

Narrow Passage Watcher for Safe Motion Planning by Using Motion Trend Analysis of C-Obstacles

Hong Liu and Yan Wang*

¹Key Laboratory of Machine Perception and Intelligence, Peking University, Beijing, China

²Engineering Lab on Intelligent Perception for Internet of Things (ELIP), Shenzhen Graduate School of Peking University, Shenzhen, China

Abstract: Finding safe paths for robots in changing environments is a significant issue for motion planning. However, it could be fairly difficult when there are narrow passages in the configuration space. Solutions to this problem can be applicable to not only mobile robots but also other domains such as computer animation and computational biology. This paper presents a novel method called Narrow Passage Watcher (NPW) to cope with narrow passage issues in changing environments. It approximately predicts the variation trend of narrow passages and analyzes their security and thus guides to safe path planning. Meanwhile, a supporting hybrid boost strategy is presented to increase the sampling density inside narrow passages with different variation trend. Compared with existing work, the predictive mechanism provided by NPW gives the planner foresight so that it can find safer paths in changing environments with a higher success rate. Experiments conducted with a dual-manipulator system with 12 DOFs show that NPW can reduce the number of replanning times and total planning time remarkably as well as improving the success rate of path planning.

Keywords: Manipulator planning, Narrow passage, Robot and Mechanism, Safe motion planning.

1. INTRODUCTION

Probabilistic Roadmaps (PRMs) [1] and Rapidly Exploring Random Trees (RRTs) [2] are two typical sampling-based methods of robot motion planning. They have shown spectacular planning power for many-DOF robots, but narrow passages and small volume regions in the free configuration space (C-free) are still bottlenecks because of the difficulty to sample inside them [3].

Some methods aiming at increasing sampling density within difficult regions have been proposed. For narrow passages, both filtering strategies [4, 5] and retraction strategies [6, 7] have been researched. Methods proposed in [8-10] try to exploit the information about C-obstacles to construct the roadmap. They have notable effects on boosting sampling density within narrow passages in static environments but are helpless in changing environments.

Dynamic Roadmaps (DRMs) [11, 12] are put forward to cope with motion planning problems in changing environments. However, they are still confronted with difficulties from narrow passages of C-free and only a few researches have paid attention to this work [13, 14]. In changing environments, considering the indeterminacy of the variation trend of narrow passages, it is hard for the planner to consciously seek out a safe path passing through narrow passages which keep narrowing down.

This paper introduces Narrow Passage Watcher (NPW) to anticipate in the variation trend of narrow passages and take appropriate measures to guide to the path planning. In changing environments, NPW analyzes the motion trend of C-obstacles on the basis of the validity toggle of points. Narrow passages formed by heading-toward C-obstacles will have a shrinking effect, they can be identified by NPW and are thought to be more and more dangerous. Incidence edges of points within them are given a higher cost as the penalty. Contrarily, narrow passages that are formed by backing-away C-obstacles have a widening trend and they are deemed to be safer to pass through, incidence edges of points within them are given a lower cost, thus the planner will choose the safer path with a higher priority. For increasing the sampling density within narrow passages, a hybrid boost strategy based on the NPW's analysis is proposed to implement different boost intensity for narrow passages with different variation trends. Safe narrow passages will contain more activated boost points so that the success rate of finding a collision-free path is increased.

2. RELATED WORKS

DRM, It is a direct descendant of PRM method. It precomputes a roadmap that only encodes valid motions in the preprocessing phase. Beyond that, a uniform rectangular cell decomposition of workspace is built and the representation that encodes the mapping from cells in the discretized workspace to nodes and edges in the roadmap is computed. This representation is called the W-C Map. In the online planning phase, the planner first identifies cells occupied by obstacles

*Address correspondence to this author at the No. 2199, Lishui Road, Shenzhen, China. Postcode: 518055; Tel: (86)15815538959; E-mail: yanwang@sz.pku.edu.cn

in the discretized workspace, and then disables the corresponding nodes and edges in the roadmap according to the encoded mapping. However, the connectivity improvement of the roadmap stays to be a difficult problem when narrow passages exist.

