge test +ymin).

4.1 Experiment Settings

Simulations were run on a Linux operating system (Ubuntu 12.04) on a 3.1GHz Core i5 processor with 4GB of memory. Our methods were implemented in OMPL and the implementations of RRT*, FMT*, the uniform sampler and the Gaussian sampler were taken from OMPL. The implementation of RRT* was a k-nearest version, including delaying collision checks, as well as pruning via branch and bound and node rejection. We took the default values of RRT* parameters from OMPL, a steering parameter of 3.5% of the maximum extent of the C-space and a goal-bias probability of 5%. The implementation of FMT* was a k-nearest version and the tree was expanded in a cost-to-come+cost-to-go space. We also took the default values of FMT* parameters from OMPL and a radius multiplier of 1.1. The implementation of aFMT* was based on FMT* and we sped up this method by reusing both existing samples and connections from previous iterations. The initial number of samples n_0 in aFMT* was set to 500 and other parameters were identical to FMT*. In our method, the initial number of samples n_0 was also set to 500 and the k' in the region construction and the region boosting was set to 10. We took the default values of the Gaussian sampler parameters from OMPL and the standard deviation was set to 10% of the maximum extent of the C-space. The bridge test sampler was implemented based on the Gaussian sampler and the standard deviation parameter was equal to the Gaussian sampler. All results were averaged over 100 runs and all methods run until an initial feasible solution was found or time ran out.

We tested the *freeRatio* and the *buRatio* in many different scenarios of OMPL. As shown in Fig.10, the *buRatio* was able to characterize whether configuration obstacles were evenly distributed. For example, the bugtrap-hardest (Fig. 9(a)) had large open areas and as such, its *buRatio* was close to zero. Obstacles were almost evenly distributed in the maze (Fig.9(e)) and as such, its *buRatio* was relatively large. According to our observations of these scenarios and the results in Fig.10, we provide an empirical value of 0.5 as the *threshold* in the guided hybrid sampling. Where the *buRatio* is smaller than the *threshold*, we believe that such a scenario contains large open areas and the proportion of uniform, Gaussian and bridge samples needs to be adjusted.

4.2 Self Comparisons

We tested the parameters of our method in Fig.11 and show that our method is not sensitive to parameters. Parameters were tested in the apartment scenario with 16 settings for two parameters, as well as other parameters in default values. Two parameters are shown: $250 \leq n_0 \leq 2000$, which is the initial number of samples; $5 \leq k' \leq 80$, which is the initial number of samples in each region. The vertical axis shows the success rate of our method in finding an initial feasible solution within 20 seconds in the apartment scenario. It can be estimated from Fig.11 that a too-small or too-large initial number of samples had an effect on the success rate of our method. The reasons for this can be explained from two perspectives. First, the results of region classification may not be accurate when the initial number of samples is too small. Second, the given time may not be enough for dealing with too many samples. However, when the initial number of samples is moderate, our method achieves better performance.

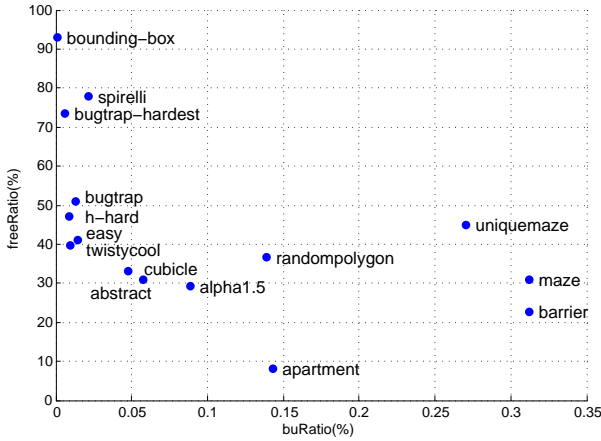


Figure 10. The *freeRatio* and the *buRatio* of different scenarios in OMPL.

4.3 Comparing with State-of-the-art Asymptotically Optimal Planners

For each scenario, we show three graphs. The first shows the success rate of finding an initial feasible solution as a function of time. The second shows the mean cost of successful attempts as a function of time. The third shows the harmonic mean cost as a function of time.

