Chapter 34
Path Planning in Changing Environments Based on “Frame” Difference

Zhi Wang, Fang Xiao and Hong Liu

Abstract An effective path planner to solve difficult region problems in changing environments is proposed in this paper. When obstacles move at uncertain speeds, difficult regions change their characteristics accordingly. Identifying difficult regions is a thorny issue. A novel method using “Frame” Difference (FD) is presented in this paper, which is motivated by the idea of moving object detection. Changing regions are detected by FD and obstacle speed can be predicted qualitatively by counting the number of toggle points in those regions. Then, in order to adapt to different speeds, hybrid difference algorithm (HDA) which is a hybrid of adjacent frame difference or K-frame difference is proposed. HDA provides enough movement information of obstacles, and leads to safe path planning. Experiments conducted with a dual-manipulator system show that our method has lower replanning times and higher success rate than related planners, such as capacitor bridge builder and dynamic bridge builder.

Keywords Path planning · Changing environment · Frame difference · PRM

34.1 Introduction

During last two decades, with the development of sampling-based framework [1], a great progress of research in static environments has been obtained, especially rapidly-exploring randomized tree (RRT) [2] and probabilistic roadmap method

Z. Wang (✉)
University of Science and Technology, Beijing 100083, China
e-mail: zhiwang0503@163.com

F. Xiao
Shenzhen Graduate School of Peking University, Shenzhen 518055, China
e-mail: 1201213677@sz.pku.edu.cn

H. Liu
The Key Laboratory of Machine Perception, Peking University, Beijing 100087, China
e-mail: hongliu@pku.edu.cn

Z. Deng and H. Li (eds.), *Proceedings of the 2015 Chinese Intelligent Automation Conference*, Lecture Notes in Electrical Engineering 338,
DOI 10.1007/978-3-662-46466-3_34

(PRM) [3]. In static environments, many variants of randomized algorithms are proposed for difficult region problems, like obstacle-based PRM [4], Bridge Test [5] and Gaussian Sampling Strategy [6]. However, in changing environments, due to the movement of obstacles in workspace (W-space), configurations occupied by obstacles (C-obstacle) will change accordingly. As a result, roadmap cannot accurately reflect the current state of the configuration space (C-space).

Dynamic Roadmap Method (DRM) [7], as a PRM derivative, computes mapping from W-space to C-space (W–C mapping) to solve problems in changing environments, instead of online collision detection. Dynamic bridge builder (DBB) [8] is a combination of DRM and bridge Test. Although DBB performs well on narrow passages identification, it cannot guarantee safety when obstacles are moving. Capacitor bridge builder (CBB) [2] has been proposed as a safe path planning method on the basis of DBB. However, CBB only identified “capacitor” bridges, it cannot cover all of difficult regions.

In this paper, “Frame” Difference (FD) [9] is used initially in path planning. In changing environments, continuous environments are discretized into static fragments. Each fragment is considered as a “frame,” and we construct and update the roadmap in it. In preprocessing phase, points are sampled hierarchically. Then W–C mapping is computed with all points. In updating phase, though FD, changing regions are detected, and the number of toggle points is counted. In order to adapt to different speeds and provide enough movement information, hybrid difference algorithm (HDA) is proposed. After detecting changing regions, difficult regions are identified by bridge planner. Incremental points are activated around the safe endpoints of bridges. The overview of our method is illustrated in Fig. 34.1.

34.2 Safe Path Planning Based on “Frame” Difference

34.2.1 Hierarchy Sampling Strategy (HSS)

In changing environments with difficult regions, planning methods need to not only identify difficult regions accurately, but also increase sampling points in those regions rapidly. HSS reduces the size of W–C mapping by sampling useful points, in order to improve efficiency of path planning (Fig. 34.2).

34.2.2 Changing Regions Detection by “Frame” Difference

By comparing information with adjacent time fragments, changing regions can be detected and movement tendency can be predicted in the next-time fragment.

