Leveraging Interactive Distance Fields for Safe and Smooth Reactive Planning

Usama Ali, Adrian Müller, Tobias Kaupp
Technical University of Applied Sciences
Würzburg-Schweinfurt (THWS)
Schweinfurt, Germany

Fouad Sukkar, Lan Wu, Teresa Vidal-Calleja
University of Technology Sydney (UTS)
Sydney, Australia

***Abstract*—Human-robot collaboration applications require safe and reactive planning. Euclidean distance fields (EDF) are a promising representation of such dynamic scenes due to their ability to reason about free space and the readily available distance to collision costs. A key challenge for the commonly used discrete EDF representations, however, is the need for differentiable distance fields to produce smooth collision costs and efficient updates of the dynamic objects. In this paper, we propose to use a Gaussian Process (GP) distance field-based framework that enables both, differentiable distance fields and fast dynamic scene updates. This framework allows implicit semantic reasoning of static and dynamic regions of the scene. Moreover, we propose to use this framework in combination with the Riemannian Motion Policies (RMP) as a local reactive planner to enable safe human-robot interactions. We design a collision avoidance policy that models the repulsive motion using the distance and gradient fields from our GP. We show the performance of our reactive planner in an experiment with a UR5e interacting safely and smoothly with a human. A detailed demonstration of the framework and the experimental setup is available at https://usaali.github.io/IDMP-RMP/.**

I. Introduction

In order for robots to collaborate in close proximity to humans, we require them to be reactive and safe. An efficient representation of the environment that facilitates this behaviour is crucial to enabling such applications. For example, a robot manipulator carrying out a task in a shared workspace must react to changes in the environment and do so in a timely, robust and natural manner.

Euclidean distance fields (EDF) are a promising representation for such tasks as they provide smooth collision costs and reason about free space which is important for safety critical applications. Recently, progress has been made towards increasing the computational efficiency of computing discrete EDFs for fast motion planning [7, 4]. However, responding to changes in the environment dynamically has remained a challenge for EDF methods. Moreover, the discrete nature of these EDF representations poses a challenge for smooth distance gradient estimation, required for smooth collision avoidance.

In this work we leverage the Interactive Distance Field Mapping and Planning (IDMP) framework [1] which utilizes Gaussian Processes (GP) to represent a continuous distance and gradient field. Key to our approach, IDMP utilises a local Gaussian Process Distance Field (GPDF) named Frustum Field which constantly takes in new observations from a depth sensor and fuses this information with a global GPDF quickly and robustly. This is achieved by determining implicit semantic information about moving objects via a simple classification process that utilises readily available distance to surface information from both the local and global representation. This is crucial to ensure that the GPDF is kept updated in scenes with dynamic objects like human-robot collaboration scenarios. Otherwise a planner will act on outdated information, which might lead to collisions.

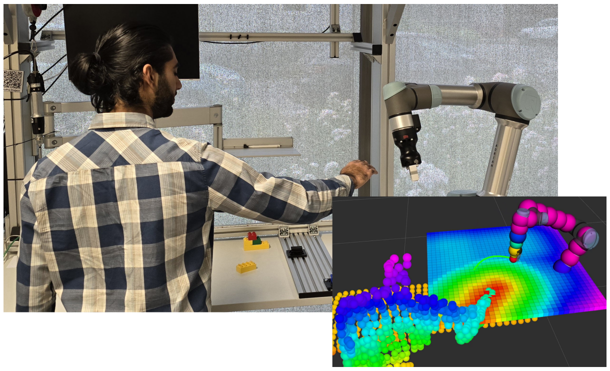


Fig. 1. Dynamic obstacle avoidance using our combined RMP and interactive GPDF framework. (inset) Robot with collision spheres and trajectory traced out in green. The sensor pointcloud is depicted as spheres, which are colored depending on their height. A slice of the distance field is visualized as colored boxes with an color spectrum from red, meaning small distances, to purple, meaning large distances.