It should be noted that the mean cost of successful attempts may not be able to describe the real performance of asymptotically optimal planners when the success rate is relatively low. As shown in Fig.12(b), the mean cost function is not monotonic, since it only calculates the cost of successful attempts. For example, the mean cost of RRT* increases from 9 to 12 seconds. It can be estimated that the success rate of RRT* increases during this period of time, while the cost of new solutions achieved by RRT* is larger than before. Therefore, the harmonic mean cost, namely the harmonic mean of cost, is also adopted to describe the

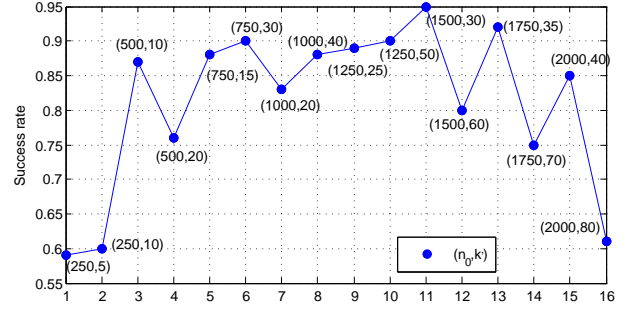


Figure 11. The success rate of our method with different parameter settings in the apartment scenario

performance of asymptotically optimal planners. This has two advantages over mean cost. First, since the harmonic mean cost can deal with the infinite cost, it considers all attempts rather than successful attempts only, which delivers a closer indication of real performance. Second, to some extent, the harmonic mean cost can reflect the success rate of algorithms. In other words, it combines two main measures of asymptotically optimal planners. If two methods have the same success rate, the harmonic mean cost only reflects the difference in terms of real cost.

4.3.1 Numerical Experiments under Challenging Scenarios with Narrow Passages

Bugtrap-hardest: the challenging bugtrap problem is proposed in [35]. To escape the bugtrap, the robot has to pass a narrow opening, while the size of free space is considerably larger than the size of narrow opening. Simulation results for this problem are depicted in Fig.12 and Tab. 3. Our method indicated a higher success rate than other methods, achieving 93% in 20 seconds, while RRT* achieved 30% and the methods with a bridge test sampler remained at 0%. As the boundary nodes were always selected for invalid extensions, RRT* was not considered useful for this scenario. It is difficult to get a sample from a bridge test sampler in large open areas and as such, aFMT*+bridge and aFMT*+bridgetest+ymin were unable to find a solution after 20 seconds. Although the success rates of aFMT* and aFMT*+ymin were similar to our method at 20 seconds, Tab.3 shows that the average number of states for our method were about 11% of the other two methods. The reason for this is that the uniform sampler wasted many samples in the large open areas of the bugtrap-hardest scenario, while our method solved this problem by adjusting the proportion of three types of samples. Taking advantage of FMT*, our method had a lower mean cost than RRT* and reached the lowest harmonic mean cost.

Apartment: the particularly challenging apartment scenario is highly cluttered with obstacles. As its *freeRatio* is low, it is difficult to obtain a free sample. Even worse, the goal position of this problem lies in a narrow corridor. Fig. 13 and Tab.3 present simulation results. Compared to other methods, our method achieved significant improvement

