2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

A Predictive Model for Narrow Passage Path Planner by Using Support Vector Machine in Changing Environments

Hong Liu, Fang Xiao* and Can Wang

Abstract— **Narrow passages in changing environments create huge difficulties, since locations and shapes of narrow passages in Configuration Space(C-space) change frequently. It is very important for a planner to identify narrow passages in real time and boost valid points within them effectively. A novel narrow passage predictive model for designing a path planner in changing environments is proposed in this paper. Firstly, an Expanded Dynamic Bridge Builder is presented to identify narrow passages rapidly with validity-toggle sampling points in C-space. Secondly, the predictive model is adopted to sample possibly free points within these narrow passages without invoking any collision detection in order to avoid intense computational complexity. The predictive model is obtained by the famous classification method of Support Vector Machine (SVM). A new feature, which includes a group of points' distance and validity information, is proposed in SVM training process to capture approximate structure of local narrow passages. Therefore, the predictive model can excavate the hidden similar structure of local narrow passages. Experiments carried out with two 6-DOFs manipulators show that our approach gain higher success rate of planning and time efficiency than other related methods.**

I. Introduction

Path planning has many applications in games, robotics, etc. During the past several decades, some widely used randomized algorithms have been proposed, such as Probabilistic Roadmap (PRM) [1] and Rapidly exploring Random Tree (RRT) [2]. Lazy evaluation is adopted by several PRM variants [3], [4],which can use much less preprocessing time and computing resources than PRM. Nevertheless, uniform sampling methods sample few points in difficult regions, which may lead to low planning successful rate. In static environments, many non-uniform sampling methods have been proposed to solve difficult region problem [5], such as Obstacle-Based PRM (OBPRM) [6], Gaussian Sampling [7], Bridge Test [8], Toggle Probabilistic Roadmap (Toggle PRM) [9], etc. Besides, Machine learning is another solution for difficult regions with capturing the structure of Configuration Space (C-space) [10], [11], [12]. However, almost all of them are generally ineffective with moving obstacles.

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).

Hong Liu is with Faculty of Key Laboratory of Machine perception (Ministry of Education), Peking University, Shenzhen Graduate School, Beijing, 100871 CHINA. E-mail: *hongliu@pku.edu.cn*

Fang Xiao and Can Wang are with the Engineering Laboratory on Intelligent Perception for Internet of Things (ELIP), Peking University, Shenzhen Graduate School, 518000, CHINA. E-mail: *1201213677@sz.pku.edu.cn* and *canwang@pku.edu.cn*.

Dynamic Roadmap (DRM) is an extension of PRM to make planner adapt to changing environments [13],[14]. Nonetheless, DRM is still confronted with difficulties from difficult regions. There are a few researches focused on narrow passages in changing environments. Dynamic Bridge Builder (DBB) [15] and Capacitor Bridge Builder (CBB) [16] have focused on identifying narrow passages rapidly in changing environments. However, how to identify narrow passages in C-space as complete as possible and how to effectively increase the density of sampling points within them are still worth studying.

In this paper, a novel Narrow Passage Predictive model (NPP) is proposed to identify and boost narrow passages in changing environments. Our approach is motivated by the intuition that an efficient planner in changing environments should have the capacity of building approximate structure of local narrow passages, and avoiding intense computational complexity for boosting narrow passages. Therefore, NPP is proposed to capture the structure of narrow passages. This model includes two parts. Firstly, Expanded Dynamic Bridge Builder (EDBB) is proposed to identify different kinds of narrow passages rapidly and roughly capture motion information of narrow passages. Secondly, the predictive model, which trained by SVM, predicts the validity of new-generated incremental points within narrow passages. The NPP delays detecting validity of points in query phase in order to avoid unnecessary invocations of the collision checker.

There are two main contributions in this paper. One is that EDBB builds more kinds of bridges to identify more narrow passages efficiently. The other is that the predictive model captures approximate structure of local narrow passages and avoids unnecessary collision detections effectively for boosting narrow passages.

The rest of this paper is organized as follows: Section II describes related works. Details of our method are demonstrated in Section III. Section IV shows the experiments and results in different scenarios. Finally, conclusions are given in Section V.

