IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 7, 2019, accepted October 11, 2019, date of publication October 22, 2019, date of current version November 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948976

A Scalable Sampling-Based Optimal Path Planning Approach via Search Space Reduction

WENJIE LU AND DIKAI LIU
Centre for Autonomous Systems, University of Technology Sydney, NSW 2007, Australia

Corresponding author: Wenjie Lu (wenjie.lu@uts.edu.au)

This work was supported in part by the Australian Research Council (ARC) Linkage Project under Grant LP150100935, in part by the Roads and Maritime Services of NSW, and in part by the Centre for Autonomous Systems (CAS) at the University of Technology Sydney.

ABSTRACT Many sampling strategies in Sampling-Based Planning (SBP) often consider goal and obstacle population and may however become less efficient in large and cluttered 3D environments with a goal distanced away. This paper presents a search-space-Reduced optimal SBP approach (RSBP) for a rigid body. This reduced space is found by a sparse search tree, which is enabled by a Metric Function (MF) built on a neural network. The offline-learnt MF estimates the minimum traveling cost between any two nodes in a fixed small workspace with various obstacles. It allows connections of two sparse nodes without path planning, where the connections represent the traveling costs (not paths). It is proven that the asymptotic optimality is preserved in the RSBP (assuming a zero-error MF) and the optimality degeneration is bounded (assuming a bounded-error MF). The computational complexity during planning is shown linear to the Lebesgue measure of the entire search space (assuming the same sampling density across environments). Numerical simulations have shown that in tested large and cluttered environments the RSBP is at least as fast as the bidirectional fast marching tree* and informed rapidly exploring random tree*, with planned paths of similar optimality. The results also have shown the RSBP's improved scalability to large environments and enhanced efficiency in dealing with narrow passages.

INDEX TERMS Learning, path planning.

I. INTRODUCTION

This paper studies the optimal path planning problems of a rigid body in large and cluttered environments, arising from many search and inspection applications, e.g., an autonomous underwater vehicle navigates in a large cave populated with stalactites [1], an unmanned aerial vehicle flies through forests [2]. Optimal Sampling-Based Planning (SBP) methods have been state-of-the-art for planning an optimal collision-free (also referred to as feasible) path in high dimensional spaces with complex obstacles, outperforming their exact counterparts, such as visibility graph [3], Voronoi diagrams [4], and cell decomposition [5]. These SBP planners, e.g., Probabilistic RoadMap* (PRM*) [6], Rapidly-exploring Random Tree* (RRT*) [6], Informed RRT* [7], Fast Marching Tree* (FMT*) [8], and Bidirectional FMT* (BFMT*) [9], typically grow a graph (a tree or a roadmap) with collision-free random nodes, which are connected via feasible and optimal paths (also referred to as motions).

In cluttered environments, the commonly-used straight-line motion might be quite short, therefore the aforementioned planners require dense sampling in a large search space. Other choices of motions require solving optimization problems and are thus not efficient [10]. This paper seeks to reduce the search space via an offline-learnt Metric Function (MF), for improving the runtime efficiency of optimal SBP in large and cluttered environments.

After the first appearance of optimal SBP methods, tremendous effort has been made to improve their efficiency from varying the key components: (i) nearest neighbor search [11]; (ii) graph connection [12]; (iii) collision checking [13], [14]; and (iv) node sampling [15], [16]. Besides, parallelizing sampling (expanding two trees from the start and the goal) and

The associate editor coordinating the review of this manuscript and approving it for publication was Rui-Jun Yan.

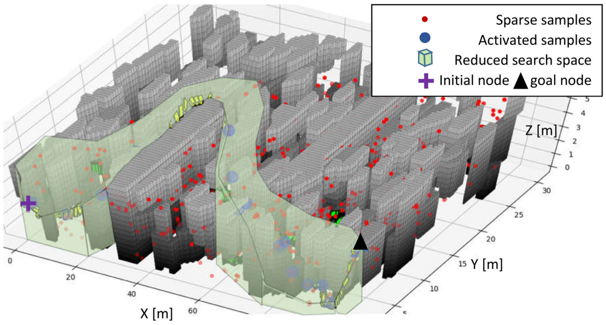


FIGURE 1. **Illustration of a reduced search space obtained from RSBP (better viewed in color).**

running multiple planners simultaneously in parallel threads have been orthogonal effort to aforementioned [17], [18].

The effectiveness of SBP is determined by how much the graph captures the information on connectivity and optimality of the planning problems. The effectiveness is therefore determined by samples' distribution and sampling strategies. Other than commonly used uniform sampling, many sampling heuristics have been designed for special scenarios or purposes, such as *"medial axis of Voronoi graphs"* for capturing the skeleton of the feasible configuration space [19], *"boundary"* for increasing probability in having samples near boundaries [20], *"bridge-test"* for generating samples in narrow passages [21], and *"goal biasing"* for favoring search towards the goal region. Multiple aforementioned heuristics are combined for comprehensive purposes.

Some sampling strategies iteratively and adaptively change their bias based on the statistics of the nodes previously sampled. Expansive space trees explore the search space by first measuring the density of the explored space, and then biasing exploration towards some less explored space. Resolution-complete SBP estimates the failure frequency of tree expansions [22] for more effective exploration. A generalized Voronoi diagram is often employed to search for free space in BFMT* [23]. Some other strategies iteratively sample in the region confined by the current-best cost (via the triangle inequality in Euclidean spaces) and then search for a better solution [7].

Limiting or avoiding sampling in the non-optimal regions is essential. However, such requirement ends in a dilemma: knowing that some regions are non-optimal requires sampling in these regions. Due to short motions of straight lines, dense sampling is necessary in cluttered environments (as shown in Fig. 1) to sufficiently reject some non-optimal regions. As pointed out earlier, other options in connecting samples involve directly solving a number of non-trivial control or optimization problems and are not efficient for optimal SBP.

Contributions: In this study, we seek the possibility of searching for optimal regions (vice versa non-optimal regions) via building a sparse tree, before planning an optimal and detailed path. This detailed path is the output of the RSBP algorithm and is usually termed as "path" in any SBP (See Definition 1 in Section II). Here the prefix "detailed" is used to distinguish a sparse path that is introduced later in (ii) and formally defined in Section IV. The optimal region is a reduced search space that is guaranteed in probability one to contain a path that is optimal or sub-optimal (with bounded optimality degeneration). The contributions of this paper are itemized as follows.

(i) *An offline-trained Metric Function (MF).* It is assumed that the obstacles in a 3D environment are known without error. Rather than using the obstacle information purely in a black-box fashion, we represent the obstacles by 3D occupancy grids and directly use them as a part of the input to the MF. The limitations of this assumption and the grid representation will be discussed in Section VII. The proposed MF uses the information of the obstacle occupancy within a small fixed-size workspace (also referred to as a cell) to evaluate the minimum traveling distance and reachability between any two nodes in that cell. The reachability between two nodes is "true" if a feasible path exists to connect them.

The core of the MF consists of a Contractive AutoEncoder (CAE) and a Fully Connected Network (FCN). The CAE encodes an arbitrary environment (obstacle occupancy) in a lower-dimensional space. The CAE is a widely-used technique in representation learning of 2D images and 3D geometries [24]. The FCN takes in the environment encoding with the configurations of a node pair as the input. It outputs the evaluations of the minimum traveling distance and reachability. This MF is offline trained based on the planning results of a large number of randomly-generated problems in a cell (i.e., randomizing the obstacles and the node pair). The planning results are obtained by the PRM* method [6], [25].

(ii) *Search-space-Reduced SBP (RSBP).* The RSBP is enabled by the offline-trained MF. In runtime, the RSBP consists of two stages. Stage I first decomposes the large workspace into cells of a same size, then uses the learnt MF to generate a sparse tree, where sparse nodes are sampled on the surfaces of these cells, and at last searches globally for an optimal sparse path. The sparse path is composed of some sparse nodes. The sparse tree is a graph capturing the connectivity and traveling cost between sparse nodes. The reduced search space is the union of cells that the found sparse path goes through, as shown in Fig. 1 (the space is smoothed for the illustration purpose). Stage II plans a detailed path within the obtained reduced space by any classical SBP.

These nodes on this found sparse path are referred to as the activated nodes. In contrast to the aforementioned detailed path, the sparse path does not provide collision-free and sufficiently-dense interpolations between consecutive activated nodes. The straight-line connections between the consecutive activated sparse nodes are very likely to collide with obstacles. In Stage I, instead of using optimal and feasible motions to connect sparse nodes, we only utilize the learnt MF to estimate the value of minimum traveling distance and the reachability between them. The detailed collision-free motions between the consecutive sparse nodes are neglected in Stage I, which significantly reduces the number of calls to sample nodes, check collisions, and connect nodes.

(iii) *Properties of the RSBP planner (in runtime)*. 1: It is shown that the search space can be reduced by a factor of $v/(uM)$, where v (u) is the Lebesgue measure of the search space in the entire workspace (in the cell) and M is the number of the cells the sparse path goes through. 2: The computational complexity of the RSBP is $O(v)$, given a fixed sampling density. The state-of-the-art optimal SBP typically offers $O(v \log v)$ [6]. 3: In addition, the resultant RSBP is proven to offer typical asymptotic optimality by assuming the trained MF has zero estimation error. 4: The optimality degeneration due to the MF estimation error (assumed bounded) is shown bounded.

(iv) *Validation through numerical simulations*. 1: Simulation results have demonstrated that in tested large and cluttered environments the proposed RSBP is at least as fast as the BFMT* and the Informed RRT* while producing paths of similar optimality. 2: The scalability of the RSBP is demonstrated through tests in cluttered environments of growing sizes. 3: Through introducing additional data on cells with narrow passages during offline training, the MF could estimate costs of optimal paths through narrow passages, making the RSBP more efficient than BFMT* and Informed RRT* in finding narrow passages. This results also show that the capability of the RSBP could be potentially enhanced by including more data. It may allow reducing efforts in designing and combining scenario-specific sampling strategies.

Related work: The Informed RRT* restricts search in a subspace confined by the cost of the current-best solution and iteratively increases the density of samples in the subspace [7]. Batch Informed Trees* use an informed search strategy to only consider the subspace that could provide a better solution [16], which requires an initial feasible solution. However, it might be difficult to have a feasible initial solution in a cluttered environment. The heuristics often result in a conservative reduction in the search space since the obstacles are not directly taken into account. Recently, a deep neural network has been employed to offer better sampling allocations in more likely regions, through biasing sampling towards these regions via learnt sampling distributions [15]. However, it requires to encode all obstacles in the entire workspace at once and is thus not suitable for large environments.