Bridge Builder Planner. Random Bridge Builder (RBB) [5] boosts the sampling density inside narrow passages in static environments. Non-uniform sampling is conducted around the midpoint of bridge and thus connectivity of the roadmap is improved. Dynamic Bridge Builder (DBB) [13] is a method to identify narrow passages in changing environments. It is a combination of DRM and RBB and performs well on narrow passages identification, but it cannot guarantee safety for the path through the narrow passage and the path is apt to collide with moving obstacles.

Safe Motion Planning. Safe motion planning has been studied in several works and is important to guarantee the robot's security and reduce total planning cost. As a variant of DBB, Capacitor Bridge Builder (CBB) is proposed to identify difficult but relatively safe regions in changing environments [14]. Nevertheless, CBB does not pay attention to the variation trend of narrow passage in the C-space, which is crucial for the security of planning.

3. HIERARCHICAL SAMPLING STRATEGY

In preprocessing phase, a hierarchical sampling strategy is employed to generate point set G_n and edge set G_a of the roadmap. Meanwhile, W-C Map is computed in the same way with DRM. Sampling points are divided into three levels. Points in the first level are generated uniformly at random and denoted by P , describing the general structure of C-space. For each $p \in P$ and its B nearest neighbors $p_i \in P, i = 1, \dots, K$, create edges between P and P_i , thus the initial roadmap is built. Edges in the initial roadmap consist of G_{ap} , which is defined with:

$$e \in G_{ap} \Leftrightarrow \forall e \in G_a : p_1, p_2 \in P \quad (1)$$

where P_1 and P_2 are two endpoints of edge e . For each edge in G_{ap} , their midpoints are put into the roadmap and comprise the second level, denoted by M . Points in M are used to conduct the bridge test. Within radius R of each point $p \in P \cup M$, K points are produced, which is denoted by T_p . Here, R is chosen by Gaussian distribution and we say that each point in T_p belongs to P . All T_p makes up the third level, which are added into the roadmap and denoted by T . For each $t \in T$, we identify its K nearest neighbors in the roadmap and connect them. Points in T are set inactive initially and they cannot take part in path query, which is completed by A* algorithm, until they are activated.

Points in P record the fundamental structure of C-space. In order to describe the validity toggle of points in P , we follow a similar definition of P_+ and P_- to that introduced in [14]:

$$p \in P_+ \Leftrightarrow \forall p \in P : p.\text{validity turns from false to true} \quad (2)$$

$$p \in P_- \Leftrightarrow \forall p \in P : p.\text{validity turns from true to false} \quad (3)$$

4. NARROW PASSAGE WATCHER

Narrow Passage Watcher (NPW) is designed to predict the variation trend of narrow passages by analyzing the motion trend of C-obstacles. Moving obstacles in the workspace will validate or invalidate some points in P . P_+ and P_- consist of these points, respectively. Points keeping their validity unchanged usually indicate obstacles without motion or the interior of obstacles with slight motion and they make up P_0 .

P_+ points are usually on the backside of moving C-obstacles while P_- points always lie ahead of them. Therefore, P_+ points would keep valid for some time while P_- points couldn't turn valid very soon. During a very short time slice, the movement and deformation of C-obstacles are tiny, points in P_+ are much safer than points in P_- .

The edge e whose midpoint is valid and two endpoints are on the C-obstacle or near the C-obstacle's boundary is called bridge. Particularly, NPW are concerned with two bridge sets whose variation trends are salient: B_r and B_s , which are defined with: P_+

$$e \in B_r \Leftrightarrow \forall e \in G_{ap} : p_1, p_2 \in P_-, m.\text{validity} = \text{true} \quad (4)$$

$$e \in B_s \Leftrightarrow \forall e \in G_{ap} : p_1, p_2 \in P_+, m.\text{validity} = \text{true} \quad (5)$$

