Learning Implicit Environment Field

Xueting Li[1], Sifei Liu[2], Shalini De Mello[2], Xiaolong Wang[3], Ming-Hsuan Yang[1], and Jan Kautz[2]

[1]UC Merced
[2]NVIDIA
[3]UC San Diego

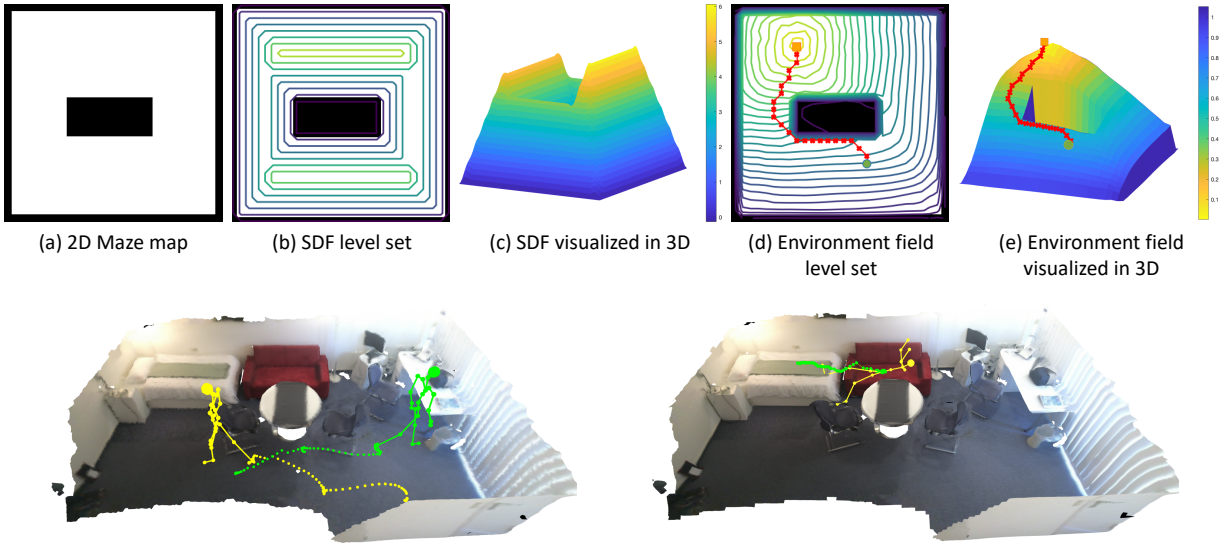

Figure 1: **Implicit environment fields in the 2D and 3D space.** By utilizing implicit functions, we learn a continuous environment field that encodes the reaching distance between any pair of points in the 2D and 3D space. We present utilization of the environment field for 2D maze navigation in (d)-(e) and human scene interaction modeling in 3D scenes in (f). Note that we flip the environment field in (e) upside down for visualization purposes.

Abstract

In this work, we propose a new representation for scenes dubbed as the environment field that encodes the "reaching distance" – the distance between any position in the scene to a goal along a feasible trajectory. We demonstrate that this new representation can directly guide the dynamic behaviors of agents[1] inside the environment. We learn the environment field using a neural implicit function from discretely sampled training data and showcase its application in human trajectory prediction in indoor environments and agent navigation in mazes. Furthermore, to generate semantically plausible trajectories for humans, we learn a generative model that predicts regions where humans commonly appear and constrain humans within such regions via the environment field. Extensive experiments demonstrate that the proposed method can produce both feasible and plausible trajectories efficiently and accurately.

1. Overview

In this work, we propose a new representation for scenes dubbed as the "environment field" that can directly guide the dynamic behaviors of agents in a scene, e.g. navigating an agent to reach a given goal in a physically plausible manner. We showcase the application of the proposed environment field in both human trajectory modeling in 3D indoor environments and agent navigation in 2D mazes and

[1]We use a general definition of "agent" that either represents robots in 2D mazes or humans in 3D scenes.

observe comparable if not better results than state-of-the-art methods.

