Kinodynamic Motion Planning on Gaussian Mixture Fields

Luigi Palmieri[1,2], Tomasz P. Kucner[3], Martin Magnusson[3], Achim J. Lilienthal[3], Kai O. Arras[1]

***Abstract*— We present a mobile robot motion planning approach under kinodynamic constraints that exploits learned perception priors in the form of continuous Gaussian mixture fields. Our Gaussian mixture fields are statistical multi-modal motion models of discrete objects or continuous media in the environment that encode e.g. the dynamics of air or pedestrian flows. We approach this task using a recently proposed circular linear flow field map based on semi-wrapped GMMs whose mixture components guide sampling and rewiring in an RRT* algorithm using a steer function for non-holonomic mobile robots. In our experiments with three alternative baselines, we show that this combination allows the planner to very efficiently generate high-quality solutions in terms of path smoothness, path length as well as natural yet minimum control effort motions through multi-modal representations of Gaussian mixture fields.**

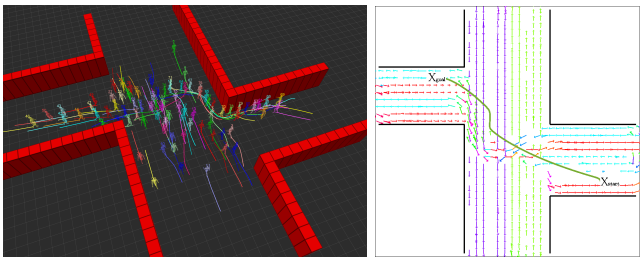

Fig. 1. We present a planning approach that generates smooth and optimal trajectories over a field of Gaussian mixtures. **Left:** The figure shows an example from the studied simulated *intersect* scenario where multiple flows of people encounter each other. **Right:** An example path (in **green**) generated among the Circular Linear Flow Field (CLiFF) map which associates a Gaussian mixture model to each location, whose components encode multiple weighted flow directions.

I. Introduction

Robot operation environments are often rich in semantics, affordances and dynamically moving objects that follow typical motion patterns. Knowledge of such features in addition to the basic geometry of the workspace represents valuable information for a motion planner to generate better solutions in terms of path quality, safety, replanning frequency or social normativeness. Mobile service robots in human environments, for example, may exploit information about typical pedestrian flows to avoid high-density areas and to take advantage of such flows to reach a destination. UAVs in robot olfaction scenarios, for example, may plan paths that help to estimate gas distributions or localize gas sources. In this paper, we present a planning approach that accounts for typical motion of dynamic objects or continuous media such as pedestrian flows or air/water currents, modeled as a field of semi-wrapped Gaussian mixtures to represent the underlying multi-modal vector field.

Past approaches have considered motion planning over regular (unimodal) vector fields: Kularatne *et al.* [11] present a graph-based approach that generates time optimal and energy efficient motion plans for autonomous surface and underwater vehicles in time-varying flow fields. The kinematic constraints of the vehicles are accounted for in the cost function. Otte *et al.* [14] describe a graph-based algorithm to solve the problem of real-time path planning in time-varying wind fields. The anytime algorithm finds an $\alpha\,\beta$ solution quickly which is then, given more time, incrementally improved. Lolla *et al.* [12] generate paths for swarms of underwater vehicles over dynamic water flow fields using a level set approach. The method, based on a 2D grid representation of the environment, finds time-optimal paths while respecting the kinematic constraints of the system. Ko *et al.* [9] present an RRT-based path planner over a vector field defined in the configuration space. To find a path, the algorithm tries to minimize an *upstream criterion* which quantifies the control effort to go against a vector field. Tree growth is guided by this criteria resulting in extensions that are more probably aligned with the vector field directions.

Recurring patterns of human motion that people typically follow in an environment have been learned and used for planning in [1, 13, 5, 18]. Such patterns can be seen as sparse vector fields as they are only defined in parts of the state space where humans have been repeatedly observed. Based on the Risk-RRT algorithm [5], Rios *et al.* [18] use Gaussian processes (GP) to predict motion of humans and generate paths with an RRT-based planner that minimize the risk of disturbing and colliding with surrounding people. O'Callaghan *et al.* [13] present a method that generates paths by following learned motion patterns of people using GPs. The method computes a navigational map based on the motion patterns whose cells incorporate velocity vectors. The robot navigates through the environment by querying the learned map and obtaining the next direction to follow. Bennewitz *et al.* [1] learn a collection of human motions patterns using Gaussian mixtures and Expectation Maximization (EM). For each observed human the most probable pattern is determined and used to compute motion predictions for planning. The method uses A* on a 2D grid with cell costs discounted by the probability that a person is in a cell at a given time.

