GMR-RRT*: Sampling-Based Path Planning Using Gaussian Mixture Regression

Jiankun Wang, Tingguang Li, Baopu Li, and Max Q.-H. Meng, *Fellow, IEEE*

***Abstract*—Mobile robot autonomous path planning is an essential factor for its wide deployment in real-world applications. Conventional sampling-based algorithms have gained tremendous success in the path planning field, but they usually take much time to find the optimal solution so that the planning quality (evaluated with time cost and path length) cannot be guaranteed. In this paper, based on Gaussian Mixture Regression (GMR) and the family of Rapidly-exploring Random Tree (RRT) schemes, we propose the GMR-RRT* algorithm to achieve fast path planning for mobile robots. The proposed GMR-RRT* consists of learning navigation behaviors from human demonstrations and planning a high-quality path for the robot. Using the GMR, the key features of human demonstrations are captured to form a probability density distribution of the human trajectory in the current environment. This distribution is further utilized to guide the RRT scheme's sampling process to generate a feasible path in the current environment quickly. We test the proposed GMR-RRT* in different environments, comparing it with three state-of-the-art sampling-based algorithms. The experimental results demonstrate that the GMR-RRT* algorithm can achieve better performance in terms of time cost, memory usage, and path length.**

***Index Terms*—Mobile robot, learning from human demonstrations, sampling-based path planning.**

I. Introduction

ROBOTS have been widely used in our daily life, and many researchers have contributed a lot to the studies of mobile robots. Among them, robot path planning is a fundamental topic [1], which aims to plan a feasible path for robots in current environments. Many algorithms have been proposed to address the path planning algorithm, such as the well-established A* algorithms [2], Artificial Potential Field (APF) algorithms [3], and sampling-based algorithms [4], [5]. The A* algorithms perform poorly as the problem scale increases, and the APF algorithms often end up in a local minimum. The sampling-based algorithms have become popular due to their capability of efficiently searching the state space. However, one of the limitations is that they usually require much time to find the optimal solution. In real-world applications, such as autonomous vehicles, it is essential to compute a high-quality path to save limited power.

The motivation of this paper is to achieve fast and high-quality path planning based on the current research on path planning and machine learning. On the one hand, for recent path planning work, sampling-based algorithms such as Probabilistic Road Map (PRM) [4] and Rapidly-exploring Random Tree (RRT) [5] have been widely used in daily life and gained great success. Many of their variants have also been proposed for different applications [6]–[8]. However, the sampling-based algorithm suffers sensitivity to the sampling approach, and its performance fluctuates significantly in different environments. On the other hand, for the latest machine learning work, a probabilistic method, namely learning from demonstrations (LfD) [9], is used to generate robot motions from demonstrated trajectories. This method utilizes the Gaussian Mixture Regression (GMR) [10] to infer robot motions from the human demonstrated trajectories for its efficient implementation and easy combination with probabilistic sampling-based planners. However, the predicted motion cannot be directly taken for the robot path planning because it does not consider the geometric constraints (do not collide with the obstacles) nor kinodynamic constraints (robot attributes). As shown in Fig. 1(b), the predicted trajectory collides with the obstacle. However, the predicted trajectory can provide proper guidance for the sampling process, which happens in sampling-based algorithms.

Inspired by the LfD technology, in this paper, we present the GMR-RRT* algorithm based on the RRT* [11] (an advanced version of the RRT) and the GMR. It is a novel sampling-based path planning algorithm that empowers mobile robots to learn navigation behavior from human demonstrated reference trajectories. The collected reference trajectories, which consist of temporal and spatial information, are modeled by the Gaussian Mixture Model (GMM), as shown in Fig. 1(a). Using the GMR, a predicted trajectory represented by the mean and corresponding

Manuscript received 9 June 2021; revised 2 November 2021; accepted 7 February 2022. Date of publication 14 February 2022; date of current version 24 October 2022. This work was supported in part by the Shenzhen Key Laboratory of Robotics Perception and Intelligence under Grant ZDSYS20200810171800001 and in part by the National Natural Science Foundation of China under Grant 62103181. *(Corresponding authors: Baopu Li; Max Q.-H. Meng.)*

Jiankun Wang is with the Shenzhen Key Laboratory of Robotics Perception and Intelligence, Shenzhen 518055 China, and also with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: wangjk@sustech.edu.cn).

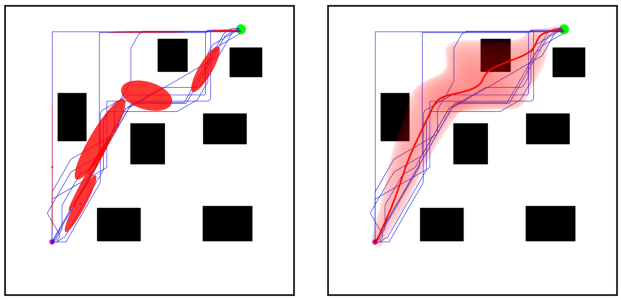
Tingguang Li is with Tencent Robotics X Lab, Shenzhen 518000, China (e-mail: tgli0809@gmail.com).

Baopu Li is with Baidu Research (US), Sunnyvale, CA 94089 USA (e-mail: bpli.cuhk@gmail.com).

Max Q.-H. Meng is with the Shenzhen Key Laboratory of Robotics Perception and Intelligence, Shenzhen 518055, China, also with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China, and also with the Shenzhen Research Institute, Chinese University of Hong Kong, Hong Kong, on leave from the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, Hong Kong (e-mail: max.meng@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2022.3150748.

Digital Object Identifier 10.1109/TIV.2022.3150748



(a) (b)

Fig. 1. Illustration of the GMR. (a) Human demonstrated trajectories (in blue) and corresponding GMM ($K = 8$) represented as red ellipses. (b) Reproduced trajectory by the GMR. The mean of the trajectory is depicted by red line, and the corresponding covariance is depicted by light tubes around the mean, respectively.

covariance at each time step can be obtained, as shown in Fig. 1(b). The covariance reflects the dispersion of the collected human trajectories, which is essential for the robot to finish the path planning work successfully. The estimated mean and covariance are utilized to guide the sampling process to construct the RRT* tree. At each time step, the GMR-RRT* chooses a state (spatial position on the map) using the Gaussian conditioning theorem, which means that the GMR-RRT* implements a nonuniform sampling function to search the state space. Then the chosen states will be used to construct a feasible path based on the RRT* scheme. The execution process of the GMR-RRT* algorithm is illustrated in Fig. 2, and a flowchart is illustrated in Fig. 3. A series of comprehensive experiments validates that the GMR-RRT* advances the state-of-the-art performance by presenting a novel machine learning-based path planning algorithm, and it achieves better performance in static and dynamic environments.

The contributions of this paper are summarized as:

1) An efficient nonuniform sampling method based on the RRT* and the GMR;
2) A novel sampling-based path planning algorithm, namely GMR-RRT*, which can quickly find a high-quality solution;
3) A proof of the probabilistic completeness and asymptotic optimality of the proposed algorithm.

The rest of this paper is organized as follows. Section II introduces the related work, and Section III presents the details of the GMR-RRT* algorithm. Section IV describes the numerical experiments and analyzes the results. The conclusions of this paper and some future research directions are discussed in Section V.

II. Related Work

Path planning through learning from demonstrations has been extensively researched and applied. It aims to plan a feasible path for the robot by learning from human demonstrations or experiences [12], [13]. Many different methods have been proposed to achieve imitative path planning, which can be broadly classified into two categories: model-based methods and learning-based methods.

