This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3230325

Real-time Neural Dense Elevation Mapping for Urban Terrain with Uncertainty Estimations

Bowen Yang[1], Qingwen Zhang[1], Ruoyu Geng[1], *Graduate Student Member, IEEE*, Lujia Wang[2], *Member, IEEE*, and Ming Liu[3], *Senior Member, IEEE*

***Abstract*—Having good knowledge of terrain information is essential for improving the performance of various downstream tasks on complex terrains, especially for the locomotion and navigation of legged robots. We present a novel framework for neural urban terrain reconstruction with uncertainty estimations. It generates dense robot-centric elevation maps online from sparse LiDAR observations. We design a novel pre-processing and point features representation approach that ensures high robustness and computational efficiency when integrating multiple point cloud frames. A generative Bayesian model then recovers the detailed terrain structures while simultaneously providing the pixel-wise reconstruction uncertainty. We evaluate the proposed pipeline through both simulation and real-world experiments. Our approach achieves high-quality terrain reconstruction with real-time performance on a mobile platform, and the uncertainty estimates may further benefit the downstream tasks of legged robots. (See https://kin-zhang.github.io/ndem/ for more details.)**

***Index Terms*—AI-Enabled Robotics, Legged Robots, Mapping.**

I. Introduction

RECENT research presents promising progress in enabling mobile robots to operate in complex environments, where the scene perception quality is essential to task performance. Although extensive research has been conducted on 3D scene reconstruction with detailed structures [1]–[4], the high computational costs limit their applications in various robotic applications like locomotion or local navigation. Therefore, it's essential to obtain an efficient online scene representation, which is challenging due to the sparsity of observations, occlusions, and the inaccuracy of sensors and odometry [5].

Due to the difficulty of directly recovering the scene from a single sparse and occluded observation frame, it's necessary to utilize the history frames. Fankhauser *et al.* [6], [7] gradually construct and update a robot-centric elevation map using the incoming observations. The raw elevation map is further fed into a neural network to recover the occluded regions [8]. However, converting the raw observations into maps as the network inputs may lose information useful to the reconstruction task. Hoeller *et al.* [5] propose a neural scene reconstruction approach that adopts dense point clouds from depth cameras as the inputs and recurrently refines the previous results, while the recurrent structure may introduce the potential risk of accumulating reconstruction errors.

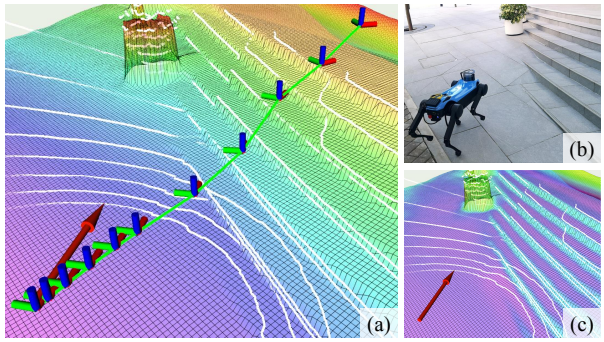

Fig. 1. We present an online neural dense elevation mapping approach that recovers detailed urban terrain structures from sparse and noisy 16-channel LiDAR observations (a)(b), while additionally providing the terrain reconstruction uncertainty (c). The framework can further benefit the downstream locomotion and navigation tasks, especially for legged robots.

To solve the challenges caused by occlusions and input inaccuracy, Miki *et al.* [9] adopt traditional inpainting approaches and filter-based methods to generate and denoise dense elevation maps. Due to the strong feature recovery capabilities, generative neural networks are introduced in [5], [8], [10], [11] to eliminate the noises and generate rational features in the occluded regions. However, directly using the inpainted maps in robotic applications may cause danger due to the lack of concrete terrain information in the occluded regions. Therefore, providing the map reconstruction uncertainty is essential for the robustness of the downstream tasks.

This letter focuses on the urban terrain reconstruction task using elevation maps. We present an end-to-end framework for online neural dense elevation mapping with measurable uncertainties using sparse LiDAR observations (Fig. 1). A novel point cloud pre-processing approach maintains a group of statistical point features to represent the height distribution of the points inside each grid, enabling efficient integration of point frames. A generative Bayesian model then returns dense elevation maps with detailed terrain structures from the noisy and sparse point features based on the prior knowledge of urban environments, outperforming the traditional approach

Manuscript received: August 4, 2022; Revised: October 31, 2022; Accepted: November 30, 2022. This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Guangdong Basic and Applied Basic Research Foundation, under project 2021B1515120032, awarded to Prof. Lujia Wang. (corresponding author: Lujia Wang.)

[1]Bowen Yang, Qingwen Zhang and Ruoyu Geng are with the Hong Kong University of Science and Technology, Hong Kong SAR, China. {byangar,qzhangcb,rgengaa}@connect.ust.hk

[2]Lujia Wang is with the Hong Kong University of Science and Technology, Hong Kong SAR, China, and also with the Clear Water Bay Institute of Autonomous Driving. eewanglj@ust.hk

[3]Ming Liu is with the Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, 511400, Guangdong, China, and HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen. eelium@ust.hk

Digital Object Identifier (DOI): see top of this page.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 20,2022 at 08:32:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3230325

[9] in several urban scenarios. Compared with the state-of-the-art neural scene reconstruction approaches [5], [8], our approach additionally provides the reconstruction uncertainty for each grid, which may further improve the robustness of various downstream tasks. Our main contributions include:

- We propose an end-to-end framework for robust online neural terrain reconstruction to generate dense elevation maps from noisy and sparse LiDAR observations.
- We design a computational-efficient point cloud pre-processing approach to represent and maintain essential point features for high-quality mapping.
- We develop a generative Bayesian model that recovers detailed urban terrain structures and provides pixel-wise reconstruction uncertainty.
- We deploy our approach on an AGX Xavier for high-quality terrain mapping in real-world scenarios with real-time performance and present motivating examples to show its benefits to downstream robotic tasks.

II. Related Work

A. 3D Scene Reconstruction Approaches

Extensive research has been conducted on 3D scene reconstruction. Neural Radiance Fields (NeRFs) are constructed using multi-view images [12]–[14] to represent large-scale urban environments [1], [2]. However, they normally require extensive observation of the scene for better performance. Truncated Signed Distance Fields (TSDFs) are widely used to represent detailed structures and can be used to incrementally build Euclidean Signed Distance Fields (ESDFs) as proposed in Voxblox [3] for navigation. VDBFusion [4] improves the data structure of Voxblox [3] for better efficiency and more accurate ray-casting. Nevertheless, it still brings heavy computation and memory burdens to mobile devices and might be unsuitable for applications with online mapping requirements.