II. Related Works

A. SVM

The task of predicting validity of a sampling point can be regarded as a classification task, in which sampling points in C-space are classified as valid or invalid. The machine learning provides various methods for classification task [17], [18]. Support Vector Machine(SVM) is considered since the learner can select the data from which it learns [17]. SVM only needs much less samples to learn an applicable model,

978-1-4799-6923-4/15/$31.00 ©2015 IEEE

which is widely used in many fields. In order to find the optimal classified hyperplane, the basic training principle of SVM is that the expected classification error for test samples is minimized. According to the structural risk minimization inductive principle [15], a function classifies the training samples accurately. Although there is no guarantee that a solution will always exist in space, it is quite feasible to construct an useful solution. The liner SVM classifier is defined as:

$$f(x) = \omega^T x + b \tag{1}$$

where (ω, b) is classified hyper plane and x is feature as the input. Here, ω and b are computed by training samples.

B. DRM

Dynamic Roadmap (DRM) is used in changing environments [13], [14]. In preprocessing phase, DRM generates sampling points randomly in C-space. Then, W-space is decomposed into basic cells. Finally, DRM will give the mapping between basic cells in W-space and roadmap in C-space. The most important part of DRM is two kinds of mapping, point mapping $\Phi_p(w)$ and edge mapping $\Phi_e(w)$ (Eq.3):

$$\Phi_p(w) = \{q \in G_p | \Omega(q) \cap w \neq \phi\} \tag{2}$$

$$\Phi_e(w) = \{e \in G_e | \Omega(q) \cap w \neq \phi \text{ for some } q \in e\} \tag{3}$$

We suppose that the roadmap constructed in C-space is $G = \{G_p \cup G_e\}$. Here, G_p is the set of configuration points and G_e is the set of roadmap's edges. Points and edges of the the roadmap are labeled as invalid, if the basic cell w of W-space is occupied by obstacles. The set $\Omega(q)$ defines a subset of basic cells which are occupied by a robot whose configuration is q. However, mappings $\Phi_p(w)$ and $\Phi_e(w)$ are too complex to compute. Instead, we generally compute inverse mappings $\Phi_p^{-1}(w)$ and $\Phi_e^{-1}(w)$. In order to calculate $\Phi_p^{-1}(w)$, the robot in W-space is set to be the first set of configurations in C-space, and then a seed cell is put inside the robot and expanded in each direction until all cells $\Omega(q)$ occupied by the robot are found by collision detection [11]. It needs to update the roadmap timely when obstacles are moving.

C. Bridge Builder Planners

Identifying narrow passages sometime has decisive factor on whether a planner can find a free path. Dynamic Bridge Builder (DBB) is applied to changing environments [15] for narrow passages. In preprocessing phase, DBB generates main points and constructs new roadmap like DRM. Then, DBB computes midpoints of roadmap's edges and generates incremental points around midpoints. Afterwards, W-C mapping of all sampling points is calculated for ensuring validity of sampling points. If the midpoint of a edge is valid and two endpoints are invalid, this edge is a bridge which identifies a narrow passage. Capacitor Bridge Builder (CBB) is an extension DBB, which considers time information for path planning [16]. Moreover, a "capacitor" bridge is built between positive and negative validity-toggled endpoints in C-Space when obstacles' regions change in updating phase. Although DBB and CBB present good performance on identifying narrow passages, there are still so many limitations for these methods. For example, DBB cannot instantly identify changing narrow passage and CBB can only identify moving narrow passages.

III. Narrow Passage Predictive Model

A. Overview of Narrow Passage Predictive Model

In this paper, narrow passage predictive model is proposed to design an effective planner. Firstly, EDBB dynamically identifies narrow passages based on both validity-unchanged and validity-toggled sampling points, which takes both space and time information of C-space into account. Secondly, the predictive model is designed to boost narrow passages by learning SVM. The validity of incremental points are estimated instantly by the predictive model.As shown in Fig.1, there are three working phases for NPP algorithm. In preprocessing phase, The roadmap is initialized with a hierarchical sampling strategy. Afterwards, the predictive model is trained by offline SVM in preprocessing phase. Offline SVM training is discomposed into two parts. One is that training samples should be obtained firstly. The other is that the predictive model is learned by SVM with these training samples.