Learning generative models or extract features for 3D objects from data has been an active field. It seeks a lower-dimensional representation of object geometries (also referred to as encoding) [24], [26]. While point clouds might be a more general representation for obstacles, this paper employs a grid representation. The grid representation is an ordered structure and yields an easier loss function for training. A point cloud is often an unordered set and requires solving optimization problems to calculate the training losses. Thus it is more challenging to train the MF if a point cloud is adopted [27]. We seek a model on 3D grid objects for generating environment encodings that are suitable for training the MF.

Using learning-based tools for planning has been explored in VF-RRT [28], which is a tree-based motion planner that minimizes the so-called cost-to-go (upstream cost). The upstream cost is integrated over a vector field and kinodynamic constraints are encoded in the estimated cost. However, no obstacle information is considered, therefore this cost estimation is not feasible to generate a sparse tree for large cluttered environments. The proposed study is orthogonal to [28].

Navigation of a robot using networks has also been studied, where many approaches take in raw images from cameras or distance data from sonar sensors. Then through fuzzy inference systems or recurrent neural networks, the system outputs high-level commands, e.g., searching for a goal, avoiding obstacles, and/or low-level control, e.g., maintaining heading, turning, adjusting forward speed [29]. Graph neural network models have also been investigated to solve planning problems, which often require a graph representation of the environment and encode it into a neural network structure [30]. Vector-based navigation using grid-cell representations has been discussed in [31].

This paper is organized as follows. The problem formulation is given in Section II, followed by the MF structure and its training procedure in Section III. Then, the proposed RSBP is presented and discussed in Section IV. After that, the analysis on the asymptotic optimality, the bounded optimality degeneration, and the computational complexity are given in Section V. Results on numerical simulation are summarized and discussed in Section VI. At last, discussions on limitations and future work can be found in Section VII, followed by conclusion in Section VIII.

II. PROBLEM FORMULATION

This paper considers the problems of planning an optimal path of a rigid body from its initial configuration $\mathbf{q}_0 \in \mathbb{SE}(3)$ to its goal region $\mathcal{G} \subset \mathbb{SE}(3)$ in an obstacle-populated 3D workspace. A workspace with an obstacle population is referred to as an environment. The workspace is defined as $\mathcal{W} \triangleq [0,\ L] \times [0,\ D] \times [0,\ H] \subset \mathbb{R}^3$, where $L \in \mathbb{R}^+$, $D \in \mathbb{R}^+$, and $H \in \mathbb{R}^+$.

Let the robot configuration be denoted as $\mathbf{q} = [\mathbf{x}^T, \boldsymbol{\epsilon}^T, \eta]^T$ in the 6-dimensional manifold $\mathbb{SE}(3)$ (embedded in a $d = 7$-dimensional Euclidean space), where $\mathbf{x} = [x, y, z]^T$ denotes the body's translational coordinates, and $\boldsymbol{\epsilon}$ and η are the quaternion representation of the body's orientation in the inertial frame. Let $\mathcal{A}(\mathbf{q})$ denote the closed subset of $\mathcal{W}$ occupied by the rigid body at the configuration $\mathbf{q} \in \mathcal{C}$, where $\mathcal{C} \triangleq \{\mathbf{q} \in \mathbb{SE}(3) \mid \mathcal{A}(\mathbf{q}) \subset \mathcal{W}\}$ is the entire configuration space (also referred to as the entire search space). The Lebesgue measure v of the search space $\mathcal{C}$ is typically determined by L, D, and H, assuming no restriction on the body orientation.

The workspace $\mathcal{W}$ is populated by N rigid obstacles, $\mathcal{B}_1, \ldots, \mathcal{B}_N$, where N increases as $\mathcal{W}$ grows. The obstacle geometries and locations can be represented by a point cloud from laser scans and SLAM algorithms [32]. Their geometries and locations are assumed known without error in this study, the limitations from this assumption will be discussed in Section VII. A configuration $\mathbf{q}$ is collision-free or feasible if the rigid body placed at $\mathbf{q}$ does not intersect with any obstacle in the workspace. Let $\mathcal{C}_{free} \triangleq \{\mathbf{q} \in \mathcal{C} | \mathcal{B}_i \cap \mathcal{A}(\mathbf{q}) = \emptyset, \forall\, i = 1, \cdots, N\}$ denote the collision-free configuration space, and $\mathcal{C}^c_{free} \triangleq \{\mathbf{q} \in \mathcal{C} | \mathbf{q} \notin \mathcal{C}_{free}\}$ the collision-not-free configuration space. In the remainder of this paper, a set with the superscript c denotes the complementary set.

Definition 1 (Path or Detailed Path): A path $\mathbf{s}(\tau)$ *is a mapping from* $[0, 1]$ *to* $\mathcal{C}$. *This path is also referred to as a detailed path.*

Notice that the prefix "detailed" is used to distinguish a sparse path that is formally defined in Section V. A sparse path is not suitable for low-level controllers. It is because that the nodes of a sparse path are not sufficiently dense and the straight-line connection between two consecutive nodes are very likely to collide with obstacles.

Definition 2 (Strong and Weak Clearance) [6]: A path has strong δ *clearance if the minimum distance from the rigid body along this path to the obstacles is* δ. *A path* $\mathbf{s}$ *is said to have weak* δ *clearance if there exits a homotopic transform* $\pi(\alpha)$, *where* $\alpha = [0, 1]$, *such that* $\pi(\alpha = 0) = \mathbf{s}$, $\pi(\alpha = 1)$ *has strong* δ*-clearance, and* $\pi(\alpha \in (0, 1])$ *has strong* δ_α*-clearance.*

This study seeks a planning approach that scales well to L, D, and H and that solves a number of the following problems for a given rigid-body robot and a fixed objective function $J(\mathbf{s})$ to minimize, where $J(\mathbf{s})$ is the accumulated weighted translational and rotational distances.

Problem 1: Given a workspace $\mathcal{W}$ *with cluttered obstacles, an initial configuration* $\mathbf{s}(0) = \mathbf{q}_0 \in \mathcal{C}$, *and a goal region* $\mathcal{G} \subset \mathcal{C}$, *find a collision-free path* $\mathbf{s}$, *i.e.,* $\mathbf{s}(\tau) \in \mathcal{C}_{free}, 0 \leq \tau \leq 1$ *such that* $\mathbf{s}(1) \in \mathcal{G}$ *and* $J(\mathbf{s})$ *is minimized. The optimal solution* $\mathbf{s}$ *of the problem is assumed weak-*δ*-clearance if exists.*

III. RSBP: OFFLINE TRAINED MF

Building a sparse search tree requires estimating minimum traveling distance between two sparse nodes, which are most likely interfered by multiple obstacles. This distance estimation may be obtained by solving constrained optimization problems or by using a feedback controller, which is either inefficient, non-optimal, or unreliable for real-time applications.

In this section, we present a neural network that is trained offline and is suitable to learn the MF for estimating minimum distance and reachability between any two nodes within a smaller workspace (also referred to as a cell) $\boldsymbol{\omega} \triangleq [-l,\ l] \times [-d,\ d] \times [-h,\ h]$, where $l \in \mathbb{R}^+$, $d \in \mathbb{R}^+$ and $h \in \mathbb{R}^+$. We define the configuration space associated with this cell as $\boldsymbol{\zeta} \triangleq \boldsymbol{\omega} \times \mathbb{SO}(3)$. The Lebesgue measure u of $\boldsymbol{\zeta}$ is determined by l, d, and h since the range of the body orientation is fixed.

Notice that the discussion in this section is restricted to such a cell. The minimum traveling distance between two nodes in this cell is determined by four factors. They are (i) the geometry of the rigid body $\mathcal{A}$, (ii) the obstacle geometries in $\boldsymbol{\omega}$, (iii) the configurations of this two nodes ($\mathbf{p}_1$ and $\mathbf{p}_2$ in $\boldsymbol{\zeta}$), and (iv) a definition of the minimum distance (i.e., the Lagrangian in $J(\mathbf{s})$). Notice that $\mathbf{p}_1$ and $\mathbf{p}_2$ are any pair of randomly sampled nodes in this cell. In this paper, the Lagrangian in $J(\mathbf{s})$ is a given weighted sum of the translational distance and the rotational distance. Whether this study can be extended to other Lagrangian needs further investigation. It is practical to assume a fixed body geometry and a fixed Lagrangian in $J(\mathbf{s})$ for a given robot. These fixed factors are referred to as internal characteristics. Factors (ii) and (iii) vary among planning problems and they are referred to as external factors.

In addition to considering two nodes ($\mathbf{p}_1$ and $\mathbf{p}_2$), the MF takes in the obstacle occupation as a part of the input. In this paper, the obstacle occupation is represented by 3D grids. It is straight-forward and quick to convert points clouds to occupancy grids. Each grid can take the value from 0 to 1, with 1 being 100%-sure occupied. In this paper no uncertainties are associated with obstacles, therefore each grid can only be 0 or 1. Given a fixed cell and the number of the grids, the resolution of the obstacle representation is determined. Here the obstacle representation is also referred to environment representation. Let $\mathbf{o} \in \chi$ denote the obstacle occupation for $\boldsymbol{\omega}$, where χ is the set of all possible occupations. Then, the proposed MF offers an estimation of minimum traveling distance between two nodes and is denoted as

$$d : \chi \times \boldsymbol{\zeta} \times \boldsymbol{\zeta} \to \mathbb{R}_{\geq 0}, \tag{1}$$

where $\mathbb{R}_{\geq 0}$ denotes the set of all non-negative real numbers.

A. METRIC FUNCTION STRUCTURE

The encoding of the environment in $\boldsymbol{w}$ is critical to learning the MF. In this paper, the small environment in $\boldsymbol{w}$ is converted into $a \times a \times a$ occupancy grids, where a is the number of grids along each axis. The limitations and the resolution loss from the grid representation will be discussed in Section VII. However, the grid representation results in an ordered structure in a high-dimensional space. The concatenation of this representation and the sample configurations ($\chi \times \boldsymbol{\zeta} \times \boldsymbol{\zeta}$) is not suitable as the input to the MF. In this paper, we use a Contractive AutoEncoder (CAE) to capture an encoding $\mathbf{e}$ of $\mathbf{o}$ in a b-dimensional vector space $\mathcal{E}$ (latent features) [24]. The magnitude of the representation is also penalized (contracted) during the network training. Then the encoding (highlighted in orange in Fig. 2) is aggregated with two sample configurations, resulting in a $(b + 2d)$-dimensional encoding of $\mathcal{E} \times \boldsymbol{\zeta} \times \boldsymbol{\zeta}$, where $d = 7$ is the dimensions of $\boldsymbol{q}$. It is possible that the distance between two nodes is infinite when no feasible path between them is available. As a result, the network outputs two elements, one estimating the possibility of a feasible path and the other estimating the minimum distance.

