Hierarchical Random Exploring with Multiple Linking Modes

Zhengguo Li, Wenchao Gao, Albertus Hendrawan Adiwahono, and Wei Yun Yau

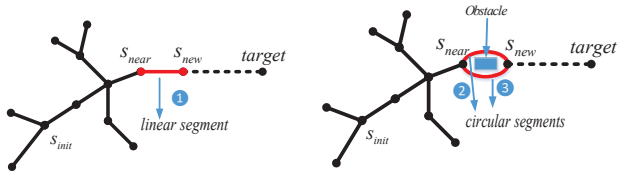
***Abstract*—Narrow passages in the configuration space pose challenge for sampling-based planners. In this paper, the conventional wisdom of "divide and conquer" is adopted to design a hierarchical strategy to address this challenging problem. Rapidly-exploring random tree (RRT) is taken as an example to illustrate the proposed strategy. A hybrid map which consists of a topological map and a set of metric maps is built up to represent the configuration space and information on the narrow passages is included in the topological map. The path from the starting configuration to the goal one is divided into segments using the information on the narrow passages. The RRT runs independently to search for a sub-path for each segment. The overall path is obtained by concatenating all the sub-paths. In addition, to improve the possibility of avoiding obstacles, two more circle-based linking modes are introduced to the RRT planner. Experimental results show that the planning time can be reduced significantly using the proposed planning methods. As a result, the success rate of path planning can be improved if the time for planning is limited.**

I. Introduction

Path planning problems involve searching the system configuration space for a collision-free path which connects a given start configuration with a given goal configuration, while satisfying constraints imposed by complicated obstacles. Sampling-based planners such as rapidly-exploring random trees (RRT) [1], [2] and regionally accelerated batch informed trees (RABIT*) [3] are usually more efficient to high-dimensional problems than graph-search based planners [4], [5]. However, narrow passages in the configuration space pose challenge for the sampling-based planners. This is because the narrow passages have small volumes, and the probability of sampling from small sets is low [6]. Therefore, to enhance the efficiency of these planners in presence of narrow passages becomes the major contribution of this paper.

In this paper, two novel ideas are proposed to simplify and improve the existing sampling-based planners. One is a hierarchical strategy. The strategy is inspired by the hybrid-map-based localization method in [7] and the switched linear system in [8]. Instead of using a single metric map to represent the whole space, a hybrid map was adopted in [7] and it is composed of a topology map and a group of metric maps. Each metric map corresponds to an edge in the topological map. The location of the robot can be determined based on the relative pose with respective to each topological edge rather than the initial pose. As such, the complexity of localization algorithm is reduced significantly. The complexity of analysis and design can be reduced significantly. Intuitively, the same wisdom can be adopted to reduce the complexity of sampling-based planners in presence of narrow passages. Based on this observation, the state space is partitioned into a group of subspaces, and a hybrid map is adopted to represent the state space. Instead of searching a path from the starting configuration to the goal one as in the existing RRTs [1], [2], a sub-path in each segment is searched by the RRT independently. The overall path will be obtained by concatenating all the sub-paths. As such, the complexity of the RRT can be reduced significantly in presence of narrow passages.

The other is to introduce multiple linking modes in order to avoid obstacles more efficiently. This idea is inspired by the continuous-curvature based path planning for obstacle avoidance in [9], [10]. Collision detection plays an important role in RRT and RABIT*, while only the line segment is used in the mentioned algorithms (red line in in Figure 1(a)). Sampling times could be reduced if there are more linking modes for two given vertices. As such, there can be more modes to expand the tree so as to avoid obstacles more efficiently. Based on this observation, two new circle modes are introduced to the sampling-based planners, illustrated by the up and down arcs in Figure 1(b). Again, RRT is taken as an example to illustrate the efficiency of the proposed new modes. Experimental results show that the planning time is reduced and the success rate is increased by the multiple linking modes compared to the conventional straight line extension, given the same limited exploration time and extension parameters.



(a) Conventional RRT with linear segment (b) Proposed RRT with circular segment

Fig. 1. The different extend operations.

The rest of this paper is organized as follows. In the following section, kinematics of nonholonomic robots is presented. Two ideas to improve the sampling-based planners are formulated and presented in the Section III. Experimental results are provided in the Section IV to verify the efficiency of the proposed path planners. Finally, concluding remarks are given in Section V.