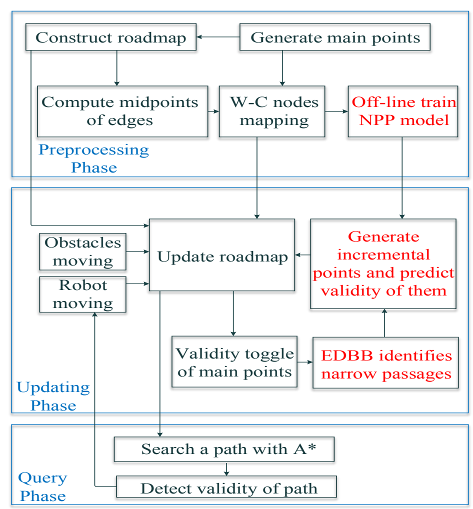

Fig. 1: Flowchart of narrow passage predictive model.

In updating phase, start and goal configurations are given. All the edges of the roadmap are traveled to find different kinds of bridges, which can identify narrow passages. Once a narrow passage is identified, new sampling points called the incremental points are generated and the validity of these points are predicted by the predictive model. Then, the roadmap is updated by adding valid incremental points. In query phase, A* algorithm is employed to plan a free path and necessary collision detections are delayed to verify it.

B. Preprocessing Phase

It is necessary to choose Hierarchy Sampling Strategy (HSS) [15], which divides all the points into three levels. As shown in Fig.3, main points $P = \{p_1, p_2, \ldots, p_n\}$ are generated as first-layer points to construct roadmap. The roadmap is constructed by a straight-line local planner using manhattan distance [19], [20]. The midpoints of edges $M =$

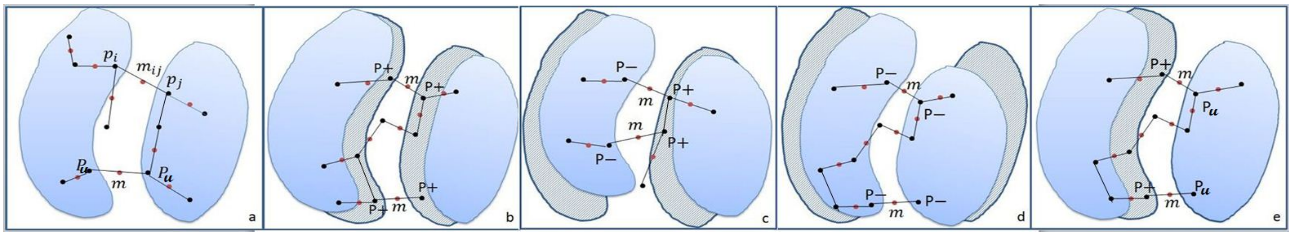

Fig. 2: Different narrow passages are identified by EDBB. The blue region is current obstacle's region and the grey region represents its previous position. (a) Static narrow passage. (b) Disappearing narrow passage. (c) Moving narrow passage. (d) Forming narrow passage. (e) Disappearing narrow passage. The figure is best viewed in color.

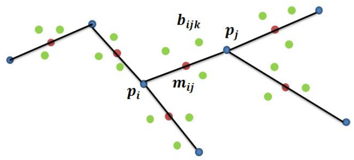

Fig. 3: Blue points are main points. Red points are midpoints. Green points are incremental points.The figure is best viewed in color.

$\{m_{12}, \ldots, m_{ij}, \ldots\}$ are the second-layer points. Here, m_{ij} is the midpoint of edge p_ip_j. Incremental points are third-layer points, which are generated to increase density of sampling points within narrow passages in updating phase. Let $B_{ij} = \{b_{ij1}, \ldots, b_{ijk}\}$ represents the set of incremental points around m_{ij}. The predictive model is proposed to predict whether an incremental point is valid or not. Algorithm 1 shows HSS in details.

Algorithm 1 Hierarchy Sampling Strategy

Input: $P = \{p_1, p_2, \ldots.., p_n\}$
Output: W-C mapping files of P and M

```
for each point p_i ∈ P do
    Connect K-nearest neighbors of p_i
    Edge set E = all the edges around p_i
    for each edge e_i ∈ E do
        Pick another endpoint p_j ∈ e_i
        Compute m = (p_i + p_j)/2
        Set m.validity = true
    end for
end for
M =all the middle points
Compute W − C Mapping for P and M
```

C. Expanded Dynamic Bridge Builder

The frequently-used notations are defined as follows: $P+$: positive points are defined as points whose validity turns from false to true. $P-$: negative points are defined as points whose validity turns from true to false. P_u: points are defined as invalid points whose validity are unchanged. The Algorithm 2 shows how to identify narrow passages with EDBB. The validity of some sampling points change when obstacles moving. Under this situation, EDBB can effectively re-identify narrow passages. For example, in Fig.2, obstacles' regions move to new positions in updating phase. By using W-C point mapping, validity and modification of P can be obtained. For each $p \in P$, if $p.validity$ turns from false to true, add p to $P+$. Otherwise, if $p.validity$ turns from true to false, it will be put into $P-$. Other points are belong to P_u.