Some approaches describe the volumetric occupancy or terrain height to simplify the calculation. Hornung *et al.* [15] propose OctoMap for probabilistic 3D occupancy estimation. Jia *et al.* [16] develop an accelerator for real-time OctoMap building on embedded platforms. Stepanas *et al.* [17] adopt GPU for online occupancy map generation. Elevation maps are widely used as an efficient representation of terrain features. Fankhauser *et al.* [6], [7] present a robot-centric probabilistic elevation mapping approach that also returns the uncertainty from the sensor noise and state estimation inaccuracy. Miki *et al.* [9] implement a GPU pipeline for efficient elevation mapping with various post-processing functions. However, these approaches may fail to recover dense and detailed terrain features when the observation is quite sparse. Our approach provides online dense elevation mapping from sparse LiDAR frames using a neural network. Different from the uncertainty in [6], [7], our approach additionally considers the uncertainty from map inpainting as it generates dense maps, which may better benefit the downstream tasks for robust performance.

B. Scene Completion and Map Inpainting

Various map inpainting and scene completion methods leverage data-driven approaches or the prior information of terrain structures to solve the occlusion problem and optimize the reconstruction results. Existing works use learning-based approaches for 2D occupancy maps inpainting [11], [18], or 3D semantic scene completion [19], [20]. Hoeller *et al.* [5] implement an online neural scene reconstruction method to generate 3D occupancy voxels for structured terrains using point frames from depth cameras. Different from [5], our approach generates 2.5D elevation maps using sparse LiDAR frames and also provides the reconstruction uncertainty.

Some recent work focuses on online elevation map inpainting. Miki *et al.* [9] provide various traditional inpainting methods such as Telea [21], Navier-Stokes [22], and minimum filter [23], and perform plane segmentation to recover the empty grids and extract geometric surfaces. Stölzle *et al.* [8] adopt self-supervised learning to train a neural network that takes raw elevation maps as input and recovers the occluded regions. We present an end-to-end framework that efficiently pre-processes the raw sparse LiDAR observations to obtain the point features and then adopts a generative Bayesian model for real-time local dense elevation mapping. Our approach better recovers the details of urban terrain structures compared with the traditional inpainting methods [21], [23] in [9] and further provides the reconstruction uncertainty compared with [8].

III. Methodology

This section explains our problem formulation and how we design the dataset generation environment, data pre-processing approach, network architecture, and training objectives.

A. Problem Formulation

We focus on the online mapping scenarios, where the mobile robot continuously observes the static environment and receives sparse 3D LiDAR point clouds $\mathbf{P} = \{\mathbf{p}_k | \mathbf{p}_k = (x_k, y_k, z_k)^T \in \mathbb{R}^3\}$. The terrains are represented as robot-centric 2D grid maps $\mathbf{y}_h$, where each grid contains a height value of that location. Our elevation mapping process is to obtain dense height estimations $\hat{\mathbf{y}}_h$ that recover the terrain features given sequential LiDAR frames $\{\mathbf{P}^{(i)}\}$ and robot's motion estimates $\mathbf{e}^{(i)} = (\mathbf{R}^{(i)}, \mathbf{t}^{(i)})$, $\mathbf{R} \in \mathbb{R}^{3\times3}$, $\mathbf{t} \in \mathbb{R}^3$.

Generating dense elevation maps to represent detailed terrain features is an ill-posed problem due to observation inaccuracy and occlusions. Therefore, we specifically concern the urban scenarios with the assumption that the terrains are structured with sharp edges, flat surfaces, or smooth slopes. The point frame $\mathbf{P}^{(i)}$ at time i is processed to maintain $\mathbf{x}^{(i)}$ which is a 7-channel feature map containing the statistical

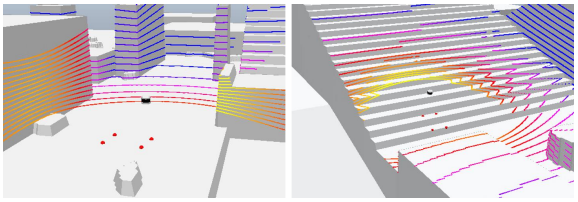

Fig. 2. Examples of the simulation environment. The generated maps contain diverse types of urban terrains. A 16-channel LiDAR moves and observes the environment. The LiDAR's orientations change with the local topography considering the robot's foot configuration (red balls).

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 20,2022 at 08:32:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3230325

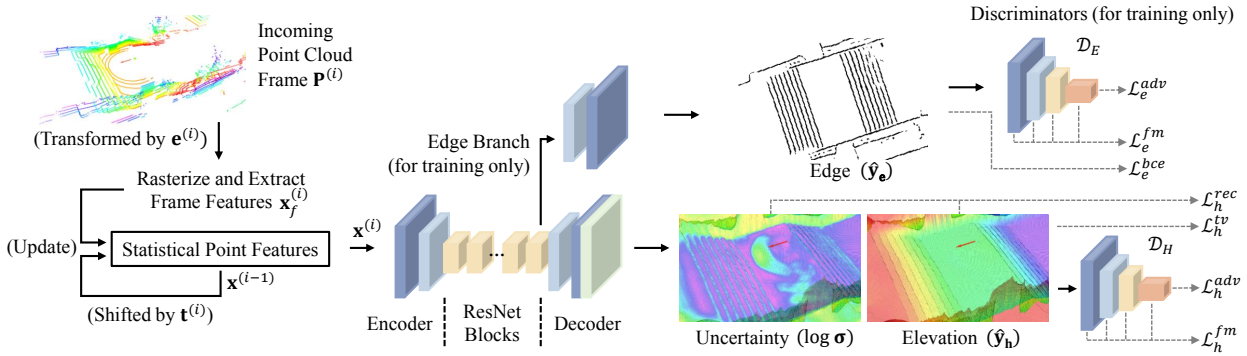

Fig. 3. The proposed elevation mapping pipeline. It first pre-processes the observed sparse point clouds to update the statistical point features. A generative Bayesian model then encodes the current point features to generate the dense elevation map and the pixel-wise reconstruction uncertainty while simultaneously returning a binary edge map for multitask learning in the training phase. Two discriminators are adopted to guide the learning of elevation map and edge map generation using adversarial loss $\mathcal{L}^{adv}$ and feature matching loss $\mathcal{L}^{fm}$ (See Section III-E for details of the loss terms).

features of the history frames (explained in Section III-C). Then a generative model recovers the urban terrain features and generates dense maps. The concept of Bayesian learning [24] is introduced to measure the mapping uncertainty.