Unlike [11, 14, 12, 9] we use a more powerful probabilistic

[1]L. Palmieri and K. O. Arras are with Bosch Corporate Research, Stuttgart, Germany, luigi.palmieri, kaioliver.arras@de.bosch.com,

[2]L. Palmieri is also with University of Freiburg, Computer Science Department, palmieri@informatik.uni-freiburg.de

[3]T. P. Kucner, M. Magnusson and A. J. Lilienthal are with Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden. tomasz.kucner, martin.magnusson, achim.lilienthal@oru.se

representation than vector fields named Circular Linear Flow Field (CLiFF) map [10]. It associates a Gaussian mixture model to each location whose components encode multiple weighted flow directions. The model captures the dependency between motion speed (a linear variable) and direction (a circular variable) using semi-wrapped Gaussian mixture models introduced by Roy et al. in [20]. In addition to the model to represent environment dynamics, and contrarily to the previously described approaches that use discrete search, we use an asymptotically optimal sampling-based motion planner that implicitly considers the robot's kinematic and its non-holonomic constraints by using a steer function to plan in a continuous state space.

The combination leads to a novel algorithm, named *CLiFF-RRT**, that plans kinodynamically feasible paths under a CLiFF-map model, trading off classical path quality metrics with the compliance to the environment dynamics. In the experiments, we compare our approach to RRT, RRT* and an uninformed variant of the algorithm and show that CLiFF-RRT* is significantly faster than the baselines and produces solutions that best comply to the flow directions as modelled by the map. The algorithm also achieves shorter and smoother paths and retains the probabilistic completeness and asymptotic optimality properties of RRT*.

The paper is structured as follows: in Sec. II we briefly present the CLiFF-map model and describe the algorithm and its properties in Sec. III. We present experiments in Sec. IV and discuss their results in Sec. V.

II. CLiFF-Map Model

The Circular Linear Flow Field map (CLiFF-map) [10] describes motion patterns as a field of Gaussian mixtures whose local elements are probability distribution of (instantaneous) velocities $\boldsymbol{V} = (\theta, \rho)$, where $\theta \in [0, 2\pi)$ is the orientation and $\rho \in \mathbb{R}^+$ the speed. This is a heterogeneous vector with one circular random variable (θ) and one linear (ρ). For their representation we choose the *semi-wrapped normal distribution*: a bivariate normal distribution on a cylinder where one of the dimensions is defined along the cylinder's height while the other is wrapped around it's circumference,

$$\mathcal{N}^{\mathcal{SW}}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{V}) = \sum_{k\in\mathbb{Z}} \mathcal{N}_{\boldsymbol{\mu},\boldsymbol{\Sigma}}\left(\begin{bmatrix}\theta\\ \rho\end{bmatrix} + 2\pi\begin{bmatrix}k\\ 0\end{bmatrix}\right). \tag{1}$$

To model multi-modal events such as human motion patterns, CLiFF-maps employ semi-wrapped Gaussian mixture models (SWGMM),

$$p(\boldsymbol{V}|\boldsymbol{\xi}) = \sum_{j=1}^{J} \pi_j \mathcal{N}^{\mathcal{SW}}_{\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j}(\boldsymbol{V}) \tag{2}$$

with $\sum_{j=1}^{J}\pi_j = 1$. A SWGMM ($\boldsymbol{\xi}$) is a weighted sum of J semi-wrapped normal distributions, that capture the local distribution of velocities. A CLiFF-map (see Fig. 2), denoted as $\mathcal{D}$, is a field of $N_{\mathcal{D}}$ tuples that characterize local motion patterns of dynamic obstacles as

$$\mathcal{D} = \{(\boldsymbol{\xi}_s, \boldsymbol{l}_s) | s \in \mathbb{Z}^+ \wedge \boldsymbol{l}_s \in \mathbb{R}^2\}, \tag{3}$$

where $\boldsymbol{\xi}_s$ denotes a SWGMM that describes a local motion pattern at position $\boldsymbol{l}_s$. We use Mean Shift (MS) to estimate the initial position of clusters for EM, which estimates the parameters of SWGMM ($\boldsymbol{\xi}$).

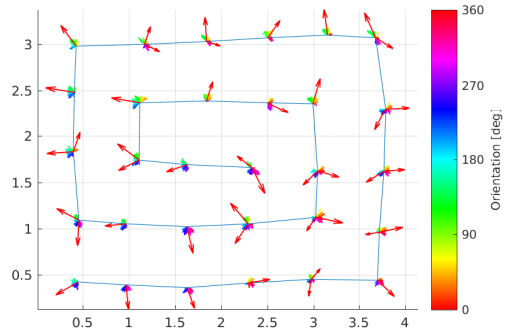

Fig. 2. Visualisation of CLiFF distributions obtained for a set of wind measurements. The red arrows represent the directions of the modes while colour coded ones represent the raw measurements. [22]

The von Mises distribution, broadly used for modeling uncertain circular data (e.g. [4]), is not suitable for this kind of heterogeneous variables. Attempts to overcome this include e.g. building Independent von Mises–Gaussian distributions (Roy et al. [19]) but such distributions still assume no correlation between magnitude and orientation of velocity vectors – an invalid assumption in most real world cases.

III. Our Approach

In this section we describe our method to plan a robot's motion under a CLiFF-map model. To this end, we introduce an *extended upstream criterion* which measures the effort to navigate through a CLiFF-map, followed by the description of the algorithm and its properties.