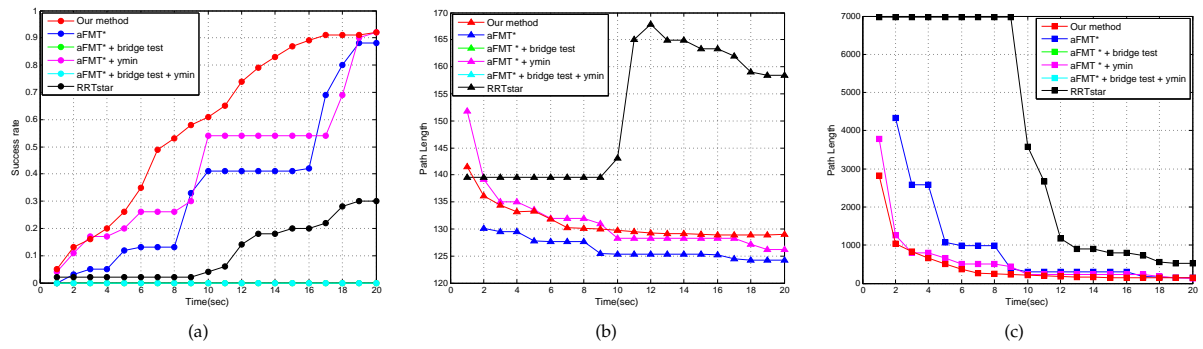


Figure 12. Simulation results for bugtrap-hardest in 2D space

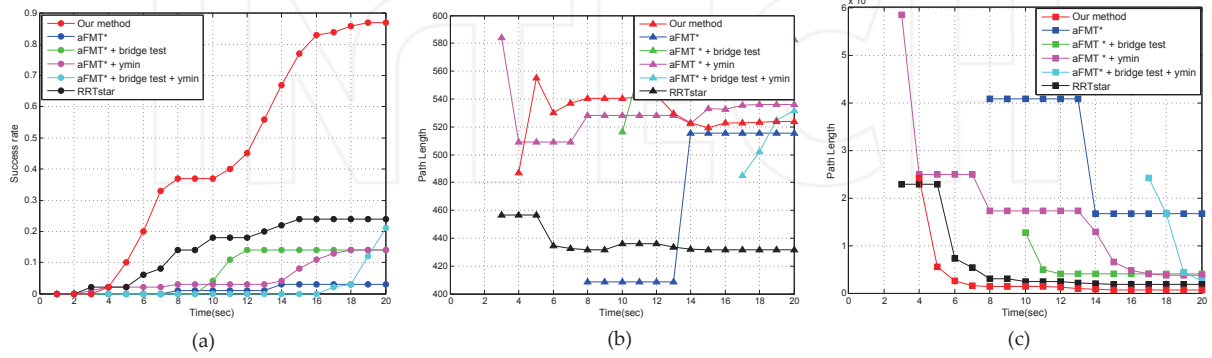


Figure 13. Simulation results for apartment in 3D space

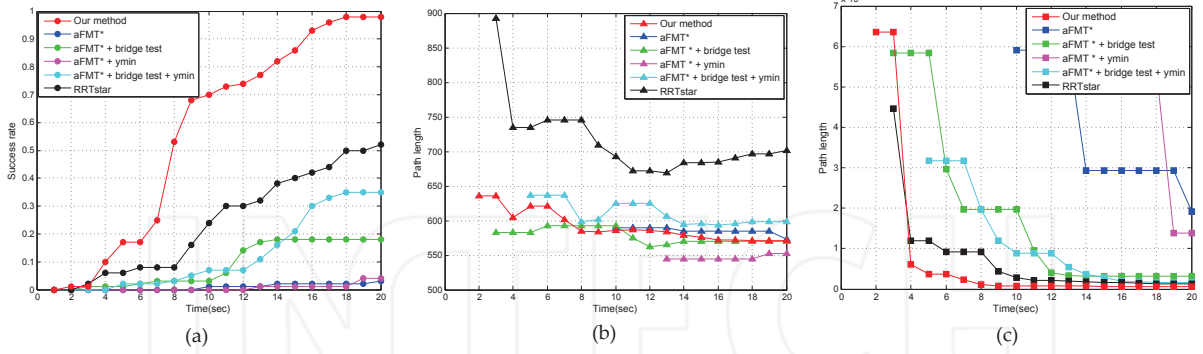


Figure 14. Simulation results for abstract in 3D space

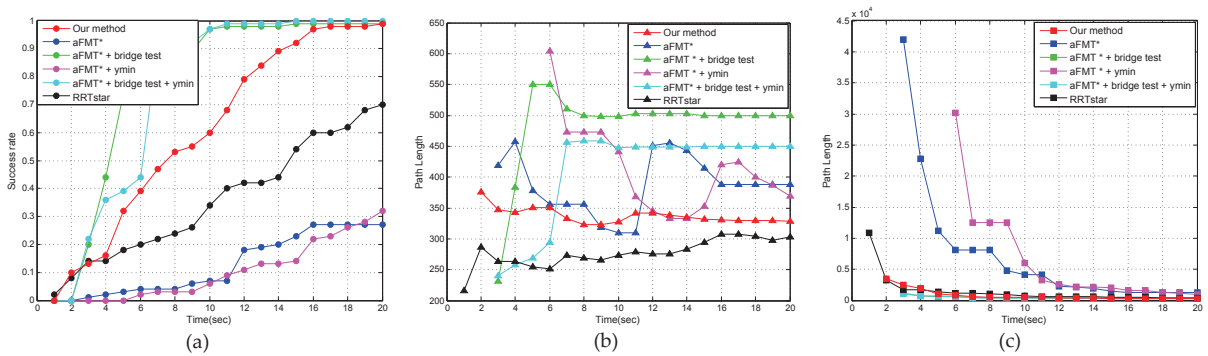


Figure 15. Simulation results for twistycool in 3D space

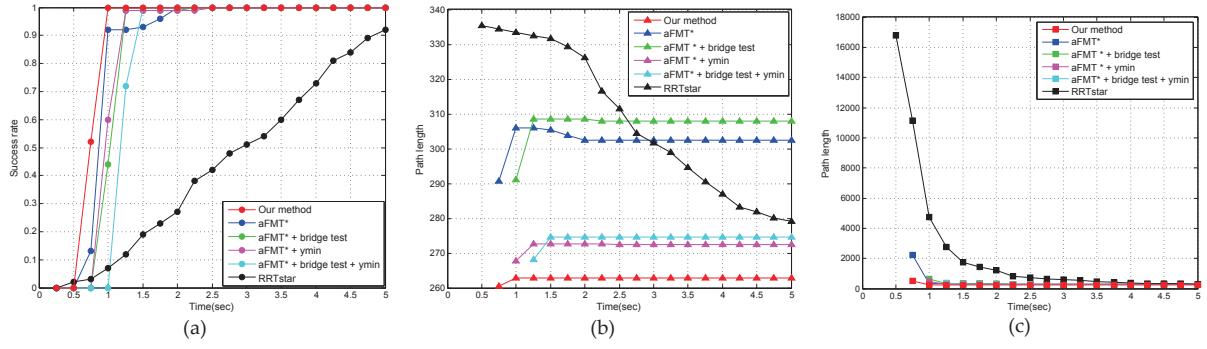


Figure 16. Simulation results for Maze in 2D space

on success rate over 20 seconds, i.e., 84% and 63% higher than aFMT* and RRT*, respectively. This improvement can be attributed to our region classification and boosting, which can effectively identify and boost difficult regions, and save a significant number of samples in the useless regions. As this scenario is cluttered with obstacles, which is suited for building bridges, aFMT*+bridge and aFMT*+bridgetest+ymin also had higher success rates than methods employing a uniform sampler. However, without a boosting strategy, their success rates were still lower than our method. Although the mean cost of our method was somewhat higher than RRT*, we can easily make up this difference by applying a post-processing method, e.g., a short cut method [36]. Additionally, as shown in Tab.3, our method required roughly 35% less states than RRT*. Again, our method reached the lowest harmonic mean cost.

Abstract: the main difficulty of the abstract scenario is that the start position of the robot is quite restricted. Moving out of the horizontal cylinder is difficult and requires several attempts. The plots for this scenario, given in Fig.14, show similar conditions to those found for the apartment scenario. Again, our method achieved the highest success rate over 20 seconds, i.e., 94% and 46% higher than aFMT* and RRT*, respectively. In this scenario, RRT*, aFMT*+bridge