



Original Article

Adaptive Region Boosting method with biased entropy for path planning in changing environment

Risheng Kang^{a,*}, Tianwei Zhang^b, Hao Tang^a, Wenyong Zhao^a^a Engineering Laboratory on Intelligent Perception for Internet of Things (ELIP), Peking University, Shenzhen Graduate School, China^b Nakamura-Takano Lab, Department of Mechano-informatics, The University of Tokyo, 113-8685, Japan

Available online 6 October 2016

Abstract

Path planning in changing environments with difficult regions, such as narrow passages and obstacle boundaries, creates significant challenges. As the obstacles in W-space move frequently, the crowd degree of C-space changes accordingly. Therefore, in order to dynamically improve the sampling quality, it is appreciated for a planner to rapidly approximate the crowd degree of different parts of the C-space, and boost sample densities with them based on their difficulty levels. In this paper, a novel approach called Adaptive Region Boosting (ARB) is proposed to increase the sampling density for difficult areas with different strategies. What's more, a new criterion, called biased entropy, is proposed to evaluate the difficult degree of a region. The new criterion takes into account both temporal and spatial information of a specific C-space region, in order to make a thorough assessment to a local area. Three groups of experiments are conducted based on a dual-manipulator system with 12 DoFs. Experimental results indicate that ARB effectively improves the success rate and outperforms all the other related methods in various dynamical scenarios.

Copyright © 2016, Chongqing University of Technology. Production and hosting by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Keywords: Motion planning; DRM; Biased entropy classification; Hybrid boosting strategy

1. Introduction

As one of the most fundamental researches in robotic field, path planning has been applied in many other disciplines such as aviation, industrial manufacturing, games/virtual reality and so on. The generalized motion planning problem has been proved to be PSPACE-hard [1]. Therefore, many sampling based methods have been proposed to reduce computational complexity. Probabilistic Roadmap method (PRM) [2] and rapidly exploring Random Tree (RRT) [3] are two of the most successful ones with probabilistic complete guarantee. Though in the past two decades many variants of PRM [4,5] have gained great success in solving path planning problems even in high-

dimensional C-space, it still remains hard in changing environments.

Difficult region creates a significant challenge for sampling-based methods to pass through, because it is hard to put enough samples in difficult region to construct a sufficient roadmap. In static environments, many non-uniform methods [6–9] are proposed to increase sampling density in difficult regions. Some heuristic measures are also proposed to solve the difficult region problem, such as predictive models [10,11], Approximate Cell Decomposed (ACD) [12], Region-sensitive method [13] and so on. Here, Region-sensitive method identifies difficult regions with probability density estimation to obtain approximate structure of C-space. Although all the methods above perform well in static environments, there are still lots of problems required to be solved under changing environments.

Path planning in changing environments encounters great challenges since the validity of sampling points changes

* Corresponding author.

E-mail address: kang@sz.pku.edu.cn (R. Kang).

Peer review under responsibility of Chongqing University of Technology.

frequently in C-space. Dynamic Roadmap Method (DRM) [14–16] is an extension of PRM for path planning in changing environments, which is an uniform sampling method up against many troubles from difficult regions. In the past several years, only a few works [17–20] have focused on this conundrum. In changing environments, shapes and positions of regions change frequently according to the motion of obstacles. Particularly, some regions become more and more difficult and some regions become freer and freer based on the moving trend of obstacles. Hereby, the crucial issue is how to efficiently boost sampling quality in different difficult-level regions in order to increase the connection degree of the roadmap.

In this paper, a novel method named Adaptive Region Boosting (ARB) is proposed for path planning in changing environments, which could adaptively boost the sampling quality of regions based on the different crowd-degrees. The motivation of ARB is to find a heuristic measure for the “difficulty-degree” of the region around every valid sampling point. Biased entropy is proposed to evaluate difficult degree of a region. Low biased entropy represents low difficult degree, therefore the higher the biased entropy is, the more difficult the local region is.

By considering the difference between two adjacent frames, not only space information is captured, but also time information is taken into account to evaluate changing degree of a local region in updating phase. Our approach includes two steps. In preprocessing phase, a hierarchical sampling strategy is proposed to construct the roadmap by two layers of samples. In updating phase, more samples will be adaptively activated in difficult regions in order to boost sampling density within these regions.

Our contributions can be concluded as follows:

- 1) Biased entropy is proposed to evaluate the difficult degree of a specific C-space region by considering both temporal and spatial information.
- 2) Various boosting strategies are presented to adaptively increase the sampling density within difficult regions based on the various difficult degree.

The rest of this paper is organized as follows. Section 2 describes related works. Details of our method are given in Section 3. Three groups of experiments are conducted and analyzed in Section 4. Finally, conclusions are drawn in Section 5.

2. Related works

2.1. DRM

Dynamic Roadmap method (DRM) [14–16], which is a variant of PRM, has been successfully applied in changing environments. Initially, DRM generates sampling points randomly in C-space with no obstacles. Then the W-space is

decomposed into basic cells to construct the mappings between the cells in W-space and the roadmap in C-space. The core of DRM includes two kinds of mappings, node mapping and edge mapping:

$$\Phi_p(w) = \{q \in G_p | \Omega(q) \cap w \neq \emptyset\} \quad (1)$$

$$\Phi_e(w) = \{e \in G_e | \Omega(q) \cap w \neq \emptyset \text{ for some } q \in e\} \quad (2)$$

Here, $G = \{G_p, G_e\}$ is the roadmap constructed in C-space, where G_p and G_e are the nodes and edges set of the roadmap, respectively. Functions $\Phi_p(w)$ and $\Phi_e(w)$ are the mappings of nodes and edges. Nodes and edges of the roadmap are labeled as invalid if the basic cell w of W-space is occupied by obstacles. $\Omega(q)$ denotes a subset of basic cells which are occupied by a robot whose configuration is q .

However, it is too complex to compute the nodes mapping $\Phi_p(w)$ and the edges mapping $\Phi_e(w)$. Instead, we generally compute the inverse mappings $\Phi_p^{-1}(w)$ and $\Phi_e^{-1}(w)$. The computing of $\Phi_p^{-1}(w)$ [18] is that the robot in the W-space is first set to the configuration q in C-space, and then a seed cell is put inside the robot and expanded in each direction until all cells $\Omega(q)$ are found by collision detector. In general, vast time is needed to compute edge mapping in order to verify whether edge e is valid or not. Therefore, edge mapping is not used frequently in computing, while node mapping is the core of DRM method.