The entire network structure is shown in Fig 2, which consists of two subnetworks: a CAE and a Fully Connected

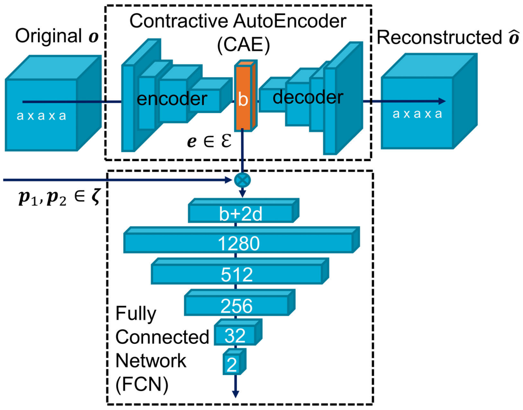

Original $\boldsymbol{o}$
Contractive AutoEncoder (CAE)
Reconstructed $\hat{\boldsymbol{o}}$
a x a x a
encoder
b
decoder
a x a x a
$\boldsymbol{e} \in \mathcal{E}$
$\boldsymbol{p}_1, \boldsymbol{p}_2 \in \zeta$
b+2d
1280
512
256
Fully Connected Network (FCN)
32
2

FIGURE 2. Neural network structure of the MF.

TABLE 1. CAE parameters ($a = 32, b = 48$).

	number of filters	kernel size
conv1	16	4
conv2	32	4
conv3	64	4
conv4	128	2
deconv1	128	2
deconv2	64	4
deconv3	32	4
deconv4	16	2
deconv5	1	1

Network (FCN). The CAE takes in the $a \times a \times a$ grid representation of $\boldsymbol{w}$. The encoder of the CAE consists of four convolutional modules followed by a fully connected layer that produces an encoding. Each convolutional module starts with a 3D convolution layer, followed by a batch normalization layer, a ReLU activation layer, and a max pooling layer. The decoder takes in this encoding and maps it back to $a \times a \times a$ grids through five deconvolution modules, which starts with a deconvolution layer, followed by a bias layer, a batch normalization layer, and a ReLU activation layer (except the last module). The last convolution layer employs a sigmoid activation layer over a ReLU layer.

In the convolution and deconvolution modules, the stride in all directions is 1, the padding of style is "same", the max pool size is 2 in all directions. The number of the filters and the kernel sizes vary and are summarized in Table 1, with a and b being 32 and 48, respectively. The choice of a and b will be discussed in the next subsection.

The encoding is concatenated with two node configurations to create an input to the FCN. This network consists of 5 dense layers. Each layer is followed by a dropout layer. The sizes of the dense layers and the dropout rates are summarized in Table 2. Notice that the dropout rate is set to 0.2 in order to avoid overfitting in training and to suppress arbitrary errors in the MF evaluations as shown in Fig. 5.

B. TRAINING DATA

We assume that the number of obstacles is a Poisson random variable with its intensity proportional to the volume of $\boldsymbol{w}$, e.g., expected number of trees in a forest might be proportional to the size of the forest. Therefore, increasing the size of $\boldsymbol{w}$ (i.e., $\boldsymbol{\zeta}$) results in more obstacles. With the same grid resolution, the number of grids increases, which in return requires more training data and introduces more difficulty in training. The advantages from a larger $\boldsymbol{w}$ include allowing more sparse samples and fewer MF evaluations in Stage I. On the other hand, decreasing the size of $\boldsymbol{w}$ leads to fewer obstacles and less-various environments, thus fewer training data and less-difficult training. However, samples in Stage I will be less-sparse, requiring more samples and more calls of the MF evaluations. This paper uses a small size of $\boldsymbol{w}$ such that most of the optimal paths between any two nodes on surfaces of $\boldsymbol{w}$ are not straight lines as shown in Fig. 3. In this way, the MF captures optimal detour information, and the size of $\boldsymbol{w}$ considered is still small for the CAE.

TABLE 2. FCN parameters ($a = 32, b = 48$).

	unit size	dropout rate
dense1	1280	0.2
dense2	512	0.2
dense3	256	0.2
dense4	32	0.2
dense5	2	N/A

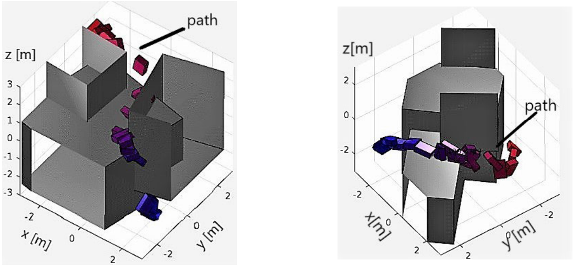


FIGURE 3. Examples of paths from PRM* and obstacles.

In this study, the robot body is box-shaped with its dimensions being $0.75 \times 0.5 \times 0.25$ $[m^3]$, which is similar to the underwater robot developed at the University of Technology Sydney. Only box-shaped obstacles are considered. For each box, the lengths of three sides and the positions are independently uniformly sampled from $[1, 5]^3$ $[m^3]$ and $[-6, 6]^3$ $[m^3]$ respectively. While the roll and pitch angles of each obstacle are zeros, the yaw angle is uniformly sampled from $[0, 2\pi]$. The number of boxes is randomly generated from a Poisson distribution with its intensity being 5. Also notice that due to a high possibility of boxes being overlapping, the cell contains various concave obstacles.

In this paper, we chose the cell size as $[-3, 3]^3$ $[m]^3$, then after trial and error, we set a and b as 32 and 48, where a is designed based on the obstacle complexity and computational power, and b is determined by the accuracy of the obstacle reconstruction and the compactness of the encoding. Limitation of this setting will be discussed in Section VII. Such choice of obstacle populations might not be realistic but makes obstacle generation easy and collision query quick when the PRM* is applied to create training data.

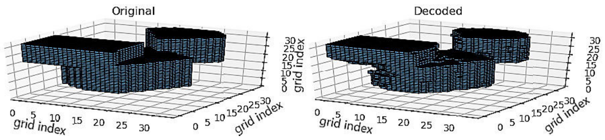


FIGURE 4. An example showing the similarity between the original and the reconstructed environments.

C. TRAINING

It is not trivial to train this neural network from scratch. We follow a three-phase training approach used in [24]. In the first phase of training, we only train the CAE to reconstruct the grid representation. Then we fix the trained CAE to train FCN, followed by a joint training on the FCN and the encoder in CAE, which are already partially trained.

Phase I: The loss function used for training the CAE is a summation of a cross-entropy loss on the mismatch between the reconstructed grid output and the grid input, and a contractive term for penalizing the magnitude of the encoding:

$$E_c = -\frac{1}{M_g}\sum_{m=1}^{M_g}[p_m \log \hat{p}_m + (1-p_m)\log(1-\hat{p}_m)] + \|\mathbf{e}\|_2, \tag{2}$$

where $M_g = a^3$ denotes the number of all grids, m iterating over every grid, and p_m ($\hat{p}_m$) is the (estimated) probability of Grid m being occupied. The regularization on the encoding $\mathbf{e}$ is its L2 norm.

About 90k environments $\boldsymbol{o}$ were randomly generated for training the CAE. This network was initialized randomly and trained with eq. (2). We initialized the kernel with a mean being 0 and a standard deviation being 0.02. The bias was again randomly generated from $\mathcal{N}(0, 0.02)$. We trained the MF with a fixed learning rate of $1e-4$ using Adam [33], for about 300 epochs. The training took 9 hours on a cluster with an Intel Xeon Gold 2.7GHz CPU and a NVIDIA Quadro P5000 graphics card. The error was reduced to 0.73, where 0.03 being the cross-entropy loss and 0.7 being the contractive loss on the length of the encoding, approximately. One example comparing the reconstructed and original environments is shown in Fig. 4.

Phase II: The training data for the second and third phases was obtained by the PRM* with dense sampling (approximately $20K$ samples per environment in a cell). For each randomly generated environment of the cell $\boldsymbol{w}$, we need to obtain minimum distance values between a number of node pairs. For this purpose, we randomly generated 46k environments $\boldsymbol{o}$ in cells and 100 node pairs for each environment.

The outputs from the FCN consist of one for the feasibility and one for the minimum distance value. The loss function on feasibility is given as

$$E_f = -[p_f \log \hat{p}_f + (1-p_f)\log(1-\hat{p}_f)], \tag{3}$$

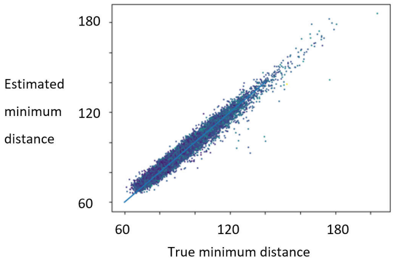


FIGURE 5. Estimation performance of a trained MF.

where p_f ($\hat{p}_f$) is the (estimated) probability of feasibility. The loss function on minimum distance is given as

$$E_d = \|d - \hat{d}\|_2, \tag{4}$$

where d ($\hat{d}$) is the (estimated) minimum distance between nodes if a feasible path between is available. In the second phase, we initialized the weights and bias of FCN with a Xavier initializer and then trained FCN for about 100 epochs by Adam, which took about 6 hours on the same machine, with the CAE trainable parameters fixed. The training rate was $1e-4$. Note that when back propagating training errors from the minimum distance values, the samples with infinite values were not considered for training stability.

Phase III: In the final phase, we fine-tuned the entire network jointly with all three loss equations (2) - (4). The learning rate was reduced to $1e-5$. The error in estimating minimum distance is reduced from 11% to 7%. The scatter plot of the true minimum distance values from the PRM* against the estimated values from the MF is shown in Fig. 5, where a closer distance from a point to the diagonal axis represents a better estimation. The limitations on grid representation and artificially-generated environments for training will be discussed in Section VII.

IV. RSBP: PATH PLANNING

In runtime, the proposed RSBP consists of two stages. In Stage I, a sparse tree is built by using the learnt MF and a reduced search space is then found, which contains a sub-optimal/optimal path. In Stage II, a classical SBP (this paper uses BMFT*) densely sample nodes within the found reduced search space to generate a detailed path. The algorithm is summarized in the Algorithm 1. Details about Stages I and II are given in the following subsections, respectively.