A. Extended Upstream Criterion

The goal of our algorithm is to find planning solutions that trade off classical motion planning metrics such as path length and path smoothness with the compliance to the environment dynamics. In order to quantify the latter, we extend the *upstream criterion*, proposed by Ko *et al.* [9] for unimodal vector fields, to fields of Gaussian mixtures so as to account for the multi-modal nature of the CLiFF-map representation.

Given a state $\mathbf{x}_i$ that falls into a cell $\boldsymbol{l}_i$ to which a mixture $\boldsymbol{\xi}_i$ with J_i semi-wrapped normal components is associated, the upstream metric is computed as

$$U_d(\mathbf{x}_i, \boldsymbol{\xi}_i) = \sum_{j=1}^{J_i} \Big(\|\boldsymbol{\mu}_{ji}\| - \langle \boldsymbol{\mu}_{ji}, \mathbf{x}'_i \rangle \Big)$$

with $\langle , \rangle$ being the inner product, $\boldsymbol{\mu}_{ji}$ the first-order moment of the jth component of mixture $\boldsymbol{\xi}_i$, and $\mathbf{x}'_i$ the unit vector describing the direction of the path at $\mathbf{x}_i$. When $\mathbf{x}_i$ is mapped to a cell $\boldsymbol{l}_i$ that has no distributions, we perform nearest-neighbor interpolation and compute $U_d(\mathbf{x}_i, \boldsymbol{\xi}_i)$ using the closest mixture $\boldsymbol{\xi}_i$. The criterion yields low costs for paths that comply to the directions of CLiFF-map mixture components and high costs for paths in opposite directions.

B. CLiFF-RRT*

For planning, we choose (and modify) RRT* as a natural choice for optimal motion planning under kinodynamic constraints. Let $\mathcal{X} \in \mathcal{R}^d$ be the configuration space and $\mathcal{U} \in \mathcal{R}^m$ the control space, the dynamics of the robot can be described by the differential equation $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}(0) = \mathbf{x}_0$, with $\mathbf{x}(t) \in \mathcal{X}$, $\mathbf{u}(t) \in \mathcal{U}$ and f describing the system's kinematic constraints.

RRT* [7] is a probabilistically complete single-query sampling-based planner that asymptotically finds optimal solutions for a motion planning problem. Given an obstacle space $\mathcal{X}_{obs} \in \mathcal{X}$, a free space $\mathcal{X}_{free} \in \mathcal{X} \setminus \mathcal{X}_{obs}$, a start state $\mathbf{x}_{start} \in \mathcal{X}_{free}$ and a goal state $\mathbf{x}_{goal} \in \mathcal{X}_{goal} \subset \mathcal{X}_{free}$, the algorithm expands into $\mathcal{X}_{free}$ a tree τ whose edges are trajectories $\boldsymbol{\sigma}_i$ (with $\boldsymbol{\sigma}_i(j)$ being state j of trajectory i) that satisfy the kinematic constraints of the considered system.

In summary, we approach the task as a hierarchical motion planning problem in that we first generate a discrete path $\mathbf{P}_\mathcal{D}$ that selects mixtures at relevant locations, and then use those mixtures to bias the sampling and rewiring procedures in RRT*. The first step makes sure that an initially feasible path is found quickly given a CLiFF map while the second step, generates and incrementally improves a trajectory $\mathbf{x}_\mathcal{D}$ that satisfies the kinodynamic vehicle constraints. The result is CLiFF-RRT* in Alg. 1 whose steps are explained next.

Algorithm 1 *CLiFF-RRT**

function CLiFF-RRT*($\mathbf{x}_{start}$, $\mathbf{x}_{goal}$)
$\mathbf{P}_\mathcal{D} \Leftarrow$ SelectMixtures($\mathbf{x}_{start}$, $\mathbf{x}_{goal}$)
if $\mathbf{P}_\mathcal{D} = \emptyset$ **then**
  return *failure*
end if
τ.AddNode($\mathbf{x}_{start}$)
$g(\mathbf{x}_{start}) \Leftarrow 0$
$n \Leftarrow 1$
while $n \leq N_{Iter}$ **do**
  $\mathbf{x}_{rand} \Leftarrow$ CLiFFSampling($\mathcal{X}, \mathbf{P}_\mathcal{D}$)
  $\mathbf{x}_{near} \Leftarrow$ NearestSearch($\tau, \mathbf{x}_{rand}, \mathbf{P}_\mathcal{D}$)
  $\mathbf{u}_{new}, \boldsymbol{\sigma}_{new}, \mathbf{x}_{new} \Leftarrow$ Steer($\mathbf{x}_{near}, \mathbf{x}_{rand}$)
  if $\boldsymbol{\sigma}_{new} \in \mathcal{X}_{obs}$ **then**
    continue
  end if
  τ.AddNode($\mathbf{x}_{new}$)
  τ.AddEdge($\mathbf{x}_{near}, \mathbf{x}_{rand}, \boldsymbol{\sigma}_{new}, \mathbf{u}_{new}$)
  $g(\mathbf{x}_{new}) \Leftarrow g(\mathbf{x}_{near}) +$ Cost($\mathbf{x}_{near}, \mathbf{x}_{new}$)
  $\tau \Leftarrow$ Rewire($\tau, \mathbf{x}_{new}, \mathbf{x}_{near}$)
  if $\mathbf{x}_{new} \in \mathcal{X}_{goal}$ **then**
    $\mathbf{x}_\mathcal{D} =$ ExtractTrajectory($\mathbf{x}_{new}$)
  end if
  $n \Leftarrow n + 1$
end while
return *failure*