The model-based method considers modeling the robot kinematic and dynamic characteristics, the current environment, and humans to plan a feasible path for robot execution. A geometric model is often used to reflect human behavior in the current environment. For example, in [14], Chi *et al.* propose the Comfort and Collision Risk (CCR) map to model human behavior. Then a sampling-based algorithm is applied on the CCR map to achieve socially-compliant path planning. In [15], the Social Force Model (SFM) is introduced to achieve human-aware navigation in crowded environments. Later in [16], the Extended SFM is taken in proactive kinodynamic planning. Recently, in [17], Wang *et al.* utilize the potential function to model the relationship between robots and humans to achieve socially-compliant path planning. By combing membrane computing rules and potential filed methods, MemEAPF [18] and MemPBPF [19] are proposed to handle mobile robot path planning. An electrostatic potential field method [20] is proposed for mobile robot path planning in scattered environments. These model-based methods approximately model human behavior, but the robot still considers the human as a *different* obstacle. The robot cannot directly imitate human navigation behavior.

The learning-based method aims at generating trajectories that account for human behaviors by learning collected data from real demonstrations. The Inverse Reinforcement Learning (IRL) technique is often used to find a novel cost function to guide the robot path planning. In [21], Noe *et al.* use the IRL to learn RRT*'s cost function from demonstrations. In [22], Henrik *et al.* consider modeling cooperative navigation behavior of humans using the IRL technique to achieve socially-compliant mobile robot navigation. Deep Learning (DL) technique is also widely applied in this area. Mavrogiannis *et al.* [23] apply the Long Short-Term Memory (LSTM) architecture to learn multi-agent path topologies for socially competent planning. Convolutional neural network (CNN) is used in [24] to train a drone to fly in the civilian environment autonomously. Generative Adversarial Network (GAN) [25] can be also used to learn heuristics for sampling-based path planning. Meanwhile, a recurrent generative model [26] is proposed to achieve efficient heuristic generation for robot path planning. The learning-based method can help the robot directly imitate human navigation behavior, which has gained great popularity. However, the learning-based method requires complex network models and long offline training sessions.

Unlike the methods mentioned above, the GMR-RRT* employs the GMR as a learning tool to learn from human behavior with the Expectation Maximization (EM) algorithm. Then the learned distribution acts as a nonuniform sampling function to guide the sampling-based path planner to generate a feasible solution quickly. The simplicity of the GMR-RRT* allows it to be more efficient and affordable in practical applications.

III. Algorithm

In this section, we first introduce how to use the GMR to learn navigation behavior from human demonstrations. The learned behavior is then applied in the GMR-RRT* to achieve fast and

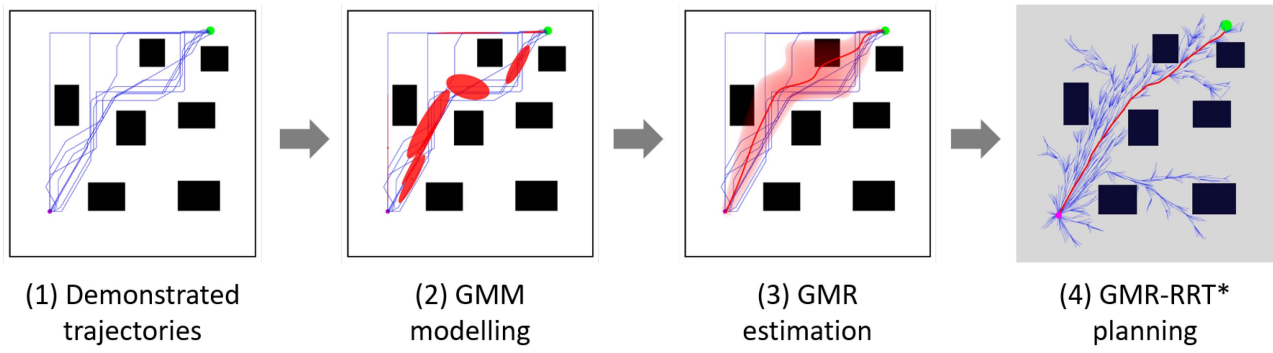


Fig. 2. Execution process of the GMR-RRT*. (a) Collection of the human demonstrated trajectories. (b) GMM modeling of the collected trajectories. (c) GMR estimation of the possible trajectory in the current environment. (4) GMR-RRT* planning with the GMR estimation as a nonuniform sampling function.

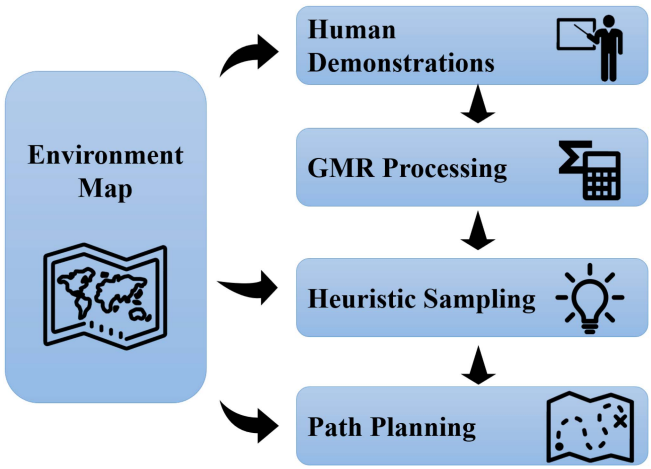


Fig. 3. Flowchart of the GMR-RRT*.

high-quality path planning. At last, we prove that the GMR-RRT* guarantees probabilistic completeness and asymptotic optimality.

A. Gaussian Mixture Regression

In this subsection, we follow the previous work to give a preliminary introduction of the GMR [9] and discuss how to adapt it to the sampling-based path planning. The GMR utilizes the Gaussian conditioning theorem to compute the distribution of output given input. Firstly, the GMM encodes the joint distribution of input and output using the EM algorithm to calculate the model parameters. Secondly, the output given the input (spatial data given temporal data in this paper) is computed through a linear combination of conditional expectations. Therefore, the GMR relies on the learned joint distribution instead of deriving the regression function directly. An illustration of the GMR calculation process is provided in Fig. 4.

For each human demonstrated trajectory, the length is rescaled to a fixed value N. The demonstrated trajectory $\xi = \{\xi_i\}_{i=1}^{N}$ is defined by N observations $\xi_i \in \mathbb{R}^3$. Each datapoint $\xi_i = \{\xi_t, \xi_s\}$ encodes time $\xi_t \in \mathbb{R}$ and spatial position $\xi_s \in \mathbb{R}^2$. All trajectories are fed into the GMM with K components, and the probability density function is defined as

$$\begin{aligned} p(\xi_i) &= \sum_{k=1}^{K} p_k p\left(\xi_i|k\right) \\ &= \sum_{k=1}^{K} p_k \mathcal{N}\left(\xi_i; \mu_k, \Sigma_k\right) \\ &= \sum_{k=1}^{K} p_k \frac{1}{\sqrt{(2\pi)^3 |\sum_k|}} e^{-\frac{1}{2}\left((\xi_i-\mu_k)\sum_k^{-1}(\xi_i-\mu_k)\right)}, \end{aligned} \tag{1}$$

where $\{p_k, \mu_k, \Sigma_k\}$ are the prior, mean, and covariance matrix of the Gaussian component k. When applying the GMR to learn the navigation behavior, the temporal value ξ_t is used as the query point, and the corresponding spatial value ξ_s is estimated through regression. Therefore, μ_k and Σ_k can be represented separately as

$$\mu_k = \{\mu_{t,k}, \mu_{s,k}\}, \ \Sigma_k = \begin{pmatrix} \Sigma_{tt,k} & \Sigma_{ts,k} \\ \Sigma_{st,k} & \Sigma_{ss,k} \end{pmatrix}. \tag{2}$$