For avoiding collisions with moving objects we use a local reactive motion generator called Riemannian motion polices (RMP). RMPs remain to be demonstrated in combination with EDFs in online and dynamic settings [8] and have mainly been used in conjunction with voxel-based occupancy representations [9, 11].

In this paper we formulate motion policies that utilise our interactive GPDF's distance and gradient information. Our dual-GPDF approach in combination with RMPs allows for fast and direct semantic reasoning about moving and static regions in the scene which facilitates reactive and robust downstream control.

We demonstrate the use of RMPs in combination with GPDFs in a human robot collaboration setting where a UR5 robot arm must carry out a task in a shared workspace. We show that our method achieves smoother and more natural reactive behaviour compared to an occupancy-based baseline method.

II. Preliminaries

A. Interactive Distance Field Mapping

For mapping dynamic scenes we utilise the IDMP framework [1] which implicitly models a continuous Euclidean distance field via a reverting GP function [6]. Let S be a surface in a Euclidean space $\mathbb{R}^n$ which is observed with Q noisy measurements $\mathbf{y} = \{y_i\}_{i=1}^{Q}$ at positions $X = \{\mathbf{x}_i\}_{i=1}^{Q}$. We can now regress the occupancy $o(\mathbf{x}) : \mathrm{R}^n \mapsto \mathrm{R}$ by modeling it as a GP

$$o(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')), \tag{1}$$

with $k(\mathbf{x}, \mathbf{x}')$ being the covariance kernel function of the distribution, which is chosen as

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right). \tag{2}$$

We define the value of our measurement on the surface to satisfy

$$y(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in S, \tag{3}$$

with this the occupancy at any point $\mathbf{x}_*$ can be recovered by

$$o(\mathbf{x}_*) = \mathbf{k}(\mathbf{x}_*, X) \left(\mathbf{K}(X, X) + \sigma_y^2 \mathbf{I}\right)^{-1} \mathbf{1}\,, \tag{4}$$

where $\mathbf{k}(\mathbf{x}_*, X) = [k(\mathbf{x}_*, \mathbf{x}_1), ..., k(\mathbf{x}_*, \mathbf{x}_Q)]$ represents the covariance vector between the input points and the query point and $\mathbf{K}(X, X) = \left[\mathbf{k}(\mathbf{x}_1, X)^\top, ..., \mathbf{k}(\mathbf{x}_Q, X)^\top\right]$ the covariance matrix between the input points.

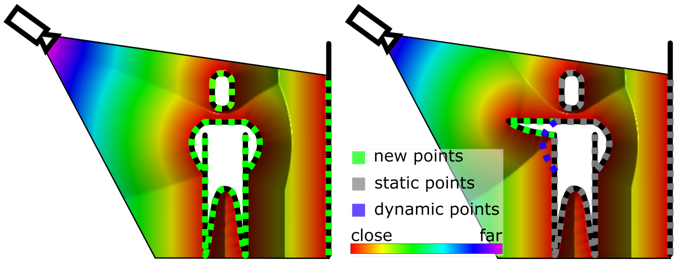
To derive the distance $d(\mathbf{x}_*)$ from the occupancy $o(\mathbf{x}_*)$, the kernel function has to be reverted, resulting in

$$d(\mathbf{x}_*) = \sqrt{-2l^2 * \log\left(\frac{o(\mathbf{x}_*)}{\sigma^2}\right)}\,. \tag{5}$$

Särkkä [12] shows that a GP linearly operated remains a GP and the operation can be directly applied to the kernel itself. Using this fact, we can compute the gradient field $\nabla d(\mathbf{x}_*)$ by differentiating Eq. 5 with respect to the distance. Since the direction of the distance field gradient $\nabla d(\mathbf{x}_*)$ aligns with the direction of the GP gradient $\nabla o(\mathbf{x}_*)$, we can simplify as,

$$\nabla d(\mathbf{x}_*) = \nabla \mathbf{k}(\mathbf{x}_*, X) \left(\mathbf{K}(X, X) + \sigma_y^2 \mathbf{I}\right)^{-1} \mathbf{1}\,. \tag{6}$$