A. STAGE I

Before building a sparse search tree, the entire workspace is decomposed into many small workspaces (cells), the size of which is determined by the size of $\boldsymbol{w}$ used in obtaining data and in training the MF, as shown in Fig. 6. In this paper, a straight forward cubic-cell decomposition is adopted, dividing the entire workspace into an array of cubic cells, as shown in Fig. 6. Such decomposition might not be ideal. It worked for the tested cases in this study. Limitations on this

Algorithm 1 RSBP: Planning

```
Input: Workspace $\mathcal{W}$, initial $\mathbf{q}_0$, goal $\mathcal{G}$, the trained MF $d$.
Output: An optimal (or sub-optimal) path if exists.
Initialize
    Decompose $\mathcal{W}$ into $\{\boldsymbol{w}_i\}_{i \in I_c}$, as well as the obstacle occupation.
    for $\boldsymbol{w} \in \{\boldsymbol{w}_i\}_{i \in I_c}$ do
        Calculate obstacle encoding and restore.
    end
Stage I
    while $n <$ a predefined number do
        Randomly pick a cell $\boldsymbol{w}$ from $\{\boldsymbol{w}_i\}_{i \in I_c}$.
        Sample a node $\phi_{i,n} \in \boldsymbol{\Psi}_i$.
        Calculate minimum distance between $\phi_{i,n}$ and any existing node in $\Phi_i$.
        Connect $\phi_{i,n}$ to the node that leads to optimal accumulated cost $J_{i,n}$.
        Rewire nodes in $\Phi_i$ if better costs exist due to $\phi_{i,n}$.
    end
    Find an optimal sparse path $\{\phi_i^*\}_{i \in I_q}$.
Stage II
    for $i \in \{1, \cdots, M_n\}$ do
        Plan an optimal path $\mathbf{s}_i^*$ connecting $\phi_i^*$ and $\mathcal{G}_i^*$ via BFMT*.
    end
    Concatenate $\mathbf{s}_i^*$ for all $i \in I_p$ to generate $\mathbf{s}^*$.
    return $\mathbf{s}^*$.
```

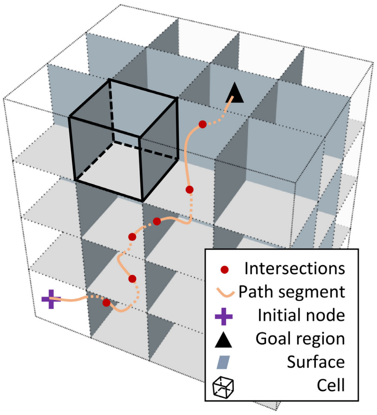


FIGURE 6. Illustration of the workspace decomposition.

decomposition for $\mathcal{W}$ with a arbitrary shape will be discussed in Section VII.

These cells are closed sets and are denoted as $\{\boldsymbol{w}_i\}_{i \in I_c}$, where I_c denote the index set of all cells. Let $\Omega_i = \{\mathbf{x} | \mathbf{x} \in \Omega(\boldsymbol{\omega}_i)\}$ denote the set of positions on the surfaces (boundaries) of the ith cell, where $\Omega(\cdot)$ denotes the operator to obtain the boundaries of a closed set. Then let $\Psi_i = \Omega_i \times \mathbb{SO}(3)$ denote the set of all configurations whose positions are on the surfaces of Cell i. Similarly let $\mathcal{D}$ denote all configurations whose positions are on any cell edge. In Stage I, random samples are generated only in $\boldsymbol{\Psi} = \{\Psi_i\}_{i \in I_c}$. With $\boldsymbol{\Psi}$ formally defined, we give the formal definition of an (optimal) sparse path.

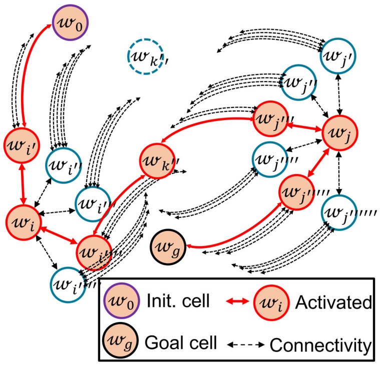


FIGURE 7. The topology graph of the decomposed workspace.

Definition 3 (Sparse Path): A sparse path $\boldsymbol{\phi}$ *of a continuous path* $\boldsymbol{s}$ *is a sequence of the configurations where* $\boldsymbol{s}$ *intersects with cell surfaces* $\boldsymbol{\Psi}$*. The configuration set is denoted as* $\{\boldsymbol{\phi}_i\}_{i \in I_\phi} = \boldsymbol{s} \cap \boldsymbol{\Psi}$*, where* I_ϕ *is the index set of the ordered intersected configurations.*

Definition 4 (Optimal Sparse Path): An optimal sparse path is a sparse path associated with an optimal continuous path $\mathbf{s}^*$*. The configuration set is denoted by* $\{\boldsymbol{\phi}_i^*\}_{i \in I_\phi^*}$*, where* I_ϕ^* *is the index set of the ordered intersected configurations.*

For simplicity, now assume that the optimal path in the large workspace transverses a limited number of cells and that it only transverses through the faces of these cells (not cell edges). In addition, the length measure of path segments on the cell faces is assumed zero. These assumptions do not affect the convergence or the optimality properties of the proposed RSBP and will be discussed with details in Lemma 1 in Section V. Then we have a topological graph, shown in Fig. 7. Each cell is represented by a disk, whose periphery represents the set of all configurations on the cell surface (not cell edges). Note that each connection of the graph in Fig. 7 is bidirectional. A cell is adjacent to other cells up to 6. The connections in Fig. 7 represent a path transversing from one cell to an adjacent cell. Also in the graph, the cells contain the initial node and goal nodes are illustrated and they are denoted as $\boldsymbol{\omega}_0$ and $\boldsymbol{\omega}_g$ (the later could have more than one cells), respectively.

Now consider an optimal path transversing through cells, starting from the initial cell and reaching the goal cell. These cells are referred to as *activated cells*, which are highlighted in red and bold in Fig. 6 (better viewed in color). In this topology, the actual path segment within a cell is not shown and is not planned in Stage I.

Stage I builds a sparse tree in a way same to RRT* [6]. However, the sampling only occurs on the surfaces $\boldsymbol{\Psi}$ of all cells (i.e., peripheries of all disks in Fig. 7), resulting in sparse samples in the workspace. Each node $\phi_{i,j}$ is associated with an accumulated cost value from the initial node, where i and j are the cell index and the sampling sequence index j. Let $J_{i,j}$ denote the accumulated cost of $\phi_{i,j}$ and $J_{i,j}$ is infinite if a path from the initial node to $\phi_{i,j}$ is not available yet.

Let $\Phi_i \subset \Psi_i$ denote the set of existing samples on the surfaces of Cell i ($\boldsymbol{\omega}_i$). Each new node is sampled by randomly choosing a cell index $i \in I_c$ and then by randomly generating a collision-free node $\mathbf{q}_{i,n}$ in Ψ_i. If the cell is the goal cell or the initial cell, the optimal traveling cost and reachabilities from the new node to the goal region and the initial node are estimated by the trained MF. In other cases, the cost and the feasibility between $\phi_{i,n}$ and any node $\phi \in \Phi_i$ are evaluated by the MF.

Notice that $\phi_{i,n}$ can also be on the surface of an adjacent Cell i' (one surface could be shared by two cells). Similarly, we check the cost and the feasibility between $\phi_{i,n}$ and any node $\phi \in \Phi_{i'}$ by using the learnt MF.

Similar to the classical RRT* algorithm, we need to rewire connections to update the sparse tree with more optimal costs. After searching, the new node is connected to the node that results in the most optimal $J_{i,n}$. Then all nodes in Φ_i and $\Phi_{i'}$ are checked. Reconnected Node $\phi_{i,j} \in \Phi_i$ to $\phi_{i,n}$ if

$$J_{i,j} > J_{i,n} + d(\boldsymbol{o}_i, \phi_{i,n}, \phi_{i,j}). \tag{5}$$

where $\boldsymbol{o}_i$ is the obstacle occupancy of Cell i and then set $J_{i,j}$ as the $J_{i,n} + d(\boldsymbol{o}_i, \phi_{i,n}, \phi_{i,j})$.

After a number of samples, a sparse tree is obtained, which connects the initial node to the goal nodes in Fig. 7. Stage I terminates if the number of samples reaches a predefined one. Better termination strategy must be explored in the future study. Then, an optimal sparse path is obtained, which is composed of the sparse nodes and initial and goal nodes. Let I_ω denote the index set of the activated cells and I_q denote the index set of the activated nodes. By connecting these cells together, we find a reduced search space, as illustrated in Fig. 8, denoted as $\mathcal{R} = \{\boldsymbol{\omega}_i\}_{i \in I_\omega} \times \mathbb{SO}(3)$.

Implementation Details on the MF Evaluations: Encoding the environment of a cell via the encoder consists of many convolution operations. Using the MF directly for each combination of a cell environment and two node configurations is inefficient on a device without a GPU. Therefore, after decomposing the entire workspace as shown in Fig. 6, each environment is encoded and the encoding is saved before the step of the sparse sampling. The obstacles and cells are fixed, so are the encodings. Then during the sparse sampling steps, only the FCN is frequently used to estimate the feasibility and the minimum distance between two nodes, which substantially enhances efficiency. Such arrangements tremendously limit the number of calling the CAE encoder, as shown in the "Initialize" block of Algorithm 1. By doing these, the number of calls to generate encodings is restricted to v/u approximately.

B. STAGE II

Stage I outputs the reduced search space $\mathcal{R}$ and the intermediate samples $\{\phi_i^*\}_{i \in I_q}$. One straight forward planning is to use an existing SBP approach and sample in the resultant $\mathcal{R}$. Here, we utilize the additional information $\{\phi_i^*\}_{i \in I_q}$ to enhance the planning efficiency. We design a ball region in configuration space for each $\phi \in \{\phi_i^*\}_{i \in I_q}$ with a small radius γ, resulting $\{\mathcal{G}_i^*\}_{i \in I_q}$. Denote the number of activated cells as M_n. Or equivalently let $M_n \triangleq \mathrm{card}(I_q)$ denote the cardinality of I_q. Then, M_n sub-planning problems are generated, as shown in Fig. 8. Let $\boldsymbol{\zeta}_i \triangleq \boldsymbol{\omega}_{I_\omega(i)} \times \mathcal{SO}(3)$ denote the search space for Subproblem $i \in I_p$, where I_p is the sub-problem index set. Re-index $\{\phi_i^*\}_{i \in I_q}$ and $\{\mathcal{G}_i^*\}_{i \in I_q}$ by the sub-problem index I_p, resulting $\{\phi_i^*\}_{i \in I_p}$ and $\{\mathcal{G}_i^*\}_{i \in I_p}$. Then the first sub-problem is to find a path from the initial node to the ball region $\mathcal{G}_1$ within $\boldsymbol{\zeta}_1$. The last sub-planning problem is to find a path from $\phi_{M_n}^*$ to the goal region $\mathcal{G}$ within $\boldsymbol{\zeta}_{M_n}$. In between, we have $M_n - 2$ sub-problems, where in ith sub-planning problem the body starts at ϕ_i^* and targets at $\mathcal{G}_i^*$ within the search space $\boldsymbol{\zeta}_i$.