Given ξ_t, the conditional distribution of ξ_s in each Gaussian model k is

$$\begin{aligned} p(\xi_s|\xi_t, k) &= \mathcal{N}(\xi_s; \hat{\xi}_{s,k}, \hat{\Sigma}_{ss,k}), \\ \hat{\xi}_{s,k} &= \mu_{s,k} + \Sigma_{st,k}\left(\Sigma_{tt,k}\right)^{-1}\left(\xi_t - \mu_{t,k}\right), \\ \hat{\Sigma}_{ss,k} &= \Sigma_{ss,k} - \Sigma_{st,k}\left(\Sigma_{tt,k}\right)^{-1}\Sigma_{ts,k}, \end{aligned} \tag{3}$$

where $\hat{\xi}_{s,k}$ is the conditional expectation and $\hat{\Sigma}_{ss,k}$ is the estimated conditional covariance. The complete conditional distribution of ξ_s given ξ_t is

$$p(\xi_s|\xi_t) = \sum_{k=1}^{K} \beta_k \mathcal{N}(\xi_s; \hat{\xi}_{s,k}, \hat{\Sigma}_{ss,k}), \tag{4}$$

where $\beta_k = p(k|\xi_t)$ is the so-called responsibility of the Gaussian component k

$$\beta_k = \frac{p_k p(\xi_t|k)}{\sum_{i=1}^{K} p_i p(\xi_t|i)} = \frac{p_k \mathcal{N}(\xi_t; \mu_{t,k}, \Sigma_{tt,k})}{\sum_{i=1}^{K} p_i \mathcal{N}(\xi_t; \mu_{t,i}, \Sigma_{tt,i})}. \tag{5}$$

With (3) and (5), the conditional expectation of ξ_s given ξ_t and the corresponding conditional covariance can be obtained using

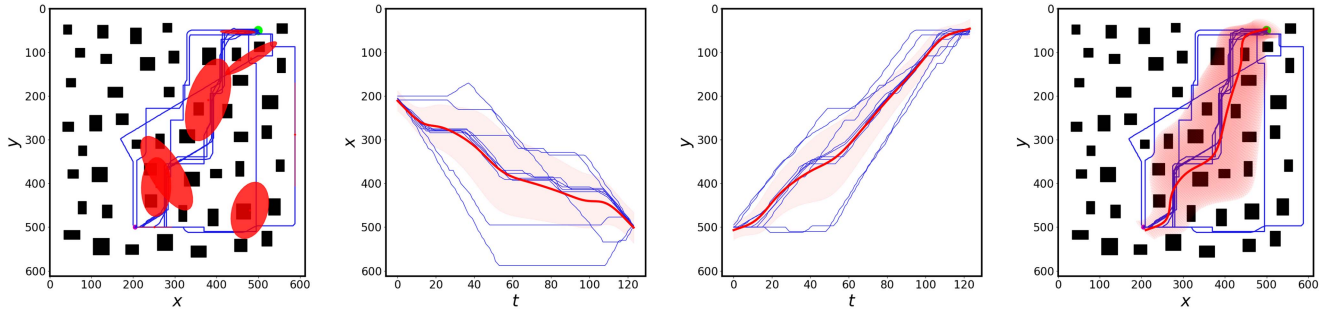


Fig. 4. Illustration of the GMR calculation process. (a) Human demonstrated trajectories (in blue) and corresponding GMM ($K = 8$) represented as red ellipses. (b), (c) Reproduced trajectories by the GMR. The spatial positions x and y, considered as outputs, are computed as a function of the time t, considered as input. The mean of the trajectory is depicted by red line, and the corresponding covariance is depicted by light tubes around the mean, respectively. (d) 2D representation of the reproduced trajectory.

the linear transformation property of Gaussian distributions

$$\hat{\xi}_s = \sum_{k=1}^{K} \beta_k \hat{\xi}_{s,k},\ \hat{\Sigma}_{ss} = \sum_{k=1}^{K} \beta_k^2 \hat{\Sigma}_{ss,k}. \tag{6}$$

By computing $\{\hat{\xi}_s, \hat{\Sigma}_{ss}\}$ at different time steps ξ_t, the expected spatial position $\hat{\xi}_s$ and associated covariance matrix $\hat{\Sigma}_{ss}$ can be obtained. In the following, we will use $\{\hat{\xi}_s, \hat{\Sigma}_{ss}\}$ as two parameters of Gaussian distribution to achieve nonuniform sampling in the path planning problem. The details can be found in Algorithm 2.

*B. GMR-RRT**

In Section III-A, we use the GMR to estimate the spatial position ξ_s of a human at different time steps ξ_t. In this subsection, the estimated spatial position ξ_s is utilized in the GMR-RRT* to guide the robot path planning, allowing the robot to imitate human behavior.

Let $\mathcal{X}$ denote the whole state space, $\mathcal{X}_{obs}$ the state space occupied by the obstacles, and $\mathcal{X}_{free} = \mathcal{X} - \mathcal{X}_{obs}$ the free space. The path planning problem aims to find a collision-free path σ connecting the start state $x_{init} \in \mathcal{X}_{free}$ and the goal region $\mathcal{G}(x_{goal}) \subseteq \mathcal{X}_{free}$, where $\mathcal{G}(x_{goal}) = \{x \big| ||x - x_{goal}|| < r\}$ is a circle centered at x_{goal} with radius r. Let Σ denote the set of all feasible paths and map each path σ to a positive number $\mathbb{R}$ with a cost function $c : \sigma \to \mathbb{R}_{\geq 0}$. The optimal path planning problem can be defined as

$$\begin{aligned} \sigma^* = &\arg\min_{\sigma \in \Sigma} c(\sigma) \\ s.t.\ &\sigma(0) = x_{init} \\ &\sigma(T) \in \mathcal{G}(x_{goal}) \\ &\sigma(t) \in \mathcal{X}_{free}(t), \forall t \in [0, T]. \end{aligned} \tag{7}$$

The cost function between two states x_1 and x_2 is defined as

$$\text{cost}(x_1, x_2) = ||x_2 - x_1||, \tag{8}$$

where $||\cdot||$ represents the Euclidean distance between two states. Therefore, the cost function of the planned path is

$$c(\sigma) = \sum_{t=0}^{T-1} \text{cost}(\sigma(t), \sigma(t+1)). \tag{9}$$

The GMR-RRT* algorithm solves the aforementioned path planning problem by maintaining a tree to cover the state space, which builds on the pipeline of RRT* scheme [11]. When the start state x_{init} and goal state x_{goal} (or a state located in the goal region $\mathcal{G}(x_{goal})$) are added to the tree as two nodes, a feasible path can be found by connecting these two nodes through edges. As shown in Algorithm 1, in the GMR-RRT*, the tree $\mathcal{T}$ is initialized with x_{init} as the root node. Then the GMR-RRT* iteratively samples random states to grow the tree until a node $x \in \mathcal{G}(x_{goal})$ is added to the tree. Otherwise, the GMR-RRT* will report *failure* when the algorithm has run for the specified $maxIter$ number.

In the sampling process, if the function **Random(1)** returns a number larger than 0.5, the uniform sampling strategy is implemented. Otherwise, the nonuniform sampling strategy incorporating the GMR is implemented. The nonuniform sampling provides a strong bias to the region that humans walk through. The feasible path probably lies in this region, but some parts of the path (Fig. 2 and Fig. 4) collides with the obstacles. Besides, this region may be occupied by moving pedestrians in dynamic environments. If we only use the nonuniform sampling strategy, sometimes it may be hard for the path planner to find a feasible path. To solve this problem, a uniform strategy is used simultaneously, which can help the path planner quickly escape the trap when the nonuniform strategy struggles to find the feasible path. Through a number of experiments, we find that the parameter of 0.5 provides the best balance between leveraging the biased sampling region and ensuring full exploration of the state space.