B. Dataset Generation

We prepare 100 urban maps in simulation (Fig. 2). Each map is $24 \times 24\,\mathrm{m}$ with $60\,\%$ being flat grounds, $10\,\%$ being stairs, $10\,\%$ being slopes, and $20\,\%$ being boxes or irregular steps to represent other obstacles. The terrain generation parameters are uniformly sampled in pre-specified ranges: $[0.20, 0.35]\mathrm{m}$ and $[0.08, 0.25]\mathrm{m}$ for stairs' width and height, $[0.18, 0.60]\mathrm{rad}$ for the slope angle, $[0.08, 3.20]\mathrm{m}$ and $[-0.50, 2.00]\mathrm{m}$ for the side length and height of boxes and steps. A 16-channel LiDAR is placed $0.50\,\mathrm{m}$ above the ground using the configuration of a Jueying Mini [25] quadrupedal robot (Fig. 1(b)). We move the robot along a predefined trajectory around the map center with a linear velocity uniformly sampled in $[0.0, 1.0]\mathrm{m/s}$. We calculate the robot's foot positions using the robot configuration and the local topography, and obtain the corresponding LiDAR positions and orientations, which are slightly perturbed to simulate the body vibrations from locomotion. The ground truth height maps $\mathbf{y}_h$ are $5 \times 5\,\mathrm{m}$ robot-centric patches with a resolution of $0.04\,\mathrm{m}$. We also generate the ground truth binary edge maps $\mathbf{y}_e$ using the Canny edge detector as done in [24] for multitask learning [26] in Section III-D.

The observations are augmented to reduce the sim-to-real gap [5]. When a point cloud is received, we add uniformly sampled values in $[-0.02, 0.02]\mathrm{m}$ on point coordinates to mimic the sensory noise. The ground truth robot odometry is also perturbed in each step with a uniformly sampled translation in $[-0.02, 0.02]\mathrm{m}$ and a rotation in $[-0.04, 0.04]\mathrm{rad}$ to simulate the inaccuracy from pose estimation.

We define a grid cell inside the mapping region to be observed if at least one data point is found at that cell during the robot motion. Since the observations right after initialization are too sparse for meaningful results, we calculate the "observation ratio" of a patch, which is the number of observed grids divided by the total number of grids in that patch, to filter out invalid samples. A sample will be dropped if the patch's observation ratio is less than $25\,\%$.

C. Representation of Point Features

We design an efficient data pre-processing approach to represent the distributions of the point height values in each grid, as they reflect the terrain structures. For example, the height values of points on the flat ground gather around the height of the plane, which approximately follows a Gaussian distribution. While on a vertical edge, the points may distribute uniformly along the vertical surface. We adopt Gaussian models to represent the point distribution properties and obtain the statistical features for the points in each grid to generate a 7-channel feature map $\mathbf{x}^{(i)}$ at time i as the network input. The statistical features (in upper case) contain the number of points $C^{(i)}$, the mean and variance of height values $E^{(i)}(Z)$ and $Var^{(i)}(Z)$, and the mean and variance of the maximum and the minimum height over time $E^{(i)}(Z_{max})$, $Var^{(i)}(Z_{max})$ and $E^{(i)}(Z_{min})$, $Var^{(i)}(Z_{min})$.

As shown in the left part of Fig. 3, when we receive a point frame $\mathbf{P}^{(i)}$, we first transform it to the map coordinate with the motion estimates $\mathbf{e}^{(i)}$ and rasterize it inside the mapping region to extract grid-wised features (in lower case) for this single frame. Each grid in the frame feature $\mathbf{x}_f^{(i)}$ contains the number of points $c^{(i)}$, the sum of height values $\sum z^{(i)}$, the sum of the squared height values $\sum(z^2)^{(i)}$, and the maximum and minimum height values $z_{max}^{(i)}$ and $z_{min}^{(i)}$. The empty grids are filled with zeros. Next, the previous statistical features $\mathbf{x}^{(i-1)}$ are shifted by the translational terms of $\mathbf{e}^{(i)}$ to compensate for the ego-motion and then updated using $\mathbf{x}_f^{(i)}$. For instance, the mean of the height values is updated as the overall sum of height values divided by the new total number of points:

$$C^{(i)} = C^{(i-1)} + c^{(i)}, \tag{1}$$

$$E^{(i)}(Z) = \frac{C^{(i-1)} \times E^{(i-1)}(Z) + \sum z^{(i)}}{C^{(i)}}, \tag{2}$$

and based on the relationship $Var(z) = E(z^2) - E^2(z)$, the variance is updated as:

$$E^{(i-1)}(Z^2) = Var^{(i-1)}(Z) + (E^2)^{(i-1)}(Z), \tag{3}$$

$$\sum(Z^2)^{(i)} = C^{(i-1)} \times E^{(i-1)}(Z^2) + \sum(z^2)^{(i)}, \tag{4}$$

$$Var^{(i)}(Z) = \frac{\sum(Z^2)^{(i)}}{C^{(i)}} - (E^2)^{(i)}(Z). \tag{5}$$

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 20,2022 at 08:32:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3230325

The mean and the variance values of Z_{max} and Z_{min} are obtained in the same way. The LiDAR frames can be pre-processed through parallel computing and the point features are updated with simple rules. That's how we achieve high efficiency on GPU-enabled mobile devices.

Although combining multiple LiDAR frames suffers from sensory noise and odometry inaccuracy, the maintained point distributions will converge to stable states as the number of observations increases, assuming that the odometry drifts are negligible in local mapping scenarios. This also improves the stability of neural map generation. One side effect is that the influence of new frames will gradually decrease as the number of points accumulates. Therefore, we append a final step of the update that decays the point count $C^{(i-1)}$ of a grid with a factor γ and limits the number of points with C_{max} if there are new points observed at that grid:

$$C^{(i)} = \min(C_{max},\ \gamma C^{(i-1)} + c^{(i)}), \tag{6}$$

where we set the maximum point count $C_{max} = 100$ and $\gamma = 0.90$ to encourage the updates using new data frames.

D. Network Architecture

One of our major objectives is to recover detailed terrain structures with high-quality edges because edges contain important terrain features that are quite essential to improve the safety and efficiency of downstream tasks. As explained in [27], traditional image inpainting pipelines suffer from blur or artifacts due to the difficulties in recovering the high-frequency component. In this case, they present EdgeConnect which sequentially adopts an edge generator and an image completion network to separately conquer the edge recovery and image inpainting. We reference the architecture of EdgeConnect while adopting multitask learning [26] for simultaneous edge generation and elevation mapping. This assists the model in maintaining high-frequency components in the bottleneck features while achieving higher inference speed compared with the serial structure of EdgeConnect.