1.1. Environment Field Formulation

Given a scene with an assigned goal position, we aim to learn a continuous environment field, whose value at each position represents its reaching distance towards a given goal.

To compute the reaching distance, we model the environment field as a wave propagation [8], which states that the distance between the initial position and the goal is equivalent to the minimal amount of time required by a wave, starting from the initial position, to propagate its front boundary to reach the goal. Specifically, we denote all positions that can be reached by an agent as the "accessible region", while other positions in the scene as "obstacles". Formally, given a spreading wave τ, i.e., a closed curve in this environment, the way this curve spreads depends on a speed function $f(x)$ defined on each location. The current position of this spreading wave τ can be modeled by an arrival time function $u(x)$ w.r.t. x, starting from the goal. When $f(x) > 0$, we can formulate it as a continuous shortest-path problem [8] through the Eikonal equation:

$$\|\nabla u(x)\| f(x) = 1, \quad x \in \Omega, \tag{1}$$

where Ω denotes the feasible region, ∇ denotes the gradient, $\|\cdot\|$ is the Euclidean norm, and $u(x_e) = 0$ at the goal location x_e. We specify the environment layout with the speed function as follows: at obstacles, we set the wave expansion speed $f(x)$ to an extremely small positive value, as the wave cannot go through them. Within the accessible region, the speed is set to a constant of 1. The time at which the wavefront reaches a point x, i.e., $u(x)$ is obtained by solving (1), thus forming the continuous landscape.

1.2. Environment Field Learning

We represent the environment field as a mapping from location coordinates $x \in \mathcal{R}^2$ to an arrival time $u(x) \in \mathcal{R}$ using implicit functions. This neural implicit function can be learned using discretely sampled pairs of $\{x, u(x)\}$ and yet represent a continuous field once trained. Therefore, we can safely resort to an analytical method, e.g., the fast marching method (FMM) [8], that solves $u(x)$ in a regular grid.

We then learn an implicit function (see Fig. 2 (a)) to regress the reaching distance at each grid cell given the cell's coordinates (normalized to $[-1, 1]$) as inputs.

1.3. Trajectory Search using Environment Field

Given the learned implicit function in Section 1.2, we search a trajectory from a start position x_s to the goal position x_e.

Taking 2D mazes as an example, we assume that an agent moves by one grid cell along one of $2^3 = 8$ possible directions at each time, as shown in Fig. 5(a). We then query the learned implicit function for the reaching distance values at all possible positions that the agent can reach in one time step and moves the agent to the position with the smallest reaching distance. We keep updating the agent position until it reaches the goal.

1.4. Conditional Environment Field

In this section, we present how to generalize the environment field to arbitrary goal positions and scenes.

Environment field for arbitrary goal positions. To learn an implicit function that predicts different environment fields for different goal positions in the same environment, we utilize a conditional implicit function [6] as shown in Fig. 2(b). The input to the implicit function is the concatenation of both the goal coordinates and the query position coordinates. The output is the reaching distance from the query position to the goal, i.e., the environment field value. During training, we randomly sample a pair of empty grid cells and set one of them as the goal. We train the implicit function with the reaching distance computed by FMM.

Environment field for arbitrary goal and environment. To further extend the implicit function to an arbitrary environment, we propose a context-aligned implicit function as shown in Fig. 2(c). Given a scene context (i.e., a 2D maze map), we first extract scene context features by a fully convolutional environment encoder. Then, we forward the concatenation of (a) the goal coordinates, (b) the query position coordinates, and (c) the scene context features aligned to the query position into the implicit function and regress the reaching distance value for the query position.

To summarize, for the maze navigation task, we learn the context-aligned implicit function (Fig. 2(c)) using different mazes, along with the reaching distance computed by FMM as training data. During inference, we predict an environment field for an unseen maze with a given goal position and search for a feasible trajectory from any start position to the goal using the strategy discussed in Section 1.3. Fig. 5 demonstrates learned environment fields along with searched trajectories for unseen maze maps.