In the nonuniform sampling strategy, the GMR-RRT* implements the GMR to choose a random state at each iteration. Firstly, the function **Random**(N) determines ξ_t by randomly choosing a value from 1 to N, where N is the length of the demonstrated trajectory in Section III-A. According to ξ_t, the

Algorithm 1: GMR-RRT*.

Input : $x_{init}, \mathcal{G}(x_{goal}), N, \text{Map}$
Output: $\mathcal{T}$
$V \leftarrow x_{init}, E \leftarrow \emptyset, \mathcal{T} = (V, E)$;
for $i = 1...\text{maxIter}$ do
  if $\text{Random}(1) > 0.5$ then
    $x_{rand} \leftarrow \text{UniformSample}()$;
  else
    $\xi_t \leftarrow \text{Random}(N)$;
    $x_{rand} \leftarrow \text{GMRSample}(\xi_t)$;
  $x_{nearest} \leftarrow \text{Nearest}(\mathcal{T}, x_{rand})$;
  $x_{new} \leftarrow \text{Steer}(x_{nearest}, x_{rand})$;
  if $\text{ObstacleFree}(x_{nearest}, x_{new})$ then
    $\mathcal{T} = \text{Extend}(\mathcal{T}, V, E, x_{new})$;
    $\text{Rewire}(\mathcal{T}, x_{new})$;
    if $x_{new} \in G(x_{goal})$ then
      return $\mathcal{T}$;
return failure;

Algorithm 2: GMRSample().

Input : ξ_t
Output: ξ_s
$\hat{\xi}_s = 0$;
$\hat{\Sigma}_{ss} = 0$;
for $k = 1...K$ do
  $\beta_k = \frac{p_k p(\xi_t|k)}{\sum_{i=1}^{K} p(i) p(\xi_t|i)}$;
  $\hat{\xi}_{s,k} = \mu_{s,k} + \Sigma_{st,k}(\Sigma_{tt,k})^{-1}(\xi_t - \mu_{t,k})$;
  $\hat{\Sigma}_{ss,k} = \Sigma_{ss,k} - \Sigma_{st,k}(\Sigma_{tt,k})^{-1}\Sigma_{ts,k}$;
  $\hat{\xi}_s = \hat{\xi}_s + \beta_k \hat{\xi}_{s,k}$;
  $\hat{\Sigma}_{ss} = \hat{\Sigma}_{ss} + \beta_k^2 \hat{\Sigma}_{ss,k}$;
$\xi_s = \text{NormalSample}(\hat{\xi}_s, \hat{\Sigma}_{ss})$;
return ξ_s;

function **GMRSample(ξ_t)** outputs ξ_s to x_{rand}, as shown in Algorithm 2.

In Algorithm 2, we use the GMR to obtain ξ_s. As described in Section III-A, the GMR relies on the Gaussian conditioning theorem to compute the distribution of output given input. Hence, for each Gaussian component k, we first compute the conditional distribution of ξ_s (the mean $\hat{\xi}_{s,k}$ and the covariance matrix $\hat{\Sigma}_{ss,k}$) given ξ_t. Secondly, using the linear transformation property of Gaussian distributions, the complete expectation $\hat{\xi}_s$, and covariance matrix $\hat{\Sigma}_{ss}$ are obtained. Then ξ_s is computed through the Gaussian distribution with the mean $\hat{\xi}_s$ and covariance matrix $\hat{\Sigma}_s$. Finally, the value of ξ_s is transmitted to x_{rand}.

After x_{rand} is chosen, the function **Nearest($\mathcal{T}, x_{rand}$)** attempts to find the nearest node $x_{nearest}$ on the tree to connect x_{rand}. Meanwhile, x_{rand} is adjusted to x_{new} through the function **Steer($x_{nearest}, x_{rand}$)**. It is noted that when the GMR-RRT* connects x_{new} to $x_{nearest}$, the nodes located in a circle of a certain radius centered at x_{new} will be searched to select the best parent node according to the cost function. If this connection is successful, the function **Extend($\mathcal{T}, V, E, x_{new}$)** add x_{new} to the node set V and add the link of x_{new} and $x_{nearest}$ to the edge set E.

When x_{new} is added to the tree, the function **Rewire**$(\mathcal{T}, x_{new})$ will *rewire* the neighbor nodes around x_{new} if there is a path passing through x_{new} that has a lower cost than the current path. The algorithm will stop when the goal state (or a state located in the goal region $\mathcal{G}(x_{goal})$) is added to the tree, or the number of iterations reaches the threshold.

C. Probabilistic Completeness and Asymptotic Optimality

Most of RRT-based path planning algorithms are proved to be probabilistic completeness and asymptotic optimality, and the GMR-RRT* can also guarantee these two properties.

The GMR-RRT* uses both the uniform and nonuniform sampling strategies. As the number of iterations goes to infinity, the whole state space will be explored. So the following equation holds

$$\lim_{n\to\infty} \mathbb{P}(V \cap \mathcal{G}(x_{goal}) \neq \emptyset) = 1, \tag{10}$$

which means that if there is a feasible path, it must be found by the GMR-RRT*. Therefore, the probabilistic completeness is guaranteed.

The GMR-RRT* implements the same *Choose Parent* and *Rewire* strategies as the RRT*. It means that if the rewiring radius γ in *Choose Parent* and *Rewire* processes satisfies

$$\gamma > 2(1 + 1/d)^{1/d} \left(\frac{\mu(\mathcal{X}_{free})}{\xi_d} \right)^{1/d}, \tag{11}$$

where d is the dimensionality of the workspace, $\mu(\mathcal{X}_{free})$ is the Lebesgue measure of $\mathcal{X}_{free}$ and ξ_d is the volume of unit ball in current workspace. In reference to Lemma 56, 71 and 72 in [11], the following equation holds

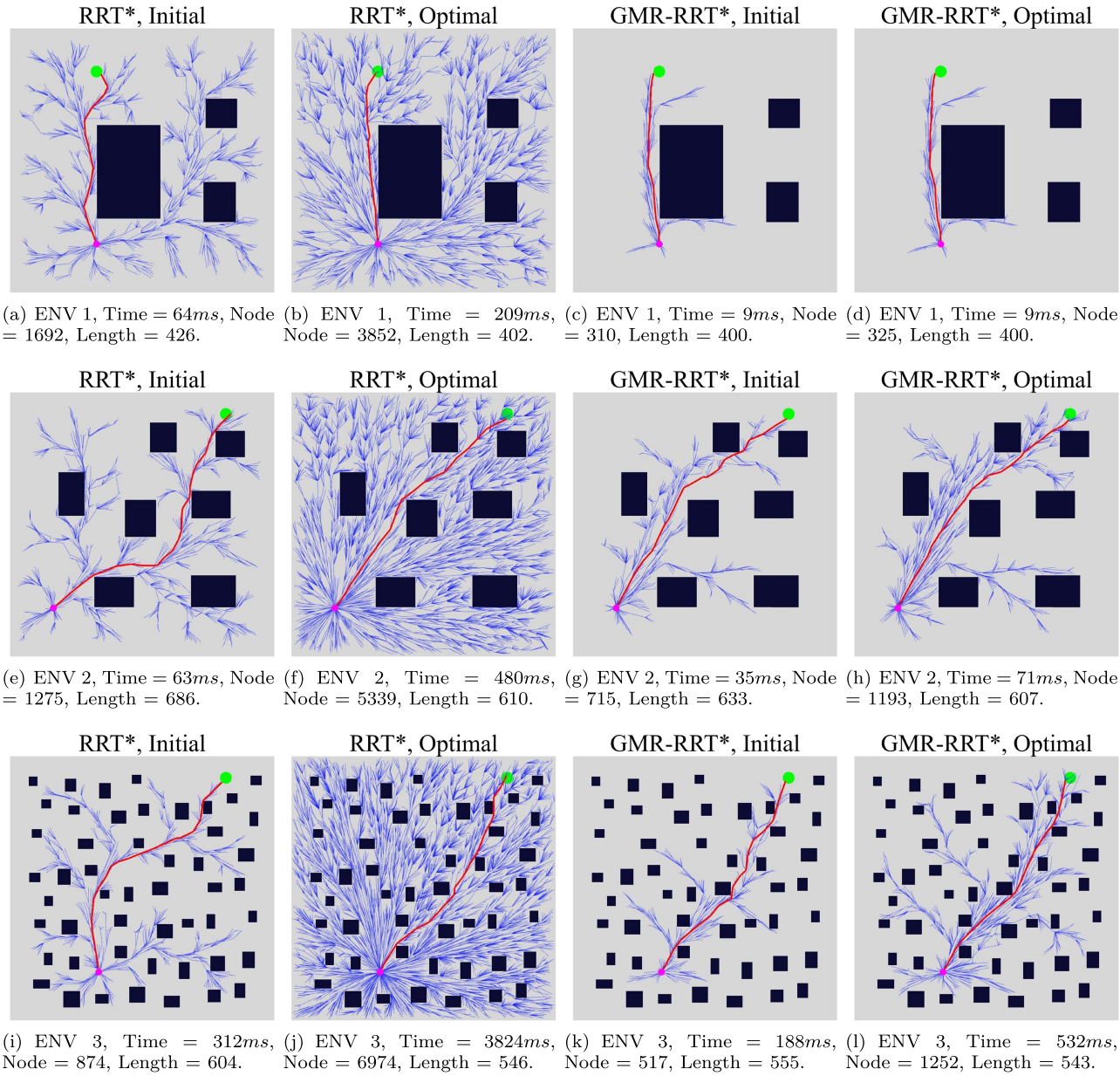
$$\lim_{n\to\infty} \mathbb{P}(c = c*) = 1, \tag{12}$$