II. Kinematics of Nonholonomic Robots

A. A Hybrid Map of Nonholonomic Robots

For simplicity, let us take nonholonomic robots as an example. Let $s(t) = \left[\begin{array}{ccc} x(t) & y(t) & \theta(t) \end{array}\right]^T$ be the state and $S \subset R^3$ be the state space, $u(t) = \left[\begin{array}{cc} v(t) & \omega(t) \end{array}\right]^T$ be the control input and $U \subset R^2$ the control space, $S_{obs} \subset S$ the obstacle space, and $S_{free} = S - S_{obs}$ the free-state space. A system on the state space S is given as [11]

$$\left[\begin{array}{c} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{array}\right] = \left[\begin{array}{cc} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{array}\right] \left[\begin{array}{c} v(t) \\ \omega(t) \end{array}\right] \; ; \; s(0) = S_{start}, \quad (1)$$

where $s_{start} \in S$, and, for all t, $s(t) \in S$, and $u(t) \in U$.

A hybrid map is built up and it is composed of a topological map and a group of metric maps. As described above, the topological map only considers places and relations between them. The map is then a graph, in which the nodes correspond to places and edges correspond to the paths. The length and width as well as the starting and ending configurations of each sub-path are stored. A logical path will be firstly found from the starting configuration to the goal one. As such, all the starting and ending configurations of the narrow passages in the logical path will be utilized to divide the whole workspace into segments using edges. Here, narrow passages are small regions whose removal changes the connectivity of S_{free}: they may change the way different regions are connected, or even change the number of connected components [6]. The edge, which corresponds to each narrow passage, is helpful to reduce the complexity of sampling-based path planners. Therefore, besides the continuous state $s(t)$, a discrete state $m(t)$ is adopted to denote the sub-paths which the robot is located at time t [7]. The continuous state $s(t)$ is the relative configuration with respect to the starting configuration of each sub-path rather than the initial configuration. Let t_k denote the kth switching instance, t_k^- and t_k^+ are the time instance just before and after the time t_k. The discrete state $m(t)$ satisfies that

$$m(t_k^-) \neq m(t_k^+). \quad (2)$$

For simplicity, suppose that $m(t_k)$ corresponds to a narrow passage. Its two configurations are represented by $s(t_k^+)$ and $s(t_{k+1}^+)$.

B. Linking Modes of Nonholonomic Robots

Consider an interval $[t_j, t_j + \Delta T]$ and assume that the values of $v(t)$ and $\omega(t)$ are constant in the interval. In other words, the robot continues its movement in the interval. If the value of ω_j is not 0, the trajectory of the robot in the interval is then given as

$$\theta_p(t) = \theta(t_j) + \omega_j(t - t_j), \quad (3)$$

$$x_p(t) = x_c(t_j) + \frac{v_j}{\omega_j}\sin\theta_p(t), \quad (4)$$

$$y_p(t) = y_c(t_j) - \frac{v_j}{\omega_j}\cos\theta_p(t), \quad (5)$$

where the center $(x_c(t_j), y_c(t_j))$ is

$$x_c(t_j) = x(t_j) - \frac{v_j}{\omega_j}\sin\theta(t_j), \quad (6)$$

$$y_c(t_j) = y(t_j) + \frac{v_j}{\omega_j}\cos\theta(t_j). \quad (7)$$

Otherwise, the trajectory of the robot in the interval is

$$\theta_p(t) = \theta(t_j), \quad (8)$$

$$x_p(t) = x_c(t_j) + v_j\cos\theta(t_j)(t - t_j), \quad (9)$$

$$y_p(t) = y_c(t_j) + v_j\sin\theta(t_j)(t - t_j). \quad (10)$$

Clearly, the robot will follow a straight line if the value of ω_j is 0. Otherwise, it follows a circle with the center as $(x_c(t_j), y_c(t_j))$ and the radius as v_j/ω_j. This implies that possible modes for the path planning of nonholonomic robots include: 1) a straight line; and 2) circular arcs. They are defined as follows:

The straight line mode: Given two different points $\left[\begin{array}{cc} x_i & y_i \end{array}\right]^T$ and $\left[\begin{array}{cc} x_{i+1} & y_{i+1} \end{array}\right]^T$, the corresponding line segment is represented as

$$\left[\begin{array}{c} x \\ y \end{array}\right] = \left[\begin{array}{c} x_i \\ y_i \end{array}\right] + \tau \left[\begin{array}{c} x_{i+1} - x_i \\ y_{i+1} - y_i \end{array}\right], \quad (11)$$

where $0 \leq \tau \leq 1$.