2. Human Navigation in Indoor Environments

Going beyond the toy example of agent navigation in 2D mazes, in this section, we discuss how to apply the environment field to real-world 3D scenes. Given the point cloud of each scene, we aim to generate a geometrically feasible and semantically plausible trajectory for a human from a start position to the goal. We then align an existing sequence of human poses onto the generated trajectory (see Fig. 4). While the "accessible" and "obstacle" locations in 2D mazes are well-defined, the "accessible region" for humans in 3D scenes is hard to identify and usually requires prior knowledge of human behavior. We propose a learning-based solution that generates the accessible region

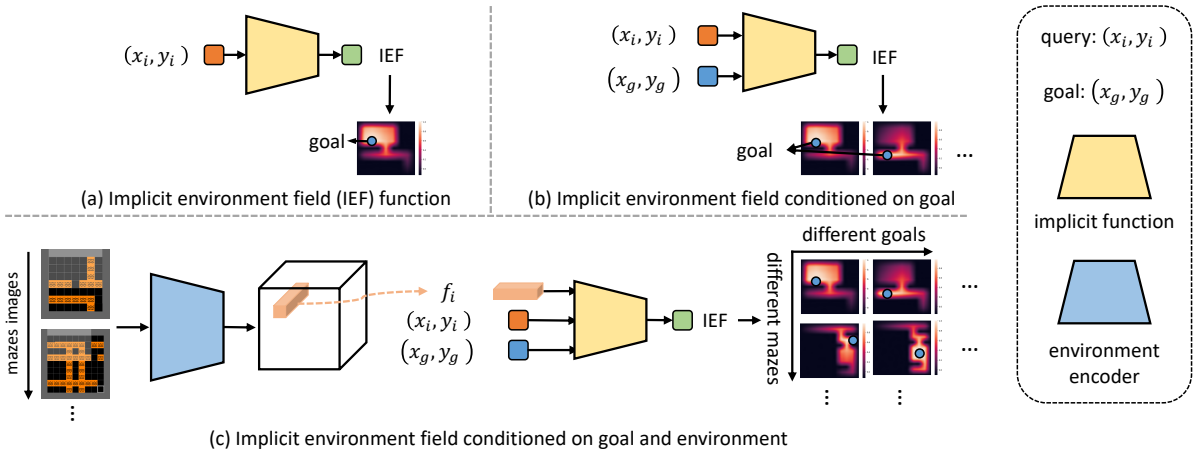

Figure 2: **Variants of implicit environment functions (IEFs).** The IEF in (a) assumes a fixed environment and a fixed goal position; (b) generalizes to different goal positions in the same environment; (c) further generalizes to different goal positions in different environments. Goal positions are represented by blue circles. In this figure, we use the 2D maze as an example, but models in (a) and (b) can be generalized to 3D scenes by simply changing the inputs from 2D coordinates to 3D coordinates.

in 3D scenes and models complex human actions, e.g., sitting down, etc. (see Section 2.1).

2.1. 3D Scene Environment Field

Data-driven accessible region in 3D scenes. We learn the distribution of human torso locations in 3D scenes, conditioned on the scene context, through a variational autoencoder (VAE) [4]. As shown in Fig. 3, given a scene point cloud, we first take the global feature produced by the pooling layer of a PointNet [7] as the scene context feature. We map the context feature together with human torso location observations to a normal distribution using an encoder. A decoder then reconstructs the human torso location given the concatenation of a sampled noise from the normal distribution and the scene context feature. We learn this conditional VAE using the reconstruction objective on human torso locations together with the Kullback–Leibler (KL) divergence objective [4]. During inference, we sample random noise from a standard normal distribution to generate feasible locations for human torsos to formulate the accessible region in a 3D scene.