and aFMT*+bridgetest+ymin performed much better than methods employing a uniform sampler. The reasons for this can be explained as follows. First, it is much easier for RRT* to explore a narrow passage from the inside to the outside than to find the entrance to a narrow passage. Second, there are relatively open areas in this scenario, while more samples focus on the narrow passages by using a bridge test sampler. Moreover, the methods applied in the modified locally optimal one-step connection strategy (aFMT+ymin, aFMT+bridgetest+ymin) had a higher success rate than methods without this strategy (aFMT, aFMT+bridgetest). Fig.14(b) shows that our method had a lower cost than RRT*. As expected, our method had the lowest harmonic mean cost.

Twistycool: twistycool contains a wall with a small hole and useless open areas, and the difficulty of this scenario arises from the small hole that the robot must move through. Results in Fig.15(a) and Tab.3 show few differences from those in the previous three scenarios. First, the methods with a bridge test sampler (aFMT*+bridgetest,

aFMT*+bridgetest+ymin) had a higher success rate than our method over 20 seconds and the states they needed were less than for our method. Most of the samples achieved by the bridge test sampler were adjacent to the wall, which was sufficient for FMT* to find a path through the hole. In other words, the bridge test sampler is particularly well-suited to this scenario, while our method wasted some samples in open areas. Even so, our method still had a higher success rate and fewer states than RRT*. Second, the mean cost of our method was lower than for methods with a bridge test sampler and higher than RRT*. Therefore, the methods with a bridge test sampler had the lowest harmonic mean cost over 12 seconds, while our method had the lowest harmonic mean cost after 12 seconds.