By quickly updating the roadmap to indicate invalid nodes and edges of the roadmap when obstacles are moving, DRM performs well in changing environments. However, difficult region problem is also a tremendous challenge for DRM planner.

2.2. Dynamic Bridge Builder method

Difficult region problem creates tremendous difficulty for path planning, as there are so few sampling points within these regions that planners can not find a collision-free path. This problem becomes more urgent in changing environments, because the positions and the shapes of narrow passages change frequently. How to increase sampling density within difficult areas is worth to be studied.

Some bridge builder planners are proposed to work out difficult regions in changing environments, such as Dynamic Bridge Builder (DBB) [18]. In preprocessing phase, DBB generates sample points and constructs a new roadmap like DRM. Then, DBB computes midpoints of edges in the roadmap and generates incremental points around those midpoints. Afterward, W-C mapping of sampling points is calculated to ensure validity of sampling points and edges. If the midpoint of an edge is valid and two endpoints are invalid, this edge is regarded as a bridge which identifies a local region of a narrow passage. In updating phase, DBB rebuilds new bridges online by traversing all the edges.

When difficult regions are identified by DBB in updating phase, two different strategies are used to boost these regions. The first one is to activate incremental points around those bridges, which means more time will be consumed to compute W-C mapping for those incremental points during preprocessing phase. This strategy is written as DBB-I [18]. The other is to predict the validity of incremental points by using Parzen-Window estimation in updating phase and delaying collision detections to query phase. In this way, it can decrease the preprocessing time. This strategy is written as DBB-II [19].

Although DBB has achieved great performance in changing environments with difficult regions, it still has some deficiency. It is difficult for DBB to identify narrow passages completely as there will be some situations when no bridges could be found in difficult regions.

2.3. Parzen Window density estimation

Various machine learning algorithms [21] have been applied to estimate the probability density of valid sampling points in a local area. In this paper, active learning algorithms are utilized since the learner can classify different kinds of regions from which it learns. Parzen-Window is a well-known density estimation method which has been extensively used in pattern recognition, classification, prediction and so on. Parzen-window estimates the Probability Density Function (PDF) which is derived from a large amount of samples.

Assume that x is a random sample, a window function is used to determine how many observation samples $\{x_1, \dots, x_n\}$ fall within the window around x . Then, the Probability Density Function (PDF) value $P(x)$ is defined as the sum of the contributions from the observations to this window. The Parzen-window estimates can be defined as follows:

$$P(x) = \frac{1}{n} \sum_{i=0}^n \frac{1}{h^d} K\left(\frac{x - x_i}{h}\right), \quad (3)$$

where $K(\cdot)$ is the window function of d -dimensions, and $h^d (> 0)$ is the window width. If the window is small enough, it will actually enclose zero points within the window. Therefore, that is the drawback of Parzen-Window Estimation. Therefore, in this paper, the K-nearest neighbors strategy is chosen to compute biased entropy, which sets the size of the window depending on the biggest distance between two points.

3. Adaptive Region Boosting method

3.1. Overview of our method

In changing environment, a time period is divided into a series of slices. In each time slice, the environments are viewed as static environments. For convenience, a time slice is called a frame. Obviously, the more slices in a time period, the more accurate we simulate the real environment.

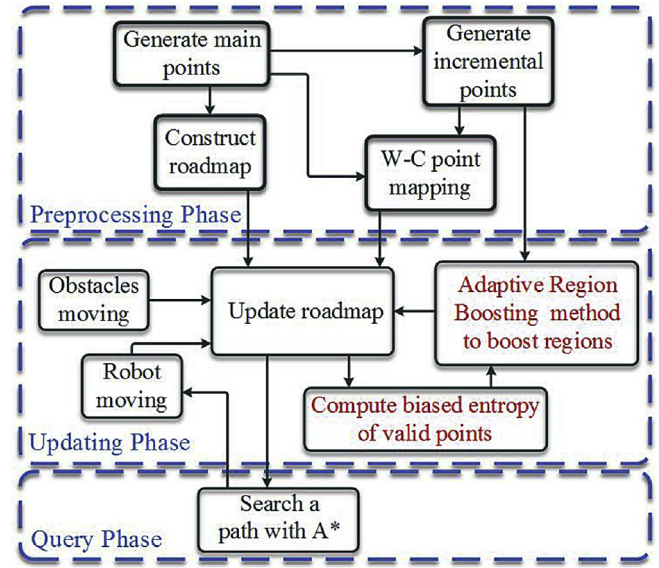


Fig. 1. Framework of Adaptive Region Boosting Method.

It is one of the most crucial problems to estimate how difficult a region is in real time. In this paper, the difficult degree of a region can be roughly estimated with information of a few sampling points. Biased entropy of a valid point represents the difficult degree of the local region around this point. The biased entropy difference between two adjacent frames can evaluate the changing degree of a region through time. With this difference, an Adaptive Region Boosting (ARB) method is proposed to boost difficult regions in order to find a free path effectively and efficiently.

During preprocessing phase, the roadmap is constructed by Hierarchical Sampling Strategy (HSS). After this, W-C nodes mapping of the roadmap is calculated for obtaining validity of these points online. During updating phase, the roadmap is updated through W-C mapping and then the biased entropy value of the valid points is computed to evaluate difficult degree of regions. Our approach computes biased entropy to evaluate how many incremental points need to be activated to boost the difficult regions. The A* algorithm is used to search a safe path. The safety of the path is also weighted by the entropy value. The smaller the entropy value of the sampling point is, the safer the path is. The flow chart of the proposed method is also shown in the Fig. 1.

3.2. Preprocessing phase

Hierarchical Sampling Strategy (HSS) is used in this paper to construct the roadmap in preprocessing phase. As shown in Fig. 2, the roadmap consists of two level points in C-space, which are main points and incremental points. Initially, the main points are generated in C-space, then, the roadmap is constructed by a straight-line local planner using Manhattan distance [22]. Finally, the incremental points are generated around main points.