The circle modes: Given two different points $\left[\begin{array}{cc} x_i & y_i \end{array}\right]^T$ and $\left[\begin{array}{cc} x_{i+1} & y_{i+1} \end{array}\right]^T$, there are two different circle modes to link them. One is the arc above the line segment and the other is the arc below the line segment. The radiuses of the two arcs have the same value, which can be computed by

$$\rho = \alpha\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}, \quad (12)$$

where $\alpha (\geq 0.5)$ is a constant.

The arc above the line segment is represented by

$$\left[\begin{array}{c} x \\ y \end{array}\right] = \left[\begin{array}{c} x_{c,a} \\ y_{c,a} \end{array}\right] + \rho \left[\begin{array}{c} \sin\theta \\ \cos\theta \end{array}\right], \quad (13)$$

while the arc below the line segment is represented by

$$\left[\begin{array}{c} x \\ y \end{array}\right] = \left[\begin{array}{c} x_{c,b} \\ y_{c,b} \end{array}\right] - \rho \left[\begin{array}{c} \sin\theta \\ \cos\theta \end{array}\right], \quad (14)$$

The angle θ of both the arcs is in the same range of $[L, U]$. The values of the centers $(x_{c,a}, y_{c,a})$ and $(x_{c,b}, y_{c,b})$ as well as L and U are given as in the following two cases:

1) When the value of α is equal to 0.5. The centers $(x_{c,a}, y_{c,a})$ and $(x_{c,b}, y_{c,b})$ are the same and they are defined as

$$\left[\begin{array}{c} x_{c,a} \\ y_{c,a} \end{array}\right] = \left[\begin{array}{c} x_{c,b} \\ y_{c,b} \end{array}\right] = \left[\begin{array}{c} \frac{x_i + x_{i+1}}{2} \\ \frac{y_i + y_{i+1}}{2} \end{array}\right]. \quad (15)$$

The values of L and U are computed as

$$L = \left\{\begin{array}{ll} \theta_i - \pi; & \text{if } \theta_i > 0 \\ \theta_i; & \text{otherwise} \end{array}\right. , \quad (16)$$

$$U = \left\{\begin{array}{ll} \theta_i; & \text{if } \theta_i > 0 \\ \theta_i + \pi; & \text{otherwise} \end{array}\right. , \quad (17)$$

where the value of θ_i is $\arctan\frac{x_{i+1} - x_i}{y_{i+1} - y_i}$.

2) When the value of α is larger than 0.5. The center $(x_{c,a}, y_{c,a})$ is

$$\begin{bmatrix} x_{c,a} \\ y_{c,a} \end{bmatrix} = \begin{bmatrix} \frac{x_i+x_{i+1}}{2} \\ \frac{y_i+y_{i+1}}{2} \end{bmatrix} + \eta\sqrt{\alpha^2-0.25}\begin{bmatrix} y_{i+1}-y_i \\ x_i-x_{i+1} \end{bmatrix}, \tag{18}$$

where the value of η is defined as

$$\eta = \begin{cases} 1; & \text{if } x_{i+1} \geq x_i \\ -1; & \text{otherwise} \end{cases}. \tag{19}$$

The center $(x_{c,b}, y_{c,b})$ is

$$\begin{bmatrix} x_{c,b} \\ y_{c,b} \end{bmatrix} = \begin{bmatrix} \frac{x_i+x_{i+1}}{2} \\ \frac{y_i+y_{i+1}}{2} \end{bmatrix} - \eta\sqrt{\alpha^2-0.25}\begin{bmatrix} y_{i+1}-y_i \\ x_i-x_{i+1} \end{bmatrix}. \tag{20}$$

The values of L and U are calculated as

$$L = \begin{cases} \min\{\theta_i, \theta_{i+1}\}; & \text{if } |\theta_i - \theta_{i+1}| < \pi \\ \max\{\theta_i, \theta_{i+1}\}; & \text{otherwise} \end{cases}, \tag{21}$$

$$U = \begin{cases} \max\{\theta_i, \theta_{i+1}\}; & \text{if } |\theta_i - \theta_{i+1}| < \pi \\ \min\{\theta_i, \theta_{i+1}\} + 2\pi; & \text{otherwise} \end{cases}, \tag{22}$$

where the values of θ_i and θ_{i+1} are determined by the pixels $\begin{bmatrix} x_i & y_i \end{bmatrix}^T$ and $\begin{bmatrix} x_{i+1} & y_{i+1} \end{bmatrix}^T$, and the corresponding two arcs, respectively. Figure 2 illustrates those extendable paths between two vertices with different α values from 0.5 to 1.5, including a linear segment shown in red color (only the arcs above the line are drawn).