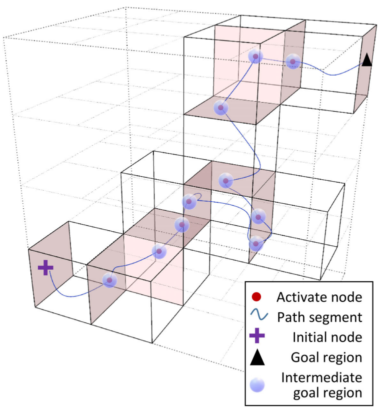


FIGURE 8. Sub-planning problems and optimal path segments.

Once the path segment in each cell is planned, the path segments are concatenated together to produce the optimal path, since γ is set to a sufficiently small number. Note that solving these sub-planning problems can be parallelized for better efficiency. However, in the tests in Section VI these sub-problems were solved one by one.

V. ANALYSIS

In this section, the probabilistic optimality convergence is first proven held in the proposed RSBP, under the assumption that the MF can accurately estimate optimal cost and feasibility between any two nodes in a cell. In fact, as shown in [34], the network work used has the capability to achieve arbitrary accuracy given a sufficient number of neurons. Then, the optimality degeneration is shown bounded under the assumption that the trained MF has bounded errors. After that, the computational complexity is shown linear to the Lebesgue measure of the entire search space, given a fixed sampling density.

A. ASYMPTOTIC OPTIMALITY

It is sufficient to only study the cases that the number of intersections between the optimal path and surfaces is limited and that the optimal path does not transverse through any cell edge, since there exists a path $\mathbf{s}^{*\prime}$ that is arbitrarily close to $\mathbf{s}^*$ (Lemma 1).

Lemma 1: For any $\vartheta > 0$, there exists a strong-δ'-clearance path $\mathbf{s}^{*\prime}$ that such $|\mathbf{s}^{*\prime}(\tau) - \mathbf{s}^{*}(\tau)| \leq \vartheta$ for $\tau \in [0, 1]$, $\mathbf{s}^{*\prime}(\tau)$ intersects with $\boldsymbol{\Psi}$ limited times, and $\mathbf{s}^{*\prime}(\tau) \cap \mathcal{E} = \emptyset$ for $\tau \in [0, 1]$. $\delta' < \delta/2$, where δ is given in Definition 2.

Proof: According to Definition 2, we have $\pi(\alpha)$ that is able to continuously transform $\mathbf{s}^*$ into a path that has strong δ-clearance. Therefore, there exist an α, such that $\mathbf{s}' = \pi(\alpha)$ has an arbitrary δ' clearance such that $2\delta' < \delta$. The maximal difference between $\mathbf{s}^*$ and $\mathbf{s}'$ is a continuous function that is zero when $\alpha = 0$. Thereafter, there exists α for arbitrary ϑ such that $|\mathbf{s}'(\tau) - \mathbf{s}^*(\tau)| \leq \vartheta/2$ and $\mathbf{s}'(\tau)$ has δ'-clearance.

Now without loss of generality, we assume that only one segment ρ sits on a cell surface. We can construct a $\mathbf{s}^{*\prime}$ that is within $\vartheta/2$ to $\mathbf{s}'$ by adding variation to the path segment, which is $\iota \min(\vartheta/2, \delta')\overrightarrow{\mathbf{n}}$. The unit vector $\overrightarrow{\mathbf{n}}$ is the normal of the cell surface. The function $\iota \in [0, 1]$ continuously converges to zero at both endings of ρ. As a result, for any $\vartheta > 0$, we can find $\mathbf{s}^{*\prime}$ that is within ϑ distance to $\mathbf{s}^*$, which has δ'-clearance. Clearly, $\mathbf{s}^{*\prime}$ does not have a segment on a cell surface. The same construction also applies to a point going through an edge by operating on a segment containing this point. □

Therefore it is sufficient to study the probabilistic convergence to $\mathbf{s}^{*\prime}$. In the remainder of this paper $\mathbf{s}^{*\prime}$ is denoted as $\mathbf{s}^*$. Let n and k denote the number of samples in Stage I and Stage II, respectively. To simplify proof, here we assume the initial node and goal nodes are also in $\boldsymbol{\Psi}$. The analysis is conducted in two phases. It is first proven that in Stage I the probability of the activated nodes converging to the optimal sparse path is almost one, as $n \to \infty$ (Lemma 2). Then the asymptotic optimality is achieved by the property of the asymptotic optimality from classical SBP (Lemma 3).

Lemma 2 (Convergence to an Optimal Sparse Path): As $n \to \infty$, $P(\{\{\boldsymbol{\phi}_{n,i}\}_{i \in I_{\phi,n}} = \{\boldsymbol{\phi}^*_i\}_{i \in I^*_\phi}\}) = 1$, where the most outer $\{\cdot\}$ denotes an event.

Here we adopted a norm to denote the difference between two detailed paths from [6],

$$||\mathbf{s} - \mathbf{s}'||_{BV} = \int_0^1 ||\mathbf{s}(\tau) - \mathbf{s}'(\tau)||d\tau. \qquad (6)$$

Let $\mathbf{s}_{k,i}$ denote path generated by Stage II given $\boldsymbol{\phi}^*_i, \mathcal{G}^*_i$. Let $\mathbf{s}^*_i$ denote path segment in the cell associated with the ith sub-planning problem, which is re-parameterized by $\tau \in [0, 1]$. Then, in the second phase of the proof, we show the following lemma.

Lemma 3: As $\eta \to 0$, n and $k \to \infty$, then, for any $i \in I^*_\phi$ and $\varepsilon > 0$, $P(\{||\mathbf{s}_{k,i} - \mathbf{s}^*_i||_{BV} < \varepsilon\}) = 1$.

The proof of Lemma 2 is outlined here, analogous to [6]. We first prove the existence of a sequence of scalars δ_l and sparse paths $\{\boldsymbol{\phi}_{l,i}\}_{i \in I_{\phi,l}}$ with l being the sequence index, such that $\{\boldsymbol{\phi}_{l,i}\}_{i \in I_{\phi,l}}$ has strong δ_l-clearance. In addition, $\lim_{l \to \infty} \delta_l = 0$, and $\lim_{l \to \infty} \{\boldsymbol{\phi}_{l,i}\}_{i \in I_{\phi,l}} = \{\boldsymbol{\phi}^*_i\}_{i \in I^*_\phi}$ (Lemma 4). By construction, the samples within a cell are connected. After that, we prove that the difference between a sparse path and its closest one converges to zero, as $n \to \infty$ (Lemma 5). Then, considering Lemmas 4 and 5 yield that the sparse path converges to an optimal sparse path associated with a weak-δ-clearance optimal path, as $n \to \infty$ (Lemma 2).

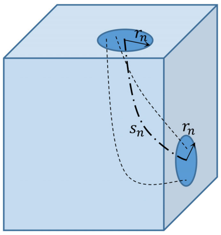


FIGURE 9. Projection of Balls B'_n covering ϕ_n.

Since samples are obtained in an incremental manner in Stage I, the connections between samples also depend on their appearance sequence. Here we adopted a notion of Marked Point Process from [6], where samples $\{\phi_1, \phi_2, \cdots\}$ are associated with $\{\xi_1, \xi_2, \cdots\}$, respectively. Notice that the cell index associated with ϕ_i is dropped here and i is the sampling index (served as the identity). Besides, $\{\xi_1, \xi_2, \cdots\}$ are independently uniformly sampled from the support $[0, 1]$ in order to represent sampling order of samples. For example, ϕ_i is sampled before $\phi_{i'}$ if $\xi_i < \xi_{i'}$.

Here in Stage I, we resort to a new graph, termed as *random cell graph*, the cell size of which is fixed, different to the varying r in a random r-disc graph used in [6].

Definition 5 (Random r-Disc Graph [6]): Let $r \in \mathbf{R}^+$, and $n, d \in \mathbb{N}$. A random r-disc graph $G_n(r)$ in d dimensions is a graph whose n vertices, $\{q_1, q_2, \cdots, q_n\}$, are independent, uniformly distributed random variables in $(0, 1)^d$, and such that $(\mathbf{q}_i, \mathbf{q}_{i'})$, $i \neq i'$, is an edge if and only if $||\mathbf{q}_i, \mathbf{q}_{i'}|| \leq r$ and $\xi_i < \xi_{i'}$.

Definition 6 (Random Cell Graph): Given a cell decomposition, and the union $\boldsymbol{\Psi}$ of their surfaces. A random cell graph G_n in d dimensions is a graph whose n vertices from $\boldsymbol{\Psi}$, $\{\phi_1, \phi_2, \cdots, \phi_n\}$, are independent, uniformly distributed random variables in G_n, and such that $(\phi_i, \phi_{i'})$, $i \neq i'$, is an edge if and only if ϕ_i and $\phi_{i'}$ are from a same cell and $\xi_i < \xi_{i'}$.

As in [6], now consider a random cell graph G_n and another graph $G'_n \subset G_n$. Let $J_{i,j}$ denote the accumulated optimal cost from $\mathbf{q}_0$ in reaching ϕ_j (i being the cell index of ϕ_j). In G'_n, each vertex has a single parent leading to an optimal cost $J_{i,j}$. Since the graph is built incrementally, the cost showing ϕ_j in both G'_n and G_n is same. Apparently, G'_n is from the planning in Stage I.

The following lemma is a variation from Lemma 50 in [6].

Lemma 4: Let $\mathbf{s}^*$ be an optimal path that has strong δ-clearance (see Definition 2). Let $\{\delta_l\}_{l \in \mathbb{N}}$ be a sequence of real numbers, such that $\lim_{l \to \infty} \delta_l = 0$. Since $0 \leq \delta_l \leq \delta$, for all $l \in \mathbb{N}$. Then, there exists a sequence of sparse paths $\{\boldsymbol{\phi}_l\}_{l \in \mathbb{N}}$ such that $\lim_{l \to \infty} \boldsymbol{\phi}_l = \boldsymbol{\phi}^*$.

Proof: The existence of $\boldsymbol{\phi}_l$ can be guaranteed by the existence of $\mathbf{s}_l$, which is guaranteed by Lemma 50 in [6]. □

Let $\mathbf{\Sigma}_n$ denote the set of sparse paths in G_n and let $\boldsymbol{\phi}'_l$ be the sparse path that is closest to $\boldsymbol{\phi}_l$ in the set $\mathbf{\Sigma}_n$.