SelectMixtures($\mathcal{X}$, $\mathbf{x}_{start}$, $\mathbf{x}_{goal}$): in this step we select the semi-wrapped mixtures $\mathcal{N}^{SW}$ that allow the system to move from $\mathbf{x}_{start}$ to $\mathbf{x}_{goal}$ while respecting the learned environment's dynamics. We run a Dijkstra search over the graph $\mathcal{G}$ in which each node $n_{\mathcal{G},i}$ is associated to each mixture component $\big(\forall(\boldsymbol{\xi}_i, \boldsymbol{l}_i) \in \mathcal{D}\big)$: for each map cell $\boldsymbol{l}_i$ we compute edges that go between all pairs of the SWGMM components of $\boldsymbol{l}_i$ and those of the cells in its 4-neighborhood. To each edge $e(n_{\mathcal{G},i}, n_{\mathcal{G},j})$, with $n_{\mathcal{G},i}$ and $n_{\mathcal{G},j}$ being two neighboring nodes, we associate the following cost:

$$c(e(n_{\mathcal{G},i}, n_{\mathcal{G},j})) = d(e(n_{\mathcal{G},i}, n_{\mathcal{G},j})) + U_d(\mathbf{x}_{n_{\mathcal{G},j}}, \boldsymbol{\xi}_{n_{\mathcal{G},j}}) \quad (4)$$

where $d(e(n_{\mathcal{G},i}, n_{\mathcal{G},j}))$ is the squared Euclidean distance between the nodes, $\mathbf{x}_{n_{\mathcal{G},j}}$ and $\boldsymbol{\xi}_{n_{\mathcal{G},j}}$ respectively the state and the mixture associated to the node $n_{\mathcal{G},j}$.

The search generates a concatenation $\mathbf{P}_\mathcal{D}$ (i.e. a path) of $N_{\mathbf{P}_\mathcal{D}}$ CLiFF-map tuples $(\boldsymbol{\xi}_i, \boldsymbol{l}_i)$ from $\boldsymbol{l}_0$ (with $\mathbf{x}_{start} \in \boldsymbol{l}_0$) to $\boldsymbol{l}_{N_{\mathbf{P}_\mathcal{D}-1}}$ (with $\mathbf{x}_{goal} \in \boldsymbol{l}_{N_{\mathbf{P}_\mathcal{D}-1}}$) which is forwarded to the sampling unit.

CLiFFSampling($\mathcal{X}, \mathbf{P}_\mathcal{D}$): it draws $\mathbf{x}_{rand}$ samples in $\mathcal{X}$. The parameter $\alpha \in [0, 1]$ sets the probability of the biasing towards the $N_{\mathbf{P}_\mathcal{D}}$ CLiFF-map mixtures $\boldsymbol{\xi}_i$ of $\mathbf{P}_\mathcal{D}$:

$$\mathbf{x}_{rand} \sim \sum_{i=0}^{N_{\mathbf{P}_\mathcal{D}}-1} \sum_{j=1}^{J_i} \pi_j \mathcal{N}^{SW}_{\boldsymbol{\xi}_i}(\boldsymbol{\mu}_{j\boldsymbol{\xi}_i}, \boldsymbol{\Sigma}_{j\boldsymbol{\xi}_i})$$

With a probability of $(1 - \alpha)$, samples are drawn from a uniform distribution defined on the entire state space $\mathcal{X}$.

NearestSearch(τ, $\mathbf{x}_{rand}$, $\mathbf{P}_\mathcal{D}$): it returns the node $\mathbf{x}_{near}$ that connects to $\mathbf{x}_{rand}$ with minimum cost-to-go $C(\mathbf{x}_{near}, \mathbf{x}_{rand}, \mathbf{P}_\mathcal{D})$ within distance δR (as parameter) from the latter:

$$\mathbf{x}_{near} = \arg \min_{\mathbf{x} \in \mathcal{X}_{\delta R}} g(\mathbf{x}) + \mathtt{Cost}(\mathbf{x}, \mathbf{x}_{rand}) \quad (5)$$

with $g(\mathbf{x})$ being the cost-to-come to vertex $\mathbf{x}$ from root $\mathbf{x}_{start}$ through the current tree τ. If no nodes are found within this distance, the closest vertex in terms of Euclidean distance is returned.