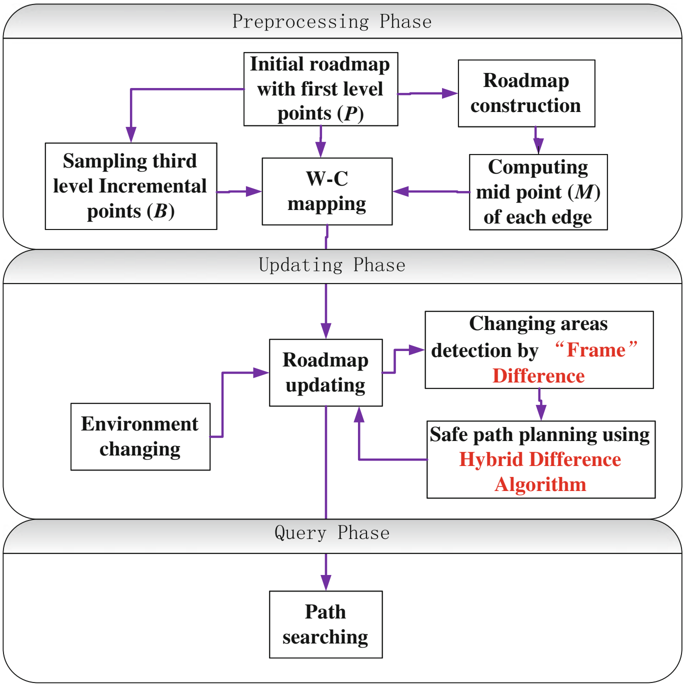

Fig. 34.1 Overview of our method

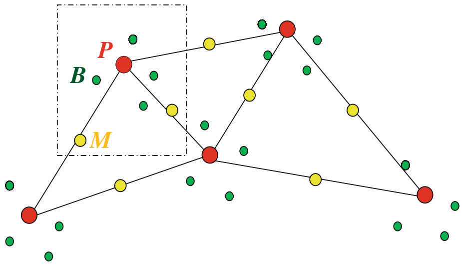

Fig. 34.2 Hierarchical sampling strategy. *Red*, *yellow*, *green* points are in first level (P), second level (M), and third level (B), respectively

Inspired by the moving object detection based on differenced frame process [9], continuous dynamic environments are discretized into static fragments. Each fragment is considered as a "frame," and we build and update the roadmap in it.

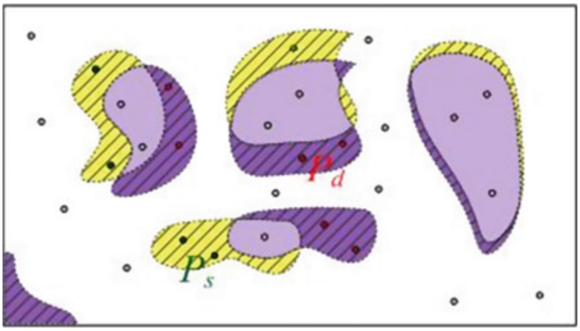

Fig. 34.3 Changing region detection. The regions in *purple* represent current C-obstacle and *yellow* represents their previous position. *Purple* with shadow presents the dangerous region and *yellow* with shadow presents the safe region. P_s is the set of safe points. P_d is the set of dangerous points

The simple way to detect changing regions may be to calculate the difference between previous frame and current frame, called Adjacent-FD. Changing regions are filled with shadow in Fig. 34.3.

$$D^t(p) = V^t(p) - V^{t-1}(p) \tag{34.1}$$

Here, for each $p \in P$, $D^t(p)$ represents the difference of validity of p between frame t and $t-1$. If $D^t(p) = 2$, it means that $V^t(q)$ toggles from -1 to 1. On the contrary, $D^t(p) = 2$ means $V^t(q)$ toggles from 1 to -1. After Adjacent-FD, the number of the toggle points is counted, denoted by N_t.

From another point of view, N_t indicates the motion amplitude of obstacles. If the number of toggle points N_t is larger than threshold T, it means movement speed is relatively fast. If N_t is smaller than T, it means slow and moderate.

34.2.3 Hybrid Difference Algorithm (HDA)

Adjacent-FD computed above is simple and easy to implement, but it cannot provide enough information for slower moving obstacles. Moving slowly means inadequate toggle points with a little information of movement tendency. For the purpose of making motion tendency to be obvious, K-FD is chosen as an improvement.

The new algorithm called K-FD uses current frame to minus previous K frames, respectively,

$$D_k^t(p) = V^t(p) - V^{t-k}(p) \quad (\text{for each}\, p \in P,\ k = 1, 2, \ldots, k) \tag{34.2}$$

$$D^t(p) = \bigcup_{k=1}^{K} D_k^t(p) \tag{34.3}$$

Here, $D_k^t(p)$ represents the difference of validity of p between frame t and $t-k$. $D^t(p)$ is the final difference we want. As long as the difference of adjacent K frames does exist, our method can detect it. If K is too small, motion regions are not that obvious. However, if K is too large, the detection will be too sensitive. Drawn from the experiments, K is set to be 3.