where c refers to the cost function in Section III-B. It indicates that the GMR-RRT* can find an optimal path, if it exists, as the number of iterations go to infinity. Therefore, the asymptotic optimality is guaranteed.

IV. Experiments

In this paper, we utilize the PseudoSlam simulator [27] to collect human demonstrated trajectories. PseudoSlam is an efficient simulation platform that can simulate various mobile robot tasks. The obstacles are randomly generated in square rooms with a size of 612×612(unit: *pixel*). We invited 10 volunteers to control the robot through a keyboard to navigate the environment to the goal state and then collected the trajectories. The linear step size of the robot is 10, and the angular step size is 30°.

To evaluate the algorithm performance, five numerical experiments are conducted in static and dynamic environments to demonstrate the efficiency and extensibility of the GMR-RRT* compared with three state-of-the-art path planning algorithms including the RRT* [11], the Informed RRT* (IRRT*) [28] and the Real-time RRT* (RT-RRT*) [29]. We adopt Visual Studio



(a) ENV 1, Time = 64ms, Node = 1692, Length = 426. (b) ENV 1, Time = 209ms, Node = 3852, Length = 402. (c) ENV 1, Time = 9ms, Node = 310, Length = 400. (d) ENV 1, Time = 9ms, Node = 325, Length = 400.

(e) ENV 2, Time = 63ms, Node = 1275, Length = 686. (f) ENV 2, Time = 480ms, Node = 5339, Length = 610. (g) ENV 2, Time = 35ms, Node = 715, Length = 633. (h) ENV 2, Time = 71ms, Node = 1193, Length = 607.

(i) ENV 3, Time = 312ms, Node = 874, Length = 604. (j) ENV 3, Time = 3824ms, Node = 6974, Length = 546. (k) ENV 3, Time = 188ms, Node = 517, Length = 555. (l) ENV 3, Time = 532ms, Node = 1252, Length = 543.

Fig. 5. Illustration of the RRT* and the GMR-RRT* on finding the initial solution and converging to the optimal solution. The magenta and green disks denote the start state and goal region, respectively. The blue lines denote the tree branches, the red line denotes the generated trajectory, and the black rectangles denote the obstacles.

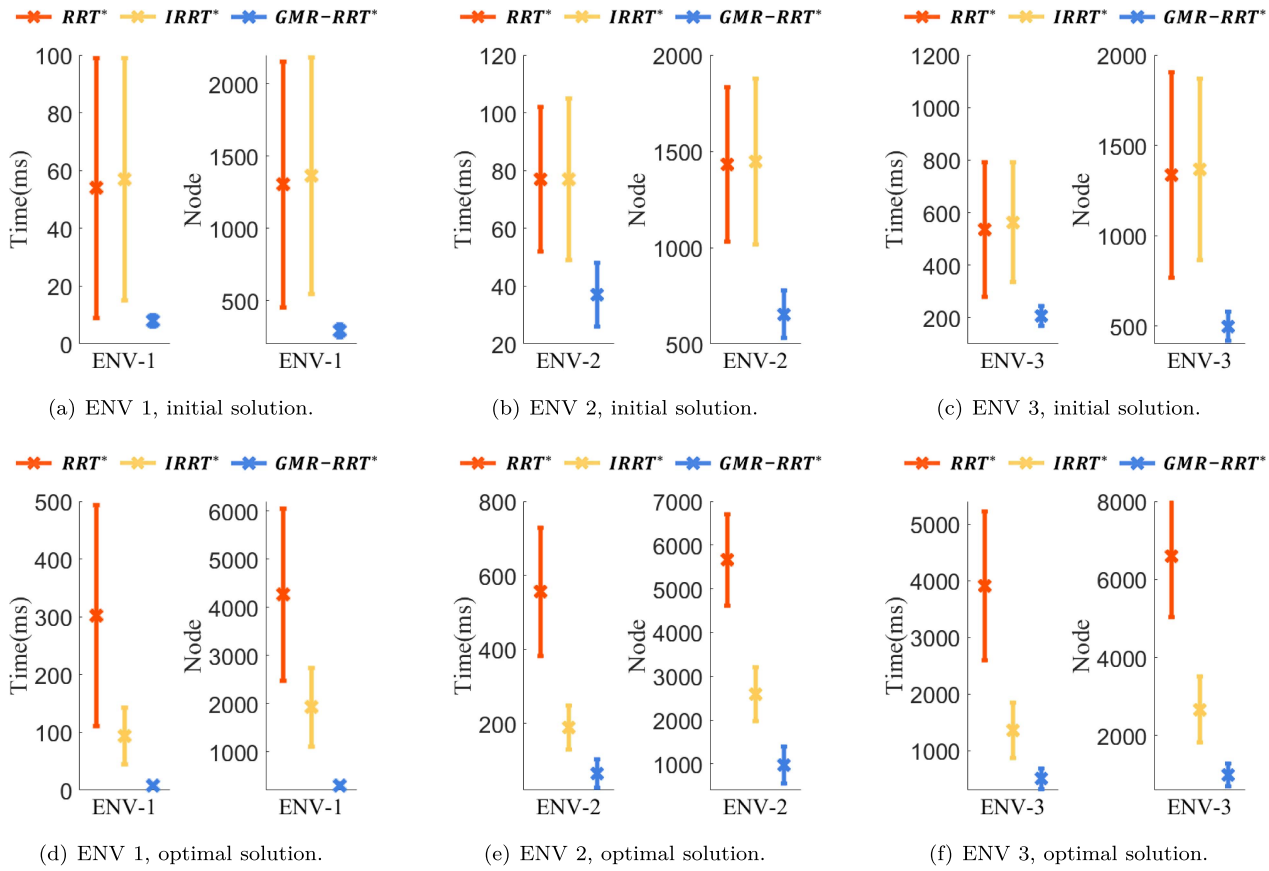
2015 as the experiment platform, which runs on an Intel i5-4590 with 8 GB RAM.