Cost($\mathbf{x}_i$, $\mathbf{x}_j$): returns the cost of the trajectory $\boldsymbol{\sigma}$ that connects node $\mathbf{x}_i$ to node $\mathbf{x}_j$. Our algorithm aims to find trajectories which are smooth and short, respect the environment dynamics and minimize the upstream criteria (the control effort to move with the vector field) with respect to the off-line learned mixtures $\forall \boldsymbol{\xi}_i \in \mathcal{D}$. For these reasons we use the following cost function:

$$C(\mathbf{x}_p, \mathbf{x}_z, \mathbf{P}_\mathcal{D}) = \sum_{i=1}^{N_p} \|\boldsymbol{\sigma}(i) - \boldsymbol{\sigma}(i-1)\| + |(1 - |\mathbf{q}_i \cdot \mathbf{q}_{i-1}|)| + \sum_{i=1}^{N_p} U_d(\boldsymbol{\sigma}(i), \boldsymbol{\xi}_i)$$

where $\boldsymbol{\sigma}(i)$ are intermediate states of the trajectory $\boldsymbol{\sigma}$ connecting $\mathbf{x}_p$ to $\mathbf{x}_z$, $\mathbf{q}_i$ are related quaternions, and $U_d(\boldsymbol{\sigma}(i), \boldsymbol{\xi}_i)$ being the upstream functional value at $\boldsymbol{\sigma}(i)$. A supervised learning approach can be used to improve the efficiency of the cost computation as in [16].

Steer($\mathbf{x}_i$, $\mathbf{x}_j$): it generates a trajectory $\boldsymbol{\sigma}$ and the set of controls $\mathbf{u}$ needed to steer the system from $\mathbf{x}_i$ to $\mathbf{x}_j$. The analytical steer function connects any pair of states and respects the *topological property* as described in [7, 17].

Rewire($\tau, \mathbf{x}_{new}, \mathbf{x}_{near}$): rewires the tree τ as in the original RRT* [7], using the above described Steer and Cost functions. The rewiring is done at each iteration on a set of vertices found by a near neighbor search as in [8]: it finds the set of all the states in τ that lie within a box centered

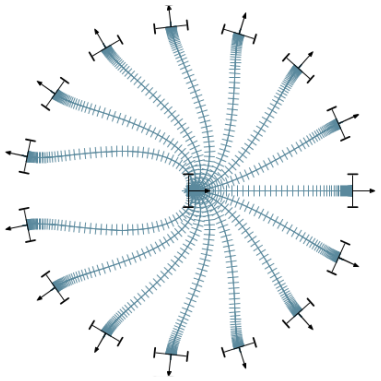

Fig. 3. Example smooth paths generated by the POSQ steer function when steering the robot from the center to the poses on the circle.

on $\mathbf{x}_{near}$ whose volume scales as $\gamma^{D\frac{\log(n)}{n}}$, with D being the Hausdorff dimension of the distribution generated by the system dynamics.

C. Steer Function

We consider wheeled mobile robots with a differential drive kinematic configuration with state $\mathbf{x} = (x, y, \theta, v)$, where $(x, y) \in \mathbb{R}^2$ is the Cartesian position, $\theta \in [-\pi, \pi)$ is the heading orientation and v its translational velocity. After a Cartesian-to-polar coordinate transformation, the equations of motions are $\dot{\rho} = -\cos\alpha\ v, \dot{\alpha} = \frac{\sin\alpha}{\rho}v - \omega, \dot{\phi} = -\omega$ where ρ is the Euclidean distance between the Cartesian coordinates of the robot pose (x, y, θ) and of the goal state, ϕ the angle between the x-axis of the robot reference frame $\{X_r\}$ and the x-axis of the goal state frame $\{X_g\}$, α the angle between the y-axis of the robot reference frame and the vector connecting the robot with the goal position, v the translational and ω the angular robot velocity. Thanks to the polar representation we overcome the obstruction to stabilizability for such system described in the Theorem of Brockett [3]. To exactly connect any pairs of states smoothly and efficiently for this description of a wheeled mobile robot, we use and extend the POSQ steer function [15], see Fig. 3. Each time when the steer function is called in Alg. 1 to connect two sampled states $\mathbf{x}_1 = (x_1, y_1, \theta_1, v_1)$ to $\mathbf{x}_2 = (x_2, y_2, \theta_2, v_2)$, we plan an initial extension by using POSQ as described in [15] and then modify the velocity profile so that the initial velocity is equal to v_1 and final one is v_2. The velocity profile is generated using an efficient third-order polynomial time-law [2].

D. Algorithm Properties

RRT* has favorable properties such as probabilistic completeness and asymptotic optimality. In this section we briefly analyze how the alterations of the proposed algorithm impact those properties. RRT and RRT* are probabilistically complete as their sampling procedure draw samples from a uniform distribution over the state space. This applies also to CLiFF-RRT* which generates, at a given probability, uniformly distributed random samples, none of which are rejected. Regarding asymptotic optimality, Karaman and Frazzoli have shown that for n uniformly distributed random samples, a steer function that connect two poses exactly, an admissible cost function and a specific constant γ in the selection of the neighboring nodes, RRT* almost surely converges asymptotically to the optimal solution as n goes to infinity [8]. CLiFF-RRT* uses the same rewiring and neighbor nodes selection procedures of RRT*. It uses a steer function that exactly connects two nodes and its cost function $C(\mathbf{x}_i, \mathbf{x}_j, \mathbf{P}_\mathcal{D})$ is an admissible cost function for RRT*: it is monotonic, additive and Lipschitz continuous. Moreover, it generates, at a given probability (1-α), uniformly distributed random samples. Therefore CLiFF-RRT* retains the asymptotic optimality property of RRT*.