Algorithm 1 Hierarchical Sampling Strategy.**Input:** $P = \{p_1, p_2, \dots, p_n\}$.**Output:** W-C mapping files of P and B .

```

1: for each point  $p_i \in P$  do
2:   Connect K-nearest neighbors of  $P_i$ ;
3:   Generate incremental points  $B_i$  around  $p_i$ ;
4:   for each node  $b \in B_i$  do
5:     Connect  $b$  with  $p_i$ ;
6:     Connect K-nearest neighbors of  $b$ ;
7:     Set  $b.activated = false$ ;
8:   end for
9: end for
10:  $B = \{B_1, \dots, B_n\}$  is the set of all the incremental points;
11: Compute  $W - C$  Mapping for  $P$  and  $B$ .
```

As shown in Algorithm 1, each node is connected to their centers which are the nodes of P initially. Then each node of B is connected to its k-nearest neighbors. This roadmap is denoted by $G = \{G_p(P, B), G_e\}$, in which $P = \{p_1, \dots, p_n\}$ is the set of main points, $B = \{B_1, \dots, B_n\}$ is the set of incremental points and G_e is the set of edges in the roadmap. Here, B_i is the set of the incremental points around p_i . At the beginning, all points of B will be marked as inactive, which means they will be ignored in the course of searching a path. The W-C node mapping for points of P and B should be computed in the preprocessing phase, in order to update their validity fast for each query. In updating phase, when incremental points around the main points need to be activated, these incremental points are used to search a path.

3.3. Biased entropy

In this paper, new statistic information of regions are utilized to classify regions around sampling points. The biased entropy, which is a measure of the disorder of the samples in a region, is used to evaluate difficult degree of the region. The higher the biased entropy is, the more difficult the region is.

In order to calculate the biased entropy value, Parzen Window estimation, which is an activate machine learning method, is adopted to estimate probability density. Since entropy represents the information of a probability distribution,

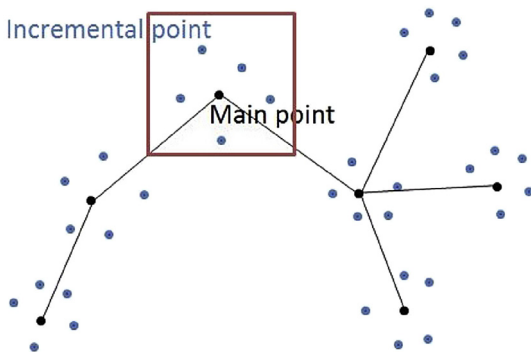


Fig. 2. Hierarchical Sampling Strategy (HSS) is used to construct the roadmap. Black points are main points. Blue points are incremental points.

Parzen Window is needed to provide the probability information of sampling points in a region.

As shown in Fig. 3, $N_q = \{q_1, \dots, q_k\}$ is the set of K-nearest neighbor points of point q . The region R_q is the smallest area which contains $N_q \cup q$. The P_{valid} and $P_{invalid}$ denote the set of valid and invalid sampling points, respectively. The observation probability is defined as the probability of valid sampling points. The probability estimation is defined as:

$$P(q) = \frac{|P_{valid} \cap N_q|}{|P_{valid} \cap N_q| + |P_{invalid} \cap N_q|}. \quad (4)$$

Assume initial point q to be a valid sampling point, then $P_{valid} \cap N_q \neq \emptyset$ and $P(q) \in (0, 1]$. However, $|P_{valid} \cap N_q| + |P_{invalid} \cap N_q| = K + 1$ is constant and $P(q)$ is a linear function, which needs a large amount of samples to ensure high accuracy. Therefore, simple K-nearest neighbors method is not applied to local regions.

Thus, information entropy is introduced to be a measure of the disorder of the samples in a specific region. The entropy value of the initial point is calculated by $P(q)$. In C-space, $P(q) \in (0, 1]$, because all sampling points in R are valid or invalid. Then, the entropy is defined as:

$$H(P(q)) = -P(q) \log P(q), \quad (5)$$

when $P = 0.5$, $H(P(q))$ gains the maximum. However, it is not a monotone function. It is difficult to measure difficult degree of regions, such as, approximately free regions, obstacle boundaries and narrow passages. A new estimating function called biased entropy is defined as:

$$e(q) = \begin{cases} 1 + P(q) \log P(q), & P(q) < 1/2; \\ -P(q) \log P(q), & \text{else.} \end{cases} \quad (6)$$

Function $e(q)$ is a decreasing function, which denotes information entropy of a valid sampling point. The higher $e(q)$ is, the more difficult the region R_q will be considered. As shown in Fig. 4, there are fewer valid points within narrow passages than in regions of obstacle boundaries, the biased entropy of the point within narrow passages is higher than

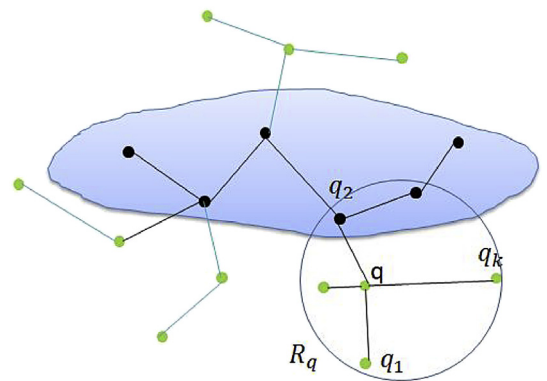


Fig. 3. Sampling with K-nearest strategy. Point q is the center point and q_1, \dots, q_k are K-nearest neighbor points.

regions of obstacle boundaries. In conclusion, biased entropy is an effective measure to evaluate difficult degree of regions.

3.4. Region boosting strategy

During updating phase, the roadmap needs to be updated all the time. In every frame, the roadmap is updated by W-C mapping according to which cell in W-space is occupied by obstacles or robots. In our method, the roadmap is also updated with an adaptive boosting region method, which is shown in Algorithm 2.

Algorithm 2 Update roadmap online.

Input: Old roadmap $G_{old} = \{G_p(P, B), G_e\}$.

