Supplementary Materials for Occupancy Grid Mapping without Ray-Casting for High-resolution LiDAR sensors

Yixi Cai, Fanze Kong, Yunfan Ren, Fangcheng Zhu, Jiarong Lin, Fu Zhang

I. Proof of Theorem 1

Proof. The volume of free space $\boldsymbol{V}$ is composed of two parts: center free space $\boldsymbol{V}_{\texttt{center}}$ and free space of rays $\boldsymbol{V}_{\texttt{ray}}$, as shown in Fig. 1. The radius r of the center sphere is obtained in consideration of vertical FoV $\Phi \in [-\alpha, \alpha]$ as follows:

$$r = \sqrt{3}\sin(\min\{\text{atan}(\frac{1}{\sqrt{2}}) + \alpha, \frac{\pi}{2}\})R/\gamma \tag{1}$$

An illustration to calculate the radius r is presented in Fig. 2. Let $\gamma_0 = \sqrt{3}\sin(\min\{\text{atan}(\frac{1}{\sqrt{2}}) + \alpha, \frac{\pi}{2}\})$, then $r = \frac{\gamma_0 R}{\gamma}$. The volume of center space $\boldsymbol{V}_{\texttt{center}}$ can be obtained in proportion to the entire free space $\boldsymbol{V_I}$ as

$$\boldsymbol{V}_{\texttt{center}} = (\frac{r}{R})^3 \boldsymbol{V}_I = (\frac{\gamma_0}{\gamma})^3 \boldsymbol{V}_I \tag{2}$$

Since the volume of center space $\boldsymbol{V}_{\texttt{center}}$ is smaller than the entire free space, γ_0/γ should be smaller than 1, leading to $\gamma > \gamma_0$. Otherwise, we have $f(\gamma) = 1.0,\ 0 < \gamma \leqslant \gamma_0$.

For the volume of free space of rays $\boldsymbol{V}_{\texttt{ray}}$, we consider the points $\mathcal{P} = \{p_{i,j} | i = 1, \ldots, N, j = 1, \ldots, \mathcal{M}_{\texttt{ray}}\}$ on the depth image $\mathcal{M}$ where N and M are the sizes of two dimension of the depth image. For each point $p_{i,j}$, the number of cells traversed by a ray from the origin to the point is the same as counting the cells intersected with the diagonal of a cuboid $\mathcal{C}$ as:

$$\begin{aligned}\mathcal{N}_{\texttt{cell}} =& a + b + c - \gcd(a, b) \\ & - \gcd(b, c) - \gcd(a, c) + \gcd(a, b, c)\end{aligned} \tag{3}$$

where a, b, c are the three dimensions of the cuboid. Suppose the point $p_{i,j}$ is projected to the depth image $\mathcal{M}$ with the spherical angle of (ϕ_i, θ_j), the size of the equivalent cuboid $\mathcal{C}$ is

$$\begin{aligned} a = \frac{R-r}{d}\cos(\phi_i)\cos(\theta_i),\ b = \frac{R-r}{d}\cos(\phi_i)\sin(\theta_i) \\ c = \frac{R-r}{d}\sin(\phi_i)\end{aligned} \tag{4}$$

Thus, the number of cells traversed by a ray from the origin to a point $p_{i,j}$ is obtained by approximation of (3) as:

$$\mathcal{N}_{\texttt{cell},i,j} = \frac{(R-r)}{d}\big(\text{c}(\phi_i)(\text{c}(\theta_j) + \text{s}(\theta_j)) + \text{s}(\phi_i)\big) \tag{5}$$

Yixi Cai and Fanze Kong contributed equally to this work.
Corresponding author: Fu Zhang.
The authors are with Mechatronics and Robotic Systems (MaRS) Laboratory, Department of Mechanical Engineering, University of Hong Kong, Hong Kong SAR, China. (email: {yixicai, kongfz, renyf, zhufc, jiarong.lin}@connect.hku.hk; fuzhang@hku.hk)

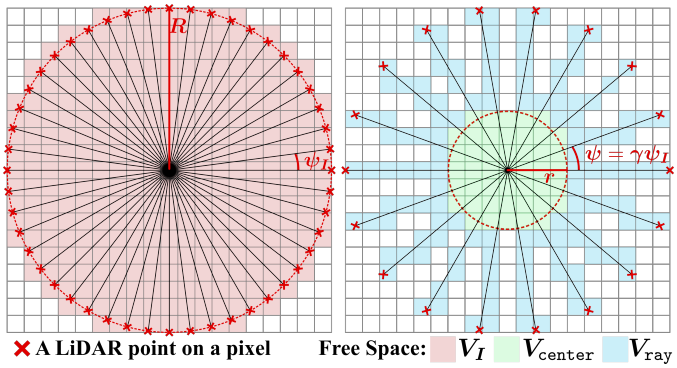

Fig. 1. (a) Free space $\boldsymbol{V}_I$ determined by projecting to $\mathcal{M}_I$ with a resolution of ψ_I. (b) Free space $\boldsymbol{V}$ determined by projecting to$\mathcal{M}$ with a resolution of $\psi = \gamma\psi_I$, which is composed of $\boldsymbol{V}_{\texttt{center}}$ and $\boldsymbol{V}_{\texttt{ray}}$.