IV. Experiments

The purpose of the experiments is to evaluate the performance of the proposed CLiFF-RRT* algorithm with respect to the baselines of regular RRT and RRT* and an uninformed variant of the algorithm, called All-Mixtures-RRT*, that generates samples from a distribution composed of *all* CLiFF-map mixtures and not on a subset as it the case with CLiFF-RRT*. All methods use the steer function described in Sec. III-C and cost function described in Sec. III-B.

We run the experiments on a single core of an ordinary PC with a 2.80 GHz Intel i7 processor and 32 GB RAM using C++. After a set of informal validation runs we set the parameters α to 0.95 and δR to 4 m, while γ is set in a way to satisfy the requirements of RRT*, see [8].

A. Environments

We study how the planners behave in environments of varying complexity. Given our interest in wheeled mobile service robots, environments are generated by exploiting off-line learned motion models of pedestrian traffic. We have designed four simulated test environments shown in Fig.1 and Fig.4-5. In all cases there are different flow dynamics between the start and goal, and different planning solutions (homotopy classes) are possible. The *L* and *P* environments contain a few obstacles. Here the planners have less geometric constraints to better follow the upstream criterion in the free space. The *maze* environment has many different homotopy classes and narrow passages: the environment has many different flows that go against each other. The *intersect* scenario has many flows of pedestrians coming from different corridors intersecting in a junction: also here there are few geometric constraints but several flows are present. All the CLiFF-maps have been generated with the help of the pedestrian simulator Pedsim [21]. The grid cell size for the CLiFF-map is set to 1 m in all the environments.

B. Metrics

For each planner and environment, we perform 50 runs. For the *L*, *P* and *intersect* scenarios each run lasts 60 s, and for the *maze* scenario 120 s. We compute the means and standard deviations of the following metrics: planning time T_s (measured in seconds), resulting trajectory length l_p (measured in meters) and final cost C_s. Furthermore, to measure smoothness, we use a metric introduced in our previous work [17]: roughness R, defined as the square of the change in curvature κ of the robot, integrated along the

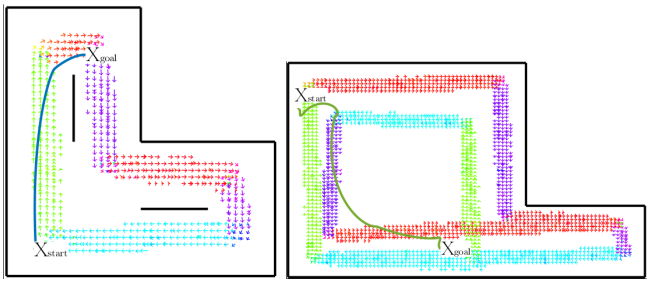

Fig. 4. **Left:** an example CLiFF-RRT* path (**in blue**) generated in the *L* scenario. **Right:** an example CLiFF-RRT* path (**in green**) generated in the *P* scenario. The **arrows** represent the learned mixtures. In these environments just a few obstacles are present. The algorithm finds the best solution that optimizes path length and the upstream criterion: the solutions follow the learned flows.

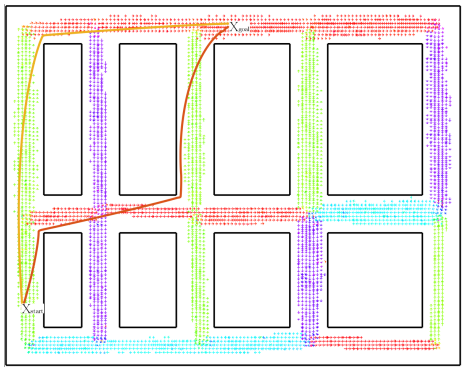

Fig. 5. An example CLiFF-RRT* path generated in the more complex *maze* scenario. In **red** the RRT* path generated by minimizing only path length. The **arrows** describe the learned mixtures. CLiFF-RRT* computes a path (**in orange**) that better minimizes the upstream criterion, without encountering or crossing flows going in opposing direction.

trajectory and normalized by the trajectory length L, $R = \int_{t_0}^{t_l} \left| \frac{1}{L} \frac{d\kappa}{dt} \right|^2 dt$. Smoother trajectories have smaller roughness. We also report the percentage of trajectories found (problems solved) within the planning time limit.

V. Results And Discussion

The experimental results for CLiFF-RRT* and the three baseline planners are given in Table II. The best values are highlighted in boldface, smaller values are better for all performance metrics excepts for the percentage of the problems solved.