Output: New roadmap $G_{new} = \{G_p(P, B), G_e\}$.

```

1: Update roadmap  $G_{old} = \{G_p(P, B), G_e\}$  with W-C mapping;
2: for  $i = 1 \rightarrow n$  do
3:   if  $p_i.validity = true$  then
4:     Compute entropy value  $e_{new}(p_i)$ ;
5:     Compute  $d = e_{new}(p_i) - e_{old}(p_i)$ ;
6:     Get the number of activated points  $c_{aold}$ ;
7:     Compute new number of activating points
       $c_{anew} = c_{aold} + d * |B_i|$ ;
8:     if  $c_{anew} > c_{all}$  then
9:        $c_{anew} = c_{all}$ ;
10:    end if
11:    if  $c_{anew} < 0$  then
12:       $c_{anew} = 0$ ;
13:    end if
14:    if  $d > 0$  and then
15:      Increase  $0 \rightarrow |d * c_{all}|$  activated points to
      make  $c_{anew}$  incremental points are activated in total
      around  $p_i$ ;
16:    else  $d < 0$ 
17:      Decrease  $0 \rightarrow |d * c_{all}|$  activated points to
      make  $c_{anew}$  incremental points are activated in total
      around  $p_i$ ;
18:    end if
19:     $e(p_i)_{old} = e(p_i)_{new}$ ;
20:  end if
21: end for

```

When obstacles move in W-Space, obstacle regions in C-space will change accordingly. Since the motion of obstacles is unknown, the changing degree of regions in C-space is quite different. For example, if some regions become narrow passages from free regions between two frames, these regions have a more intensity change than the other regions in contrast. Thus, the difference of $e(p)$ can represent the changing degree of region R_p , which is defined as:

$$d = e(p_i)_{new} - e(p_i)_{old}, \quad (7)$$

where $d \in [-1, 1]$. Here, $e_{new}(p)$ and $e_{old}(p)$ are $e(p)$ in previous frame and $e(p)$ in current frame respectively.

We assume that c_{aold} and c_{anew} represent how many incremental points have been activated in previous frame and in

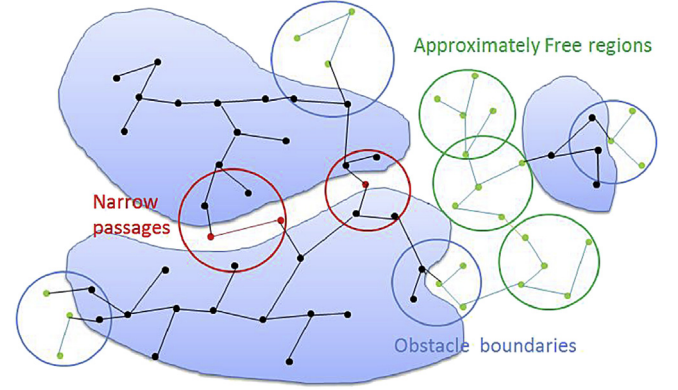


Fig. 4. This figure illustrates how the biased entropy works. The higher $e(q)$ is, the more difficult the region R_q will be considered. Narrow passages have high value of biased entropy. Obstacle boundaries have middle value of biased entropy. Approximately free regions have low value of biased entropy.

current frame, respectively. c_{aold} and c_{anew} are initialized with 0, which represent all the incremental points B_i that have not been activated. If $d > 0$, the region R_i becomes more and more difficult. Thereby, the number of activated incremental points needs to be increased. On the contrary, if $d < 0$, the difficult degree of region R_i becomes much smaller. Hereby, the number of activated incremental points needs to be decreased. The absolute value $|d|$ represents changing degree of R_i . The higher $|d|$ is, the more incremental points need to be activated or inactivated for updating the roadmap. The number of incremental points in the new roadmap is represented by:

$$c_{anew} = c_{aold} + d * |B_i|, \quad (8)$$

where $|B_i|$ represents the number of total incremental points around p_i . However, when $c_{anew} > |B_i|$ or $c_{anew} < 0$, we reset $c_{anew} = |B_i|$ or $c_{anew} = 0$ respectively. In other words, if $c_{anew} > |B_i|$, all incremental points in B_i are activated. If $c_{anew} < 0$, all incremental points in B_i are inactivated. As shown in Fig. 5(a), the difficult degree of region R_p becomes smaller, some activated points are inactivated to constrain scale of the roadmap. As shown in Fig. 5(b), the region R_p becomes more and more difficult, some inactivated points are activated to boost this region.

With the adaptive region boosting method, more points are sampled in more difficult regions. Moreover, changing degree of regions is also taken into account to represent how many incremental points are needed to be activated or inactivated, in order to balance the density of sampling points in local areas with different motion of obstacles.

During query phase, the roadmap is updated by adaptive region boosting method with moving obstacles and moving robots in every frame. After that, A* algorithm is adopted to search a collision-free path, and robot moves to the next configuration.

4. Experiments and analysis

In order to evaluate the proposed method, extensive experiments are implemented with two manipulators modeled by

parameters of practical 6-DoF Kawasaki manipulators (FS03N) in 3D W-space. The two manipulators are mounted on two fixed bases which amount to 12-DoFs. 12 dimensional C-space is constructed, because the 12-DoFs of manipulators are considered simultaneously. The reachable workspace of two manipulators is decomposed into 406134 grids, and each grid is a cube with the size of $4 \times 4 \times 4$. Collision detection in our experiments is implemented by a free 3D Collision Detection Library, ColDet 1.1. All the experiments are carried out on an Intel Dual-core CPU 3.00 GHz with 4 GB memory.

4.1. Scenarios of experiments

Three groups of experiments are arranged to evaluate the proposed method, as shown in Fig. 6. The planning tasks are to finish a hand shaking movements between two manipulators while passing through variant obstacles. The start and goal positions of each scenarios are given in Figs. 7–9, respectively. As shown in Fig. 6(a), two horizontal small boards are moving up and down in a big board with two rectangular holes, creating four dynamic holes for robots to pass through. Scenario II consists of a square board with a hole moving up and down in the air, the start and goal positions can be found in Figs. 6(b) and 8, respectively.

Six bars flying in the air construct a more complex environment which is shown in Fig. 6(c). All the bars are horizontal, while four of them are in the periphery of two manipulators and the other two are between two manipulators. The outside 4 bars move up and down and the other 2 bars move up and down in opposite direction. During the run time, two vertical bars first move toward each other. When they are close enough, they drift apart in the opposite direction. The start and goal positions can be found in Figs. 6(c) and 9, respectively. The start configuration is constant. In all scenarios, each bar has a move range, once the maximum distance is reached, the corresponding direction is reversed.