Environment field learning for 3D scenes. Different from the 2D maze scenarios, here we model the implicit function on the full 3D space. Due to the limited representation capacity of implicit functions [9], we consider an arbitrary goal position in a fixed scene environment. As shown in Fig. 2 (b), the input to the implicit function is the concatenation of both the goal coordinates and the query position's coordinates. The output is the reaching distance from the query position to the goal. To utilize the FMM discussed in Section 1, we discretize a 3D scene to a $64 \times 64 \times 64$ voxel grid and mark all voxel cells within the generated accessible region as "accessible" and other voxel cells as "obstacles". During inference, we utilize the search strategy discussed

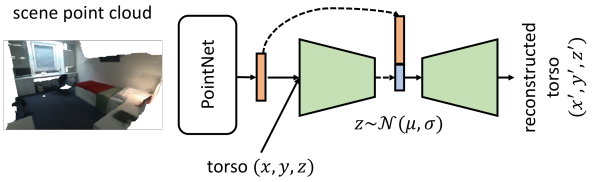

Figure 3: **A conditional VAE for accessible space generation.**

in Section 1.3 to predict a trajectory for humans from the start location to the goal. We set the step size as average human step size and the possible directions as the $3^3 = 27$ locations the human can take in the 3D space.

Pose-dependent trajectory search. We further optimize the predicted trajectory based on a given pose sequence and guarantee that the human is navigated towards physically feasible locations. To this end, at each step, we use the real step size computed from adjacent locations in the pose sequence instead of the predefined average step size. For each possible location the human can reach within one step size, we check if (a) the human is well supported and (b) the human is not colliding with other objects at these locations and move the human to the location with the smallest reaching distance value while obeying these two constraints. Similar to [5], to determine if a pose is well supported, we check if the torso, the left lap, and the right lap joints of sitting poses, as well as the feet joints of standing poses have non-positive signed distances to the scene surface. To determine if a pose collides with other objects, we check if all joints of the pose except the aforementioned support joints have non-negative signed distances. Fig. 4 (b), (d) and (e) illustrate the searched trajectory along with the aligned human poses.

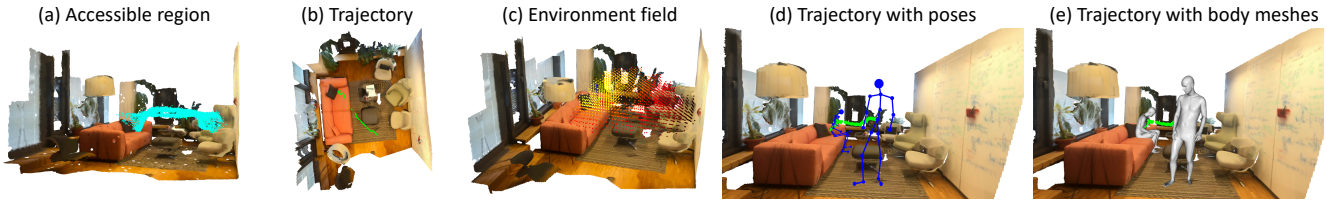

Figure 4: **Dynamic human motion modeling on PROX [3]** The accessible region in (a) includes generated torso locations by the VAE model discussed in Section 2.1. In the environment field in (c), the brighter a point is, the closer it is to the goal.

3. Experiments

We evaluate the proposed environment field on agent navigation in 2D mazes (Section 3.1) and dynamic human motion modeling in 3D scenes (Section 3.2).

3.1. Agent Navigation in 2D Mazes

We evaluate the proposed environment field on the Minigrid [2] and Gridworld maze datasets [10], which include 2D mazes with randomly placed obstacles (see Fig. 5(a)). Fig. 5(b) and (c) show the learned environment field and the level sets (each level set consists of locations of the same reaching distance to the goal) respectively. Figure 5(a) further demonstrates the effectiveness of using the learned environment field in navigating an agent to the goal while avoiding collision with obstacles.