Fig. 3 presents our pipeline and network architecture. The maintained point features $\mathbf{x}^{(i)}$ are first passed into convolutional layers for encoding, which are then down-sampled before entering 6 ResNet blocks for feature extraction. Each ResNet block contains a dilated convolutional layer for larger receptive fields and a normal convolutional layer for output. Two decoding blocks respectively up-sample the bottleneck features and generate the binary edges $\hat{\mathbf{y}}_e$ and height map $\hat{\mathbf{y}}_h$. Based on the heteroscedastic aleatoric uncertainty model in [24], our height map generation block returns an additional channel that contains a log scale variance map $\log \boldsymbol{\sigma}$ which is integrated into the reconstruction loss in Section III-E and jointly optimized with the height map. The model is encouraged to return high $\log \boldsymbol{\sigma}$ values in the places where it's difficult to reconstruct the terrain due to strong noises or occlusions, and thus it reflects the reconstruction uncertainty. Two discriminators with convolutional layers $\mathcal{D}_E$ and $\mathcal{D}_H$ generate 4-time down-sampled binary classification maps to guide the training of edge and height map generation. During deployment, only the height map branch in the generator is activated for fast inference. Because the network is fully convolutional, it can deal with different input sizes to adapt to the mapping requirements of various downstream tasks.

E. Training Objectives

During the training process, the edge generation is optimized using a pixel-wise binary cross-entropy loss $\mathcal{L}_e^{bce}$:

$$\mathcal{L}_e^{bce} = -\sum \left(\mathbf{y}_e \log \hat{\mathbf{y}}_e + (1-\mathbf{y}_e)\log(1-\hat{\mathbf{y}}_e)\right). \tag{7}$$

For height map generation, we use the heteroscedastic model in [24] to formulate our reconstruction loss $\mathcal{L}_h^{rec}$ and measure the data-dependent reconstruction uncertainty, where L1 norm is adopted due to its robustness to outlying residuals:

$$\mathcal{L}_h^{rec} = \sum \frac{\sqrt{2}}{\boldsymbol{\sigma}} ||\mathbf{y}_h - \hat{\mathbf{y}}_h|| + \log \boldsymbol{\sigma}. \tag{8}$$

Based on the prior information on structured urban terrains, we introduce the unsupervised total variation loss $\mathcal{L}_h^{tv}$ to remove the noises on the surfaces while preserving the edges in the generated height maps:

$$\mathcal{L}_h^{tv} = \sum \sum_{j,k} |\hat{y}_{(h)\,j+1,k} - \hat{y}_{(h)\,j,k}| + |\hat{y}_{(h)\,j,k+1} - \hat{y}_{(h)\,j,k}|, \tag{9}$$

where j, k are map indices in rows and columns.

Both the edge and the height map generation additionally adopt the adversarial loss $\mathcal{L}^{adv}$ and the feature matching loss $\mathcal{L}^{fm}$ calculated from the respective discriminators. The generation adversarial loss is defined as:

$$\mathcal{L}^{adv} = \sum \log(1 - \mathcal{D}(\hat{\mathbf{y}})). \tag{10}$$

Similar to [27], the feature matching loss is calculated by comparing the feature maps from multiple intermediate convolutional blocks $\mathcal{D}^{(l)}$ in the discriminators:

$$\mathcal{L}^{fm} = \sum ||\mathcal{D}^{(l)}(\mathbf{y}) - \mathcal{D}^{(l)}(\hat{\mathbf{y}})||. \tag{11}$$

The overall optimization objective of the generation model is the weighted sum of all the above terms, where the generated edge $\hat{\mathbf{y}}_e$, height map $\hat{\mathbf{y}}_h$, and the map of log variance $\log \boldsymbol{\sigma}$ are jointly optimized:

$$\mathcal{L}^{total} = \mathbf{w}^T [\mathcal{L}_e^{bce}, \mathcal{L}_e^{adv}, \mathcal{L}_e^{fm}, \mathcal{L}_h^{rec}, \mathcal{L}_h^{tv}, \mathcal{L}_h^{adv}, \mathcal{L}_h^{fm}]^T. \tag{12}$$

IV. Experiments

A. Implementation Details

1) Setup: We adopt CoppeliaSim [28] to collect the dataset using the approach described in Section III-B. The final dataset contains 40k valid samples with an average observation ratio of 60 %. The train, validation, and evaluation split ratios are 0.7, 0.1, and 0.2 respectively. Both the simulation and the real-world experiments adopt the same network model trained solely on this simulation dataset.

In simulation experiments, all the mapping approaches are evaluated using $5 \times 5\,\mathrm{m}$ patches in *urban* environments that contain regions of *stairs*, *slopes* as well as other *obstacles* (boxes and irregular steps). We additionally conduct the experiment on 2k samples of irregular *hills* with rough surfaces (the second row of Fig. 4), which are quite different from

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 20,2022 at 08:32:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3230325