The key aspect of the IDMP framework (see Figure 3) is that it uses a Frustrum Field to fuse and identify the dynamic regions locally before passing the information to the Fused Field that contains the global information. Figure 2 shows the internal update process of IDMP. The background displays the distance field within the sensor's field of view generated by the frustum GPDF. While the fused GPDF is trained on all points from the internal global map, the frustum GPDF only utilizes the latest observations, capturing changes in the scene. By querying the frustum GPDF with the fused GPDF's training points, we can directly retrieve implicit semantics based on distance metrics.



(a) Observation of unknown scene

(b) Update of a moving object

Fig. 2. The Frustum Field is used to extract semantics from the new observation. (a) A completely new observation is added to the Fused Field. (b) For every following observation, the Frustum Field is queried with the points from the Fused Field and semantics are determined according to the resulting distances.

Training points in the fused GPDF are classified as *static* if their queried distance in the frustum GPDF is below a certain threshold, indicating the object has not moved. Training points are classified as *dynamic* when this distance exceeds the sensor noise threshold, indicating that the object has moved. For the final case we query newly observed sensor points with the fused GPDF. Those points with distances greater than a certain threshold are classified as *new* and are fused into the global GPDF.

B. Riemannian Motion Policies

Riemannian Motion Policies (RMP) [10] is a local reactive motion generator that combines multiple simple task-based policies to achieve potentially complex high-level behaviours that would otherwise be difficult to design directly in the configuration space of the robot. Policies are formulated as second-order dynamical systems and weighted by their corresponding Riemannian metric resulting in smooth and expressive paths.

A policy $\mathcal{P}$ is modeled on a Riemann Manifold $\mathcal{M}$ and is defined by the tuple $(f, \mathcal{A})^{\mathcal{M}}$, where f outputs an acceleration based on the states $\mathbf{x}$ and $\dot{\mathbf{x}}$:

$$\ddot{\mathbf{x}}_d = f(\mathbf{x}, \dot{\mathbf{x}}) \quad \text{with} \quad \mathbf{x} \in \mathcal{M}. \tag{7}$$

$\mathcal{A}(\mathbf{x}, \dot{\mathbf{x}})$ is a Riemann Metric for $\mathcal{M}$ that also varies with the state.

Multiple policies can be optimally combined with the weighted sum

$$\mathcal{P}_c = (f_c, \mathcal{A}_c)^{\mathcal{C}} = \left(\left(\sum_i \mathcal{A}_i\right)^+ \sum_i \mathcal{A}_i f_i, \sum_i \mathcal{A}_i\right). \tag{8}$$

To transform policies between the task and configuration-space manifolds, push- and pullback operations are performed, see Ratliff et al. [10] for more details.

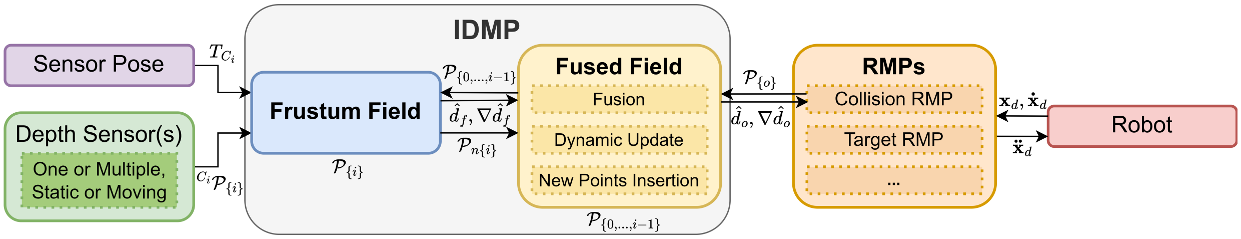


Fig. 3. System diagram of our proposed Framework. IDMP processes the sensor data and fuses this into a global GPDF which is then queried by our RMP policy.