4.2. Analysis of parameters

Firstly, the value of a crucial parameter is discussed. In ARB, entropy value is calculated by K-nearest neighbors method. Thereby, parameter K has a significant influence on the performance of the planner. If K is very small, no enough sampling points to estimate accurately makes region classification unreliable. On the contrary, if K is very large, the region R_q becomes too large leading to an increasing error rate. In order to obtain optimal K , 100 experiments are carried out with different K in different scenarios.

As shown in Fig. 10, success rate of the planner is low with both large and small K . However, when $K = 15$ or $K = 20$, the success rate is higher than others. As to different scenarios, the best choice of parameter K is different. Nevertheless, $K = 15$ is the best choice to consider both computing cost and success rate. Therefore, $K = 15$ is selected in ARB.

The size of region R_q also depends on parameter K . The diameter of R_q is the furthest distance between two points which are K-nearest neighbors of q . Furthermore, as the center of region R_q is not always point q , all the regions are restricted in the boundaries of C-space.

4.3. Analysis of preprocessing phase

During preprocessing phase, some samples are generated to construct the roadmap at first, called the main points. Then, the incremental points are generated around main points and connected to the roadmap. However, these incremental points are inactivated initially, which means these points and edges will not be searched until they are activated. In this paper, there are eight incremental points around every main point. After that, W-C nodes mapping of all points are computed to obtain validity of these points in updating phase.

Table 1 summarizes the number of sampling points in different methods. Here, column N_p is the number of main points which are initial vertex of roadmap. The cardinality of

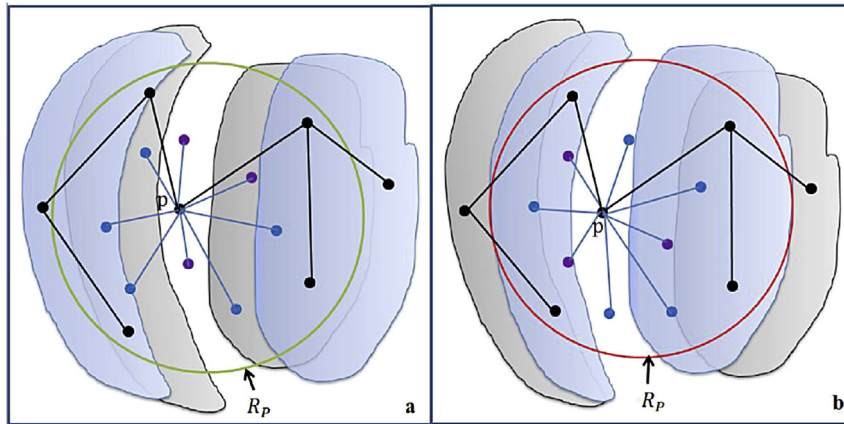


Fig. 5. Two different situations for narrow passage. The blue region is current obstacle's region and the gray region represents its previous position. The black points are main points. The purple points are incremental points which are activated in previous frame. The blue points is newly activated or inactivated incremental points. (a) The difficult degree of region R_p becomes smaller, blue points are inactivated to constrain scale of the roadmap. (b) The difficult degree of region R_p becomes higher, blue points are activated to boost the region.

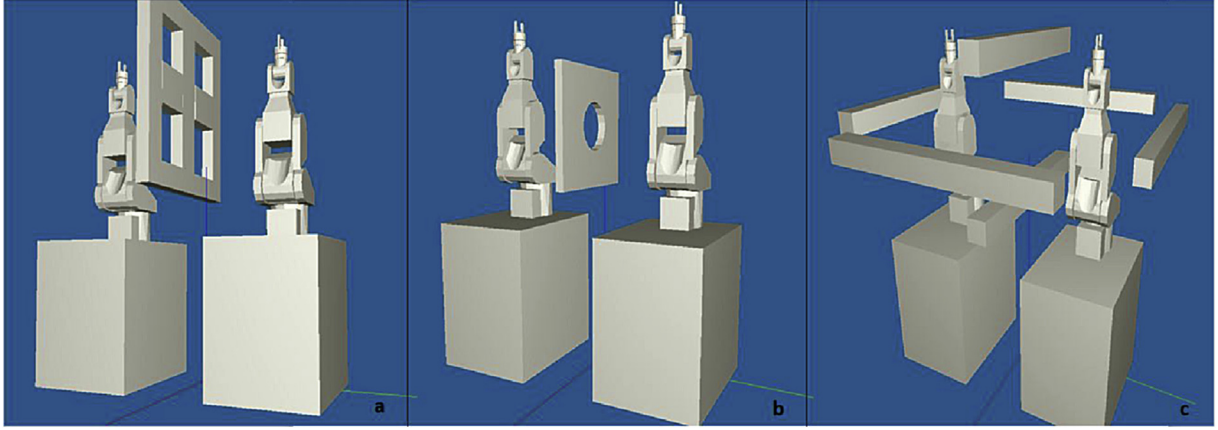


Fig. 6. Experiment scenarios I(a), II(b), III(c) are shown in this figure, each consist of 12 DoFs.