Some methods can achieve a higher success rate over a shorter period of time in these challenging scenarios, e.g., kinodynamic motion planning by interior-exterior cell exploration (KPIECE) [37] in the abstract scenario and RRT-connect [38] in the apartment scenario. However, since path quality is not considered by these methods, comparisons were not made to them.

4.3.2 Numerical Experiments under Scenarios without Narrow Passages

Maze: navigating a maze is a prototypical problem for path planners. Simulation results for this problem are shown in Fig.16 and Tab.3. Since there were no narrow passages in this scenario, our method and other aFMT* methods had similar success rates, all higher than RRT*. Although the success rates of all the methods rose rapidly to 100%, the methods with a bridge test sampler were slightly slower than methods employing a uniform sampler and our method. The reason for this is that a uniform sampler performs fewer collision checks per sample than a bridge test sampler. Shown in Tab.3, the number of states for our method was higher than for other aFMT* methods, which can be explained as having resulted from our boosting strategy. Moreover, as our method employed more states, we had a lower mean cost and a lower harmonic mean cost than other aFMT* methods.

Barriers: barriers present another prototypical scenario for path planners. The results for this scenario, given in Fig.

| scenario | method | iteration | | | state | | | collision check | | | time(s) | | | success rate |
|-----------------|----------------------------|-----------|----------|-------------|------------|--------------|--------------|-----------------|--------------|--------------|-------------|--------------|--------------|--------------|
| | | min | max | avg | min | max | avg | min | max | avg | min | max | avg | |
| bugtrap hardest | aFMT* | 6 | 10 | 9.29 | 16002 | 256002 | 184550 | 3559 | 18267 | 13496 | 1.51 | 18.81 | 12.49 | 88% |
| | aFMT* + bridge test | - | - | - | - | - | - | - | - | - | - | - | - | 0% |
| | aFMT* + ymin | 2 | 10 | 8.63 | 1002 | 256002 | 153900 | 1183 | 19515 | 12807 | 0.26 | 19.16 | 11.09 | 92% |
| | aFMT* + bridge test + ymin | - | - | - | - | - | - | - | - | - | - | - | - | 0% |
| | RRT* | 3014 | 210649 | 147540 | 2005 | 4042 | 3037 | 6994 | 220176 | 154930 | 1 | 19 | 13.40 | 30% |
| | Our method | 1 | 7 | 5.43 | 572 | 54368 | 21702 | 939 | 13239 | 6869 | 0.13 | 20.07 | 7.88 | 93% |
| apartment | aFMT* | 2 | 3 | 2.67 | 1002 | 2002 | 1669 | 4364 | 7516 | 6412 | 7.01 | 13.86 | 11.34 | 3% |
| | aFMT* + bridge test | 1 | 1 | 1 | 502 | 502 | 502 | 4217 | 6319 | 5310 | 9.05 | 11.51 | 10.26 | 14% |
| | aFMT* + ymin | 1 | 3 | 2.64 | 502 | 2002 | 1716 | 4025 | 16197 | 12520 | 2.97 | 17.34 | 12.99 | 14% |
| | aFMT* + bridge test + ymin | 1 | 1 | 1 | 502 | 502 | 502 | 16197 | 19684 | 18557 | 16.38 | 19.96 | 18.73 | 21% |
| | RRT* | 10287 | 53714 | 31014 | 718 | 3387 | 2032 | 11911 | 59849 | 34814 | 3 | 15 | 9 | 24% |