Algorithm 2 Expanded Dynamic Bridge Builder

Input: $W - C$ point mapping for P and M
Output: All the bridges $E_B = \{p_ip_j, \ldots\}$

```
for each point p_i ∈ P do
    if p_i.validity turns from false to true then
        p_i ∈ P+
    else if p_i.validity turns from true to false then
        p_i ∈ P−
    else p_i.validity = false and p_i ∉ P−
        p_i ∈ P_u
    end if
end for
for p_i ∈ P+ or p_i ∈ P− or p_i ∈ P_u do
    Pick each edge p_ip_j connected with p_i
    Get middle point m_ij ∈ M
    if m_ij.validity is true then
        Get the other endpoint p_j
        if p_j ∈ P+ or p_j ∈ P− or p_i ∈ P_u then
            Mark edge p_ip_j as a bridge
        end if
    end if
end for
```

EDBB considers more types of bridges to identify more narrow passages. Assume that $p_i, p_j \in P$ are endpoints of an edge respectively and m_{ij} is midpoint of p_ip_j. Only m_{ij} is validity which can build a bridge. As shown in Fig.3(a), if $p_i, p_j \in P_u$, the edge p_ip_j is a normal bridge which can identify static narrow passages. As shown in Fig.3(b), if $p_i, p_j \in P+$, one obstacles' region is moving far away from another and the narrow passage is disappearing. As shown in Fig.3(c), if $p_i \in P+$, $p_j \in P-$, obstacles' regions are moving roughly towards one direction and the moving narrow passage could be identified more rapidly than DBB. As shown in Fig.3(d), if $p_i, p_j \in P-$, two obstacles' regions are more and more close and the narrow passage is gradually forming. With the similar situation, if $p_i \in P-$, $p_j \in P_u$, the forming narrow passages can also be identified. As shown in Fig.3(e), if $p_i \in P+$, $p_j \in P_u$, the edge can identify disappearing narrow passage as well. EDBB builds more kinds of bridges in order to locate narrow passages more completely.

D. Offline SVM Training Model

In C-space, invalid point is labeled with -1 and 1 if the point is valid. Predictive model traditionally concerns itself with the approximation of a function $f(x) \to y$. Here x is the input feature vector and y is prediction result. The output y is a continuous value in the range $[-1, 1]$. Feature for classification is comprised by the distance matrix of points $\{n_1, n_2, \dots, n_k\}$. The distance matrix D of these points are given by:

$$\begin{pmatrix} d_{11} & d_{12} & \dots & d_{1k} \\ d_{21} & d_{22} & \dots & d_{2k} \\ \dots & \dots & \dots & \dots \\ d_{k1} & d_{k2} & \dots & d_{kk} \end{pmatrix} \tag{4}$$

where d_{ij} is distance between n_i and n_j. Because $d_{ij} = d_{ji}$ and $d_{ii} = 0$, matrix D is a diagonal matrix. Assume that $v_i = 1$ if n_i is valid, and $v_i = -1$ if n_i is invalid. The $\frac{k^2-k}{2}$ dimension feature vector x is presented as follow:

$$x = (\ d_{12}v_2, \dots, d_{ij}v_j, \dots, d_{(k-1)k}v_k\ ) \tag{5}$$

The set $d_{12}, \dots, d_{1k}$ can be seen as weight of estimation and the set $v_2, \dots, v_k$ are validity information of these points. In addition, the feature represents a directed graph of $n_1, \dots, n_k$. Hence, vector x is an available feature to represent structure of local narrow passage.

