GMR-RRT* summary

Mano Srijan battula (118546490) email: mbattula@umd.edu

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. In this paper,based on Gaussian Mixture Regression (GMR) and the family of Rapidly-exploring Random Tree (RRT) schemes,It proposes 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.**

I. INTRODUCTION

Many algorithms have been proposed to address the path planning algorithm.such as the well-established A* algorithms, Artificial Potential Field (APF) algorithms,and sampling-based algorithms. 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 such as such has Probabilistic Road Map (PRM) and Rapidly exploring Random Tree (RRT) have become popular due to their capability of efficiently searching the state space.But they usually require much time and high computational power to find the optimal solution therefore In real-world applications, such as autonomous vehicles, its not useful.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.Even though they have been widely used in daily life and gained great success, The sampling-based algorithm suffers sensitivity to the sampling approach, and its performance fluctuates significantly in different environments.The latest machine learning work, a probabilistic method, namely learning from demonstrations (LfD),is used to generate robot motions from demonstrated trajectories. This method utilizes the Gaussian Mixture Regression (GMR) 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).However, the predicted trajectory can provide proper guidance for the sampling process, which happens in sampling-based algorithms. In this paper,it presents the GMR-RRT* algorithm based on 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). Using the GMR, a predicted trajectory represented by 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 it 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 flowchart (Fig. 1) The experiments below validates that the GMR-RRT* advances the state-of-the-art performance by presenting a novel machine learning-based path

Fig. 1. Flowchart of the GMR-RRT*..

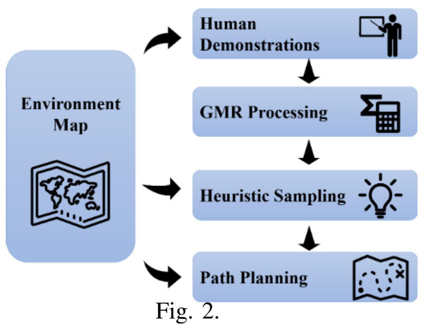


Fig. 2.

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

planning algorithm, and how it achieves better performance in static and dynamic environments.

II. Algorithm

we first introduce how to use the GMR to learn navigation behavior from human demonstrations.The learned behavior is then appliedees in the GMR-RRT* to achieve fast and high-quality path planning. At last, we prove that the GMR-RRT* guarant probabilistic completeness and asymptotic optimality.

A. *Gaussian Mixture Regression*

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.By computing ϵ_s , $\sum_{ss}$ from the equation in fig 2 at different time steps ϵ_t, the expected spatial position ϵ_s and associated covariance matrix $\sum_{ss}$ can be obtained. In the following, we will use ϵ_s $\sum_{ss}$, ˆss as two parameters of Gaussian distribution to achieve nonuniform sampling in the path planning problem.

B. *GMR-RRT**

As shown in Algorithm 1, in the GMR-RRT*, the tree τ 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 $\in$ $G_{x_{goal}}$ is added to the tree. Otherwise, the GMR-RRT*

Fig. 3.

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

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

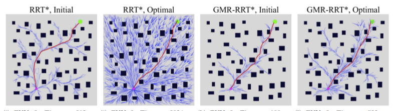
will report failure when the algorithm has run for the specified maxIter number. In the sampling process, if the function Random(1) returns a number larger than 0.5, the uniform sampling strategy is implemented. Otherwise, the nonuniform sampling strategy incorporating the GMR is implemented. The GMR-RRT* implements the same Choose Parent and Rewire strategies as the RRT*.

In Algorithm 2, we use the GMR to obtain ϵ_s.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 ϵ_s.Given ϵ_t. Secondly, using the linear transformation property of Gaussian distributions, the complete expectation ϵ_s, and covariance matrix $\sum_{ss}$ are obtained. Then $\epsilon_s$is computed through the Gaussian distribution with the mean ϵ_sˆ and covariance matrix $\sum_s$. Finally, the value of ϵ_s is transmitted to x_{rand}.After $x_{rand}$is chosen, the function Nearest(T, x_{rand}) attempts to find the nearest node $x_{nearest}$ on the tree to connect x_{rand}.

C. *Probabilistic Completeness and Asymptotic Optimally*

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.if there is a feasible path, it must

Fig. 4. 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.



(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.

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*.

III. Experiments

GMR-RRT* with the RRT* and the IRRT* in static environments and 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.Therefore, the asymptotic optimality is guaranteed.

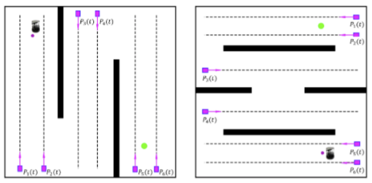
A. *Experiments on static*

The first environment is cluttered with more static obstacles as shown in fig 4. From fig 4. it can seen that GMR-RRT* can quickly find the initial solution with a small number of nodes in the environment. 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. it is because GMR-RRT* utilizes a nonuniform sampling method. Due to distribution it conveys the information on how a human walks to the goal region in the current environment. By learning from human behavior, it is capable of fast navigating to the goal region

B. *experiments on dyminc obstacles*

fig 5. llustrates 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.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. 5. 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 goal position. Pedestrians move along the black dotted line and change the moving direction when they arrive at the border of the map.



IV. Discussion

A. *Comparison with GMM-RRT**

GMM based nonuniform sampling strategy has some limitations.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.The GMR method can further utilize the calculated Gaussian distributions to form a promising region which indeed accelerates the path planning process.

B. *Limitations*

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,because of it cannot perform like the conventional RRT* to adapt to a general class of path planning algorithms.

V. Conclusions and Future Work

The GMR-RRT* algorithm performs like a human to find the trajectory. This 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 human like manner but requires collected human demonstration for a given environment. To overcome the this problem is to use Inverse Reinforcement Learning (IRL) 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 or the combination of GMR and GP to learn human navigation behavior.