| | Our method | 1 | 2 | 1.57 | 657 | 1965 | 1314 | 9302 | 53125 | 27457 | 3.74 | 18.65 | 10.30 | 87% |
| abstract | aFMT* | 4 | 6 | 5.25 | 4002 | 16002 | 11002 | 7647 | 9744 | 8590 | 9.21 | 20.02 | 15.58 | 4% |
| | aFMT* + bridge test | 1 | 3 | 2.78 | 502 | 2002 | 1808 | 3221 | 13178 | 9674 | 2.69 | 13.30 | 10.55 | 18% |
| | aFMT* + ymin | 5 | 6 | 5.75 | 8002 | 16002 | 14002 | 13229 | 15717 | 14817 | 12.28 | 18.81 | 17.05 | 4% |
| | aFMT* + bridge test + ymin | 1 | 3 | 2.74 | 502 | 2002 | 1773 | 8267 | 34382 | 24100 | 4.04 | 17.64 | 13.23 | 35% |
| | RRT* | 10893 | 231394 | 67195 | 1298 | 8129 | 4646 | 15598 | 235942 | 77440 | 3 | 20 | 11.65 | 52% |
| | Our method | 1 | 4 | 3.07 | 660 | 6906 | 3440 | 9544 | 54050 | 30721 | 1.97 | 17.82 | 8.91 | 98% |
| twistycool | aFMT* | 2 | 5 | 4.07 | 1002 | 8002 | 4817 | 7473 | 23861 | 19452 | 2.85 | 15.78 | 11.21 | 27% |
| | aFMT* + bridge test | 1 | 3 | 1.26 | 502 | 2002 | 638 | 505 | 17786 | 5962 | 2.24 | 14.7 | 4.73 | 99% |
| | aFMT* + ymin | 2 | 5 | 4.06 | 1002 | 8002 | 4752 | 21687 | 51424 | 41217 | 5.26 | 19.95 | 13.97 | 32% |
| | aFMT* + bridge test + ymin | 1 | 2 | 1.06 | 502 | 1002 | 532 | 963 | 51789 | 14762 | 2.29 | 14.85 | 5.42 | 100% |
| | RRT* | 1097 | 43305 | 24175 | 1120 | 12465 | 7223 | 3540 | 61188 | 34707 | 1 | 20 | 10.66 | 70% |
| | Our method | 1 | 4 | 2.61 | 524 | 14206 | 2768 | 7514 | 63742 | 39517 | 1.17 | 19.12 | 8.11 | 99% |
| maze | aFMT* | 1 | 2 | 1.08 | 502 | 1002 | 542 | 2677 | 7429 | 3544 | 0.66 | 1.89 | 0.88 | 100% |
| | aFMT* + bridge test | 1 | 2 | 1 | 502 | 1002 | 507 | 2907 | 7506 | 3635 | 0.84 | 2.22 | 1.02 | 100% |
| | aFMT* + ymin | 1 | 2 | 1.01 | 502 | 1002 | 507 | 3712 | 9730 | 4304 | 0.83 | 2.43 | 1.00 | 100% |
| | aFMT* + bridge test + ymin | 1 | 1 | 1 | 502 | 502 | 502 | 3976 | 5259 | 4551 | 1.02 | 1.43 | 1.21 | 100% |
| | RRT* | 1901 | 15373 | 8146 | 245 | 3092 | 1598 | 3574 | 27436 | 16000 | 0.5 | 5 | 2.88 | 92% |
| | Our method | 1 | 1 | 1 | 590 | 805 | 685 | 4566 | 6433 | 5441 | 0.64 | 0.87 | 0.75 | 100% |
| barrier | aFMT* | 1 | 4 | 1.54 | 502 | 4002 | 807 | 1207 | 10710 | 2706 | 0.33 | 3.90 | 0.79 | 100% |
| | aFMT* + bridge test | 1 | 2 | 1.14 | 502 | 1002 | 572 | 1760 | 5557 | 2814 | 0.69 | 2.10 | 1 | 100% |
| | aFMT* + ymin | 1 | 3 | 1.25 | 502 | 2002 | 637 | 1380 | 6774 | 2866 | 0.37 | 2.06 | 0.72 | 100% |
| | aFMT* + bridge test + ymin | 1 | 1 | 1 | 502 | 502 | 502 | 2543 | 4311 | 3629 | 0.81 | 1.22 | 1.06 | 100% |
| | RRT* | 5700 | 28114 | 15382 | 248 | 2100 | 926 | 7854 | 35764 | 19602 | 0.75 | 4 | 1.92 | 98% |
| | Our method | 1 | 2 | 1.01 | 571 | 1478 | 689 | 2135 | 4594 | 3563 | 0.37 | 0.93 | 0.57 | 100% |

Table 3. Simulation Results

17, are similar to those of the maze method. Again, our method and other aFMT* methods reached a success rate of 100% faster than RRT*. Our method had the lowest mean cost and the lowest harmonic mean cost.