Algorithm 34.1: Hybrid Difference Algorithm

Input: $P = \{p_1 \cdots\cdots p_n\}$, Frame t, Frame $t\text{-}1$
Output: P_s , P_d
for each point $p \in P$ **do**
  Compute Adjacent-FD:
   $D^t(p) = V^t(p) - V^{t-1}(p)$
  Compute the number of toggle points
   $N_t = \sum|D^t(p)|/2$
end for
if $\boldsymbol{N_t}$ < threshold T **then**
  Obstacles move fast
  Compute K-FD:
  for $k \leqslant K$ **do**
   $D_k^t(p) = V^t(p) - V^{t-k}(p)$
   $D^t(p) {+}{=} D_k^t(p)$
  end for
else $N_t \geqslant$ threshold T
  Continue to use Adjacent-FD
end if
if $D^t(p) > 0$ **then**
  $p \in P_s$
else if $D^t(p) < 0$ **then**
  $p \in P_d$
end if

HDA is a hybrid of Adjacent-FD and K-FD. The purpose of using it is to guarantee the number of toggle points. When obstacles move slowly, HDA enables N_t to be enough to build bridges by choosing K-FD. When obstacles move fast, HDA choose Adjacent-FD as before. After HDA, we get $D^t(p)$, which is used to classify p. If $D^t(p)$ is positive, add p to P_s. If $D^t(p)$ is negative, add p to P_d. $D^t(p) = 0$ indicates no difference among K frames. Details are shown in Algorithm 34.1.

34.2.4 Bridge Builder and Boosting Strategy

A Dynamic Bridge Builder with HDA is used in this paper. As shown in Fig. 34.4, after classifying p, this kind of bridges will be flagged: One of the endpoints belongs to P_s, colored in green, the other endpoint belongs to P_d, colored in red, and the midpoint is in C-free, colored in yellow.

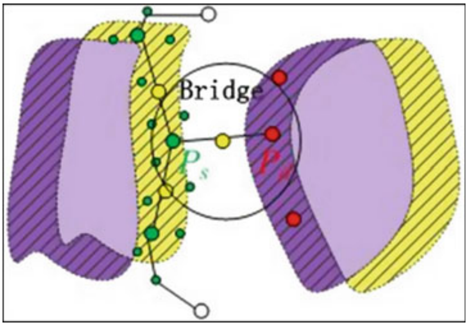

Fig. 34.4 Bridge builder and boost strategy. Bridges are built between P_s and P_d, and incremental points are activated around P_s

According to safe path planning, endpoints $p \in P_S$ are safer than other points, because they always behind obstacles. These points are chosen to be boost, which means incremental points around them will be activated, as shown in Fig. 34.4. Little green points are the third-level points around safe points. This boosting strategy makes more sampling points in safe regions, so that the path we searched will be relatively safe in a short time.

34.3 Experiment and Discussions

To evaluate the proposed method, simulated experiments are implemented in 3D scenario in hundreds of times. The proposed algorithm is mainly for the narrow passages in changing environments. Only when the narrow passages are included can reflect the superiority of it. In 3D W-space, there are two manipulators modeled by parameters of practical 6-DOF Kawasaki manipulators (FS03 N). The dual-manipulator system with 12-DOFs is planned to check the efficiency of algorithm in high-dimensional C-space. Here, we use Coldet2.1, which is popular and free, to conduct collision detection. All our experiments are performed on an ordinary personal computer with 3.00 GHz CPU and 2 GB memory. Experimental results are based on an average of 500 executions.

Generally, when a significant difficult region appears in W-space, C-space will be a corresponding difficult region. Hence, the experimental scenario (Fig. 34.5a) we set includes a number of difficult regions. It involves a rectangular board with a hole in the middle which is placed between two manipulators. The start configuration is randomized (Fig. 34.5b), and the goal configuration is a grasper docking motion through the hole which is difficult to complete (Fig. 34.5c). The rectangular board is always moving up and down at different speeds.

Table 34.1 shows the number of bridges built at different speed using our method. Because of the adaptability to speed, HDA can obtain enough toggle points

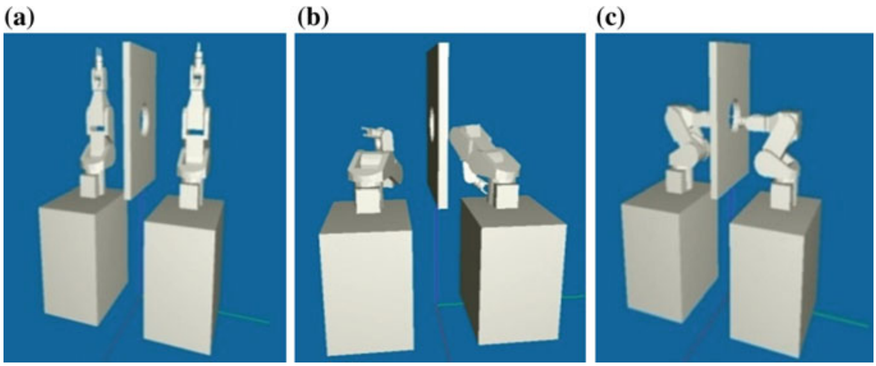

Fig. 34.5 **a** Simulation scenario. **b** A randomized start configuration. **c** A goal configuration