How to obtain training samples is described in detail in this paragraph. In preprocessing phase, W-C mappings of the main points and the midpoints are calculated after the roadmap has been constructed. When obstacles are moving, EDBB is used to identify narrow passages. If p_ip_j is a bridge and m_{ij} is the midpoint of p_ip_j, algorithm randomly obtains point $n_1 \in P \cup M$ in a SampleArea around m_{ij}. Radius of SampleArea is $\frac{3}{4}|p_ip_j|$. Here, $|p_ip_j|$ is the length of edge p_ip_j. Then, (K-1)-nearest points $\{n_2, \dots, n_k\} \subset P \cup M$ around n_1 are obtained. Validity of all above points can be acquired with W-C mapping. The output of training sample is the validity of n_1. The input is feature x which is computed by these points. New training sample can be gained by repeat above steps, which has different n_1 with all the previous samples.

As shown in Fig.4, points $\{n_1, n_2, \dots, n_k\}$ are obtained to compute a training sample. If point n_1 is valid, output y is set as 1 and sample (x, y) is called positive sample, which is shown in Fig.4(a). On the contrary, output y is set as -1 and sample (x, y) is called negative sample, which is shown in Fig.4(b).

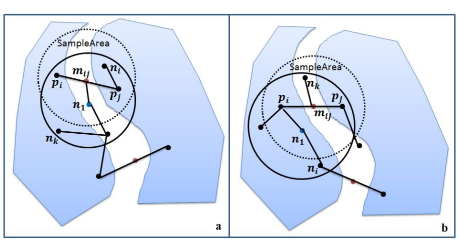

Fig. 4: The red points are midpoints, the black points are main points and the blue point is center point n_1. (a)positive sample. (b) negative sample. The figure is best viewed in color.

After training samples are obtained, the SVM is required to learn the predictive model. The SVM classifier is presented by:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i x_i^T x + b, \tag{6}$$

where x_i and y_i are the input and the output of training sample respectively, and x_i^T is transposition of vector x_i. The input$x$ is feature vector. Parameters α_i is computed by follow equation:

$$\begin{aligned} &\max_{\alpha} L = \max_{\alpha}(\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j x_i^T x_j), \\ &s.t. \qquad \alpha_i \geq 0, i = 1 \dots, n \qquad and \sum_{i=1}^{n} \alpha_i y_i = 0, \end{aligned} \tag{7}$$

where $\{\alpha_1, \dots, \alpha_n\}$ is computed to make L gains maximum. Then, parameter b is computed by equation:

$$b = -\frac{1}{2}(\max_{j, y_j = -1} \sum_{i=1}^{n} \alpha_i y_i {x_i}^T x_j + \min_{l, y_l = 1} \sum_{i=1}^{n} \alpha_i y_i {x_i}^T x_l), \tag{8}$$

In this process, the robot do not begin to plan until sufficient amount of training samples are acquired and the predictive model is learned. Then, how to predict validity of new sampling points in updating phase will be discussed in next section.

E. Online Prediction

The similarity of local narrow passages could be learned by SVM through training a large amount of samples. Then, in updating phase, the predictive model predicts the validity of incremental point online. The precess that the predictive model predicts validity of incremental points is shown in Algorithm 3.

Algorithm 3 Online Narrow Passages Predictive

Input: All the bridges $E_B = \{p_ip_j, \dots\}$
Output: New roadmap
for Each midpoint m_{ij} of bridge p_ip_j **do**
  while $l < n$ **do**
    Generate b_l in SampleArea around m_{ij}, $r_{SampleArea} = \frac{3}{4}|p_ip_j|$
    Get $k-1$ neighbor points $n_2, \dots, n_k$ around b_l
    Compute feature x
    Compute $f(x)$
    if $f(x) > 0$ **then**
      Add b_l to the roadmap G
    end if
  end while
end for

The input of algorithm 3 is the set of all the bridges that are built by EDBB. For each bridge p_ip_j, we suppose that m_{ij} is the midpoint of it and $|p_ip_j|$ is the length of it. Then, the algorithm generates incremental points $B_{ij} = \{b_1, b_2, ..., b_n\}$ around m_{ij} in SampleArea. The radius of SampleArea is $\frac{3}{4}|p_ip_j|$. Then, the $k-1$ neighbor points of b_l whose validity are known when the W-C mapping are obtained. Then, feature x is computed as the input of predictive model. The predictive model computes prediction $f(x)$. If $f(x) > 0$, the new sampling point is viewed as valid and is added into the roadmap through connecting b_l with $n_2, \dots, n_k$. Otherwise, this point is discarded. Finally, the output of algorithm is a new roadmap. In query phase, the A* algorithm searches a path, which may be free with high probability. Then, validity of the incremental points on the path are checked with collision detection. If there is an invalid point on the path, planner will replan a new path. The predictive model could be fast enough to capture the approximate structure of local narrow passages and to predict validity of incremental points.