Lemma 5: The variation between $\boldsymbol{\phi}_{l'}$ and $\boldsymbol{\phi}_l$ converges to zero almost surely, i.e.

$$P(\{\lim_{n\to\infty}\sum_{i\in I_c}||\phi_{l',i}-\phi_{l,i}||=0\})=1. \tag{7}$$

Proof: As shown in Fig. 9, construct a number of balls $B_l = \{B_{l,1}, B_{l,1}, \cdots, B_{l,M_l}\}$ covering $\boldsymbol{\phi}_l$. Since the surface goes through the center of the ball $B_{l,i}$ and the samples are generated on the surface, then we conclude that there exit a number of balls B'_l in a space with one dimension fewer. The projections of B'_l onto the intersected surfaces are illustrated in Fig. 9.

As in [6], the proof is based on the Borel-Cantelli lemma by showing that $\sum_{n\in\mathbb{N}} P(\{\sum_{i\in I_c}||\phi_{l',i}-\phi_{l,i}|| \geq \varepsilon\})$ is finite for any $\varepsilon > 0$, which implies that $\sum_{i\in I_c}||\phi_{l',i}-\phi_{l,i}||$ converges to zero almost surely. Again $\{\cdot\}$ denotes an event. The index n is associated with G_n. Notice that in the remainder of this proof the index l of the path sequence in Lemma 4 is dropped for simplifying notation, therefore $\phi_{',i}=\phi_{l',i}$, $\phi_i=\phi_{l,i}$, and $M_n = M_{l,n}$.

Let $\alpha, \beta \in (0, 1)$ be two constants, both of which are not functions of n. Construct a set B'_n of balls covering the path $\boldsymbol{\phi}$, whose radius is $r_n \triangleq \min\{\delta, 4(\frac{\mu(\mathbf{\Psi})}{\mu(B'_n)})^{1/(d-1)}(\frac{\log n}{n})^{1/(d-1)}\}$. The operator μ denotes a function to obtain the Lebesgue measure of a closed space. Notice that r_n is obtained from [6], with $d-1$ replacing d since B'_n is one dimension fewer than B_n.

The following is analogous to [6]. Now consider a new set B''_n of balls with radius $\beta\delta_n$. Let $I_{n,m}$ denote the event that there is no sample in B''_n, as follows.

$$I_{n,m} \triangleq \begin{cases} 1, & \text{if } (B''_{n,m}) \cap V_n = \emptyset \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Let K_n denote the number of empty balls in B''_n, i.e., $K_n \triangleq \sum_{m=1}^{M_n} I_{n,m}$. Consider the event that $I_{n,m}$ holds for at most an α fraction of the balls in B_n, i.e., $\{K_n \leq \alpha M_n\}$.

Recall that the vertices is subsequent balls in B'_n are connected by edges in G_n. If only at most α fraction of the balls in B''_n are empty, then $\sum_{i\in I_c}||\phi_{',i}-\phi_i||$ is bounded by $(\alpha+\beta)r_n M_n$.

Then taking the complement of both sides and using the monotonicity of probability measures,

$$P(\{\sum_{i\in I_c}||\phi_{',i}-\phi_i|| \geq (\alpha+\beta)\delta_n\}) \leq P(K_n \geq \alpha M_n). \tag{9}$$

It can be shown that the right-hand side of inequality above is summable for all small $\alpha, \beta > 0$. According to Lemma 55 in [6], let $\nu \in (0,\ 1)$ and $p_n \triangleq e^{-\mu(B''_n)\nu n/\mu(\mathbf{\Psi})} \leq e^{-c\beta\nu\log(n)}$ for some c, then we have

$$P(\{K_n \geq \alpha M_n\}) \leq e^{-cn} + e^{-M_n p_n}. \tag{10}$$

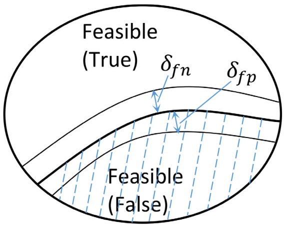


FIGURE 10. **Reachability classification error.**

Summing up both sides yields

$$\sum_{n=1}^{\infty} P(\{K_n \geq \alpha M_n\}) \leq \sum_{n=1}^{\infty} e^{-cn} + e^{-M_n p_n} < \infty, \tag{11}$$

and thus $P(\{\sum_{i\in I_c}||\phi'_{n,i}-\phi_{n,i}|| > \varepsilon\}) < \infty$. □

The proof of Lemma 2 is given as follows.

Proof: Recall that $\boldsymbol{\phi}^*$ denotes the optimal sparse path, and that $\lim_{l\to\infty}\boldsymbol{\phi}_l = \boldsymbol{\phi}^*$. By Lemma 5, $\lim_{n\to\infty}||\boldsymbol{\phi}_{l',n} - \boldsymbol{\phi}_l|| = 0$ holds with probability one. Thus, by repeated application of the triangle inequality, $\lim_{n\to\infty}||\boldsymbol{\phi}_{l',n}-\boldsymbol{\phi}^*|| = 0$. Therefore,

$$P(\{\lim_{n,l\to\infty}||\boldsymbol{\phi}_{l',n}-\boldsymbol{\phi}^*||=0\}) = 1.$$
$$P(\{\lim_{n\to\infty} J_n(\mathcal{G}) = J^*\}) = 1. \tag{12}$$

where J^* is the optimal cost to Problem 1. □

Recall Lemma 3, the optimal path in the ith sub-planning problem between ϕ^*_i and $\mathcal{G}^*_i$ can be obtained by classical optimal SBP approaches. Typical asymptotic optimality property from SBP approaches holds and its proof is neglected here [6].

To sum up, if the sparse path converges to the intersections between the optimal path and the cell surfaces, the obtained path segments from Stage II must be overlapped with the optimal path. Therefore, the path resulted from the proposed RSBP is optimal.

B. OPTIMALITY DEGENERATION

The MF network has two outputs, one being the feasibility between two nodes, and the other being the estimated cost. The optimality degeneration from the MF estimation errors is given based on the following assumptions. The feasibility estimation is a common classification problem, this error is assumed bounded by false positive δ_{fp} and false negative δ_{fn}, as shown in Fig. 10. It is also assumed that the estimated cost has bounded error δ_c, i.e., $|c-c^t| \leq \delta_c$, and that the cost is Lipschitz continuous with support L over the node configuration of the input to the MF network, for any given environment in a cell. These assumptions are commonly used in deep reinforcement learning [35]. The variables with superscript t denote the true values. In addition, it is assumed that the optimal and sub-optimal sparse paths have strong δ-clearance, where $\delta > \delta_{fn}$. Otherwise the RSBP might fail to find these paths.

Given a path $\boldsymbol{\phi} = \{\phi_i\}_{i\in I_q}$ defined as intersection between surfaces and path **s**. Recall that $M \triangleq \text{card}(I_q)$ denotes the cardinality of I_q. For the ith segment of **s**, its endings are ϕ_i and ϕ_{i+1}. Let $c^*_{i,i+1}$ denote the true cost between these two nodes.

Then, the estimated cost of the ith path segment from the MF network is bounded by

$$c_i^t - \delta_c \leq c_i \leq c_i^t + \delta_c. \tag{13}$$

Now consider a case that the optimal sparse path $\boldsymbol{\phi}^*$ transverses through false negative regions. Therefore, ϕ_i has a maximum distance δ_{fn} to ϕ_i^*, which affects the $i-1$th and ith sub-planning problems. Then, for $1 < i \leq M$,

$$c^*_{i-1} - L\delta_{fn} \leq c^I_{i-1} \leq c^*_{i-1} + L\delta_{fn}, \tag{14}$$

$$c^*_i - L\delta_{fn} \leq c^I_i \leq c^*_i + L\delta_{fn}, \tag{15}$$

where c_i^I denotes cost return from Stage I. The cost estimation of a path closest to the optimal solution from Stage I, $\sum_{i\in I_q} c_i^I$, therefore is within

$$\left[\sum_{i\in I_p} c_i^* - 2(M-1)(L\delta_{fn} + \delta_c), \right.$$

$$\left. \sum_{i\in I_p} c^*_{i-1,i} + 2(M-1)(L\delta_{fn} + \delta_c)\right].$$

Then consider the path founded by RSBP in a case where the ϕ_i is incorrectly classified as collision-free. Then, the resultant path from Stage I ends up in an infeasible choice. In Stage II, a number of samples will be generated near this infeasible samples, resulting in a new path. The maximum deviation is δ_{fp} and the misclassification of the ϕ_i^* leads to cost estimation errors of $i-1$th and ith path segments, as shown below. Then for $1 < i < M_n$,

$$c^I_{i-1} - L\delta_{fp} - \delta_c \leq c^{II}_{i-1} \leq c^I_{i-1} + L\delta_{fp} + \delta_c, \tag{16}$$

$$c^I_i - L\delta_{fp} - \delta_c \leq c^{II}_i \leq c^I_i + L\delta_{fp} + \delta_c. \tag{17}$$

The sparse path found in Stage II $\sum_{i\in I_q} c^{II}_{ii}$, therefore is within the range as follows,

$$\left[\sum_{i\in I_p} c_i^I - 2(M-1)(L\delta_{fp} + \delta_c), \right.$$

$$\left. \sum_{i\in I_p} c^I_i + 2(M-1)(L\delta_{fp} + \delta_c)\right]. \tag{18}$$

Consider the worst scenario such that the sparse path found in Stage I only yields potential optimality given by $\sum_{i\in I_q} c_i^* + 2(M-1)(L\delta_{fn} + \delta_c)$ and that feasibility error and estimation error lead to a feasible path with more cost. Equations (16) and (18) yield an bound optimality of the path obtained by RSBP, as follows,

$$\sum_{i\in I_q} c_i^* + 2(M-1)(L\delta_{fp} + L\delta_{fn} + \delta_c), \tag{19}$$

where the second summation is the bounded optimality degeneration.

C. COMPUTATIONAL COMPLEXITY

Here the computational complexity from planning (not learning) in the RSBP regarding the size of the workspace is analyzed. Recall that n denotes the number of samples in Stage I and k denotes the number of samples in Stage II. The complexity of Stage I is based on the number of calling the MF network. Each call has the complexity of $O(zn_h^3)$, where z is the number of the network layers and n_h is the maximum number of the neurons in each layer. For a given cell with n_c samples, the complexity in connecting nodes is then $O(n_c^2 z n_h^3)$. While given a fixed sampling density across all environment of different sizes, since the area of surfaces of a cell is fixed, thus the expected number of samples associated with one cell is fixed. Then the complexity in connecting nodes in each cell is fixed thus $O(1)$. The number of cells is linear to the Lebesgue measure of an entire search space $\mathcal{C}$, so is the computational complexity in Stage I.

In Stage II, the complexity of finding an optimal in a cell is given by [6], which is $O(k \log k \log^d N)$. Recall N is the number of obstacles in a cell. Again, for a cell of fixed size, it is reasonable to assume N is bounded by a constant and the required k to achieve acceptable path is also bounded by a constant.