III. Reactive Planning in Interactive Distance Fields

We propose to use the output of the fused GPDF as described in Section II-A in combination with our custom-designed collision avoidance policy for RMP (Section II-B) to safely navigate in dynamic scenes.

Figure 3 shows the proposed system architecture, where IDMP takes as input the depth sensor's data and pose. These inputs are used to generate the local Frustum Field which determines the implicit semantics of the scene and is then used to fuse the new observation with the global GPDF. Our RMP policy queries distance and gradient information from the fused global GPDF to generate accelerations which are passed to a controller for execution on the robot.

For querying distances from the robot to obstacles we use a common link-sphere approximation method [2] (see Fig. 4). We can then query distances for each sphere centre $\mathbf{x}_c$ using Eq. 5 as follows:

$$d(\mathbf{x}_c) = \sqrt{-2l^2 * \log\left(\frac{\mathbf{k}(\mathbf{x}_c, X)\left(\mathbf{K}(X,X) + \sigma_y^2\mathbf{I}\right)^{-1}}{\sigma^2}\right)}. \tag{9}$$

The gradient $\nabla d(\mathbf{x}_c)$ can then be calculated analytically via Eq. 6.

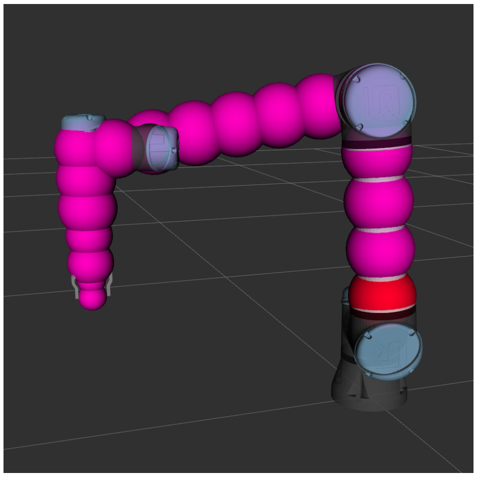


Fig. 4. Visualisation of our link-sphere approximation.

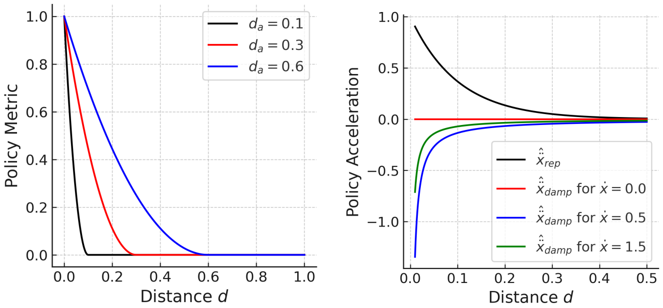


Fig. 5. Plot of the policy acceleration and metric values.

We adapt the collision avoidance policy proposed by Ratliff et al. [10] consisting of a repulsion and a damping component

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_{rep} + \ddot{\mathbf{x}}_{damp}. \tag{10}$$

We model the repulsive motion as a function of the gradient ∇d and distance d from the GPDF,

$$\ddot{\mathbf{x}}_{rep} = \eta_{rep}\nabla d(\mathbf{x})\exp\left(-\frac{d(\mathbf{x})}{\nu_{rep}}\right), \tag{11}$$

with the repulsion gain η_{rep} and the length scale ν_{rep}. The damping term reduces oscillations and is a function of the velocity $\dot{x}$ and distance d described by,

$$\ddot{\mathbf{x}}_{damp} = -\eta_{damp}\left(1 - S\left(\frac{\dot{\mathbf{x}}}{\nu_{damp}}\right)\right) * \frac{\nu_{rep}\dot{\mathbf{x}}}{d(\mathbf{x})}, \tag{12}$$

where $S(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$, η_{damp} is the damping gain and ν_{damp} the damping length scale.