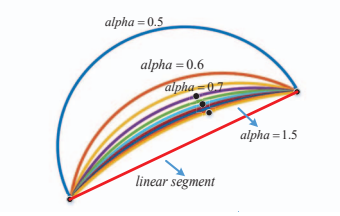


Fig. 2. Multi-mode RRT extension between two vertices.

III. Hierarchical RRT with Multiple Linking Modes

The problem studied in this section is formulated as follows:

Given an initial configuration s_{start} and a goal configuration s_{goal}. Find a path $s : [0,T] \rightarrow S_{free}$ such that $s(0) = s_{start} \in S_{free}$, $s(t) \in S_{free}$ for all $t \in [0,T]$, and $s(T) = s_{goal} \in S_{free}$.

In this section, a multi-mode RRT is first proposed to address the above problem. Two new circle-based modes are introduced to the planner so as to increase the possibility of avoiding obstacles. A hierarchical RRT is then proposed to address the cases in which narrow passages exist in the path from the initial configuration to the goal one.

A. Multi-mode RRT

The proposed algorithm given below differs subtly from the standard RRT algorithm. Additional extendable options are added to "Extend" function in *line 10* (collision detection) and *line 12* (path extension), endowing the new planner more flexibility during bypassing the obstacles. In our design, the straight line mode and two circle modes will be adopted to generate the path substitutively according to the environment. Specifically speaking, considering the fact that between two distinct vertex the straight line is the shortest, intuitively, the straight line mode is selected if it can be adopted to avoid the obstacles, otherwise the circle modes will be used. If both the linear and circular modes fail, the algorithm will choose another sampled node and repeat the "Extend" process to explore the workspace. The algorithm is presented as below.

Algorithm 1: Multi-mode RRT

main()
1. $V \leftarrow \{s_{start}\}; E \leftarrow \emptyset; i \leftarrow 0;$
2. While $i < N$ do
3. $\quad G \leftarrow (V, E);$
4. $\quad s_{rand} \leftarrow$ Sample$(i); i \leftarrow i+1;$ *Sampling*
5. $\quad (V, E) \leftarrow$ Extend$(G, s_{rand});$
6. end

Extend(G, s)
7. $V' \leftarrow V; E' \leftarrow E;$
8. $s_{nearest} \leftarrow$ Nearest$(G, s);$ *Nearest neighbor*
9. $s_{new} \leftarrow$ Steer$(s_{nearest}, s);$ *Steering*
10. if ObstacleFree$(s_{nearest}, s_{new}, j)$ then *Collision detection*
11. $\quad V' \leftarrow V' \cup \{s_{new}\};$
12. $\quad E' \leftarrow E' \cup \{(s_{nearest}, s_{new}, j)\};$
13. return $G' = (V', E')$.

All the functions in the above algorithm are elaborated as follows:

Sampling: The function Sample(i) returns an independent identically distributed sample from S_{free}.

Nearest neighbor: Given a graph $G = (V, E)$ and a point $s \in S_{free}$, the function Nearest(G, s) returns a vertex $v \in V$ that is closest to s in terms of a given distance function.

Steering: Given two points $s, s' \in S$, the function Steer(s, s') returns a point which is closer to the point s' than the point s.

Collision detection: Given two points $s, s' \in S_{free}$, the Boolean function ObstacleFree(s, s', j) returns True iff the jth mode segment between s and s' lies in S_{free}. Here, the value of j is 1, 2 or 3. As shown in Figure 1, ① stands for the line segment, ② is the circle segment above the linear path, and ③ is the circle segment below the linear path. The detection will start from the line mode and follow by the circle modes. The line mode will be chosen if there is no obstacle in the line segment between s and s'.

Compared with the original RRT, new element is introduced to each edge in the tree. The element indicates how a node and its parent are connected. For simplicity, only three modes are implemented in the current version. The same idea can be easily generated to the case that there are more than three modes.