Altogether, the computational complexity in Stage I and Stage II is $O(Mk \log k \log^d N + Mn_c^2 l n_h^3)$. Recall M is the number of cells transversed by the optimal path. Then by assuming the same sampling density across environments of different sizes, the complexity is $O(M)$ (M is linear to the Lebesgue measure of $\mathcal{C}$).

VI. NUMERICAL VALIDATIONS

The numerical simulations in this paper considered optimal planning problems of a given rigid-body robot in 3D large workspaces populated with box-shaped obstacles (possibly overlapping). Recall that the robot is box-shaped with dimensions of 0.75 × 0.5 × 0.25 $[m^3]$. The workspaces are also box-shaped, whose dimensions are a multiplication of the cell dimensions with some integers. Decompositions on the workspace of an arbitrary shape will be discussed in Section VII.

In the first set of simulations, a workspace of sizes 78 × 30 × 6 $[m^3]$ was considered to comprehensively test the proposed RSBP against Informed RRT* and BFMT* in cluttered environments, regarding the resultant path optimality and the planning time. In each of these simulations, the obstacle population was randomly generated in the same way as Section III. The second set of simulations involved a number of workspaces of increasing sizes to show the scalability of the RSBP. At last, a non-cluttered workspace with narrow passages was designed to demonstrate the potential advantage of the RSBP in finding narrow passages.

The implementations of the BFMT* and the Informed RRT* in OMPL were used [25]. In both methods, the goal threshold was set to 0.1. The parameters of BFMT* are chosen as follows. The free space volume was set to 0.1 and the number of samples was the multiplication of $1/216$ $[K/m^3]$ and the volume of the workspace (i.e., 1K samples per cell on average). Other parameters were kept as default. Regarding Informed RRT*, the resolution of state validity check was set to 0.1, the range was set to 3.0, and other parameters were kept as default. Notice that the planning time of Informed RRT* were carefully manually adjusted (in an incremental manner) to obtain a complete path for each simulation.

Notice that such comparison is unfair since BFMT* and Informed RRT* are not using any prior knowledge about the environment/obstacle distribution, the robot, or previous planning results. However, it is still promising to integrate learning techniques to summarize knowledge from previous planning and to enhance planning efficiency in a new and similar environment.

A. CLUTTERED ENVIRONMENTS

An example of obstacles' population in the workspace of sizes $78 \times 30 \times 6$ $[m^3]$ is depicted in Fig. 12, where the robot's initial configuration is indicated by a cross and its goal configuration is illustrated by a solid triangle. The parameters of cell sizes have been discussed in Section III. The training results of the MF have been illustrated in Fig. 5.

The entire workspace was decomposed into 65 cells and the obstacle occupation in each cell was encoded by the encoder of the CAE from Section III. Nodes were sampled on the surfaces of these cells and then the minimum distance between any two nodes in the same cell was evaluated by the FCN. On average 30 sparse samples per surface of a cell were generated. A sparse tree was built and searched for an optimal sparse path that is composed of the sparse samples. The results (shown in Fig. 11) were summarized over 20 runs with random obstacle populations and fixed robot initial and goal nodes. Stage I took about 23 seconds to find the reduced search space. Notice that in some tests, Stage I took nearly 70 seconds to finish building the sparse tree. The reason could be that the sampling in certain cell surface is quite challenging due to a small collision free region.

The activated samples $(\Phi*)$ are highlighted by blue spheres in Fig. 12. The obtained reduced search space is shown by the transparent boxes. Then, the planning problem in the entire workspace was decoupled into 13 sub-planning problems. Solving the 13 sub-planning problems can be easily parallelized. However, in the results given here, these 13 subproblems were solved one by one. Each subproblem took about 0.5 to 0.8 second via the BFMT*, using approximately 1K samples per subproblem.

For comparison, we solved the same planning problem by BFMT* and Informed RRT* directly. An initial path for Informed RRT* might be non-trivial in these environments. The number of samples in BFMT* was set to 65K and the parameter for Informed RRT* was set to 100 seconds. Informed RRT* could not provide complete paths in 90 seconds most of the time and it sometimes finished in less than 100 seconds. The planning time for Informed RRT* was manually picked by trial-and-error (through gradually increasing the planning time).

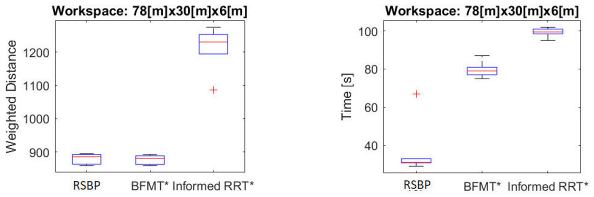


FIGURE 11. Comparison of RSBP, BFMT*, and Informed RRT*.

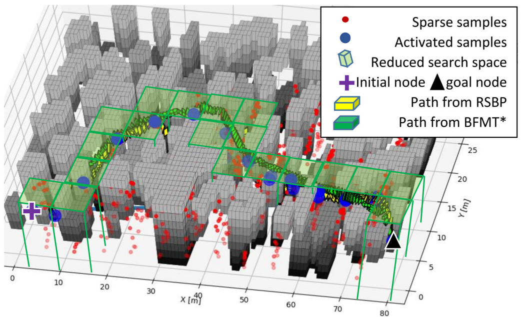


FIGURE 12. Comparison of the optimal paths generated by RSBP and BFMT*, showing similar paths were found. The union of the transparent boxes are the obtained reduced search space (better viewed in color).

The comparison results are summarized in Fig. 11. For the tested cases, the proposed RSBP method generated paths of similar optimality to BFMT*, while RSBP took less than half of the time required by BFMT*. The paths from RSBP and BFMT* are shown in Fig. 12, which are very close. The Informed RRT* was not efficient for the large cluttered environment tested in this study. The resultant path from it contains unnecessary detour.

B. SCALABILITY

A set of workspaces with varying sizes were simulated to test the scalability of RSBP. In addition to the aforementioned tests, we included another six sizes, the width and height are fixed. All together, the length of all workspaces considered are chosen in the set of $\{78, 102, 138, 174, 210, 282, 352\}[m]$. Then the volumes of all workspaces are $\{1.4, 1.84, 2.49, 3.13, 3.78, 5.08, 6.37\} \times 10^4[m^3]$, each workspace is subsequently indexed by I, $\cdots$, and VII. Intuitively, the number of samples required and computational complexity increases as the volume of the workspace increases.

As given in [6], the complexity is $n \log n$, where the sample number n is proportional to the volume of the workspace, if the sampling density stays the same. Considering the rewiring process and updates of minimum cost at each tree node, the complexity may be higher [8]. The complexity is evidenced by the results shown in Fig. 13.

By using RSBP, the number of cells that required to check is proportional to the workspace volume (or the Lebesgue

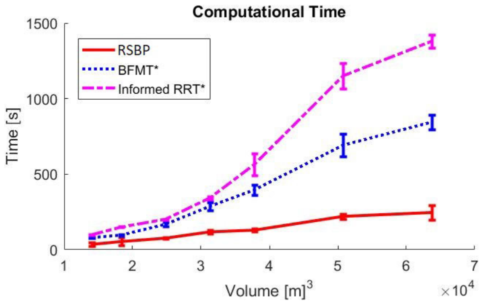


FIGURE 13. Computational time comparison of RSBP, BFMT*, and Informed RRT*.

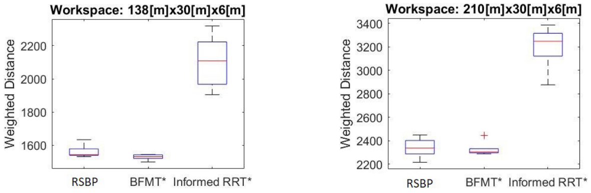


FIGURE 14. Performance comparison of RSBP, BFMT*, and Informed RRT* in the workspaces of Type III and IV.

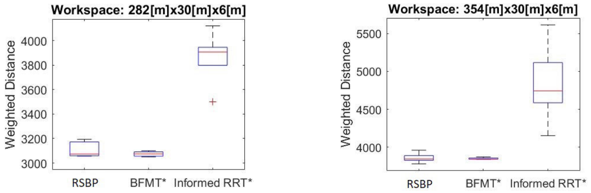


FIGURE 15. Performance comparison of RSBP, BFMT*, and Informed RRT* in the workspaces of Type V and VI.

measure of the entire search space). Thus the complexity in Stage I is linear to the workspace volume, better than $n/\log n$. In Stage II, the complexity is again proportional to the length of the path (in these experiments proportional to the volume of the workspace). Such analysis is evidenced by results shown in Fig. 13.

As for the performance among three compared methods, the observation was similar to the one in the previous subsection. The proposed RSBP approach produced paths with similar optimality to BFMT*, while Informed RRT* could not offer competitive solutions. Following figures show examples of performance comparisons on path optimality.

C. NARROW PASSAGES

In the above simulations, the distributions of obstacle population in the tests and in the MF network training process share the same one, where narrow passages hardly present. We used the trained MF for testing the narrow passage cases, the prediction error from the MF evaluation increased a lot. This is expected since the training samples did not cover the narrow passage scenarios. Therefore, we designed a distribution to randomly generate 1K narrow passages, formed by three cubes of random sizes. The optimal cost and feasibility

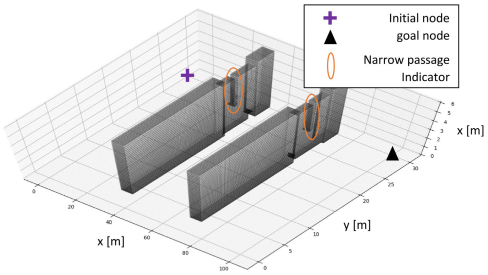


FIGURE 16. The workspace with two narrow passages.

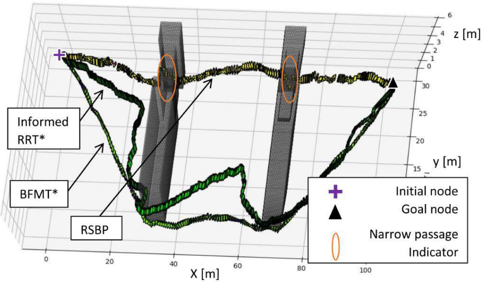


FIGURE 17. Paths from RSBP, BFMT*, and Informed RRT*.

in cells with narrow passages were obtained by PRM*. This new set of data was included in the previous training data set and was used to retrain the MF network. After training, the new network is able to estimate optimal cost with the error being about $6-9\%$ of the optimal values.