The task-space metric weights the importance of each policy in the resulting acceleration. Thus, we propose a diagonal matrix with values from a smooth activation gate which depends on the distance d and the activation parameter d_a as

$$\mathcal{A} = \mathbf{I} * \begin{cases} \frac{d^2}{d_a^2} - 2 * \frac{d}{d_a} + 1 & \text{for } d < d_a \\ 0 & \text{for } d > d_a \end{cases} \tag{13}$$

The effect of our collision avoidance policy on the output acceleration and task-space metric as a function of distance is shown in Fig. 5 with varying parameter values.

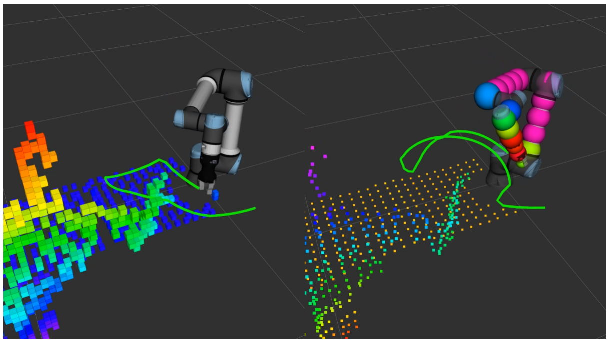


Fig. 6. Comparison of trajectories for the baseline (left) and our proposed method (right). The robot is tasked to move between two waypoints, as the human moves their arm in the way.

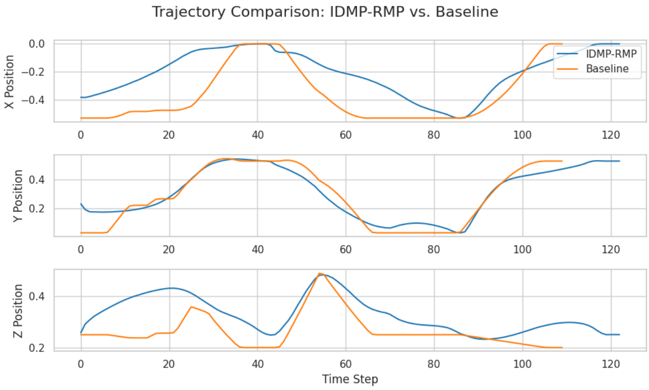


Fig. 7. Comparison of IDMP-RMP and Baseline trajectory.

Additionally to our collision policy we use a target attractor policy, a joint limit policy and a velocity limit policy as described in [10, 3].

IV. Experiments

We evaluate our method in a mock human-robot interaction scene where the robot is tasked to cycle between two waypoints. During the execution, a human enters the workspace and places their arm in the way of the robot.

We compare the behaviour of our framework against an occupancy-based reactive method implemented in ROS package MoveIt. This baseline method builds an Octomap [5] which is continuously updated with the sensor input. A trajectory is then planned using the Bi-directional Fast Marching Tree (BFMT*) algorithm [13]. During execution the trajectory is checked for possible collisions which then triggers replanning.

Figure 6 shows the resulting trajectories for both our method and the baseline. Compared to the baseline Sour method is able to handle dynamic obstacles reactively whereas the baseline method awkwardly stops as it replans.

To characterise the behaviour of our system we show trajectory plots in Fig. 7 and compute smoothness metrics in Table I. Acceleration and jerk are computed numerically as the second and third time derivatives of position. Change in curvature is computed as

$$\Delta\kappa = \frac{1}{n}\sum_{i=1}^{n-1} \|\kappa_{i+1} - \kappa_i\|, \tag{14}$$

with the curvature defined as

$$\kappa(t) = \frac{\|\dot{\mathbf{x}} \times \ddot{\mathbf{x}}\|}{\|\dot{\mathbf{x}}\|^3}. \tag{15}$$

As can be seen the trajectories produced by our method result in much smoother trajectories. Notably the mean squared jerk and change in curvature were approximately 2x and 3x lower, respectively, than the baseline.

TABLE I
Comparison of smoothness metrics between IDMP-RMP and Baseline trajectories.