B. Hierarchical RRT

Using the topology map, a logical path can be built up from the initial discrete state $m(0)$ to the goal discrete state $m(T)$. All the narrow passages included in the logical path are extracted and ordered according to their spatial location in the logical path. For simplicity, in our simulations, it is assumed that there are only two narrow passages in the logical path and their discrete dates are $m(t_{i_1})$ and $m(t_{i_2})$. Their starting and ending configurations are $s(t_{i_1}^+)$ and $s(t_{i_1+1}^+)$ as well as $s(t_{i_2}^+)$ and $s(t_{i_2+1}^+)$, respectively. Instead of finding a path from $s(0)$ to $s(T)$ as in the existing RRT, the path from $s(0)$ to $s(T)$ is divided into the following five segments:

$$s(0) \rightarrow s(t_{i_1}^+) \rightarrow s(t_{i_1+1}^+) \rightarrow s(t_{i_2}^+) \rightarrow s(t_{i_2+1}^+) \rightarrow S(T). \tag{23}$$

With the proposed hierarchical strategy, the conventional wisdom of "divide and conquer" can be adopted to reduce the complexity of RRT in presence of narrow passages. The proposed multi-mode RRT illustrated below runs independently in each of the above five segments to obtain a sub-path in the corresponding segment. The overall path is obtained by concatenating all the five sub-paths.

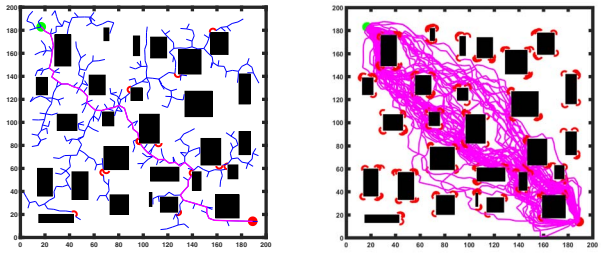
Algorithm 2: Hierarchical RRT

1. Find a logical path from $m(0)$ to $m(T)$;
2. Identify all narrow passages in the logical path;
3. Divide the path from $s(0)$ to $s(T)$ into segments;
4. Run the multi-mode RRT in each segment independently;
5. Concatenate all the sub-paths to form the path from $s(0)$ to $s(T)$.

To be noted, if there is no narrow passage in the logical path, the path from $S(0)$ to $S(T)$ is only composed of one segment. In addition, the proposed hierarchical RRT can be easily combined with the circular linking modes by employing the multi-mode RRT in each of the sub-path generation.

IV. Experimental Results

In this section, the efficiency of the proposed multi-mode RRT is first tested. The value of α is selected as 0.5 in this section. The step size is 5, the distance threshold is 5, the maximum iteration is 5000 and the goal steering probability is 0.1. Twelve maps with different layouts (random, maze, narrow-passage) in Figure 3 to Figure 5 are investigated. The created maps have the size of 200*200. The green dot indicates the starting point, while the red one denotes the goal position. 50 path planning trails using the original RRT and the proposed multi-mode RRT have been tested to compare the performance. The red arcs shown in the following figures (e.g. Figure 3(b)) represent the circular paths generated by the multi-mode RRT. It can be easily observed that those arcs are frequently used around the corners of each obstacles, resulting in a more efficient path planner to avoid the obstacles.



(a) Multi-mode result on Map 1 (b) Multi-mode results on Map 1 (50 trials)

Fig. 3. Random map simulation.

Case 1. Four different random-obstacle maps in Figure 3 (only Map1 is shown) are tested. Subfigure (a) and (b) illustrate the path planning performance for a single simulation and the 50 trials (the same idea for the following Figure 4 and 5). The corresponding exploring performances are presented in Table I. T_c and T_m are the average planning time of the original RRT and multi-mode one, respectively. The improvement is computed as $100 * (T_c - T_m)/T_c$. The same idea with the planning iterations columns. It can be shown that the planning time is reduced by up to 22.1% and the number of iterations is also reduced by the proposed multi-mode RRT.

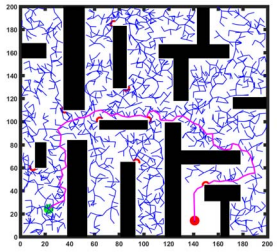
Case 2. Four different maze-like maps in Figure 4 are tested. The maximal iterations of the former three maps are 5000 while it is 10000 for the last one (marked by "$*$" on the map number in Table I). The corresponding exploring performances are presented in Table I. It can be shown that both the planning time and the number of iterations are reduced while the success rate is improved by the proposed multi-mode RRT given the same planning constraint of maximum iteration.