The maximum, minimum and average number of states, iterations, collision checks, as well as the time utilized by

each method to find an initial feasible solution in a given time (20s in the first four scenarios, 5s in the final two scenarios), are presented in Tab.3. The results show that our method performed well in all six scenarios, particularly in the challenging scenarios with narrow passages. First, the superiority of our method is revealed both in its higher

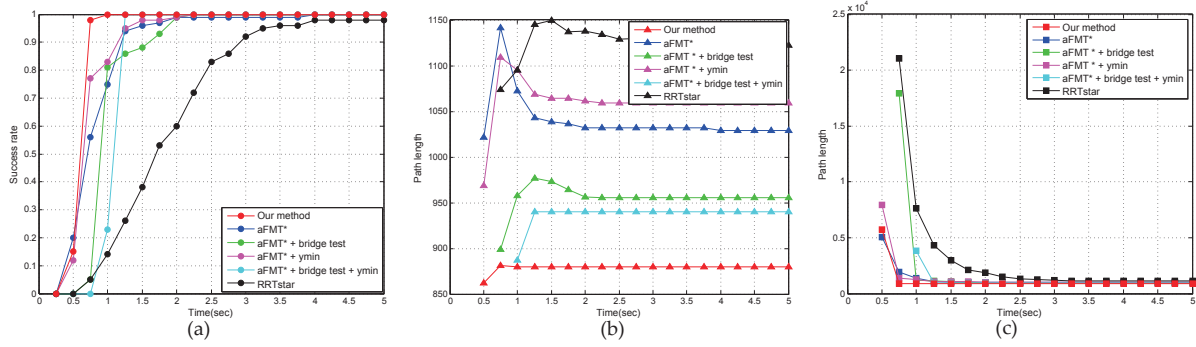


Figure 17. Simulation results for Barriers in 2D space

success rate and lower harmonic mean cost. Second, compared to other non-uniform sampling strategies and region-based methods, our method allows for the automatic tuning of parameters and is scenario-independent.

5. Conclusions and Future Work

In this paper, a region-specific hybrid sampling method is presented, which significantly enhances the performance of FMT* in the scenarios containing narrow passages. Unlike related region-based sampling methods, the configuration space is globally characterized by the proportion of uniform, Gaussian and bridge samples, and then locally captured by regions with a different radius. Our method attempts to classify regions by integrating global scenario information and local region information, and increase the number of samples in difficult regions. Where our method is unable to find a solution for the initial sample set, more samples – which are biased to difficult regions – will be added to the previous sample set. Our method outperformed other methods, since it captures rich global scenario and local region information in a statistical manner and adjusts parameters automatically according to this information. Experimental results pertaining to six scenarios validated the efficiency of our method.

Despite our method working well in simulations, there remain approaches for enhancing it further. For example, information achieved from FMT* is not employed in our method. It may be possible to use more information for updating regions into having a new status, thus increasing the efficiency of our method.

6. Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC, No.61340046, 60875050, 60675025), National High Technology Research and Development Program of China (863 Program, No. 2006AA04Z247), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20130001110011), Natural Science Foundation of Guangdong Province (No. 2015A030311034), Science and Technology Innovation Commission of Shenzhen Municipality (No. JCYJ20130331144631730, No. JCYJ20130331144716089).

7. References

- [1] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4): 566–580, 1996.
- [2] LaValle, S. M. and Kuffner J. J. Rapidly-exploring random trees: Progress and prospects. in *Proceedings of International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2000.
- [3] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.
- [4] J. D. Marble and K. E. Bekris. Towards small asymptotically near-optimal roadmaps. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2557–2562, 2012.
- [5] A. Dobson and K. E. Bekris. Sparse roadmap spanners for asymptotically near-optimal motion planning. *The International Journal of Robotics Research*, 33(1):18–47, 2014.
- [6] O. Salzman and D. Halperin. Asymptotically near-optimal RRT for fast, high-quality, motion planning. *CoRR*, abs/1308.0189v4, 2013.
- [7] O. Arslan and P. Tsotras. Use of relaxation methods in sampling-based algorithms for optimal motion planning. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2421–2428, 2013.
- [8] M. W. Otte and E. Frazzoli. RRT*: Real-time motion planning/replanning for environments with unpredictable obstacles. in *Proceedings of International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 107:461–478, 2014.
- [9] S. Koenig, M. Likhachev, and D. Furcy. Lifelong planning A*. *Artificial Intelligence*, 155(1):93–146, 2004.
- [10] J. Luo and K. K. Hauser. An empirical study of optimal motion planning. in *Proceedings of IEEE/RSJ*