First, we compare the GMR-RRT* with the RRT* and the IRRT* in static environments to evaluate their performance on finding the initial path and converging to the optimal path. The first environment (ENV 1) consists of three simple obstacles. The second and third environments (ENV 2 and 3) are cluttered with more obstacles. A collision-free path connecting the start state and the goal region is required. Secondly, we compare the GMR-RRT* with the RT-RRT* (a real-time version of the RRT*) in the dynamic environment. The robot is required to arrive at the goal region while avoiding the static and dynamic obstacles in a real-time manner.

The parameters in the simulation experiments are set as follows (unit: *pixel*): map size 612×612, step size for growing the tree 6, the radius for rewiring the tree 35, and the radius of $\mathcal{G}(x_{goal})$ 14. For the experiments in the dynamic environment, the maximum linear velocities of the people and the robot are set to 0.5 *pixel/frame* and 1.0 *pixel/frame*, respectively. In static and dynamic environments, each algorithm is executed 30 times for a statistical result.

A. Experiments in Static Environments

Fig. 5 illustrates the performance of the RRT* and the GMR-RRT* on finding the initial solution and converging to the optimal solution. Note that the IRRT* uses the same sampling and rewiring strategies as the RRT* to find the initial solution, and it differs from the RRT* on converging to the optimal solution, where the IRRT* implements an admissible ellipsoidal heuristic.



(a) ENV 1, initial solution. (b) ENV 2, initial solution. (c) ENV 3, initial solution.

(d) ENV 1, optimal solution. (e) ENV 2, optimal solution. (f) ENV 3, optimal solution.

Fig. 6. Experimental results of three algorithms on finding the initial solution and converging to the optimal solution. The time cost and the number of node are selected to evaluate the algorithm performance. The line represents the mean and the cross mark represents the standard deviation.

However, the GMR-RRT* always uses the same sampling strategy described in Section III-B in these two stages. From Fig. 5, it can be seen that the GMR-RRT* can quickly find the initial solution with a small number of nodes in three environments. The RRT* needs more time and nodes to find the initial solution. To converge to the optimal solution, the RRT* almost covers the whole map with sampled nodes while the GMR-RRT* still uses much fewer nodes.

A statistical result is provided in Fig. 6. Herein, the time cost and the number of the node are used to evaluate the algorithm performance. Less node in the tree growing process means that the algorithm occupies less memory usage because the RRT-based algorithms use node to store all information in the path planning process. From Fig. 6(a)–(c), we can see that the RRT* and IRRT* perform similarly since they use the same method to find the initial solution. Obviously, the GMR-RRT* performs much better than the RRT* and the IRRT*. The reason is that the GMR-RRT* utilizes a nonuniform sampling distribution generated from the GMR as described in Section III-A. This nonuniform distribution conveys the information on how a human walks to the goal region in the current environment. By learning from human behavior, the GMR-RRT* planner is capable of fast navigating to the goal region. It means that the GMR-RRT* can recognize the promising region where a feasible path probably lies and then bias the sampling process to this region. Therefore, the GMR-RRT* can quickly find an initial solution.

TABLE I
PATH LENGTH OF THE INITIAL SOLUTION

	RRT*	IRRT*	GMR-RRT*
ENV 1	448(±46)	455(±47)	**399(±4)**
ENV 2	667(±46)	661(±36)	**615(±13)**
ENV 3	589(±27)	601(±33)	**553(±6)**

In addition, the initial solution found by the GMR-RRT* is very close to the optimal solution. Table I shows the path length of the initial solution generated from three algorithms. The optimal solutions in the three environments are 400 ± 5, 610 ± 5, and 545 ± 5, respectively. It can be seen that the initial solution from the GMR-RRT* is much better than those from the RRT* and the IRRT*. Therefore, the GMR-RRT* can also guarantee a high-quality initial solution.

Fig. 6(d)–(f) depict the experimental results of three algorithms on converging to the optimal solution. Compared with the RRT*, the IRRT* converges faster since it uses an admissible ellipsoidal heuristic. This heuristic may bias the sampling process to the region where a better solution exists. However, the GMR-RRT* performs much better than the RRT* and the IRRT* in terms of the time cost and the number of the node, which

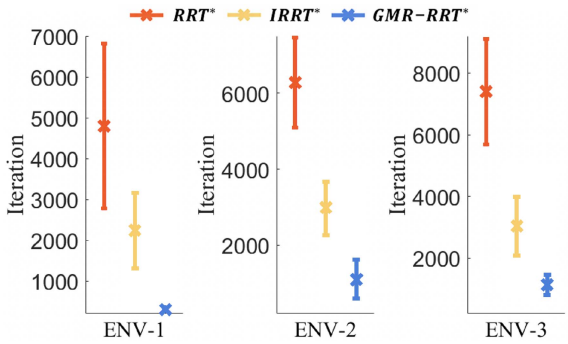
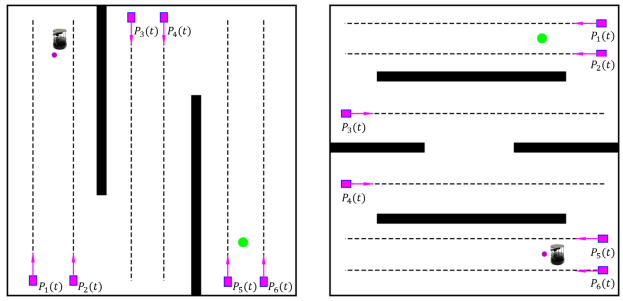
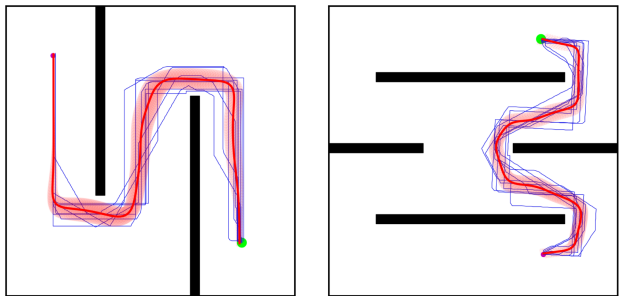


Fig. 7. Number of node used in three algorithms to find the optimal solution. The line represents the mean and the cross mark represents the standard deviation.



(a) ENV 4. (b) ENV 5.

Fig. 8. Experiment setup in two dynamic environments. A mobile robot needs to arrive at the goal region (denoted as green disk) with a real-time updated path. The black rectangle denotes the obstacle. Pedestrians move along the black dotted line and change the moving direction when they arrive at the border of the map.



(a) ENV 4. (b) ENV 5.

Fig. 9. Nonuniform sampling distributions in two dynamic environments. The human demonstrated trajectories are depicted in blue. The nonuniform sampling distribution is generated from the GMR. The mean of the reproduced trajectory is depicted by red line and the respective covariance is depicted by light tubes around the mean, respectively.

reveals that the nonuniform distribution from the GMR is a more efficient heuristic for the RRT-based algorithms. In addition, the number of iteration can directly reflect the performance difference of three algorithms on finding the optimal solution. A small number of iteration corresponds to faster convergence while a larger number corresponds to slower convergence. As shown in Fig. 7, compared with the RRT* and the IRRT*, the GMR-RRT* obviously uses much fewer iterations to converge to the optimal solution, which reveals that the GMR-RRT* significantly accelerates the convergence to the optimal solution.

It is noted that the collected trajectories require additional computation resources. First, for each environment, 10 trajectories are collected. Each trajectory consist of 82 (ENV 1), 129 (ENV 2), 124 (ENV 3), 232 (ENV 4), and 147 (ENV 5) data points. Each data point consists of three integers, one for index and the other for x and y coordinates. The memory used to store one data point is less than that used to store one node in the RRT tree, where the node consists of one pointer to the parent node, at least one pointer to the child node, collision information, and x and y coordinates. In addition, as shown in Fig. 6, the GMR-RRT* can save thousands of nodes in the path planning process. Therefore, the additional memory used to store collected trajectories is very small and even negligible. Second, using GMR to process collected trajectories is well studied and efficient [9]. It can be completed offline, and the generated mean and covariance are reusable. With the generated mean and covariance, the sampling process is direct and light-weighted, as shown in Algorithm 2.