Case 3. In addition, four different maps with narrow passages are tested and they are given in Figure 5. The maximal iterations of the two maps on the left are 5000 while they are 10000 for the maps on the right (marked by "$*$" in Table I). The corresponding exploring performances are presented in Table I. It can be shown that both the planning time and the number of iterations are reduced while the success rate is improved by the proposed multi-mode RRT. It is worth noting that the gain of this case is reduced compared with those of the former two cases, because of the narrow passages.

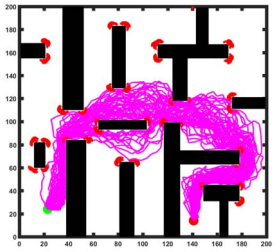
The efficiency of the proposed hierarchical strategy is then evaluated. Four other narrow-passage maps with similar layout in Figure 6 are investigated first. The blue dots indicate the starting (left) or ending (right) nodes for each sub-map within the whole topology map. The exploration performances using two approaches are shown in Figure 6(a)-(b). It can be observed that the sampling space of the proposed hierarchical RRT is much smaller than the conventional one on the left-hand-side of the obstacle "wall", resulting in a significant lower computational burden for the planner. The corresponding exploring performances are presented in Table II, where T_h is the average planning time of the hierarchical RRT. The improvement is computed as $100 * (T_c - T_h)/T_c$. It can

TABLE I
PLANNING EFFICIENCY COMPARISON BETWEEN THE CONVENTIONAL AND MULTI-MODE RRT IN DIFFERENT ENVIRONMENTS

Planning environments	Random map				Maze map				Narrow-passage map			
Map number	1	2	3	4	1	2	3	4*	1	2*	3	4*
Planning time $\mathbf{T_c}$ (s)	0.567	0.636	0.816	0.261	2.959	3.412	3.068	28.542	4.297	44.210	0.417	27.491
Planning time $\mathbf{T_m}$ (s)	0.481	0.566	0.636	0.242	2.365	3.019	2.775	26.651	3.464	42.205	0.375	26.424
Improvement (%)	15.2	11.1	22.1	7.3	20.1	11.5	9.5	6.6	19.4	4.5	10.0	3.9
Planning iteration $\mathbf{I_c}$	647	882	1033	366	2186	2732	3042	9355	2962	9113	2050	9197
Planning iteration $\mathbf{I_m}$	554	733	847	315	1892	2558	2782	8922	2704	9022	1686	8913
Improvement (%)	14.4	10.9	18.0	13.9	13.4	6.4	8.6	4.6	8.7	1.0	17.8	3.1
Success rate $\mathbf{S_c}$	1	1	1	1	1	1	0.98	0.63	0.81	0.56	0.58	0.46
Success rate $\mathbf{S_m}$	1	1	1	1	1	1	1	0.83	0.96	0.57	0.66	0.69
Improvement (%)	0	0	0	0	0	0	2	20	15	1	8	23

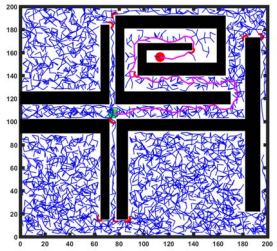


(a) Multi-mode result on Map 1

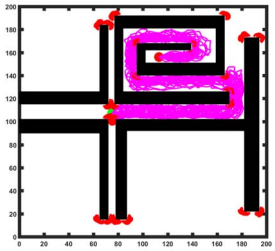


(b) Multi-mode results on Map 1 (50 trials)

Fig. 4. Maze map simulation.



(a) Multi-mode result on Map 4

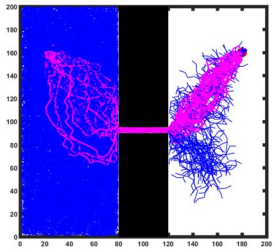


(b) Multi-mode results on Map 4 (50 trials)

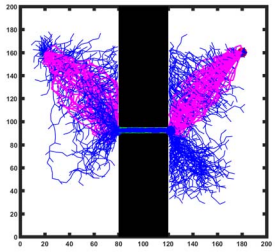
Fig. 5. Narrow-passage map simulation.

be shown that both the planning time and the number of iterations of the conventional RRT increase significantly when the passage becomes narrower while they are almost the same and keeping small values for the proposed hierarchical RRT.