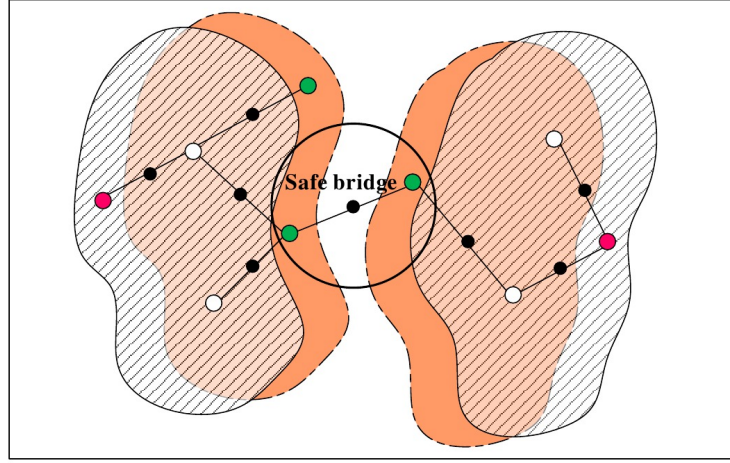
Where, B_r is the set of risky bridge and B_s is the set of safe bridge, P_1 and P_2 are two endpoints of e , m is the midpoint of e . Narrow passages identified by safe bridges tend to broaden and those identified by risky bridges tend to shrink. Any point P on the safe bridge has a good local security, i.e. points and edges in the small neighborhood of P are less probable to be invalidated by moving C-obstacles. The safe bridge and risky bridge are shown in Fig. (1), in which orange areas represent the previous locations of C-obstacles, shaded areas represent the current locations of C-obstacles.

Algorithm 1 Narrow Passage Watcher

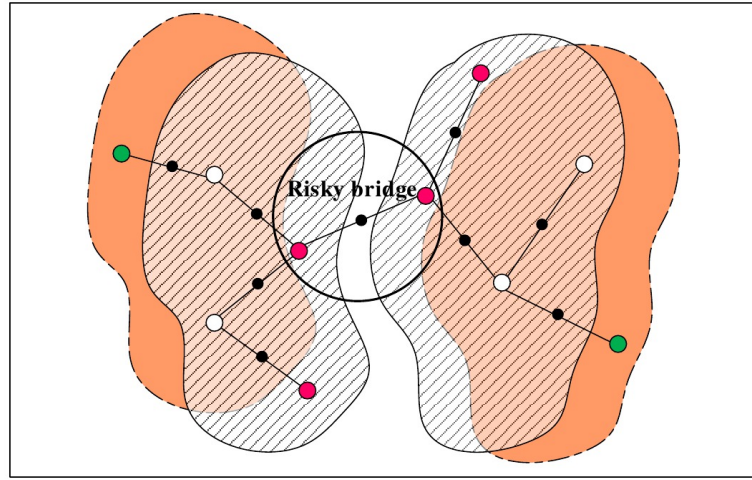
```

1: for each edge  $e \in G_{ap}$  do
2:   Pick two endpoints  $p_1, p_2$  and the midpoint  $m$ 
3:   if  $p_1, p_2 \in P_-$  and  $m.\text{validity} = \text{true}$  then
4:      $B_r \leftarrow B_r \cup \{e\}$ 
5:     for each midpoint-edge  $e'$  of  $e$  do
6:        $e'.\text{length} \leftarrow e'.\text{length} \cdot \lambda_r$ 
7:     end for
8:   end if
9:   if  $p_1, p_2 \in P_+$  and  $m.\text{validity} = \text{true}$  then
10:     $B_s \leftarrow B_s \cup \{e\}$ 
11:    for each midpoint-edge  $e'$  of  $e$  do
12:       $e'.\text{length} \leftarrow e'.\text{length} \cdot \lambda_s$ 
13:    end for
14:   end if
15: end for

```



(a)



(b)

Fig. (1). (a) Safe bridge and (b) risky bridge. White, red, green and black points are in P_0 , P_- , P_+ and M , respectively.

```

16: for each bridge  $b \in B_r \cup B_s$  do
17:   for each boost-edge  $e$  of  $b$  do
18:     Pick two endpoints  $p1$  and  $p2$ 
19:     if  $p1 \in P_+$ ,  $p2 \in T$  and  $p2.validity = true$  then
20:        $e.length \leftarrow e.length \cdot \lambda_b$ 
21:     end if
22:   end for
23: end for

```

Edges that connect the bridge's midpoint m and boost points belonging to m are called midpoint-edges. For the safe bridge, valid midpoint-edges could keep valid for a short time and they are relatively safe. A weight $\lambda_s < 1$ is given to them to make the planner choose them with higher priority when querying a path. For the risky bridge, valid midpoint-edges tend to turn invalid because C-obstacles