IV. Experiments and Analysis

In order to evaluate performance of NPP, various experiments are implemented with two manipulators modeled by parameters of practical 6-DOFs Kawasaki manipulators (FS03N) in 3D W-space. The two manipulators are mounted on two fixed bases, whose corresponding configuration space 12-DOFs. Mutual collision avoidance and coordination between two manipulators are more important than planning two manipulators respectively. Therefore, 12 dimensional C-space is constructed, because two manipulators are considered simultaneously. The reachable W-space of two manipulators is decomposed into 406134 cells, and each cell 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 of 3.00 GHz with 4GB memory.

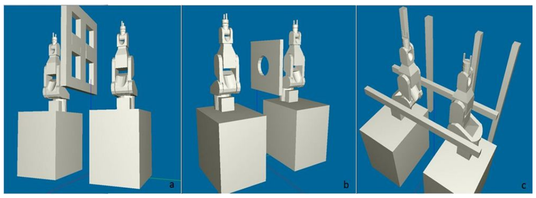

Fig. 5: (a): scenario I. (b): scenarioII. (c): scenarioIII.

As shown in Fig.5, three different kinds of scenarios are used for experiments. In scenario I, the two small boards inside two rectangular big holes in a static board are moving up and down randomly. Therefore, there are four rectangular small holes whose shapes and sizes are changing. In scenario II, there is a broad with a small round hole between the two manipulators, which moves up and down. The hole is small enough to construct narrow passages in C-space. In scenario III, there are six bars to construct a more complex environment. The two horizontal bars move up and down randomly. The four vertical bars move right and left randomly. In all scenarios, each board or bar moves at a speed of $4cm/s$ and each of them has a move range: when it reaches the maximum distance, its direction will be reversed. Since the bars and the boards move randomly, difficult areas are expected to appear in the C-space [21].

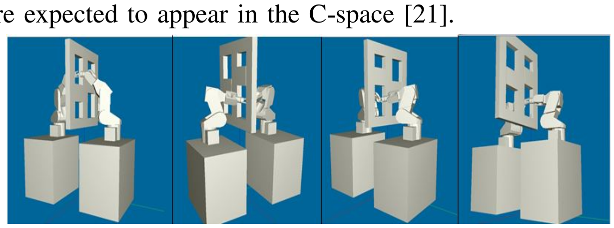

Fig. 6: Four goal configurations in Scenario I.

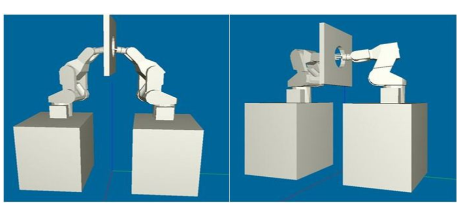

Fig. 7: Two of goal configurations in Scenario II.

As shown in Fig.6, four goal configurations are the two manipulators grasping each other in four holes respectively

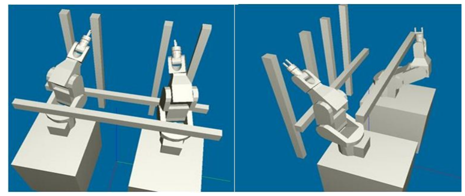

Fig. 8: Two of goal configurations in Scenario III.

in scenario I. As shown in Fig.7, it intends to find a collision-free path to complete a grasper docking motion through the hole from a random start configuration. As shown in Fig.8, two kinds of goal configurations are set. One is random goal configuration. The other is that left manipulator traverse two left bars and the right manipulator traverse two right bars.

TABLE I: Sampling Amount of Different Methods

Method	N_P	N_M	$N_{M'}$	N_B	N_S	T_c(s)
NPP	500	2244	**984**	–	**2744**	2.53
CBB	500	2244	312	–	2744	2.59
DBB	500	2244	476	7720	10464	9.18
DRM	2744	–	–	–	2744	4.57

TABLE I shows the number of sampling points in different methods. The number of main points N_P is crucial in realization. If it is too large, computation of points will cost too much time and updating phase will be time consuming. If it is too small, the resolution of final roadmap may be not large enough for successful planning. Column N_M represents the number of midpoints in roadmap. Column $N_{M'}$ represents the number of bridges in the whole planning phase. Compared with CBB and DBB, $N_{M'}$ in NPP increases by 64.71% and 46.15% respectively, because more kinds of bridges are built with EDBB. Column N_B is the number of incremental points which are used to boost narrow passages. There are no incremental points for NPP and CBB in preprocessing phase. The incremental points in CBB are also generated in updating phase and the validity of these points are predicted by using Parzen Window. N_S represents the amount of sampling points. Column T_c presents the time of constructing roadmap. T_c of NPP is less than DBB's, because incremental sampling points are not generated in preprocessing phase.