CLiFF-RRT* outperforms the baselines with respect to all the metrics. We make the following observations:

(i) CLiFF-RRT* with its focused search finds an initial solution faster than all the baselines (thus also Informed RRT* [6] which behaves as RRT* until a first solution is found). RRT and RRT* do not avoid the time-consuming exploration of the entire state space. For this reason the latter more often fails to find an initial solution in the given time. Moreover from Table I, we can see that the planning time of sub-selecting a set of mixtures with the Dijkstra search does not alter the overall planning time of CLiFF-RRT*. Additionally CLiFF-RRT*, in average, converges faster to a lower cost solution than the baselines, see Fig.6. Those results confirm the intuition that having prior knowledge of the environment's dynamics, improves planning efficiency

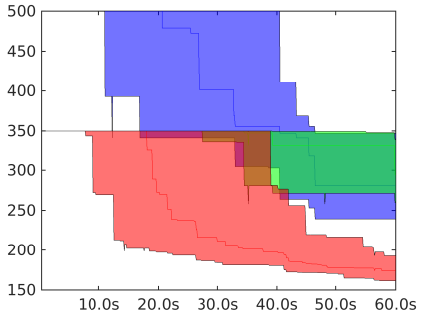

Fig. 6. Cost convergence plot (median, first and third quartiles computed over 50 runs) respect to the planning time of CLiFF-RRT* (in **red**), All-Mixtures-RRT* (in **green**) and RRT* (in **blue**), for the *Intersect* scenario. The informed sampling allows CLiFF-RRT* to quickly find an initial solution and to converge faster to a lower cost solution than the baselines.

(e.g. in our experiments knowing how people usually move in an environment allows the planner to explore a smaller part of the configuration space).

(ii) CLiFF-RRT* finds less costly solutions if compared to all the baselines. The mixtures selected via the Dijkstra search guide the tree towards areas of the state space where the found trajectory is most likely to offer a good trade-off between length and control-effort against the dynamics of the environment (the *upstream-criterion*). The uniform sampling of RRT and RRT* has not such knowledge thus those planners fail to find a better solution. All-Mixtures-RRT*, without the Dijkstra search biasing, fails to find better solution too in the allowed planning time.

(iii) The CLiFF-RRT* sampling strategy results in smoother trajectories than the baselines. Mainly because the off-line learned mixtures bias the tree towards concatenation of extensions with less velocity discontinuities. Uniform sampling generates velocities without prior knowledge about usual motions in particular portions of the state space, thus producing less correlated velocities.

VI. Conclusion

In this paper, we present CLiFF-RRT*, an algorithm that exploits prior knowledge of the environment's dynamics in order to efficiently plan smooth and short paths. Differently from previous approaches, our method plans considering a novel multi-model representation (not a simple vector field) of the dynamic obstacles' motions. We evaluate and compare the approach in four different environments to three different baseline planners, namely RRT, RRT* and an uninformed version of our algorithm that samples considering all the off-line learned CLiFF distributions. The results indicate that the CLiFF-map priors help CLiFF-RRT* to find shorter and

Environments	$\mathbf{T_{Dijkstra}}$ $[ms]$
L	4.36
P	3.36
Intersect	2.83
Maze	121.20

TABLE I

Experimental results: Planning times of Dijkstra

L environment									
Planner	Cost $\mathbf{C_s}$		Planning time $\mathbf{T_s}$ $[s]$		Traject. length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Solved in 60s
CLiFF-RRT*	$\mathbf{111.42}$	± 5.08	$\mathbf{5.30}$	± 8.08	$\mathbf{33.59}$	± 0.78	$\mathbf{0.00007}$	± 0.00004	$\mathbf{100\%}$
All-Mixtures-RRT*	130.89	± 32.92	14.97	± 17.65	35.48	± 2.31	0.00009	± 0.0001	48%
RRT	784.82	± 618.5	15.83	± 17.16	40.59	± 7.36	0.0023	± 0.0048	36%
RRT*	212.26	± 193.4	28.15	± 14.91	37.31	± 3.35	0.00043	± 0.0013	34%
Maze environment									
Planner	Cost $\mathbf{C_s}$		Planning time $\mathbf{T_s}$ $[s]$		Traject. length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Solved in 120s
CLiFF-RRT*	$\mathbf{151.51}$	± 12.62	$\mathbf{31.13}$	± 32.66	$\mathbf{123.23}$	± 1.02	$\mathbf{0.000022}$	± 0.00004	$\mathbf{90\%}$
All-Mixtures-RRT*	180.52	± 54.83	36.74	± 38.12	126.43	± 3.83	0.000038	± 0.000025	10%
RRT	1260.54	± 1278.96	41.74	± 17.87	169.94	± 25.93	0.00058	± 0.00064	10%
RRT*	560.78	± 397.98	64.29	± 35.93	176.68	± 43.67	0.00053	± 0.00062	14%
P environment									
Planner	Cost $\mathbf{C_s}$		Planning time $\mathbf{T_s}$ $[s]$		Traject. length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Solved in 60s
CLiFF-RRT*	$\mathbf{1125.16}$	± 659.0	$\mathbf{11.42}$	± 14.11	$\mathbf{59.13}$	± 4.48	$\mathbf{0.000098}$	± 0.00025	$\mathbf{56\%}$
All-Mixtures-RRT*	1688.09	± 27.48	12.38	± 7.68	103.54	± 31.1	0.0007	± 0.0024	2%
RRT	2532.96	± 798.46	21.16	± 19.8	82.87	± 23.04	0.0007	± 0.0014	16%
RRT*	1128.06	± 454.64	25.43	± 18.90	122.61	± 34.72	0.00057	± 0.00007	16%
Intersect environment									
Planner	Cost $\mathbf{C_s}$		Planning time $\mathbf{T_s}$ $[s]$		Traject. length $\mathbf{l_p}$ $[m]$		Roughness $\mathbf{R}$		Solved in 60s
CLiFF-RRT*	$\mathbf{182.52}$	± 28.77	$\mathbf{24.96}$	± 17.29	$\mathbf{34.71}$	± 1.00	$\mathbf{0.013}$	± 0.019	$\mathbf{76\%}$
All-Mixtures-RRT*	307.67	± 56.25	29.4	± 18.70	51.77	± 20.76	0.013	± 0.0104	14%
RRT	722.15	± 373.35	27.78	± 25.55	41.97	± 10.14	0.0196	± 0.012	10%
RRT*	298.75	± 69.42	27.16	± 15.77	47.61	± 16.39	0.087	± 0.0069	24%