In short, compared with the RRT* and the IRRT*, the GMR-RRT* achieves the best performance on finding the initial solution and converging to the optimal solution.

B. Experiments in Dynamic Environments

In this subsection, we implement the GMR-RRT* and the RT-RRT* in two dynamic environments, where the time cost that the robot uses to arrive at the goal region is selected as the metric to evaluate the algorithm performance. In dynamic environments, the GMR-RRT* updates the plan in real time. Specifically, the partial motion planning method [31] is used to carry out the planning and execution in parallel. In each time step, the robot moves from a parent node to its child node by following the planned path. Meanwhile, the robot updates the RRT tree and generates a new path by perceiving the environment. As this time step is over, the latest generated path is used for execution. Then a round of planning and execution begins. In this paper, the time step is set to 0.2 s. Fig. 8 illustrates the experimental setup in two dynamic environments. In each environment, there are some pedestrians moving around, and the robot needs to arrive at the goal region while avoiding the static obstacles and moving pedestrians. It requires the robot can plan a feasible path in real time. The RT-RRT* and the GMR-RRT* both find an initial path at first, and then the robot begins to move to the goal region while the found path is updated simultaneously. The difference is that they use different sampling strategies, where the RT-RRT* uses a goal-bias sampling strategy, and the GMR-RRT* uses the proposed nonuniform sampling strategy.

Fig. 9 illustrates the nonuniform sampling distributions which are used in the GMR-RRT*. Note that the human demonstrated trajectories are collected in static situations, which means there are no moving pedestrians in the data collection process. However, the generated nonuniform sampling distribution is applied in dynamic situations. Therefore, the extensibility of the GMR-RRT* can be verified in these two experiments.

Table II shows that the time cost the robot uses in two dynamic environments. It can be seen that the GMR-RRT* uses less time

TABLE II
TIME COST IN TWO DYNAMIC ENVIRONMENTS

	RT-RRT*(ms)	GMR-RRT*(ms)
ENV 4	41992(±17649)	**27575(±3837)**
ENV 5	18626(±6556)	**13341(±1916)**

TABLE III
RRT* VS. GMM-RRT* VS. GMR-RRT* IN ENV 1

	Iters(ini)	Node(ini)	Iters(opt)	Node(opt)
RRT*	1487 (±938)	1303 (±850)	4792 (±2021)	4364 (±1788)
GMM-RRT*	578 (±506)	543 (±486)	589 (±510)	554 (±490)
GMR-RRT*	**309 (±52)**	**290 (±44)**	**311 (±52)**	**292 (±45)**

TABLE IV
RRT* VS. GMM-RRT* VS. GMR-RRT* IN ENV 2

	Iters(ini)	Node(ini)	Iters(opt)	Node(opt)
RRT*	1627 (±450)	1432 (±402)	6278 (±1176)	5658 (±1050)
GMM-RRT*	1004 (±223)	891 (±197)	6241 (±1216)	5759 (±1103)
GMR-RRT*	**744 (±151)**	**654 (±122)**	**1114 (±508)**	**972 (±426)**

TABLE V
RRT* VS. GMM-RRT* VS. GMR-RRT* IN ENV 3

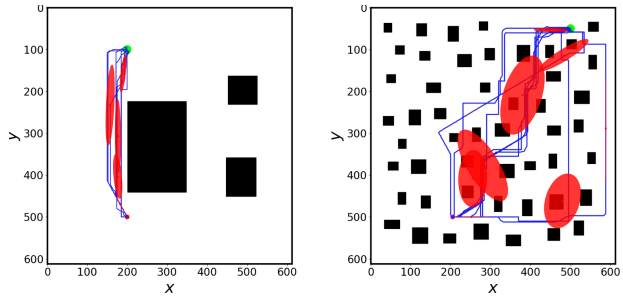
	Iters(ini)	Node(ini)	Iters(opt)	Node(opt)
RRT*	1562 (±647)	1335 (±570)	7393 (±1717)	6601 (±1564)
GMM-RRT*	1422 (±596)	1238 (±529)	5261 (±2102)	4752 (±1913)
GMR-RRT*	**572 (±87)**	**497 (±81)**	**1132 (±36)**	**1132 (±287)**

cost to arrive at the goal region compared with the RT-RRT*. The improvements in two dynamic environments are 34% and 28%, respectively. Besides, the GMR-RRT* suffers a smaller standard deviation, which reveals the stability of the GMR-RRT*. In a word, the GMR-RRT* can successfully transfer the learned human navigation behaviors from static situations to dynamic situations and achieve better performance compared with the state-of-the-art algorithm.

C. Discussion

1) Comparison With GMM-RRT:* To further demonstrate the advantages of the GMR-RRT*, we compare it with another LfD based RRT* algorithm, namely GMM-RRT* [32]. The GMM-RRT* employs the GMM method to bias the sampling process while the GMR-RRT* employs the GMR method. Numerical experiment results are shown in Table III, IV, and V, where "Iters" denotes the number of iterations, "Node" denotes the number of nodes, "ini" corresponds to the initial solution, and "opt" corresponds to the optimal solution.

From the experiment results, we can see that the GMM-RRT* performs much better than the RRT* in ENV 1. In ENV 2 and 3, the GMM-RRT* performs well on finding the initial solution but shows no advantages in converging to the optimal solution compared with the RRT*. However, the GMR-RRT* achieves



(a) ENV 1. (b) ENV 3.

Fig. 10. GMM results in different environments. Red ellipses denote the calculated GMM, and blue lines denote the human demonstrated trajectories, respectively.

the best performance among these three algorithms. The reason is that the GMM based nonuniform sampling strategy has some limitations. As shown in Fig. 10(a), when the demonstration solutions are very similar, the calculated Gaussian distributions are all along with the demonstrations. When the demonstration solutions are diverse, it is hard for the GMM to correctly model them so that the calculated Gaussian distributions seem not very good, as shown in Fig. 10(b). In this situation, when we implement the GMM-RRT* to randomly select one Gaussian distribution to generate a sample, the performance is bad. However, the GMR method can further utilize the calculated Gaussian distributions to form a promising region, as shown in Fig. 4, which indeed accelerates the path planning process.

2) Limitations: As we can see from the aforementioned algorithm analysis and experiment results, GMR-RRT* aims to teach the robot how to complete the same or similar work efficiently. GMR-RRT* learns the path distribution from provided human demonstrations. In the data collection process, for a specified environment and work, the start and goal positions remain unchanged. This usually happens in some repeated work in the same or very similar environment, such as from kitchen to bedroom in the house, or from the information desk to the toilet in a mall. In these situations, the robot completes the same work day by day. Therefore, it is critical to teach the robot how to complete the same or similar work efficiently. This is also the motivation of this paper. Some recent and related work can be found in [33], [34].

However, this is also a limitation of GMR-RRT* because it cannot perform like the conventional RRT* to adapt to a general class of path planning algorithms. This is also a fascinating topic, and we also focus on it. Related work can be found in our proposed Neural RRT* framework [35]. In the future, we plan to use the neural network driven approaches to learn from human demonstrations to equip GMR-RRT* with better extension ability in different environments.