Table 34.1 Bridge builder results based on HDA

P	Speed	Num of bridges			Time (s)
		Max	Min	Ave	
1000	2	153	57	99	0.0037
	3	171	80	118	0.0038
	4	159	68	108	0.0037
2000	2	317	172	231	0.0046
	3	384	196	278	0.0048
	4	346	190	259	0.0048

in changing regions, although obstacles move slowly. As shown in Table 34.1, to different speed, the number of bridges which our method builds is almost the same.

As shown in Table 34.2, our method contributes to higher success planning rate (SPR) and lower average replanning times (ART) than other methods in the table. When obstacles move slowly, the number of toggle points in CBB is limited,

Table 34.2 Results of comparison experiment between HDA, CBB and DRM at different speeds

Method	Speed	SPR	ART	ST
Our method	2	94.13	14.91	9.05
	3	94.09	15.37	8.54
	4	94.10	15.39	8.36
CBB	2	91.98	28.33	16.01
	3	93.10	22.19	12.28
	4	94.03	16.83	9.39
DRM	2	90.47	48.72	48.11
	3	87.69	57.36	59.83
	4	81.70	70.04	71.29

leading to insufficient number of bridges. On the contrary, HDA can obtain sufficient toggle points adaptively according to the speed. Although HDA need to activate more incremental nodes, it gives rise to good performance in replanning, as shown in the last two columns in Table 34.2. ART of our method is lower than CBB and DRM. Meanwhile, ART of our method has no obvious change in different speeds because of the adaptability. Moreover, compared with DRM, our bridge builder using HDA costs less time with higher success rate.

34.4 Conclusions

Motivated by the idea of moving object detection based on differenced frame process, our method uses "Frame" difference to detect difficult regions. A novel hybrid difference algorithm (HDA) is presented in this paper, which is designed for adapting uncertain speeds of obstacles in changing environments. Our method, combined with bridge test, can identify the difficult regions fast and easily. Experimental results in complex environment show that our method is superior to previous methods in success planning rate (SPR), average peplanning times (ART) and sum of time (ST). Generally speaking, HDA, as a method based on FD, is a novel method to solve planning problems in real time.

References

1. Kavraki LE, Svestka P, Latombe JC, Overmars MH (1996) Probabilistic roadmaps for fast path planning in high-dimensional configuration spaces. IEEE Trans Robot Autom 12(4):566–580
2. Liu H, Zhang T (2012) A capacitor bridge builder based safe path planner for difficult regions identification in changing environments. In: IEEE/RSJ international conference on intelligent robots and systems, pp 3180–3186
3. Geraerts R, Overmars MH (2004) A comparative study of probabilistic roadmap planners. In: Algorithmic foundations of robotics V. Springer, Berlin, Heidelberg
4. Amato NM, Bayazit OB, Dale LK, Jones C, Vallejo D (1998) OBPRM: an obstacle-based PRM for 3D workspaces. In: Proceeding international workshop on algorithmic foundations of robotics, pp 155–168
5. Hsu D, Jiang T, Reif J, Sun Z (2003) The bridge test for sampling narrow passages with probabilistic roadmap planners. In: Proceedings IEEE international conference on robotics and automation, vol 3, pp 4420–4426
6. Boor V, Overmars MH, van der Stappen AF (1999) The Gaussian sampling strategy for probabilistic roadmap planners. In: Proceedings IEEE international conference on robotics and automation, vol 2, pp 1018–1023
7. Leven P, Hutchinson S (2002) A framework for real-time path planning in changing environments. Int J Robot Res 21(12):999–1030

8. Ding D, Liu H, Deng X, Zha H (2007) A dynamic bridge builder to identify difficult regions for path planning in changing environments. In: IEEE/RSJ international conference on intelligent robots and systems, pp 2925–2931
9. Zhan C, Duan X, Xu S, Song Z, Luo M (2007) An improved moving object detection algorithm based on frame difference and edge detection. In: Fourth international conference on image and graphics, pp 519–523