The predictive model is the most important part in this paper. Different predictive models are learned in different scenarios in preprocessing phase. In each scenario, 100 positive samples and 100 negative samples are acquired. For every training sample, k is assigned as 7. In other words, center point n_1 has 6 neighbors. Therefore, feature vector is 21 dimension vector. Only 90% samples are required for training and the other 10% samples are required for testing.

The results of different methods are presented in TABLE II. Column SR represents the success rate of planning. Column N_{avg} and column N_l represent the number of average re-searching times and the number of largest re-searching times, respectively. Column T_p is the total preprocessing time. Column T_{avg} represents the average planning time. Twelve groups of experiments are carried out with 500 different planning for each group. Each 500 planning are carried out with the same output files of the preprocessing phase respectively.

TABLE II: Results of Different Methods

Scenario	Method	SR	N_{avg}	N_l	T_p(min)	T_{avg}(s)
I	**NPP**	**94.73%**	**21.81**	**43**	**14.57**	**0.287**
	CBB	87.14%	35.19	62	14.14	0.248
	DBB	87.96%	44.32	79	25.72	0.516
	DRM	70.65%	53.65	92	14.02	0.382
II	**NPP**	**95.27%**	**19.69**	**36**	**14.39**	**0.264**
	CBB	91.48%	32.27	57	14.14	0.241
	DBB	91.11%	44.63	71	25.72	0.518
	DRM	72.29%	50.32	87	14.02	0.385
III	**NPP**	**93.85%**	**23.74**	**49**	**15.43**	**0.253**
	CBB	86.32%	38.37	65	14.14	0.257
	DBB	86.98%	45.71	68	25.72	0.621
	DRM	71.47%	60.65	93	14.02	0.397

As shown in TABLE II, the SR of NPP, CBB and DBB improve much than DRM's with the same number of main points in all scenarios. Compared with CBB and DBB, the SR of NPP also improves in all scenarios, because EDBB builds more bridges to identify more narrow passages effectively. The NPP's N_{avg} and N_l decrease tremendously than CBB's. It is due to that not only EDBB identifies narrow passages more completely but also the validity of incremental points are estimated more accurately than CBB. The T_{avg} of NPP is not the best result as well, because many calculations with NPP lead to much longer time to predict a sampling point. However, there is not a very large growth of the T_p and the T_{avg} in NPP compared with other methods. Therefore, NPP is a better choice that considers both SR and planning time.

TABLE III: Results of Different Predictive Model

Scenario	Model	SR	N_{avg}	N_l	T_p(min)	T_{avg}(s)
I	**NPP**	**94.73%**	**21.81**	**43**	**14.57**	**0.287**
	PW	94.13%	33.59	62	14.08	0.284
II	**NPP**	**95.27%**	**19.69**	**36**	**14.39**	**0.264**
	PW	94.90%	30.27	55	14.08	0.296
III	**NPP**	**93.85%**	**23.74**	**49**	**15.43**	**0.253**
	PW	93.54%	38.15	67	14.08	0.217

Moreover, another six groups of experiments are carried out with 500 different planning to verify the predictive performance of NPP. Parzen Window (PW) is taken as the comparison task. In these groups of experiments, EDBB is used to identify narrow passages, then PW and NPP are used to predict the validity of incremental points respectively. As shown in TABLE III, under a similar SR, the N_{avg} and the N_l of NPP is much lower than PW's, because the incremental points are estimated more accurately than PW. Moreover, the N_{avg} of NPP decreases 14.41 compared with PW in scenario III, while it only decreases 10.58 in scenario II. Therefore, NPP has high universality, because it captures more accurate structure of local narrow passages. Compared with row CBB in TABLE II, the SR of PW with EDBB is higher than CBB's. It is because EDBB identifies more bridges than CBB usefully. The experiments show that NPP gains high SR without unacceptable cost of time. In conclusion, the SR and the N_{avg} of NPP outperform the other related methods.