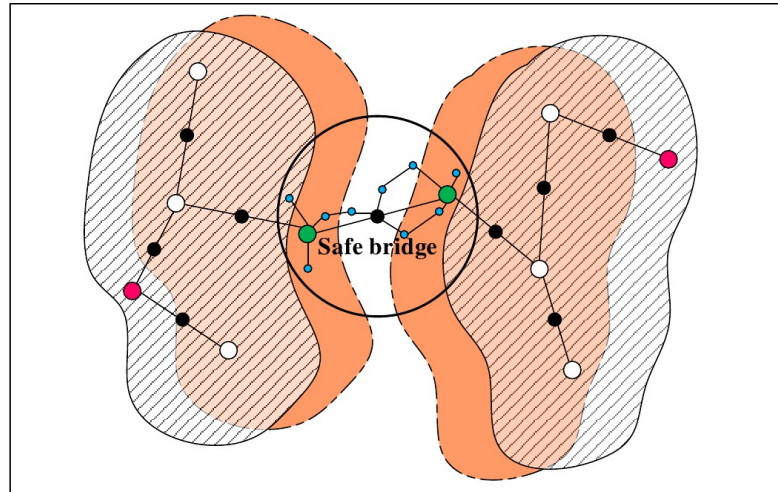
are approaching them. Hence, a weight $\lambda_r > 1$ is given to them so that the planner will avoid them as far as possible. A Gaussian function is used to estimate the influence that the bridge's security will have on edges nearby [15] and λ_s and λ_r are defined as follows:

$$\lambda_s = 1 - e^{-k(m-m)^2} \quad (6)$$

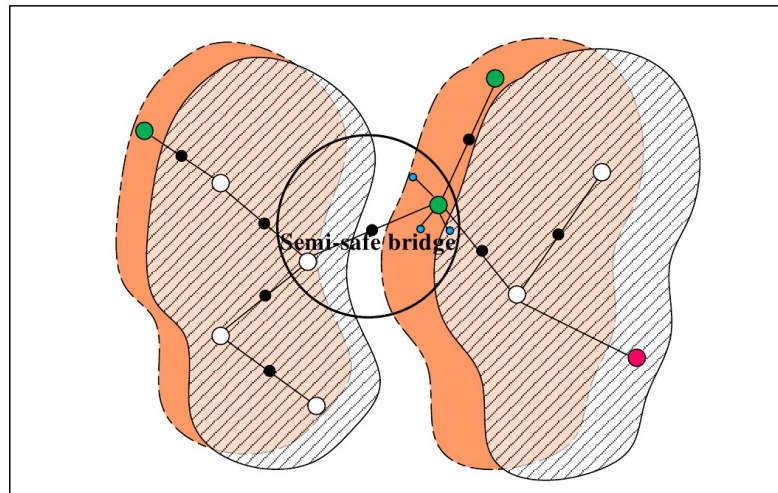
$$\lambda_r = 1 + e^{-k(m-m)^2} \quad (7)$$

Here, k is a smoothing parameter which adjusts the spread of the Gaussian function.

Each incidence edge of boost points around the bridge's midpoint (except for the midpoint-edge) is called the boost-edge. NPW estimates the security of each valid



(a)



(b)

Fig. (2). Boost the narrow passage identified by (a) safe bridge and (b) semi-safe bridge. Activated boost points are colored in blue.

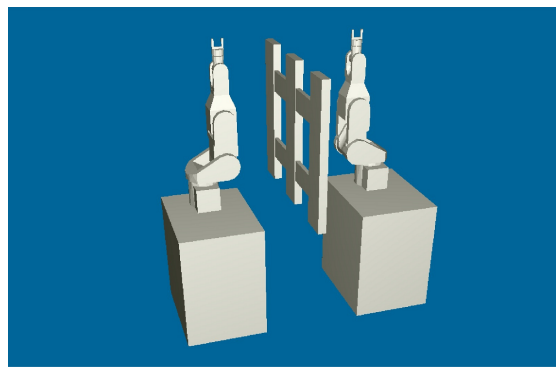
boost-edge: if one endpoint is in P_+ and another endpoint in T is valid, the boost-edge will have a relatively high security level and a small weight λ_b should be given to it, which is calculated using a similar weighting function with formula (6). See Algorithm 1 for details of Narrow Passage Watcher (NPW).