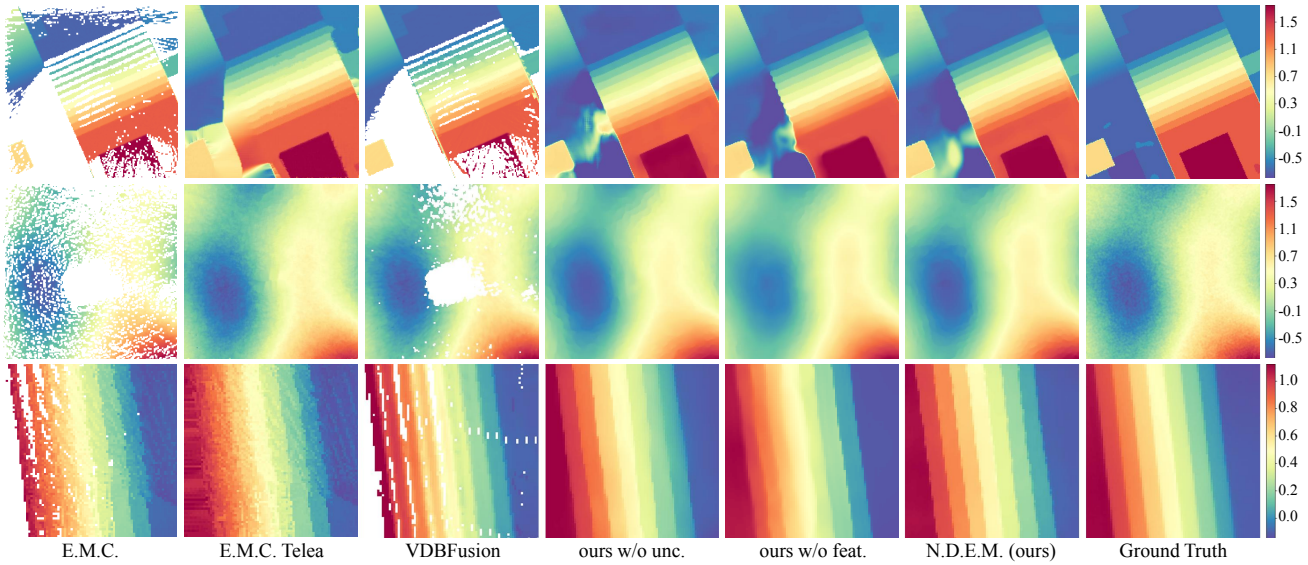

Fig. 4. Visualized examples of the elevation mapping experiments in a simulated *urban* scenario (the first row, $8 \times 8\,\mathrm{m}$), simulated *hill* scenario (the second row, $8 \times 8\,\mathrm{m}$), and on real-world stairs (the third row, $3.2 \times 3.2\,\mathrm{m}$) using different approaches, where different colors indicate the height values. For *urban* terrains and real-world stairs, our approach (N.D.E.M.) provides accurate dense elevation maps with high reconstruction quality and can further recover detailed terrain structures by introducing uncertainty estimation and adopting our statistical point features. In the *hills* scenario, our approach can still recover the terrain in a global view. However, all of our models fail to recover the local features and wrongly generate smooth and step-like terrains.

the urban scenarios in our training dataset, to evaluate the generalization capacity of our approach.

For real-world experiments, we specifically focus on the mapping performance on stairs, as it is frequently used for mapping evaluation in [5], [8], [9], and the reconstruction quality of stairs can largely affect the downstream tasks. We control a Jueying Mini [25] quadrupedal robot to approach or walk up 5 stairs with various heights and steepness. The robot receives sparse point clouds from a RoboSense RS-LiDAR-16 at $10\,\mathrm{Hz}$ to observe the environment. The robot odometry is obtained using a modified version of Normal Distribution Transform (NDT) [29] which continuously compares the current LiDAR frame with a local point cloud map containing 5 nearest frames. All the mapping approaches are evaluated on 400 patches ($2 \times 2\,\mathrm{m}$) on real-world stairs.

To obtain the ground truth maps in the real world, we collect dense point clouds using multiple point frames from an Ouster OS1 LiDAR with 128 channels. The poses of these frames are globally optimized using FAST-LIO [30]. Next, we adopt VDBFusion [4] which performs well with dense input and accurate odometry to reconstruct the terrain structures. We finally rasterize the results as the ground truth height maps. In each experiment, the relative transformation between the ground truth and the generated map origin is obtained by performing Iterative Closest Point (ICP) on the initial sparse observation frame and the dense point cloud.

2) Baseline approaches: We compare the performance of our approach (N.D.E.M.) with the CuPy implementation of elevation mapping [9] which provides the raw elevation maps (E.M.C.) and the results after using the Telea [21] inpainting method (E.M.C. Telea). We also adopt VDBFusion to incrementally reconstruct the terrain using incoming sparse observations and rasterize the results for comparison.

3) Ablation study: For ablation study, we train our model:

- without uncertainty estimation (ours w/o unc.), to show the effect of introducing Bayesian learning;
- without the edge branch (ours w/o edge.), to show the effect of adding the auxiliary edge generation task;
- without using total variation loss (ours w/o tv.), to show the influence of this regularization term;
- using valid height and variance from [9] after the "height cell update" as model inputs (ours w/o feat.), to present the importance of additional statistical features.

The models are trained and evaluated on three random seeds.

4) Evaluation metrics: The mean absolute error (MAE) is widely adopted to measure the accuracy of pixel values. In addition, we calculate the gradient of height w.r.t. the x and y directions and introduce the mean gradient difference (MGD) which is the averaged L2 norm of the gradient differences between the ground truths and the generation results to measure the capability of maintaining details at flat surfaces and edges. As it is not quite meaningful to consider the accuracy of grids in a heavily occluded region, we mask a grid if the observation ratio of a $1 \times 1\,\mathrm{m}$ patch around it is less than $50\,\%$ and obtain the masked results (mMAE and mMGD) to measure the reconstruction accuracy. We also calculate the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) without applying the masks to further evaluate the noise level and visual similarity. For the approaches without a guarantee for dense maps (E.M.C. and VDBFusion), we apply the minimum filter in [9], [23] on their results to fill in the empty region for calculation.

B. Simulation Results

Table I presents the mapping performance in simulated *urban* and *hills* scenarios. Our models achieve higher mapping accuracy in height values and their gradients compared with the traditional approaches. The high PSNR and SSIM values indicate the effectiveness of our models in generating high-quality maps and rational terrain structures. Even for *hills*

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 20,2022 at 08:32:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3230325

TABLE I
ELEVATION MAPPING PERFORMANCE ON SIMULATED URBAN TERRAINS AND HILLS

Urban	mMAE ↓	mMGD ↓	PSNR ↑	SSIM ↑
E.M.C.	3.56	0.58	57.6	0.685
E.M.C. Telea	2.62	0.34	64.1	0.751
VDBFusion	3.17	0.50	61.0	0.695
N.D.E.M.(ours)	$\mathbf{2.10} \pm 0.07$	$\mathbf{0.26} \pm 0.004$	$\mathbf{66.8} \pm 0.2$	0.738 ± 0.012
ours w/o unc.	2.82 ± 0.08	0.28 ± 0.005	66.3 ± 0.4	0.649 ± 0.022
ours w/o edge.	$\mathbf{2.10} \pm 0.07$	0.27 ± 0.006	66.1 ± 0.2	0.728 ± 0.008
ours w/o tv.	2.12 ± 0.10	0.30 ± 0.003	64.4 ± 0.2	$\mathbf{0.762} \pm 0.008$
ours w/o feat.	2.20 ± 0.05	0.29 ± 0.003	65.3 ± 0.1	0.726 ± 0.006
Hills	mMAE ↓	mMGD ↓	PSNR ↑	SSIM ↑
E.M.C.	6.87	1.11	56.0	0.461
E.M.C. Telea	3.78	0.45	70.0	0.746
VDBFusion	4.84	0.59	60.8	0.614
N.D.E.M.(ours)	2.37 ± 0.07	0.41 ± 0.002	73.2 ± 0.2	0.831 ± 0.002
ours w/o unc.	3.06 ± 0.09	0.43 ± 0.002	71.4 ± 0.2	0.821 ± 0.003
ours w/o edge.	$\mathbf{2.21} \pm 0.12$	$\mathbf{0.40} \pm 0.001$	$\mathbf{73.9} \pm 0.2$	$\mathbf{0.836} \pm 0.006$
ours w/o tv.	2.48 ± 0.06	0.42 ± 0.002	72.1 ± 0.3	0.823 ± 0.001
ours w/o feat.	2.88 ± 0.07	0.41 ± 0.001	72.1 ± 0.1	0.814 ± 0.002