TABLE II

EXPERIMENTAL RESULTS: TRAJECTORY QUALITY AND PLANNING EFFICIENCY

smoother trajectories significantly faster than all the baselines. Moreover the results show that the approach requires less control effort (smaller cost) to drive a wheeled mobile robot through a dynamic environment. CLiFF-RRT* retains the probabilistic completeness and the asymptotic optimality of RRT*. In future work, we intend to study the behavior of the algorithm in different types of dynamic environments (e.g. UAVs in robot olfaction scenarios). To further improve planning efficiency and reduce the variance of the results, we intend to investigate the use of deterministic, as opposed to random sampling sequences.

ACKNOWLEDGEMENTS

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732737 (ILIAD) and the Swedish Knowledge Foundation project under contract nr 20140220 (AIR).

REFERENCES

[1] M. Bennewitz, W. Burgard, and S. Thrun. Adapting navigation strategies using motions patterns of people. In *Int. Conf. on Robotics and Automation (ICRA)*, 2003.

[2] L. Biagiotti and C. Melchiorri. *Trajectory planning for automatic machines and robots*. Springer Science & Business Media, 2008.

[3] R. W. Brockett et al. Asymptotic stability and feedback stabilization. *Differential geometric control theory*, 27(1), 1983.

[4] S. Calderara, A. Prati, and R. Cucchiara. Mixtures of von Mises distributions for people trajectory shape analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(4), apr 2011.

[5] C. Fulgenzi, A. Spalanzani, C. Laugier, and C. Tay. Risk based motion planning and navigation in uncertain dynamic environment. *INRIA Research Report*, 2010.

[6] 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 *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[7] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. Journal of Robotics Research*, 30, 2011.

[8] S. Karaman and E. Frazzoli. Sampling-based optimal motion planning for non-holonomic dynamical systems. In *Int. Conf. on Robotics and Automation (ICRA)*, 2013.

[9] I. Ko, B. Kim, and F. C. Park. Randomized path planning on vector fields. *The International Journal of Robotics Research*, 33(13), 2014.

[10] T. P. Kucner, M. Magnusson, E. Schaffernicht, V. M. H. Bennetts, and A. J. Lilienthal. Enabling flow awareness for mobile robots in partially observable environments. *IEEE Robotics and Automation Letters (RA-L)*, 2017.

[11] D. Kularatne, S. Bhattacharya, and M. A. Hsieh. Time and energy optimal path planning in general flows. *Proc. of the Robotics: Science and Systems (RSS)*, 2016.

[12] T. Lolla, Patrick J. H. Jr, and Pierre FJ L. Time-optimal path planning in dynamic flows using level set equations: realistic applications. *Ocean Dynamics*, 2014.

[13] S. T. O'Callaghan, S. P. N. Singh, A. Alempijevic, and F. T. Ramos. Learning navigational maps by observing human motion patterns. In *Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[14] M. Otte, W. Silva, and E. Frew. Any-time path-planning: Time-varying wind field+ moving obstacles. *Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[15] L. Palmieri and K. O. Arras. A novel RRT extend function for efficient and smooth mobile robot motion planning. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[16] L. Palmieri and Kai O. Arras. Distance metric learning for rrt-based motion planning with constant-time inference. In *Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[17] L. Palmieri, S. Koenig, and K. O Arras. RRT-based nonholonomic motion planning using any-angle path biasing. In *Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[18] J. Rios-Martinez, A. Spalanzani, and C. Laugier. Understanding human interaction for probabilistic autonomous navigation using risk-rrt approach. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[19] A. Roy, S. K Parui, and U. Roy. A Mixture Model of Circular-Linear Distributions for Color Image Segmentation. *International Journal of Computer Applications*, 58(9), 2012.

[20] A. Roy, S. K. Parui, and U. Roy. SWGMM: a semi-wrapped Gaussian mixture model for clustering of circular-linear data. *Pattern Analysis and Applications*, 2014.

[21] D. Vasquez, B. Okal, and K. O. Arras. Inverse reinforcement learning algorithms and features for robot navigation in crowds: An experimental comparison. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[22] Y. Wada, M. Trincavelli, Y. Fukazawa, and H. Ishida. Collecting a Database for Studying Gas Distribution Mapping and Gas Source Localization with Mobile Robots. In *Int. Conf. Adv. Mechatronics*, pages 183–188, 2010.