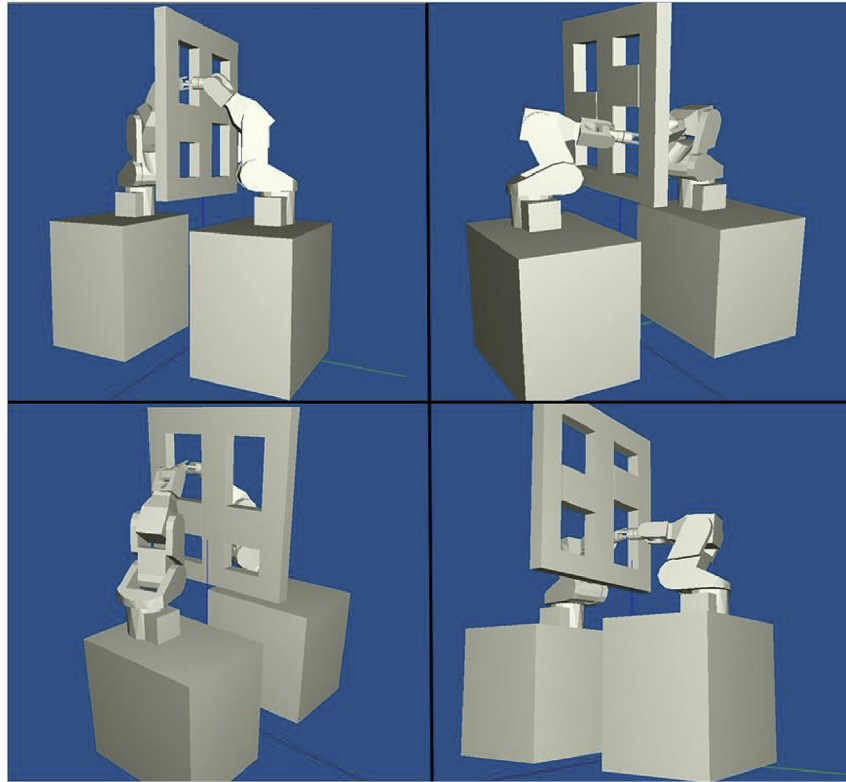


Fig. 7. Four goal configurations of scenario I are shown. The planner tends to choose the safer goal area to reach.

P is crucial in realization. If it is too large, updating phase will be time consuming. If it is too small, roadmap does not contain enough information for C-space construction. Column N_M is the number of the middle points which are used in DBB-I and DBB-II to identify narrow passages. N_B is the number of the incremental points in ARB, DBB-I and DBB-II. Column N_S is the total number of sampling points in C-space. Column T_c is the time of constructing roadmap without W-C mapping.

As shown in Table 1, ARB, DBB-I and DBB-II have the same N_P and DRM creates the same number of total sampling points with ARB. In preprocessing phase, much fewer total sampling points are generated in ARB than DBB-I with the same number of main points, because the middle points are

not computed and much fewer incremental points are generated in ARB. The N_S of DBB-II is smaller than DBB-I, because the incremental points are generated around main points instead of midpoints in DBB-I. T_c of ARB and DBB-I are similar to DRM, because ARB and DRM have similar N_S . DBB-I has more sampling points and more time of roadmap construction, because more sampling points are generated and it needs much more time to connect these points.

4.4. Analysis of updating phase

During updating phase, the obstacle region in C-space changes with the obstacles in W-space. The faster the

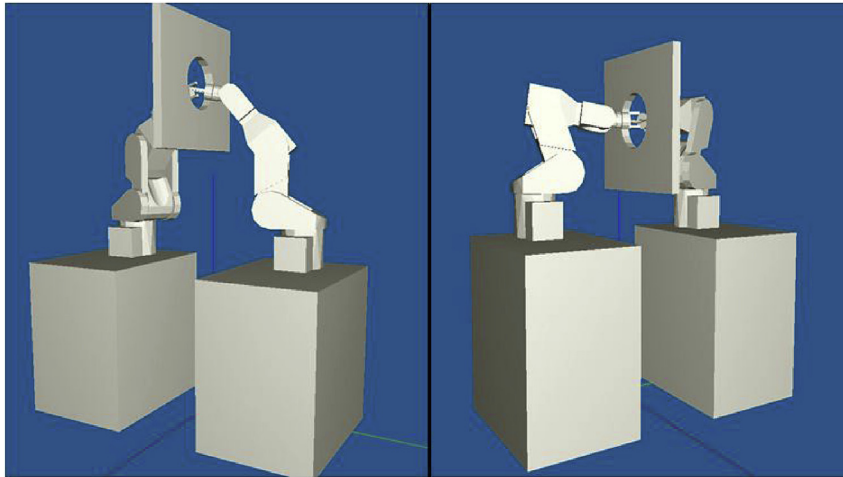


Fig. 8. Two goal configurations of Scenario II are shown.

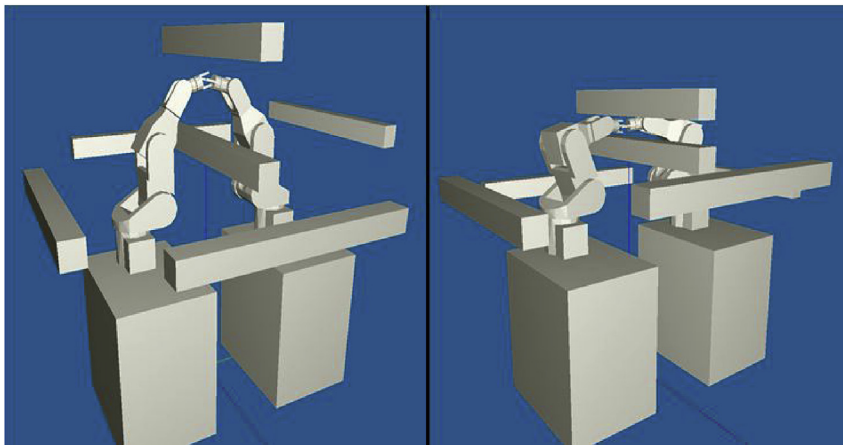
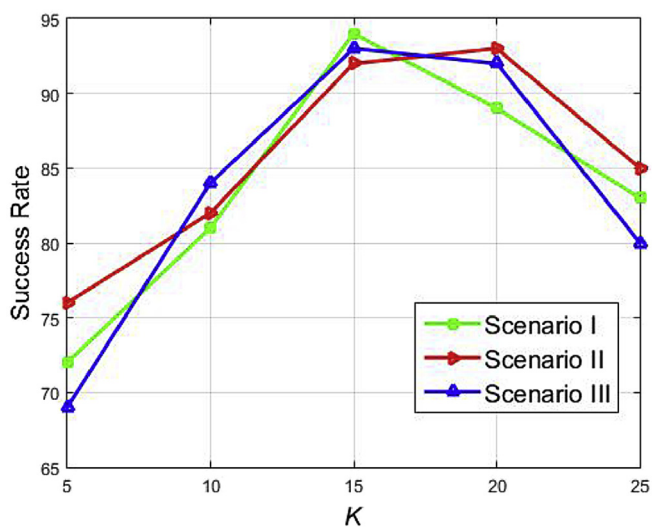


Fig. 9. Two goal configurations of Scenario III are shown.