5. HYBRID BOOST STRATEGY

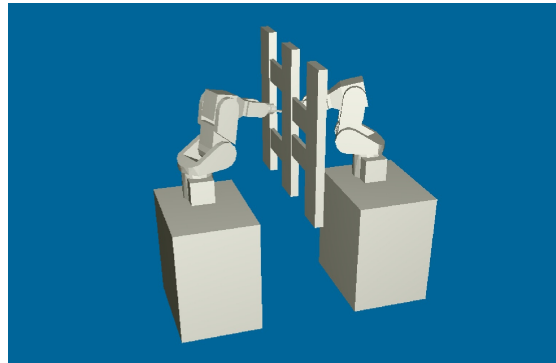
For narrow passages having different variation trends, different boost strategies ought to be employed. Narrow passages which are identified by bridges in B_S tend to widen. For each safe bridge in B_S , we can increase the

sampling density by activating boost points belonging to its midpoint and endpoints. Fig. (2a) illustrates boost effect inside narrow passage identified by the safe bridge. Narrow passages which are identified by bridges in B_r tend to shrink, the robot should keep away from them as far as possible and no boost points around them are activated.

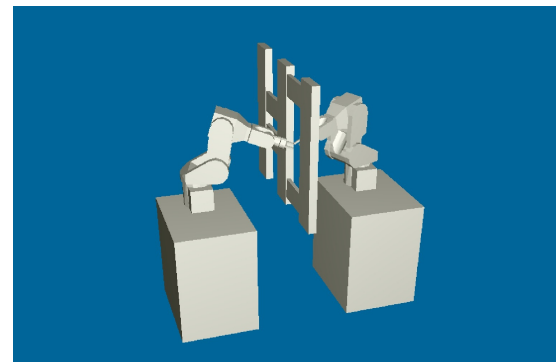
When the narrow passage is formed by a couple of obstacles of which one obstacle has little motion and another moves away from it, the semi-safe bridge of which one endpoint in P_+ and another endpoint in P_0 but invalid is always built. The endpoint in P_+ of the semi-safe bridge could be deemed safe. Consequently, boost points



(a)



(b)



(c)

Fig.3 Environment I is illustrated by (a), (b) and (c) gives two goal configurations.

belonging to this safe endpoint should be activated. Fig. (2b) shows boost effect inside narrow passage identified by the semi-safe bridge.

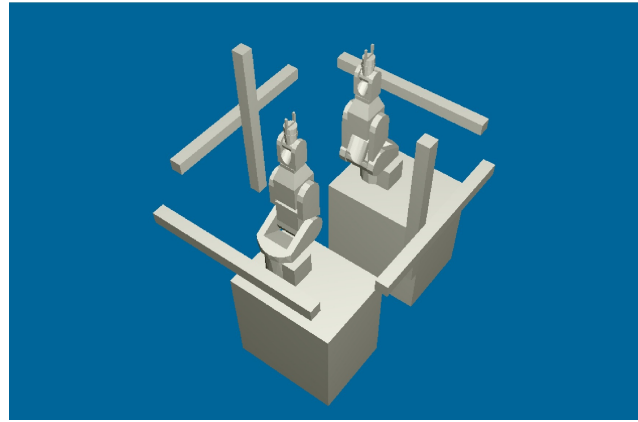
For each point $t \in T$, once it is activated, they are involved in path query and can be searched normally just like points in P and B . Once a replanning cycle is completed, all boost points are inactivated while P_+ , P_- , P_0 , B_S and B_r are cleared.

6. EXPERIMENTS AND DISCUSSIONS

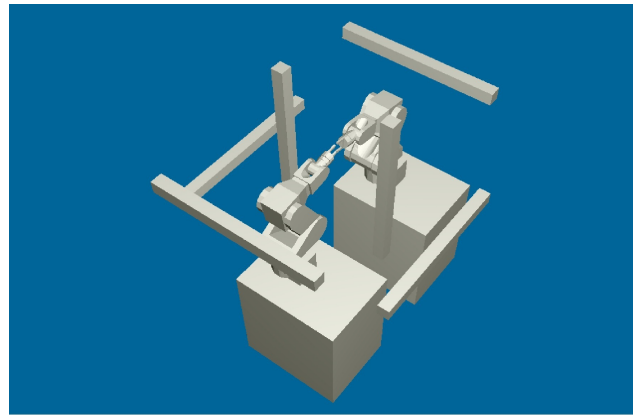
Several simulation tests are performed in 3D workspace to evaluate our method. A dual-manipulator system is made up of two manipulators which are modeled by