Metric	IDMP-RMP	Baseline	Difference
Variance of Acceleration (X)	**2.30e-05**	3.08e-05	-25.32%
Variance of Acceleration (Y)	**3.07e-05**	5.47e-05	-43.88%
Variance of Acceleration (Z)	**1.03e-05**	4.13e-05	-75.06%
Total Jerk	**0.602**	0.832	-27.64%
Mean Squared Jerk	**2.63e-05**	5.28e-05	-50.19%
Mean Curvature Change	**0.302**	0.925	-67.35%

V. Conclusion

In this paper we presented a framework that utilises interactive distance fields based on GPs in combination with RMPs to reactively plan in dynamic scenes. We demonstrated this in a human-robot collaboration scenario where a robot arm smoothly avoids colliding with a person entering its workspace while continuing to carry out its task.

While our work utilises implicit semantic information based on moving regions of the scene, future work aims to use more fine grained information such as distinguishing between human and non-human moving objects. Exploring how this can be incorporated into our RMP formulation to achieve more natural and nuanced avoidance behaviour in more complex collaborative settings is a promising avenue.

References

[1] Usama Ali, Lan Wu, Adrian Mueller, Fouad Sukkar, Tobias Kaupp, and Teresa Vidal-Calleja. Interactive Distance Field Mapping and Planning to Enable Human-Robot Collaboration. *arXiv preprint arXiv:2403.09988*, 2024.

[2] Lucian Balan and Gary M. Bone. Real-time 3D Collision Avoidance Method for Safe Human and Robot Coexistence. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 276–282, 2006. doi: 10.1109/IROS.2006.282068.

[3] Ching-An Cheng, Mustafa Mukadam, Jan Issac, Stan Birchfield, Dieter Fox, Byron Boots, and Nathan Ratliff. Rmpflow: A geometric framework for generation of multitask motion policies. *IEEE Transactions on Automation Science and Engineering*, 18(3):968–987, 2021. doi: 10.1109/TASE.2021.3053422.

[4] Luxin Han, Fei Gao, Boyu Zhou, and Shaojie Shen. Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4423–4430, 2019.

[5] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous robots*, 34:189–206, 2013.

[6] Cedric Le Gentil, Othmane-Latif Ouabi, Lan Wu, Cedric Pradalier, and Teresa Vidal-Calleja. Accurate Gaussian-Process-based Distance Fields with applications to Echolocation and Mapping. *IEEE Robotics and Automation Letters*, 2023.

[7] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1366–1373, 2017. doi: 10.1109/IROS.2017.8202315.

[8] Michael Pantic, Cesar Cadena, Roland Siegwart, and Lionel Ott. Sampling-free obstacle gradients and reactive planning in neural radiance fields. In *Proc. of Workshop on" Motion Planning with Implicit Neural Representations of Geometry" at ICRA*, 2022.

[9] Michael Pantic, Isar Meijer, Rik Bähnemann, Nikhilesh Alatur, Olov Andersson, Cesar Cadena, Roland Siegwart, and Lionel Ott. Obstacle avoidance using Raycasting and Riemannian Motion Policies at kHz rates for MAVs. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1666–1672, 2023. doi: 10.1109/ICRA48891.2023.10161365.

[10] Nathan D Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian motion policies. *arXiv preprint arXiv:1801.02854*, 2018.

[11] Victor Reijgwart, Michael Pantic, Roland Siegwart, and Lionel Ott. Waverider: Leveraging hierarchical, multi-resolution maps for efficient and reactive obstacle avoidance. *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[12] Simo Särkkä. Linear operators and stochastic partial differential equations in Gaussian process regression. In *Proc. of International Conference on Artificial Neural Networks (ICANN)*, pages 151–158. Springer, 2011.

[13] Joseph A Starek, Javier V Gomez, Edward Schmerling, Lucas Janson, Luis Moreno, and Marco Pavone. An asymptotically-optimal sampling-based algorithm for bi-directional motion planning. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2072–2078. IEEE, 2015.