Two different maps in Figure 7 (Map 5 and Map 6) are further studied to compare the proposed hierarchical RRT with the conventional one. The experimental settings are the same as before, except the maximum iteration is now 10000 (marked by the "$*$" on map number 5 and 6 in Table II). Again, a higher exploration efficiency with less sampled edges can be observed from the comparison using the hierarchical RRT. The corresponding exploring performances are presented in Table II as well. It can be shown that both the average planning time and the number of iterations are significantly reduced by up to 96.6% and 95.4% using the proposed hierarchical RRT,



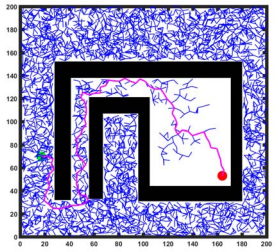
(a) Conventional RRT result on Map 4 (50 trials)



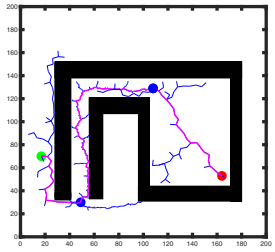
(b) Hierarchical RRT result on Map 4 (50 trials)

Fig. 6. Performance comparison on the narrow-passage map.

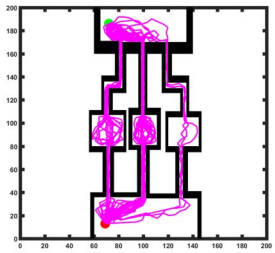
respectively.



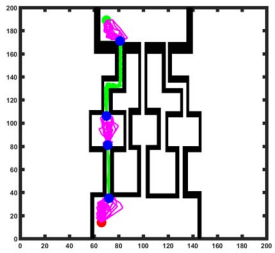
(a) Conventional RRT result on the additional Map 5



(b) Hierarchical RRT result on the additional Map 5



(c) Conventional RRT result on the additional Map 6 (50 trials)



(d) Hierarchical RRT result on the additional Map 6 (50 trials)

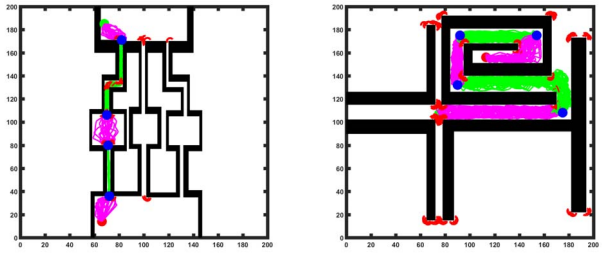
Fig. 7. Two additional maps with narrow passages.

Finally, the synergic hierarchical RRT with multiple linking modes is tested in the Figure 8. The path planning per-

TABLE II
PLANNING EFFICIENCY COMPARISON BETWEEN THE CONVENTIONAL AND HIERARCHICAL RRT IN NARROW-PASSAGE ENVIRONMENTS

Map number	1	2	3	4	5*	6*
Planning time $\mathbf{T_c}$ (s)	0.180	0.249	0.532	3.772	20.934	1.814
Planning time $\mathbf{T_h}$ (s)	0.121	0.131	0.128	0.127	0.816	0.372
Improvement (%)	32.7	47.4	75.9	96.6	96.1	79.5
Planning iteration $\mathbf{I_c}$	311	482	829	2124	6402	3942
Planning iteration $\mathbf{I_h}$	113	406	109	97	588	676
Improvement (%)	63.6	15.8	86.9	95.4	90.8	82.9

formances of the conventional, multi-mode, hierarchial and the synergic RRT are compared and shown in Table III, where T_{hm} and I_{hm} denote the average planning time and iteration of the synergic RRT, respectively. According to the table, the proposed synergic algorithm archives better planning performances than the other three comparing groups and is able to reduce the planning time by up to 93.2% and the number of iterations by up to 91.7% relative to the standard RRT in these two cases.



(a) Map 1 (b) Map 2

Fig. 8. Synergic RRT performances on two narrow-passage maps.