- International Conference on Intelligent Robotics and Systems (IROS)*, pages 1761–1768, 2014.
- [11] K. K. Hauser. Lazy collision checking in asymptotically-optimal motion planning. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2951–2957, 2015.
 - [12] B. Akgun and M. Stilman. Sampling heuristics for optimal motion planning in high dimensions. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, pages 2640–2645, 2011.
 - [13] A. Perez, S. Karaman, A. Shkolnik, E. Frazzoli, S. Teller, and M. R. Walter. Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, pages 4307–4313, 2011.
 - [14] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, Khan M., and M. S. Muhammad. RRT*-SMART: A rapid convergence implementation of RRT*. *International Journal of Advanced Robotic Systems*, 10:299, 2013.
 - [15] D. Kim, J. Lee, and S. Yoon. Cloud RRT*: Sampling cloud based RRT*. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2519–2526, 2014.
 - [16] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot. Informed RRT*: Optimal incremental path planning focused through an admissible ellipsoidal heuristic. in *Proceedings of IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, pages 2997–3004, 2014.
 - [17] A. H. Qureshi, S. Mumtaz, A. Yasar, O. Hasan, M. S. Muhammad, and M. T. Mahmood. Triangular geometrized sampling heuristics for fast optimal motion planning. *International Journal of Advanced Robotic Systems*, 12:10, 2015.
 - [18] L. Janson, E. Schmerling, A. Clark, and M. Pavone. Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. *The International Journal of Robotics Research*, 34(7):883–921, 2015.
 - [19] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime motion planning using the RRT*. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1478–1483, 2011.
 - [20] O. Salzman and D. Halperin. Asymptotically-optimal motion planning using lower bounds on cost. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4167–4172, 2015.
 - [21] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot. Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3067–3074, 2015.
 - [22] D. Hsu, L. E. Kavraki, J. C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. *Robotics: The Algorithmic Perspective*, pages 141–154, 1998.
 - [23] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. in *Proceedings of International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 155–168, 1998.
 - [24] V. Boor, M. H. Overmars, and A. F. Van Der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1018–1023, 1999.
 - [25] D. Hsu, T. Jiang, J. Reif, and Z. Sun. The bridge test for sampling narrow passages with probabilistic roadmap planners. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4420–4426, 2003.
 - [26] S. A. Wilmarth, N. M. Amato, and P. F. Stiller. MAPRM: a probabilistic roadmap planner with sampling on the medial axis of the free space. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1024–1031, 1999.
 - [27] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. H. Reif. Narrow passage sampling for probabilistic roadmap planning. *IEEE Transactions on Robotics*, 21(6):1105–1115, 2005.
 - [28] L. Tapia, S. L. Thomas, B. Boyd, and N. M. Amato. An unsupervised adaptive strategy for constructing probabilistic roadmaps. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4037–4044, 2009.
 - [29] M. Morales, L. Tapia, S. Rodriguez, and N. M. Amato. A machine learning approach for feature-sensitive motion planning. in *Proceedings of International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 361–376, 2004.
 - [30] S. Rodriguez, S. Thomas, R. Pearce, and N. M. Amato. RESAMPL: A region-sensitive adaptive motion planner. in *Proceedings of International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 285–300, 2006.
 - [31] M. Rantanen. A connectivity-based method for enhancing sampling in probabilistic roadmap planners. *Journal of Intelligent & Robotic Systems*, 64(2):161–178, 2011.
 - [32] R. J. Hyndman and Y. Fan. Sample quantiles in statistical packages. *The American Statistician*, 50:361–365, 1996.
 - [33] M. Kleinbort, O. Salzman, and D. Halperin. Efficient high-quality motion planning by fast all-pairs r-

- nearest-neighbors. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2985–2990, 2015.
- [34] I. A. Sucan, M. Moll, and L. E. Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.
- [35] A. Yershova, L. Jaillet, T. Siméon, and S. M. LaValle. Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3856–3861, 2005.
- [36] R. Geraerts and M. H. Overmars. Creating high-quality paths for motion planning. *The International Journal of Robotics Research*, 26(8):845–863, 2007.
- [37] I. A. Sucan and L. E. Kavraki. Kinodynamic motion planning by interior-exterior cell exploration. in *Proceedings of International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 449–464, 2010.
- [38] J. J. Kuffner and S. M. LaValle. RRT-connect: An efficient approach to single-query path planning. in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 995–1001, 2000.

INTECH

INTECH