V. CONCLUSION AND FUTURE WORK

In this work, we aim to achieve humanlike path planning for the service robot in human-robot coexisting environments. Sampling-based algorithms achieve great success in this field

because of their ability to efficiently search the state space. However, they suffer from slow convergence to the optimal solution. To overcome this problem, we propose the GMR-RRT* based on the RRT and GMR schemes. The GMR-RRT* algorithm performs like a human to find the trajectory. When the goal position is given, the GMR-RRT* obtains a prior knowledge about how to navigate the environment, and then quickly computes a high-quality trajectory. The experimental results demonstrate that the proposed GMR-RRT* can significantly save the time cost and memory usage on finding the initial and optimal solutions, and also enables the path planner operates in a humanlike manner. In addition, the nonuniform sampling strategy in the GMR-RRT* can also be applied in other sampling-based path planning algorithms to improve the algorithm performance further.

One of the limitations of the GMR-RRT* is that a preprocessing is required to deal with the collected human demonstration for a given environment. This is also a problem that the LfD methods need to face. A possible approach to address this problem is to use Inverse Reinforcement Learning (IRL) [36] to automatically learn how to design an appropriate cost function for robot path planning in human-robot coexisting environments.

Other possible extensions of this work include using the Gaussian Process (GP) technique [37] or the combination of GMR and GP to learn human navigation behavior. It is noted that the effective modeling of human feeling in robot path planning is also an interesting topic, where the psychology knowledge can be taken into consideration.

REFERENCES

[1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016.

[2] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[3] O. KHATIB, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[4] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[5] S. M. LaValle and J. J. Kuffner Jr., "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[6] J. Wang and M. Q.-H. Meng, "Optimal path planning using generalized voronoi graph and multiple potential functions," *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10621–10630, Dec. 2020.

[7] M. Brunner, B. Brüggemann, and D. Schulz, "Hierarchical rough terrain motion planning using an optimal sampling-based method," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 5539–5544.

[8] J. Wang, W. Chi, M. Shao, and M. Q.-H. Meng, "Finding a high-quality initial solution for the RRTs algorithms in 2D environments," *Robotica*, vol. 37, no. 10, pp. 1677–1694, 2019.

[9] N. Jaquier, D. Ginsbourger, and S. Calinon, "Learning from demonstration with model-based Gaussian process," in *Proc. Conf. Robot Learn.*, 2020, pp. 247–257.

[10] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intell. Service Robot.*, vol. 9, no. 1, pp. 1–29, 2016.

[11] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[12] G. Ye and R. Alterovitz, "Guided motion planning," in *Robotics Research*, Berlin, Germany: Springer, 2017, pp. 291–307.

[13] D. Yi, M. A. Goodrich, and K. D. Seppi, "Homotopy-aware RRT*: Toward human-robot topological path-planning," in *Proc. 11th ACM/IEEE Int. Conf. Human-Robot Interaction*, 2016, pp. 279–286.

[14] W. Chi*et al.*, "A human-friendly robot navigation algorithm using the risk-RRT approach," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot.*, 2016, pp. 227–232.

[15] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *Proc. IEEE /RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1688–1694.

[16] G. Ferrer and A. Sanfeliu, "Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 1730–1735.

[17] J. Wang and M. Q.-H. Meng, "Socially compliant path planning for robotic autonomous luggage trolley collection at airports," *Sensors*, vol. 19, no. 12, 2019, Art. no. 2759.

[18] U. Orozco-Rosas, O. Montiel, and R. Sepłveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Appl. Soft Comput.*, vol. 77, pp. 236–251, 2019.

[19] U. Orozco-Rosas, K. Picos, and O. Montiel, "Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots," *IEEE Access*, vol. 7, pp. 156 787–156 803, 2019.

[20] F. Bayat, S. Najafinia, and M. Aliyari, "Mobile robots path planning: Electrostatic potential field approach," *Expert Syst. Appl.*, vol. 100, pp. 68–78, 2018.

[21] N. Pérez-Higueras, F. Caballero, and L. Merino, "Teaching robot navigation behaviors to optimal RRT planners," *Int. J. Social Robot.*, vol. 10, no. 2, pp. 235–249, 2018.

[22] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, 2016.

[23] C. I. Mavrogiannis, V. Blukis, and R. A. Knepper, "Socially competent navigation planning by deep learning of multi-agent path topologies," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 6817–6824.

[24] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1088–1095, Apr. 2018.

[25] T. Zhang, J. Wang, and M. Q.-H. Meng, "Generative adversarial network based heuristics for sampling-based path planning," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 1, pp. 64–74, Jan. 2021.

[26] Z. Li, J. Wang, and M. Q.-H. Meng, "Efficient heuristic generation for robot path planning with recurrent generative model," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7386–7392.

[27] T. Li *et al.*, "Houseexpo: A large-scale 2d indoor layout dataset for learning-based algorithms on mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5839–5846.

[28] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2997–3004.

[29] K. Naderi, J. Rajamäki, and P. Hämäläinen, "RT-RRT*: A real-time path planning algorithm based on RRT," in *Proc. 8th ACM SIGGRAPH Conf. Motion Games*, 2015, pp. 113–118.

[30] W. Chi, C. Wang, J. Wang, and M. Q.-H. Meng, "Risk-DTRRT-based optimal motion planning algorithm for mobile robots," *IEEE Trans. Automat. Sci. Eng.*, vol. 16, no. 3, pp. 1271–1288, Jul. 2018.

[31] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 2210–2215.

[32] R. Ramón-Vigo, N. Pérez-Higueras, F. Caballero, and L. Merino, "A framework for modelling local human-robot interactions based on unsupervised learning," in *Proc. Int. Conf. Social Robot.*, 2016, pp. 32–41.

[33] G. S. Martins, R. P. Rocha, F. J. Pais, and P. Menezes, "ClusterNav: Learning-based robust navigation operating in cluttered environments," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 9624–9630.

[34] F. Tsang, R. A. Macdonald, and S. L. Smith, "Learning motion planning policies in uncertain environments through repeated task executions," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 8–14.

[35] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural RRT*: Learning-based optimal path planning," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 4, pp. 1748–1758, Oct. 2020.

[36] B. Kim and J. Pineau, "Socially adaptive path planning in human environments using inverse reinforcement learning," *Int. J. Social Robot.*, vol. 8, no. 1, pp. 51–66, 2016.

[37] S. Choi, K. Lee, and S. Oh, "Robust learning from demonstrations with mixed qualities using leveraged Gaussian processes," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 564–576, Jun. 2019.

Jiankun Wang received the B.E. degree in automation from Shandong University, Jinan, China, in 2015 and the Ph.D. degree from the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, in 2019. He is currently a Research Assistant Professor with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China.

During his Ph.D. degree, he spent six months with Stanford University, Stanford, CA, USA, as a Visiting Student Scholar supervised by Prof. Oussama Khatib. His current research interests include motion planning and control, human robot interaction, and machine learning in robotics.

Tingguang Li received the B.S. degree in control and system engineering from Nanjing University, Nanjing, China, in 2016. He is currently working toward the Ph.D. degree with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include in-hand manipulation, robot exploration, and reinforcement learning.

Baopu Li received the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong. His current research interests include deep learning and machine learning based perceptions for different applications.

Max Q.-H. Meng (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Victoria, Victoria, BC, Canada, in 1992.

He is currently a Chair Professor with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China, on leave from the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, and also with the Shenzhen Research Institute, The Chinese University of Hong Kong, Hong Kong. He holds honorary positions as a Distinguished Professor with State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin, China, a Distinguished Provincial Professor with the Henan University of Science and Technology, Luoyang, China, and the Honorary Dean of the School of Control Science and Engineering, Shandong University, Jinan, China. He has authored or coauthored more than 500 journal and conference papers and served on many Editorial Boards. His research interests include robotics, perception and sensing, human-robot interaction, active medical devices, biosensors and sensor networks, and adaptive and intelligent systems. Dr. Meng is an Elected Member of the Administrative Committee of the IEEE Robotics and Automation Society. He was the recipient of the IEEE Third Millennium Medal Award.