See Section IV-A for details of the evaluated approaches and the evaluation metrics. Our models are evaluated over 3 random seeds.

that are not contained in our training dataset, our models still present high mapping performance, indicating their strong generalization capabilities. By introducing Bayesian learning, our full version (N.D.E.M.) achieves better performance (compared with ours w/o unc.). This may be because the uncertainty term can guide the model to focus on recovering the regions with enough information and return a high $\boldsymbol{\sigma}$ to lower the importance of heavily occluded regions in the reconstruction loss. In addition, adopting the edge generation branch in training and using our statistical point features can assist to recover the terrain features and further improves the mapping accuracy (compared with ours w/o edge. and ours w/o feat.) in *urban* scenarios. While ours w/o edge. performs better in *hills* scenarios compared with other models.

Table II shows the terrain-wise results in simulated *urban* scenarios. Our models perform better than the traditional approaches on all kinds of evaluated terrains. Our full version (N.D.E.M.) achieves higher mapping performance on *stairs* and *obstacles*, while also having satisfactory results on *slopes*. This indicates that the auxiliary edge branch, the total variant regularization, and our representation of point features can assist to recover structured terrains. All the approaches present less satisfactory results for *obstacles*, probably because *obstacles* might be irregular and normally have more occlusions, which makes the inpainting difficult.

To intuitively evaluate the accuracy of our approach on map generation and uncertainty estimations, we plot the percentage of different height error levels in simulations and the corresponding averaged $\boldsymbol{\sigma}$ values. The histogram in Fig. 5 shows the percentage of mMAE values among all the *urban* results, which indicates that more than three-quarters of the grids have an error of less than $2\,\mathrm{cm}$, and around half of the grids achieve an even higher accuracy ($<1\,\mathrm{cm}$). The curve in Fig. 5 plots the change of $\boldsymbol{\sigma}$ over the unmasked errors considering all the generated height values. It shows that $\boldsymbol{\sigma}$ values are proportional to the errors, which successfully reflect the confidence of our model on mapping results.

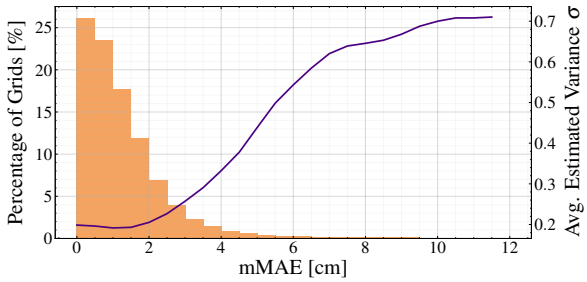

Fig. 5. The percentage (orange bar plot) of the grids with different levels of error and the corresponding estimated $\boldsymbol{\sigma}$ values (purple curve). Our approach provides high-quality mapping results that are accurate enough for various downstream tasks. The uncertainty estimations further reflect the confidence of our model on the generated maps.

The first and the second row of Fig. 4 visualize two examples in simulated *urban* and *hills* scenarios, where we use $8 \times 8\,\mathrm{m}$ patches containing multiple urban terrain features for qualitative evaluation. We select two of our ablated models (ours w/o unc. and ours w/o feat.) for visualization. Our models provide high-quality dense elevation maps in the complex *urban* scenario, and our full version (N.D.E.M.) better recovers the terrain features with straight and sharp edges. For *hills* scenarios, our models successfully recover the terrain in a global view, which explains the high SSIM scores in table I. However, they fail to recover the local features and generate step-like terrains with smooth surfaces.

C. Real-world Performance

Table III shows the performance on real-world stairs. Both raw E.M.C. with the minimum filter and E.M.C. Telea present poorly probably due to the low observation ratios of the scenes with our sensor setting, and traditional inpainting approaches may require enough observations for satisfactory performance. The VDBFusion has an acceptable performance at the expense of memory and computational efficiency as it incrementally constructs a global map, which may introduce a heavy burden to the mobile device. Our models generate maps with high accuracy and structural similarity. Our full version (N.D.E.M.) achieves the best performance among all the models, presenting the effectiveness of our approach in real-world scenarios.

We further visualize an example of the mapping results in a real-world scenario on a stair. As shown in the third row of Fig. 4, E.M.C. generates a noisy map with holes. Although E.M.C. Telea can provide dense mapping results, it fails to recover the map structures from noises and the inpainting is inaccurate if there is not enough observation. VDBFusion successfully reconstructs the stair structures for the first several steps. However, it still cannot provide dense results and the performance is unsatisfactory for the former steps. Our approach (N.D.E.M.) provides high-quality dense mapping results with clean and sharp edges compared with traditional approaches and the models for ablation study.

D. Run-time Analysis

We deploy a Python implementation of the proposed elevation mapping framework on an Nvidia Jetson AGX Xavier.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 20,2022 at 08:32:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3230325