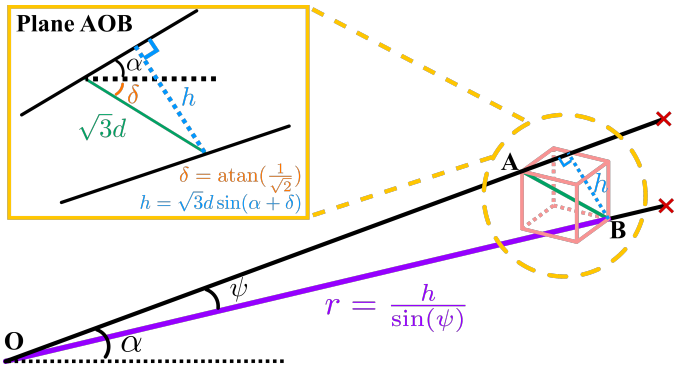

Fig. 2. When casting rays from the sensor origin **O** to the LiDAR points on two neighbor pixels, there exist points **A** and **B** on the rays that the length of **AB** is large enough to contain a cell of size d. In this case, the length **AB** equals the diagonal of the cell, which is $\sqrt{3}d$, and the calculation of **OB** is conducted on plane **AOB**. The length of **OB** is the center radius r.

where $\text{c}(\cdot)$ and $\text{s}(\cdot)$ denote $\cos(\cdot)$ and $\sin(\cdot)$, respectively. As the volume of each cell is d^3, the total volume of these cells is derived as

$$\boldsymbol{V}_{\texttt{ray},i,j} = (1 - \frac{\gamma_0}{\gamma})\big(\text{c}(\phi_i)(\text{c}(\theta_j) + \text{s}(\theta_j)) + \text{s}(\phi_i)\big)Rd^2 \tag{6}$$

When summing the volume of free space traversed by each ray, we consider $1/8$ of the points $\mathcal{P}$ for simplicity, which locates in an octant of free space $\boldsymbol{V}$ corresponding to the FoV angle of $[0, \pi/2] \times [0, \alpha]$. The depth image dimension corresponding to the octant space is calculated as:

$$N \approx \frac{\alpha}{\gamma\psi_I},\quad M \approx \frac{\pi}{2\gamma\psi_I} \tag{7}$$

The spherical coordinate (ϕ_i, θ_j) of $p_{i,j}$ can be obtained from

the depth image resolution ψ as follows:

$$\phi_i = i\gamma\psi_I,\ \theta_j = j\gamma\psi_I \tag{8}$$

Summarizing the volume of free space covered by the rays from the origin to the point cloud $\mathcal{P}$ yields:

$$\boldsymbol{V}_{\texttt{ray}} = 8\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\boldsymbol{V}_{\texttt{ray},i,j} = (4\pi(1-\mathrm{c}(\alpha))+16\mathrm{s}(\alpha))(1-\frac{\gamma_0}{\gamma})\frac{R^3}{\gamma^2} \tag{9}$$

The volume of free space $\boldsymbol{V}_I$ is equivalent to the volume of LiDAR FoV, which can be obtained by the volume of a sphere cut by two sphere caps and two cones, which is

$$\boldsymbol{V}_I = \frac{4\pi R^3}{3} - \frac{2\pi R^3}{3}(\mathrm{s}(\alpha)\mathrm{c}^2(\alpha) - (1-\mathrm{s}(\alpha))^2(2+\mathrm{s}(\alpha))) \tag{10}$$

Finally, the accuracy function for the 3-Dimensional case is

$$f(\gamma) = \frac{\boldsymbol{V}_{\texttt{center}}+\boldsymbol{V}_{\texttt{ray}}}{\boldsymbol{V}_I} = (\frac{\gamma_0}{\gamma})^3 + A(1-\frac{\gamma_0}{\gamma})\frac{1}{\gamma^2},\ \gamma > \gamma_0 \tag{11}$$

where A is a constant factor relative to LiDAR's vertical FoV range α

$$A = \frac{3(1-\mathrm{c}(\alpha)) + \frac{12}{\pi}\mathrm{s}(\alpha)}{1 - \frac{1}{2}(\mathrm{s}(\alpha)\mathrm{c}^2(\alpha) + (1-\mathrm{s}(\alpha))^2(2+\mathrm{s}(\alpha)))} \tag{12}$$