V. Conclusions

In changing environments, the problem of narrow passages is much difficult for path planning. In this paper, the Narrow Passage Predictive model (NPP) is proposed to capture approximate structure of local narrow passages, which includes following two parts. One is that EDBB, which builds bridges with both validity-unchanged sampling points and validity-toggle sampling points. EDBB can build more bridges than other related methods, because both time information and space information are taken into account. The other is the predictive model trained by SVM, which predicts validity of every generated incremental point instead of invoking collision checker online. It can decrease the average re-searching times and the largest re-searching times efficiently. Compared with other related methods applied in changing environments, experimental results of NPP show an outstanding performance in both success rate of planning and re-searching times.

References

[1] L.E.Kavraki, P.Svestka, J. C. Latombe, etc, "Probabilistic roadmaps for fast path planning in high-dimensional configuration space", *IEEE Trans.Robotics and Automation*, vol.12, no.4, pp.566-580, 1996.

[2] R.Geraerts and M.H.Overmars, "A comparative study of probabilistic roadmap Planners", *WAFR*, pp.249-264, 2002.

[3] R.Bohlin and L.E.Kavraki, "Path planning using Lazy PRM", *ICRA*, pp.521-528, 2000.

[4] H.Liu, X.Deng, H.Zha, etc, "A path planner in changing environments by using W-C Nodes Mapping coupled with Lazy Edges Evaluation", *IROS*, pp.4078-4083, 2006.

[5] D.Hsu, L.E.Kavraki, J.-C. Latombe etc, "On finding narrow passages with probabilistic roadmap planners", *WAFR*, pp,141-153, 1998.

[6] N.M.Amato, O.B.Bayazit, L.K.Dale, etc, "OBPRM: an obstaclebased PRM for 3D workspaces", *WAFR*, pp.155-168, 1998.

[7] V.Boor, M.H.Overmars, and A.F.van der Stappen, "The Gaussian sampling strategy for probabilistic roadmap planners", *ICRA*, pp.1018-1023, 1999.

[8] D.Hsu, T.Jiang, J.Reif, etc. "Bridge test for sampling narrow passages with probabilistic roadmap planners", *ICRA*, pp.4420-4426, 2003.

[9] J.Denny and N.M.Amato, "The Toggle Local Planner for Sampling-Based Motion Planning", *ICRA* , pp.1779-1786, 2012.

[10] B.Burns and O.Brock,"Information Theoretic Construction of Probabilistic",*ICRA*,pp.650-655.

[11] B.Burns and Oliver Brock, "Sampling-Based Mtion Planning Using Predictive Models", *ICRA*, pp.3120-3125, 2005.

[12] B.Burns, O.Brock, "Toward Optimal Configuration Space Sampling", *Robotics: Science and Systems, MIT Press*, pp. 105-112, 2005.

[13] P.Leven, and S.Hutchinson, "Toward real-time path planning in changing environments", *WAFR*, pp.363-376, 2000.

[14] M.Kalmann and M.Mataric, "Motion planning using dynamic roadmaps", *ICRA*, pp.4399-4404, 2004.

[15] D.Ding, H.Liu, X.Deng, etc, "A dynamic bridge builder to identify difficult regions for path planning in changing environments", *IROS*, pp.2133-2138, 2007.

[16] H.Liu, T.Zhang and Q Wang, "A "capacitor" bridge builder based safe path planner for difficult regions identification in changing environments", *IROS*, pp.3179-3186, 2012.

[17] C. Cortes and V. Vapnik. "Support vector networks", *Machine Learning*, Vol.20, Is.3, pp.273-297, 1995.

[18] Cherkassky, V. "The Nature of Statistical Learning Theory", *IEEE Tran. Neural Networks*, pp.1564, 1997.

[19] N.M.Amato, O.B.Bayazit, L.K.Dale,etc, "Choosing good distance metrics and local planners for probabilistic roadmap methods", *ICRA*, pp.630-637, 1998.

[20] J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path lanning", *ICRA*, pp.994-1011, 2010.

[21] J. P. van den Berg, and M. H. Overmars, "Using work space information as a guide to non-uniform sampling in probabilistic roadmap planners", *ICRA*, pp.453-460, 2004.