V. CONCLUSION REMARKS AND DISCUSSIONS

Two novel ideas have been introduced to improve sampling-based path planners. One is the multiple linking modes for sampling-based path planners. Besides the straight line mode used in the existing random exploring, two new circle modes are introduced. The other is a hierarchical strategy and it can be adopted to reduce the complexity of sampling-based path planners in presence of narrow passages significantly. It has been also verified that the combination of the two modifications of sampling-based planner is able to achieve an even better performance. The rapidly-exploring random tree (RRT) is selected as an example to illustrate the efficiency of the proposed new modes. More new modes can be further introduced using continuous curves including Clothoids, Bezier curve, B-spline, logistic curves, as well as those curves derived using control theory [12], [13]. All the new modes can also be incorporated into other sampling-based planners to enhance the efficiency.

TABLE III
PLANNING EFFICIENCY COMPARISON BETWEEN THE FOUR RRT VARIANTS ON THE TWO MAPS IN FIGURE 8

Map number	1	2
Planning time $\mathbf{T_c}$ (s)	1.814	27.491
Planning time $\mathbf{T_m}$ (s)	1.020	26.424
Planning time $\mathbf{T_h}$ (s)	0.372	3.033
Planning time $\mathbf{T_{hm}}$ (s)	0.124	2.102
Improvement (%)	93.2	92.4
Planning iteration $\mathbf{I_c}$	3942	9197
Planning iteration $\mathbf{I_m}$	2273	8913
Planning iteration $\mathbf{I_h}$	676	4799
Planning iteration $\mathbf{I_{hm}}$	328	2791
Improvement (%)	91.7	69.7

REFERENCES

[1] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," International Journal of Robotics Research, vol. 20, no. 5, pp. 378-400, May 2001.

[2] L. Palmieri, S. Koening, and K. O. Arras, "RRT-based nonholonomic planning using any-angle path biasing," in *2016 IEEE International Conference on Robotics and Automation*, Sweden, May 2016.

[3] S. Choudhury, J. D. Gammell, T. D. Barfoot, S. S. Srinivasa, and S. Scherer, "Regionally accelerated batch informed trees (RABIT*): a framework to integrate local information into optimal path planning," in *2016 IEEE International Conference on Robotics and Automation*, Sweden, May 2016.

[4] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Transactions on Systems Science and Cybernetics SCC, vol. 4, no. 2, pp.100-107, Apr. 1968.

[5] K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," Journal of Artificial Intelligence Research, vol. 39, no. 1, 2010.

[6] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. H. Reif, "Narrow passage sampling for probabilistic roadmap planning," IEEE Trans. on Robotics, vol. 21, no. 6, pp. 1105-1115, Dec. 2005.

[7] S. Tully, H. Moon, D. Morales, G. Kantor, and H. Choset, "Hybrid localization using the hierarchical Atlas," in IEEE/RSJ International Conference on Intelligent Robots and Systems, USA, Nov. 2007.

[8] Z. G. Li, Y. C. Soh, and C. Y. Wen, "Switched and impulsive systems: analysis, design and application," Springer-Verlag, April, 2005, ISBN: 3-540-23952-9.

[9] S. Upadhyay and A. Ratnoo, "Continuous-curvature path planning with obstacle using four parameter logistic curves," in *Proc. 2016 IEEE International Conference on Robotics and Automation*, Sweden, May 2016.

[10] T. Fraichard and A. Scheuer, "From reeds and shepps to continuous curvature paths," IEEE Transactions on Robotics, vol. 20, no. 6, pp.1025-1035, Dec. 2004.

[11] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robotics & Automation Magzine, vol. 4, no. 1, pp. 23-33, Mar. 1997.

[12] N. Ratliff, M. Zucker, J. Andrew Bagnell, and S. Srinivasa, "CHOMP: gradient optimization techniques for efficient motion planning," in *IEEE International Conference on Robotics and Automation*, pp. 489494, Japan, May 2009.

[13] J. Dong, M. Mukadam, F. Dellaert, and B. Boots, "Motion planning as probabilistic inference using Gaussian processes and factor graphs," in *Robotics: Science and Systems*, Jun. 2016, USA.

[14] K. Gochev, A. Safonova, and M. Likhachev, "Planning with adaptive dimensionality for mobile manipulation," in *IEEE International Conference on Robotics and Automation*, pp. 2944-2951, USA, May 2012.

[15] J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for proximity and collision queries," in *IEEE International Conference on Robotics and Automation*, USA, May 2012.

[16] S. Murray, W. Floyd-Jones, Y. Qi, D. Sorin, and G. Konidaris, 'Robot motion planning on a chip," in *Robotics: Science and Systems*, Jun. 2016, USA.