Fig. 10. Selection of Parameter K . The optimum value of K differs in different scenarios. However, $K = 15$ performs very well in all the scenarios.

obstacles move, the more intensely the C-space changes. In order to evaluate performance of our method, different speeds of obstacles and different methods are taken into account in experiments.

Tables 2–4 summarize the results of ARB compared with other methods in different motion speeds of obstacles. Each group of experiments is tested for 200 times from a random start configuration to the goal configurations. Here, column SR_{avg} represents average success rate of planning. Column N_{avg} and column N_l represent the number of average re-searching times and largest re-searching times, respectively. Column T_p is the time of preprocessing phase and column T_{avg} is the average planning time.

Table 1

The number of sampling points in different methods.

Method	N_P	N_M	N_B	N_S	$T_c(s)$
ARB	500	—	4000	4500	3.43
DBB-II	500	1580	2500	4580	3.87
DBB-I	500	1580	7900	9680	7.36
DRM	4500	—	—	4500	3.51

Table 2
Results of different methods with different moving speed of obstacles under Scenario I.

Method	Speed	SRavg	Navg	Nl	Tp (min)	Tavg(s)
ARB	2 cm/s	93.89%	12.81	23	14.02	0.572
DBB-II		88.93%	22.17	30	8.39	0.487
DBB-I		88.53%	21.32	32	25.72	0.516
DRM		71.92%	33.65	49	14.52	0.526
ARB	3 cm/s	93.37%	12.27	25	14.02	0.576
DBB-II		85.75%	23.18	38	8.39	0.493
DBB-I		84.79%	24.32	40	25.72	0.554
DRM		38.38%	37.37	46	14.52	0.568
ARB	4 cm/s	92.55%	12.11	20	14.02	0.581
DBB-II		82.84%	28.59	44	8.39	0.537
DBB-I		82.63%	30.53	46	25.72	0.589
DRM		61.13%	42.65	50	14.52	0.591

Table 3
Results of different methods with different moving speed of obstacles under Scenario II.

Method	Speed	SRavg	Navg	Nl	Tp (min)	Tavg(s)
ARB	2 cm/s	92.15%	16.81	25	15.34	0.559
DBB-II		87.33%	33.19	44	9.68	0.49
DBB-I		85.98%	35.32	45	25.72	0.516
DRM		74.54%	48.65	55	15.14	0.534
ARB	3 cm/s	91.95%	17.53	27	15.34	0.563
DBB-II		85.56%	39.54	48	9.68	0.516
DBB-I		85.13%	43.93	53	25.72	0.546
DRM		71.84%	52.68	61	15.14	0.587
ARB	4 cm/s	91.05%	18.85	27	15.34	0.57
DBB-II		82.34%	43.59	53	9.68	0.537
DBB-I		82.14%	47.51	58	25.72	0.563
DRM		68.29%	55.68	62	15.14	0.593

Table 4
Results of different methods with different moving speed of obstacles under Scenario III.

Method	Speed	SRavg	Navg	Nl	Tp (min)	Tavg(s)
ARB	2 cm/s	94.38%	14.27	19	14.56	0.542
DBB-II		89.36%	24.42	30	8.91	0.495
DBB-I		89.01%	27.3	32	25.72	0.546
DRM		73.96%	37.23	46	14.61	0.524
ARB	3 cm/s	93.67%	14.65	20	14.56	0.545
DBB-II		87.65%	35.85	40	8.91	0.502
DBB-I		87.16%	35.39	41	25.72	0.59
DRM		72.43%	45.59	51	14.61	0.53
ARB	4 cm/s	92.94%	14.84	20	14.56	0.544
DBB-II		85.98%	40.37	47	8.91	0.513
DBB-I		84.31%	42.54	49	25.72	0.587
DRM		70.01%	51.98	57	14.61	0.538

As shown in Table 2, with the same speed, the SR_{avg} of ARB is higher than other methods in all scenarios. The SR_{avg} of ARB achieves much improvement compared with DRM, because there are few sampling points within difficult regions in DRM. DBB-I and DBB-II are proposed to solve difficult region problems in changing environments. However, only a part of difficult regions can be identified in DBB-I and DBB-II, because only a few bridges could be built in updating phase and many difficult regions are not boosted in updating phase.

ARB computes the entropy value of valid points and boosts the regions around them. In this way, more regions can be boosted than DBB-I and DBB-II. Therefore, the SR_{avg} of ARB gains huge improvements compared with the other methods.

With the same speed, the N_{avg} and N_l of ARB are much smaller than DBB-I's and DBB-II's in all scenarios. It is due to the fact that DBB-I and DBB-II only identifies a part of difficult regions, while more local regions are boosted for increasing density of sampling points in difficult regions. Moreover, the N_{avg} and N_l of DBB-I and DBB-II in scenario II are much larger than those in scenario I and III. Because the goal configuration is within narrow passages in scenario II, scenario II is more difficult than the others. However, the N_{avg} and N_l of ARB in scenario II do not increase much more than the others. It presents that ARB can be used in various environments.

The T_p of DBB-I is larger than the other methods in all scenarios with the same speeds, because it has larger N_s and W-C mappings of all these points are calculated. The DBB-II achieves the least T_p , because the validity of the incremental points in DBB-II is predicted in updating phase and W-C mappings are not computed in preprocessing phase. The T_p of ARB is same as DRM's, because the same number of sampling points needs to compute W-C mappings. The T_{avg} of all methods are similar. ARB and DBB-I cost much time to update a large amount of sampling points. DBB-II costs some time to compute predictive model. In conclusion, ARB outperforms the other methods with the same speed of obstacles.

With different speeds of obstacles, the SR_{avg} of ARB is similar in each scenario. However, the SR_{avg} of the other methods is decreasing with higher speed. It is due to the fact that the changing degree of environments is taken into account in ARB. The number of activated points depends on the changing degree of the region. The ARB has excellent universality for various environments. In a word, the ARB gains better outstanding performance than the other related methods.