TABLE II
ELEVATION MAPPING PERFORMANCE IN SIMULATION ON DIFFERENT TYPES OF URBAN TERRAINS

Methods	*Stairs*			*Slopes*			*Obstacles*		
	mMAE ↓	mMGD ↓	PSNR ↑	mMAE ↓	mMGD ↓	PSNR ↑	mMAE ↓	mMGD ↓	PSNR ↑
E.M.C.	10.65	1.54	35.2	2.95	0.50	46.7	14.97	1.71	54.2
E.M.C. Telea	5.23	0.85	46.8	2.68	0.40	53.2	8.91	1.40	**64.7**
VDBFusion	9.03	1.37	39.3	3.04	0.40	49.3	17.28	1.76	58.3
N.D.E.M.(ours)	**3.34** $\pm$ 0.05	**0.68** $\pm$ 0.01	**47.9** $\pm$ 0.1	1.91 $\pm$ 0.10	**0.26** $\pm$ 0.005	52.7 $\pm$ 0.2	**8.83** $\pm$ 0.06	**1.33** $\pm$ 0.02	63.6 $\pm$ 0.3
ours w/o unc.	4.80 $\pm$ 0.08	0.72 $\pm$ 0.05	47.0 $\pm$ 0.5	2.36 $\pm$ 0.08	**0.26** $\pm$ 0.003	52.0 $\pm$ 0.3	10.13 $\pm$ 0.08	1.35 $\pm$ 0.05	62.6 $\pm$ 0.6
ours w/o edge.	3.47 $\pm$ 0.06	0.71 $\pm$ 0.02	**47.9** $\pm$ 0.1	**1.50** $\pm$ 0.07	**0.26** $\pm$ 0.007	**53.6** $\pm$ 0.2	9.78 $\pm$ 0.06	1.36 $\pm$ 0.03	63.0 $\pm$ 0.5
ours w/o tv.	4.49 $\pm$ 0.12	0.82 $\pm$ 0.02	47.0 $\pm$ 0.2	1.89 $\pm$ 0.08	0.28 $\pm$ 0.004	53.0 $\pm$ 0.1	8.84 $\pm$ 0.11	1.42 $\pm$ 0.02	63.2 $\pm$ 0.3
ours w/o feat.	5.14 $\pm$ 0.07	0.81 $\pm$ 0.01	46.5 $\pm$ 0.1	1.59 $\pm$ 0.03	**0.26** $\pm$ 0.002	53.3 $\pm$ 0.1	11.00 $\pm$ 0.07	1.47 $\pm$ 0.02	61.6 $\pm$ 0.2

See Section IV-A for details of the evaluated approaches and the evaluation metrics. Our models are evaluated over 3 random seeds.

TABLE III
ELEVATION MAPPING PERFORMANCE ON REAL-WORLD STAIRS

	mMAE ↓	mMGD ↓	PSNR ↑	SSIM ↑
E.M.C.	4.13	0.69	73.7	0.779
E.M.C. Telea	3.92	0.66	74.1	0.787
VDBFusion	3.41	0.52	75.5	0.839
N.D.E.M.(ours)	**2.48** $\pm$ 0.08	**0.39** $\pm$ 0.02	**77.5** $\pm$ 0.4	**0.887** $\pm$ 0.002
ours w/o unc.	2.97 $\pm$ 0.10	0.43 $\pm$ 0.02	76.6 $\pm$ 0.5	0.865 $\pm$ 0.003
ours w/o edge.	2.80 $\pm$ 0.09	0.46 $\pm$ 0.02	76.5 $\pm$ 0.4	0.859 $\pm$ 0.002
ours w/o tv.	2.84 $\pm$ 0.08	0.41 $\pm$ 0.03	76.9 $\pm$ 0.2	0.867 $\pm$ 0.002
ours w/o feat.	2.99 $\pm$ 0.08	0.40 $\pm$ 0.02	76.6 $\pm$ 0.2	0.876 $\pm$ 0.003

See Section IV-A for details of the evaluated approaches and the evaluation metrics. Our models are evaluated over 3 random seeds.

The point pre-processing stage containing all the operations in Section III-C is accelerated using GPU through CuPy element-wise kernels. It takes only around $1.6\,\mathrm{ms}$ to pre-process a point frame from a LiDAR with 16 channels (20k to 30k points) and generate the network input. It still takes a short time ($4.7\,\mathrm{ms}$) if we increase the number of points to 1M. As the generation model is fully convolutional, it can deal with input feature maps of diverse sizes and generate maps of the corresponding shape. The model inference time is irrelevant to the size of point clouds after the pre-processing stage but related to the size of the mapping region. The model takes around $24\,\mathrm{ms}$ to generate $3.2\times3.2\,\mathrm{m}$ $(80\times80\,\mathrm{px})$ local patches for locomotion as done in [5] and uses $75\,\mathrm{ms}$ to generate $12\times12\,\mathrm{m}$ $(300\times300\,\mathrm{px})$ maps for cost-optimal navigation on complex terrains as required in [31]. The whole pipeline performs mapping at $10\,\mathrm{Hz}$, achieving real-time performance on the mobile computation platform.

E. Benefits to Downstream Tasks

We then conduct path planning on real-world stairs using the approach in [31] which requires dense elevation maps for optimal path planning on complex terrains. As shown in Fig. 6, both E.M.C. Telea and our approach provide online dense elevation maps when the robot approaches the stair. However, E.M.C. Telea generates rough step surfaces and wrongly estimates the height of stairs under extremely sparse observations, resulting in the failure of the planning task. Our approach successfully reconstructs the stair once the LiDAR observes the steps. The strong capability of our approach in reconstructing urban terrain features at a glance assists the navigation tasks to achieve better performance.

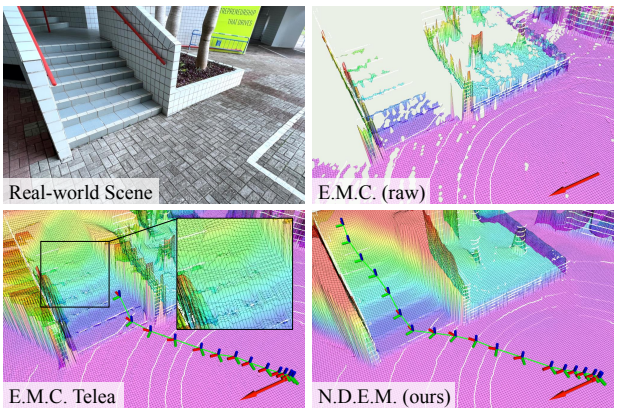

Fig. 6. Path planning experiment with the online mapping results on real-world stairs. E.M.C. Telea generates rough steps and wrong estimations of the step height, resulting in the failure of planning. Our approach successfully recovers the stair structure at a glance, presenting strong capabilities in urban terrain reconstruction with extremely sparse observation.

Adopting the uncertainty estimations in the downstream tasks may help to ensure safe and robust behaviors of the robot. Fig. 1(c) visualizes the uncertainty values on the real-world stairs. Our model returns a higher uncertainty at the edges of the stairs than the flat surfaces, which may assist the locomotion algorithms to step feet onto safe regions and prevent slips resulting from the terrain reconstruction inaccuracy. Also, when there is a hole in the ground (Fig. 7), traditional inpainting approaches (E.M.C. with minimum filter and E.M.C. Telea) [9] just fill in that region to make it a surface, resulting in potential danger to navigation tasks. Although our approach cannot provide accurate height values inside the hole due to occlusion, it returns high reconstruction uncertainty that can inform the navigation module for safe behaviors. In addition, our approach can generate the slope based on the information of adjacent grids and returns higher uncertainty values as the occlusion gets stronger.