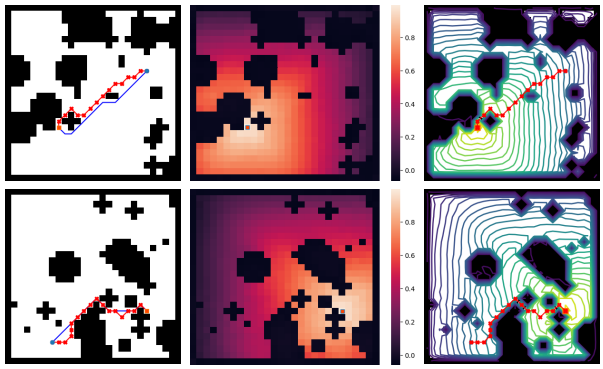

Figure 5: **Grid-world maze navigation results.** Blue paths in (a) are planned by FMM while red paths are searched by the learned environment field. Green circles and orange squares represent starting positions and goal positions. In (b)(c), the brighter an area or a contour is, the closer it is to the goal.

3.2. Indoor Human Navigation Modeling

Next, we learn an implicit environment field to model long-term dynamic human motion on the PROX dataset [3, 11] as discussed in Section 2.

Data-driven human accessible region learning. We show qualitative results of the generated accessible region discussed in Section 2.1 in Fig. 4 (a). The generated accessible region uniformly spreads out in the 3D room and is plausible for all kinds of human actions. For instance, the points on the chair and sofa are feasible locations for sitting human torsos, while the points in midair are possible locations for walking human torsos. On top of the generated accessible region, we show the learned environment field in Fig. 4 (c), where the closer a point is to the goal, the smaller its reaching distance value is, i.e., the brighter the point is in the figure. Finally, we search a trajectory for a given human pose sequence as discussed in Section 2 and shown in Fig. 4 (b). The trajectory successfully avoids colliding with furniture in the room while leading the human towards the goal. Furthermore, the human poses are also plausible at each location on the trajectory (Figure 4 (d) and (e)), demonstrating the effectiveness of the pose-dependent trajectory search process discussed in Section 2.

Table 1: Quantitative evaluations of human motion trajectory prediction on the PROX dataset. "AFF" is the same as in Table **??**. "Support" and "free' indicate the ratio of poses that are either well supported or collision-free, respectively. "Valid" indicates the ratio of poses that obey both constraints.

Method	distance ↓	support ↑	free ↑	valid ↑
HMP [1]	0.079	89.36	94.99	86.03
Ours (w/o AFF)	**0.021**	87.79	95.73	85.69
Ours	0.026	**94.44**	**96.84**	**92.14**

Comparison with HMP [1]. We quantitatively compare against the PathNet in [1] using the distance metric discussed above as well as the ratio of valid poses to all poses on the trajectory. As described in Section 2, we define a pose as valid if it is both well supported and not collision-free in the scene. The quantitative evaluations are present in Table 1. Our method is more effective at navigating humans towards goal positions compared to [1] as shown in the second column. Furthermore, after using the pose-dependent trajectory search process discussed in Section 2, our model is able to better fit the human poses to the scene, as shown in the last column in Table 1.

4. Conclusion

In this paper, we propose the environment field that encodes the reaching distance between a pair of points in either 2D or 3D space, with implicit functions. The learned environment field is a continuous energy surface that can navigate agents in 2D mazes in dynamically changing scene environments. We further extend the environment field to 3D scenes to model dynamic human motion in indoor environments. Extensive experiments demonstrate the effectiveness of the proposed method in solving 2D mazes and modeling human motion in 3D scenes.

References

[1] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qizhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *ECCV*. 2020. 4

[2] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018. 4

[3] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *ICCV*, Oct. 2019. 4

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[5] Xueting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Putting humans in a scene: Learning affordance in 3d indoor environments. In *CVPR*, 2019. 3

[6] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, June 2019. 2

[7] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 3

[8] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 1996. 2

[9] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 3

[10] Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. In *NeurIPS*, 2016. 4

[11] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3d people in scenes without people. In *CVPR*, 2020. 4