5. Conclusions

In this paper, Adaptive Region Boosting (ARB) method is proposed to boost regions with difficult degree of regions in changing environments. A new criterion called biased entropy is proposed to evaluate the difficult degree of a local region in this paper. Moreover, the absolute value of difference between two biased entropy in two continuous frames can evaluate the changing degree of the region. Our approach works in two phases. In preprocessing phase, main points and incremental points are generated to construct the roadmap. In updating phase, the biased entropy difference is used to determine the number of activated incremental points. Compared to other related works, ARB has higher success rate of planning and less re-planning time. ARB is more suitable for various changing environments than DBB-I and DBB-II, as most local regions are boosted adaptively by our approach. In conclusion, ARB outperforms the other related works, as shown in the experiments.

However, there are a few significative works on excavating information of regions in DRM method. For example, tracking the movement of obstacles and forecasting the changing of difficult regions will be our future work. Some predictive model to capture motion of regions is a much possible method.

Acknowledgements

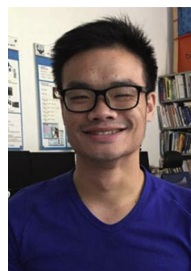
This work is supported by National Natural Science Foundation of China (NSFC, No.60875050, 60675025, 61340046), National High Technology Research and Development Program of China (863 Program, No.2006AA04Z247), Science and Technology Innovation Commission of Shenzhen Municipality (No.201005280682A, No.JCYJ20120614152234873), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20130001110011).

References

- [1] H.M. Choset, Principles of Robot Motion: Theory, Algorithms, and Implementation, MIT press, 2005.
- [2] L. Kavraki, P. Svestka, J. Latombe, M. Overmars, IEEE Trans. Robotics Automation 12 (4) (1996) 566–580.
- [3] R. Geraerts, M.H. Overmars, A Comparative Study of Probabilistic Roadmap Planners, vol. 7, Springer Tracts Adv. Robotics, 2004, pp. 43–57.
- [4] R. Bohlin, L.E. Kavraki, Path planning using lazy prm, in: IEEE International Conference on Robotics and Automation (ICRA), vol. 1, 2000, pp. 521–528.
- [5] J. Denny, N.M. Amato, The toggle local planner for sampling-based motion planning, in: International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 1779–1786.
- [6] D. Hsu, L.E. Kavraki, J.C. Latombe, R. Motwani, S. Sorkin, On finding narrow passages with probabilistic roadmap planners, in: Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics on Robotics : The Algorithmic Perspective: the Algorithmic Perspective, 2002, pp. 141–153.
- [7] N.M. Amato, O.B. Bayazit, L.K. Dale, Obprm: An Obstacle-based Prm for 3d Workspaces, 1998.
- [8] V. Boor, M.H. Overmars, et al., The gaussian sampling strategy for probabilistic roadmap planners, in: International Conference on Robotics and Automation, vol. 2, IEEE, 1999, pp. 1018–1023.
- [9] D. Hsu, T. Jiang, J. Reif, Z. Sun, The bridge test for sampling narrow passages with probabilistic roadmap planners, in: International Conference on Robotics and Automation (ICRA), vol. 3, IEEE, 2003, pp. 4420–4426.
- [10] B. Burns, O. Brock, Sampling-based motion planning using predictive models, in: International Conference on Robotics and Automation (ICRA), IEEE, 2005, pp. 3120–3125.
- [11] B. Burns, O. Brock, Toward optimal configuration space sampling, in: Proc. Robotics: Science and Systems, Cambridge, MA, June 2005.
- [12] L. Zhang, Y.J. Kim, D. Manocha, A hybrid approach for complete motion planning, in: International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, 2007, pp. 7–14.
- [13] S. Rodriguez, S. Thomas, R. Pearce, N.M. Amato, Resampl: A Region-sensitive Adaptive Motion Planner, Springer Algorithmic Foundation of Robotics VII, 2008, pp. 285–300.
- [14] P. Leven, S. Hutchinson, Toward Real-time Path Planning in Changing Environments, 2000.
- [15] H. Liu, X. Deng, H. Zha, D. Ding, A path planner in changing environments by using wc nodes mapping coupled with lazy edges evaluation, in: International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, 2006, pp. 4078–4083.
- [16] M. Kallman, M. Mataric, Motion planning using dynamic roadmaps, in: International Conference Robotics Automation (ICRA), vol. 5, IEEE, 2004, pp. 4399–4404.
- [17] H. Liu, K. Rao, F. Xiao, Obstacle guided rrt path planner with region classification for changing environments, in: International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2013, pp. 164–171.
- [18] D. Ding, H. Liu, X. Deng, H. Zha, A dynamic bridge builder to identify difficult regions for path planning in changing environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2007, pp. 2925–2931.
- [19] H. Liu, D. Ding, W. Wan, H. Zha, Predictive model for path planning by using k-near dynamic bridge builder and inner parzen window, in: International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, 2008, pp. 2133–2138.
- [20] H. Liu, T. Zhang, C. Wang, A “capacitor” bridge builder based safe path planner for difficult regions identification in changing environments, in: International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, 2012, pp. 3179–3186.
- [21] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, 2012.
- [22] N.M. Amato, O.B. Bayazit, L.K. Dale, C. Jones, D. Vallejo, Choosing good distance metrics and local planners for probabilistic roadmap methods, in: International Conference on Robotics and Automation (ICRA), vol. 1, IEEE, 1998, pp. 630–637.



Risheng Kang received the B.E. degree in computer science and technology from Inner Mongolia Agricultural University, inner Mongolia, China, in 2012. He is currently pursuing M.S. degree in computer science of Peking University, China. His research interests in robot motion planning. Recently, he also serves as reviewer for the IEEE International Conference on Robotics and Automation (ICRA).



Hao Tang received the B.E. degree in electronics and information engineering in 2013 and is working toward the Master degree at the School of Electronics and Computer Engineering, Peking University, China. His current research interests are image classification, hand gesture recognition, gender recognition, image retrieval, action recognition and deep learning. He has published several articles in ACM Multimedia Conference (MM), IEEE International Conference on Image Processing (ICIP) and International Joint Conference on Artificial Intelligence (IJCAI).



Wen-Yong Zhao got his Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2012. He received his B.Eng. and M.Eng. degrees from Tianjin University, China, in 2004 and 2007 respectively. From 2013 to 2016, he worked as a postdoctoral in ShenZhen graduate school of PeKing University. In 2016, he became a lecturer in ShenZhen institute of information technology. His research interests include path planning, image processing, computer vision and computer.