V. CONCLUSION

We proposed a learning-based dense elevation mapping framework for online urban terrain reconstruction. It maintains high reconstruction quality and accuracy in both simulation and real-world scenarios. The proposed data pre-processing method and the map generation model show robust performance that can recover detailed urban terrain structures from sparse and noisy LiDAR observations, while also achieving high efficiency to enable real-time elevation mapping on

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 20,2022 at 08:32:27 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2022.3230325

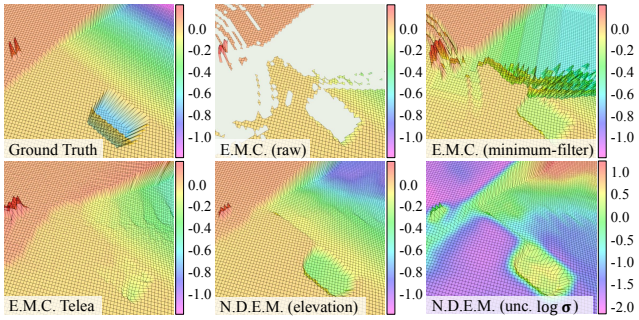

Fig. 7. When encountering the occluded regions, rather than just filling in the region as done by the traditional inpainting approaches, our mapping module returns high uncertainty at the occluded region, informing the downstream tasks of the potential danger.

mobile devices. The provided uncertainty estimations also have the potential to assist the downstream tasks for safer and more robust performance.

There are also several limitations of our approach. Our model overfits the urban scenarios and fails to recover the detailed features of irregular terrains or obstacles. The representation approach of point features needs to be improved to deal with overhanging obstacles and dynamic objects. In addition, the pipeline has no specific design to compensate for the drifts and it assumes the drifts are negligible in local mapping scenarios. Currently, a height drift of several centimeters will affect the stability of the point features and lead to degraded mapping performance. Jointly optimizing the map and the odometry might be a future direction to solve this problem.

ACKNOWLEDGMENT

We would like to thank Jianhao Jiao, Peng Yun and Xiangcheng Hu for their valuable suggestions.

REFERENCES

[1] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12932–12942, June 2022.

[2] H. Guo, S. Peng, H. Lin, Q. Wang, G. Zhang, H. Bao, and X. Zhou, "Neural 3d scene reconstruction with the manhattan-world assumption," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5511–5520, 2022.

[3] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1366–1373, IEEE, 2017.

[4] I. Vizzo, T. Guadagnino, J. Behley, and C. Stachniss, "Vdbfusion: Flexible and efficient tsdf integration of range sensor data," *Sensors*, vol. 22, no. 3, p. 1296, 2022.

[5] D. Hoeller, N. Rudin, C. Choy, A. Anandkumar, and M. Hutter, "Neural scene representation for locomotion on structured terrain," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8667–8674, 2022.

[6] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[7] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3019–3026, 2018.

[8] M. Stölzle, T. Miki, L. Gerdes, M. Azkarate, and M. Hutter, "Reconstructing occluded elevation information in terrain maps with self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1697–1704, 2022.

[9] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation mapping for locomotion and navigation using gpu," *arXiv preprint arXiv:2204.12876*, 2022.

[10] K. Katyal, K. Popek, C. Paxton, J. Moore, K. Wolfe, P. Burlina, and G. D. Hager, "Occupancy map prediction using generative and fully convolutional networks for vehicle navigation," *arXiv preprint arXiv:1803.02007*, 2018.

[11] V. D. Sharma, J. Chen, A. Shrivastava, and P. Tokekar, "Occupancy map prediction for improved indoor robot navigation," *arXiv preprint arXiv:2203.04177*, 2022.

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*, pp. 405–421, Springer, 2020.

[13] T. Chen, P. Wang, Z. Fan, and Z. Wang, "Aug-nerf: Training stronger neural radiance fields with triple-level physically-grounded augmentations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15191–15202, June 2022.

[14] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8248–8258, June 2022.

[15] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[16] T. Jia, E.-Y. Yang, Y.-S. Hsiao, J. Cruz, D. Brooks, G.-Y. Wei, and V. J. Reddi, "Omu: A probabilistic 3d occupancy mapping accelerator for real-time octomap at the edge," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 909–914, 2022.

[17] K. Stepanas, J. Williams, E. Hernández, F. Ruetz, and T. Hines, "Ohm: Gpu based occupancy map generation," *arXiv preprint arXiv:2206.06079*, 2022.

[18] M. Wei, D. Lee, V. Isler, and D. Lee, "Occupancy map inpainting for online robot navigation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8551–8557, IEEE, 2021.

[19] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 190–198, 2017.

[20] J. Zhang, H. Zhao, A. Yao, Y. Chen, L. Zhang, and H. Liao, "Efficient semantic scene completion network with spatial group convolution," in *European Conference on Computer Vision*, pp. 749–765, Springer, 2018.

[21] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of graphics tools*, vol. 9, no. 1, pp. 23–34, 2004.

[22] M. Ebrahimi and E. Lunasin, "The navier–stokes–voight model for image inpainting," *The IMA Journal of Applied Mathematics*, vol. 78, no. 5, pp. 869–894, 2013.

[23] F. Jenelten, R. Grandia, F. Farshidian, and M. Hutter, "Tamols: Terrain-aware motion optimization for legged systems," *IEEE Transactions on Robotics*, pp. 1–19, 2022.

[24] A. Kendall, *Geometry and uncertainty in deep learning for computer vision.* PhD thesis, University of Cambridge, UK, 2019.

[25] "Jueying Mini." https://www.deeprobotics.cn/en/products_jy_301.html. Accessed: 2021-08-30.

[26] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[27] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edgeconnect: Structure guided image inpainting using edge prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[28] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013. www.coppeliarobotics.com.

[29] M. Magnusson, *The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection*. PhD thesis, Örebro universitet, 2009.

[30] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[31] B. Yang, L. Wellhausen, T. Miki, M. Liu, and M. Hutter, "Real-time optimal navigation planning using learned motion costs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9283–9289, IEEE, 2021.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 20,2022 at 08:32:27 UTC from IEEE Xplore. Restrictions apply.