We then designed a new workspace, where two vertical blocks created two narrow passages, as shown in Fig. 16. In this tests, the chance that the path found by BFMT* transverses through both narrow passages was low (less than 7%) with $85k$ samples, and the chance of transversing one passage was still not satisfied (less than 15%). While the chance of Informed RRT* finding a narrow passage was a bit higher than BMFT*, since more samples were generated (more planning time was allowed to obtain a complete path). On the contrary, the proposed RSBP was able to find the narrow passage almost 100%. The paths from the three methods are illustrated in Fig. 17. One of the purposes of this test is to show that RSBP's capability in dealing with some special cases can be enhanced by more data and training. Therefore the sampling strategy called *bridge-test* for planning path in narrow passages is not compared here [21].

VII. LIMITATIONS AND FUTURE WORK

A. WORKSPACE DECOMPOSITION

In principle, the workspace decomposition does not have to align with the workspace axes. Cells can be overlapped when the workspace has an arbitrary shape. However, the computational complexity during planning might increase due to the increased number of cells. In fact, the workspace decomposition in Stage I is not necessary for building a sparse tree since the sparsity of nodes is enabled by the learnt MF. However, if the node sampling and connection is conducted in the same way as classical SBP, each MF evaluation of two nodes (within a cell) requires encoding the local environment from scratch. The encoding consists of many convolution layers in the CAE encoder, making the evaluation particularly slow on an onboard machine with no GPU. The decomposition procedure is able to significantly reduce the number of calling the CAE encoder.

In the current implementation, the size of cells has to be determined as a priori, which depends on the obstacle shapes and populations. Zoom-out-and-in networks [36], an extension of convolution networks, might be used to adjust the cell size online. However, this may require even more training data and needs further investigation.

B. INEFFICIENCY

Based on the current implementation, the proposed RSBP is slower in small and uncluttered environments than BFMT* and Informed RRT*. The proposed sparse sampling could be combined with other existing strategies on node sampling and connection. Future work involves improving the efficiency in the MF evaluation through model reduction on the trained neural network [37].

C. NETWORK GENERALITY

At shown by the narrow passage scenario in Section VI, the MF network has to be retrained if the robot geometry, the distance function, or obstacle shape and population are dramatically different from previous training. Obstacle variations during planning have to be similar to the training examples. In other cases, the proposed RSBP approach is not reliable by itself. Many techniques in transfer learning could be explored [38]. A more generative neural network model must be investigated.

D. DATA AND TESTS

In this paper, the environments for training and testing are artificially generated from the same distribution of boxes and they may not reflect real-world scenarios. Future work includes collecting data from a more realistic simulator and the real world.

E. OBSTACLE REPRESENTATION

Different than classical SBP, obstacle information is used explicitly rather than in a black-box fashion. The assumption of knowing obstacle occupancy without error is strong. The proposed algorithm is not able to deal with the uncertainties of obstacles. However, the value of each grid can represent an occupancy probability, which may extend the current approach to situations with obstacle uncertainties. In fact, the loss function in training, i.e., eq. (2) is probabilistic in nature.

Such obstacle representation can be built from Bayesian SLAM algorithms [39]. The resolution loss from the grid representation of obstacles contributes to errors of MF evaluations. Point clouds might be a better representation, which however is usually unordered and makes training of the neural networks more difficult. Some existing work on using recurrent neural networks for encoding point clouds can be explored [40].

F. ORTHOGONAL REDUCTION

Reduction on the dimensions of the configuration/state space is not considered in this paper, which has been explored in many robotic applications [41]. Non-uniform dimension reduction for motion planning will be explored and might be combined with the proposed RSBP.

G. DYNAMIC CONSTRAINTS

The dynamic constraints force the MF to take the robot velocities into account. The training data should also reflect these constraints. Therefore these SBP approaches (e.g. PRM*) built in the configuration space are not sufficient. Constraint-based planning should be adopted instead.

VIII. CONCLUSION

This paper introduces a search-space-reduced sampling and planning approach for optimal paths in large and cluttered environments. The capability of the proposed method can be enhanced by adding more data and improving the network structure. It may allow reducing efforts in designing and combining problem-specific sampling strategies. The neural network model based on a contractive autoencoder is able to estimate optimal traveling cost and reachability between two nodes. The RSBP is proven to hold asymptotic optimality and the bounded optimality degeneration. Numerical simulations have also demonstrated that RSBP outperforms the BFMT* and Informed RRT* in large and cluttered scenarios. As evidenced by analysis and numerical simulations, the computational complexity of RSBP is linear to the Lebesgue measure of the entire search space.

REFERENCES

[1] M. Chyba, D. Cazzaro, L. Invernizzi, and M. Andonian, "Trajectory design for autonomous underwater vehicles for basin exploration," in *Proc. 9th Int. Conf. Comput. IT Appl. Maritime Industries*, 2010, pp. 139–151.

[2] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, "Learning monocular reactive UAV control in cluttered natural environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2013, pp. 1765–1772.

[3] H. Wei, W. Lu, P. Zhu, S. Ferrari, G. Huang, and J. Leonard, "Visibility-based motion planning for active target tracking and localization," in *Proc. IROS*, Chicago, IL, USA, 2014, pp. 76–82.

[4] P. Bhattacharya and M. L. Gavrilova, "Roadmap-based path planning—Using the Voronoi diagram for a clearance-based shortest path," *IEEE Robot. Autom. Mag.*, vol. 15, no. 2, pp. 58–66, Jun. 2008.

[5] A. Swingler and S. Ferrari, "A cell decomposition approach to cooperative path planning and collision avoidance via disjunctive programming," in *Proc. 49th IEEE Conf. Decision Control (CDC)*, Dec. 2010, pp. 6329–6336.

[6] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[7] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 2997–3004.

[8] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 883–921, Jun. 2015.

[9] J. Starek, E. Schmerling, L. Janson, and M. Pavone, "Bidirectional fast marching trees: An optimal sampling-based algorithm for bidirectional motion planning," in *Proc. Workshop Algorithmic Found. Robot.*, 2014, pp. 1–16.

[10] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 2537–2542.

[11] M. Kleinbort, O. Salzman, and D. Halperin, "Collision detection or nearest-neighbor search? On the computational bottleneck in sampling-based motion planning," 2016, *arXiv:1607.04800*. [Online]. Available: https://arxiv.org/abs/1607.04800

[12] J. M. Phillips, N. Bedrossian, and L. E. Kavraki, "Guided expansive spaces trees: A search strategy for motion- and cost-constrained state spaces," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 4, Apr./May 2004, pp. 3968–3973.

[13] I. Garcia, J. D. Martín-Guerrero, E. Soria-Olivas, R. J. Martínez, S. Rueda, and R. Magdalena, "A neural network approach for real-time collision detection," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 5, Oct. 2002, p. 5.

[14] J. Pan and D. Manocha, "Fast probabilistic collision checking for sampling-based motion planning using locality-sensitive hashing," *Int. J. Robot. Res.*, vol. 35, no. 12, pp. 1477–1496, 2016.

[15] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 7087–7094.

[16] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 3067–3074.

[17] J. A. Starek, J. V. Gomez, E. Schmerling, L. Janson, L. Moreno, and M. Pavone, "An asymptotically-optimal sampling-based algorithm for Bidirectional motion planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2015, pp. 2072–2078.

[18] M. Otte and N. Correll, "C-FOREST: Parallel shortest path planning with superlinear speedup," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 798–806, Jun. 2013.

[19] C. Holleman and L. E. Kavraki, "A framework for using the workspace medial axis in PRM planners," in *Proc. ICRA. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia*, vol. 2, Apr. 2000, pp. 1408–1413.

[20] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. H. Reif, "Narrow passage sampling for probabilistic roadmap planning," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1105–1115, Dec. 2005.

[21] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, Sep. 2003, pp. 4420–4426.

[22] P. Cheng and S. M. LaValle, "Resolution complete rapidly-exploring random trees," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 1, May 2002, pp. 267–272.

[23] K. Ok, S. Ansari, B. Gallagher, W. Sica, F. Dellaert, and M. Stilman, "Path planning with uncertainty: Voronoi uncertainty fields," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2013, pp. 4596–4601.

[24] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 484–499.

[25] I. A. Şucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012.

[26] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," 2016, *arXiv:1608.04236*. [Online]. Available: https://arxiv.org/abs/1608.04236

[27] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "FPNN: Field probing neural networks for 3D data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 307–315.

[28] I. Ko, B. Kim, and F. C. Park, "VF-RRT: Introducing optimization into randomized motion planning," in *Proc. 9th Asian Control Conf. (ASCC)*, Jun. 2013, pp. 1–5.

[29] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3674–3683.

[30] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[31] A. Banino *et al.*, "Vector-based navigation using grid-like representations in artificial agents," *Nature*, vol. 557, no. 7705, p. 429, May 2018.

[32] C. Le Gentil, T. Vidal-Calleja, and S. Huang, "IN2LAMA: Inertial lidar localisation and MApping," in *Proc. Int. Conf. Robot. Automat.*, May 2019, pp. 6388–6394.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[34] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3577–3586.

[35] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver, "Successor features for transfer in reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4055–4065.

[36] H. Li, Y. Liu, W. Ouyang, and X. Wang, "Zoom out-and-in network with map attention decision for region proposal and object detection," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 225–238, 2019.

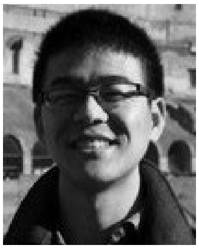
[37] A. Koul, S. Greydanus, and A. Fern, "Learning finite state representations of recurrent policy networks," 2018, *arXiv:1811.12530*. [Online]. Available: https://arxiv.org/abs/1811.12530

[38] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[39] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 647–665, 2008.

[40] F. Liu, S. Li, L. Zhang, C. Zhou, R. Ye, Y. Wang, and J. Lu, "3DCNN-DQN-RNN: A deep reinforcement learning framework for semantic parsing of large-scale 3D point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5678–5687.

[41] K. Yuan, I. Chatzinikolaidis, and Z. Li, "Bayesian optimization for whole-body control of high-degree-of-freedom robots through reduction of dimensionality," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2268–2275, Jul. 2019.



WENJIE LU received the B.S. degree in mechatronic engineering from Zhejiang University, Hangzhou, China, in 2009, and the M.S. and Ph.D. degrees in mechanical engineering from Duke University, Durham, NC, USA, in 2011 and 2014, respectively. His research interests include the learning of efficient motion planners and controllers for mobile robots that are subject to uncertainties in 3D complex environment. He has been developing such algorithms via integrating machine learning, hybrid systems, information theories, and nonparametric Bayesian models.

DIKAI LIU received the Ph.D. degree, in 1997. His main research interests include robotics including exploration, motion planning, robot teams, and physical human–robot interaction. He has been developing novel methods and algorithms that enable robots to operate in unstructured and complex 3D environments autonomously or collaboratively with human users. Example robotic systems, he developed and practically deployed include autonomous grit-blasting robots for steel bridge maintenance, assistive robots for human strength augmentation in industrial applications.

• • •