parameters of a practical 6-DOF Kawasaki FS03N manipulator mounted on the fixed base. Our method treats them as a single robot with 12 DOFs. The reachable workspace of two manipulators is decomposed into 82960 (40*31*34) grids. Each grid is a cube with the size of 4*4*4. All simulations are implemented in C++ and carried out on a 3.10GHz Intel Core i3-2100 CPU with 2GB of main memory.

Our first environment involves a changeable fence with two gaps, which consist of three vertical boards and four horizontal boards between two manipulators (See Fig. 3a). At run time, two gaps change size on reverse trend: one gap widens and another narrows. This is done by keeping vertical boards still while horizontal boards move up and down in opposite directions within a relatively small range



(a)



(b)

Fig. (4). Environment II is illustrated by (a), and (b) shows the goal configuration.

Table 1. Number of sampling points and construction time of roadmap.

Method	N_p	N_m	N_t	N_s	Construction Time(s)
NPW	500	1441	7764	9705	6.49
CBB	500	1441	2000	3941	3.02
DBB	500	1441	5764	7705	5.18
DRM	9705	-	-	9705	6.53

at a speed of 4 unit lengths per step. Two docking states are treated as goal states (See Fig. 3b, c). Two manipulators have to pass through some difficult regions that are changing before they get to the goal configuration.

Our second environment is more complicated than the first one. See Fig. (4), for Environment II: vertical bars can only move back and forth, while horizontal bars can only move up and down. The speed of each bar is 4 unit lengths per step. Each bar has a small move range: once it gets to the end of range, its direction is reversed. Bars are very close to each other to ensure the appearance of narrow passages. Manipulators intend to reach the docking position, which is the goal configuration.

NPW, CBB, DBB and DRM are tested in both Environment I and Environment II, respectively. In Table 1, N_p , N_m and N_t are the number of points in P , M , and T , respectively. N_s is the total number of points. Enough points are necessary for the planner to learn the environment. However, a large N_p could be a burden for the planner. Here, N_p is set to be 500. K is set to be 4, it decides the number of edges in the roadmap, thereby influencing N_m . For NPW, the weight of midpoint-edges and boost-edges influence the selection of path remarkably. Empirically, the smoothing parameter k of the Gaussian function in formula (6) and (7) is set to be 0.8. The time consumption of building the roadmap in preprocessing phase is also reported in Table 1.

Table 2. Comparison of success rate and number of replanning times.

Env. I					Env. II				
Method	R_{suc} (%)	N_{avg}	N_{max}	T_{avg} (s)	Method	R_{suc} (%)	N_{avg}	N_{max}	T_{avg} (s)
NPW	92.80	22.80	39	9.23	NPW	92.77	24.41	46	10.27
CBB	87.45	36.91	60	12.90	CBB	87.41	40.10	70	13.05
DBB	88.39	47.60	74	28.10	DBB	87.56	50.55	82	30.62
DRM	76.30	61.25	83	33.57	DRM	75.18	63.76	99	37.40

Table 3. Comparison of total and average replanning time.

Env. I				Env. II			
Method	T_{total}	T_{single}	T_{total} / T_{comp}	Method	T_{total}	T_{single}	T_{total} / T_{comp}
NPW	3.65	0.16	39.54	NPW	4.15	0.17	40.41
CBB	5.54	0.15	42.95	CBB	6.02	0.15	46.13
DBB	7.14	0.15	25.41	DBB	8.09	0.16	26.42
DRM	18.99	0.31	56.57	DRM	22.32	0.35	59.68

Each method is tested 100 times to compare their effectiveness and efficiency. Results are reported in Table 2. We compared the average numbers N_{avg} and the maximum numbers N_{max} of replanning for each method. The average completion time T_{avg} of planning task and success rate R_{suc} of finding a free path are also investigated.

As shown in Table 2, both the average and the largest cycles of replanning of NPW are the lowest. This is due to the prediction mechanism of NPW: it forecasts the variation trend of narrow passage and gives the safe narrow passage an intense boost. DBB cannot predict the variation trend of narrow passages, it boosts the sampling density in both risky narrow passages and safe narrow passages blindly. CBB can only build bridges whose one endpoint is in P_+ and another in P_- .

Meanwhile, success rate of NPW has about 18.37% rise compared with DRM due to the hybrid boost strategy which activates lots of boost points within safe narrow passages. In fact, NPW has the most boost points in the above four methods, but its efficiency of path query is barely reduced because it only activates some boost points that have good security so that only a part of boost points are active during online path planning and others are inactive.

The average total replanning time T_{total} in each complete planning task of each simulation is recorded in

Table 3. Meanwhile, the average time of each single replanning T_{single} and the ratio of total replanning time to the task's completion time T_{total} / T_{comp} are also reported. Obviously, NPW always takes the least average replanning time, which only accounts for about 18.93% of DRM's.

CONCLUSION

In this paper, we present Narrow Passage Watcher (NPW) to predict the variation trend of narrow passages and analyze their security. On that basis, a hybrid boost strategy is employed to carry out different boost intensity aiming at different narrow passages. NPW dramatically reduces the number of replanning times and lifts the success rate of the planner finding a collision-free path in changing environments. NPW has shown its considerable merit with resolving narrow passage problems and could work very well in high-dimensional C-space.

In the future, we will explore a general method that can process a wider range of narrow passage with different variation trend in changing environments. We also plan to research how to express the security of narrow passage more clearly and quantitatively.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

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] L.E. Kavraki, P. Svestka, J.-C. Latombe and M.H. Overmars, "Probabilistic roadmaps for fast path planning in high-dimensional configuration spaces" *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566-580, 1996.
- [2] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning", *Int. J. Robot Res.*, vol. 20, no. 5, pp. 378-400, 2001.
- [3] D. Hsu, L.E. Kavraki, J. C. Latombe, Motwani and R. Sorkin, "On finding narrow passages with probabilistic roadmap planners", *Robotics: The Algorithmic Perspective, A.K.Peters, Wellesley, MA*, pp. 141-154, 1998.
- [4] V. Boor, M. H. Overmars, A. F. V. D. Stappen, "The Gaussian sampling strategy for probabilistic roadmap planners" In: *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 1018-1023, 1999.
- [5] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners" In: *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 4420-4426, 2003.
- [6] 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: *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 1024-1031, 1999.
- [7] M. Saha and J.-C. Latombe, "Finding narrow passages with probabilistic roadmaps: the small step retraction method," In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 622-627, 2005.
- [8] N.M. Amato, O.B. Bayazit, L.K. Dale, C. Jones and D. Vallejo, "OBPRM: an obstacle-based PRM for 3D workspaces," In: *Proc. Int. Workshop on Algorithmic Foundations of Robotics*, pp. 155-168, 1998.
- [9] H.Y. Yeh, S. Thomas, D. Eppstein and N.M. Amato. "UOBPRM: a uniformly distributed obstacle-based PRM", In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2655-2662, 2012
- [10] J. Denny and N.M. Amato, "Toggle PRM: simultaneous mapping of C-free and C-obstacle - a study in 2D", In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2632-2639, 2011.
- [11] P. Leven and S. Hutchinson, "A framework for real-time path planning in changing environments" *Int. J. Robot. Res.*, vol. 21, no. 12, pp. 999-1030, 2002.
- [12] M. Kallmann and M. Mataric, "Motion planning using dynamic roadmaps," In: *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 4399-4404, 2004.
- [13] D. Ding, H. Liu, X. Deng and H. Zha, "A dynamic bridge builder to identify difficult regions for path planning in changing environments," In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Syst.*, pp. 2925-2931, 2007.
- [14] H. Liu and T. Zhang, "A "capacitor" bridge builder based safe path planner for difficult regions identification in changing environments," In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3180-3186, 2012.
- [15] B. Burns and O. Brock, "Sampling-based motion planning using predictive models," In: *Proc. IEEE Int. Conf. on Robotics and Autom.*, pp. 3120-3125, 2005.

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© Liu and Wang; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.