Summarizing the results above, the accuracy function $f(\gamma)$ is provided as

$$f(\gamma) = \begin{cases} \frac{\gamma_0^3}{\gamma} + A(1-\frac{\gamma_0}{\gamma})\frac{1}{\gamma^2} & \gamma > \gamma_0 \\ 1 & 0 < \gamma \leqslant \gamma_0 \end{cases} \tag{13}$$

■

II. Proofs for Time Complexity Analysis

A. Proof of Lemma 1

Proof. The on-tree update strategy searches the nodes inside or intersecting with the sensing area. To analyze the time complexity, we examine the worst-case scenario for updating the D-Map, which occurs when the octree splits the sensing area into the smallest size. This situation can arise when the point cloud has a serrated shape (as shown in Fig.9(a) in the primary article), leading to a depth image resembling a checkerboard where there are abrupt changes in depth values from a minimum to a maximum between neighboring pixels. In such situations, the number of leaf nodes visited in the D-Map is equivalent to the number of cells traversed by rays of points on the depth image. For each ray, the number of traversed cells is approximated as $\boldsymbol{\mathcal{O}}(R/d)$, as derived in (3) and (5). Thus, the total number of leaf nodes in the worst case is given as:

$$\mathcal{S}_{\texttt{leaf}} = \boldsymbol{\mathcal{O}}(\frac{nR}{d}) \tag{14}$$

Next, we consider the other nodes visited along the path toward the leaf nodes. The number of these accompanying nodes $\mathcal{S}_{\texttt{acc}}$ is determined by the height of the octree, which is $\log(D/d)$, as follows:

$$\mathcal{S}_{\texttt{acc}} = \boldsymbol{\mathcal{O}}(\frac{nR}{d}\log(\frac{D}{d})) \tag{15}$$

Therefore, the total number of visited nodes in the worst case of our on-tree update strategy is obtained as

$$\begin{aligned} \mathcal{S}_{\texttt{DMap}} &= \mathcal{S}_{\texttt{leaf}} + \mathcal{S}_{\texttt{acc}} \\ &= \boldsymbol{\mathcal{O}}(\frac{nR}{d}) + \boldsymbol{\mathcal{O}}(\frac{nR}{d}\log(\frac{D}{d})) \\ &= \boldsymbol{\mathcal{O}}(\frac{nR}{d}\log(\frac{D}{d})) \end{aligned} \tag{16}$$

■

B. Proof of Lemma 2

Proof. When considering the procedure of occupancy state determination, the time complexity is dominated by the range query on the 2-D segment tree. As explained in Section IV-B in the primary article, the time complexity of a range query on a 2-D segment tree is $\boldsymbol{\mathcal{O}}(\log N \log M)$ where N and M are the sizes of two dimensions. In our case, the 2-D segment tree is constructed from a depth image, with $N = \lceil \Phi_I/\psi_I \rceil$ and $M = \lceil \Theta_I/\psi_I \rceil$. Thus, the time complexity of occupancy state determination is $\boldsymbol{\mathcal{O}}(\log(\Phi_I/\psi_I)\log(\Theta_I/\psi_I))$. ■

C. Proof of Theorem 2

Proof. In the on-tree update strategy, an occupancy state determination process is applied to each node that is visited. The time complexity of this process is given by Lemma 2. Additionally, the number of visited nodes is given by $\mathcal{S}_{\texttt{DMap}}$ in Lemma 1. Thus, the overall time complexity can be expressed as the product of these two factors:

$$\boldsymbol{\mathcal{O}}\Big(\frac{nR}{d}\log(\frac{D}{d})\log(\frac{\Phi_I}{\psi_I})\log(\frac{\Theta_I}{\psi_I})\Big) \tag{17}$$

In D-Map, the depth image resolution is bounded by (2) as:

$$\psi_I = \max\{\frac{d}{R}, \psi_{\texttt{lidar}}\} \geqslant \psi_{\texttt{lidar}} \tag{18}$$

Therefore, the time complexity in (17) can be written as:

$$\boldsymbol{\mathcal{O}}\Big(\frac{nR}{d}\log(\frac{D}{d})\log(\frac{\Phi_I}{\psi_{\texttt{lidar}}})\log(\frac{\Theta_I}{\psi_{\texttt{lidar}}})\Big) \tag{19}$$

Considering that Φ_I and Θ_I are bounded by constants (e.g., π) and $\psi_{\texttt{lidar}}$ is a constant parameter related to the LiDAR sensor, the time complexity can be further simplified as $\boldsymbol{\mathcal{O}}(\frac{nR}{d}\log(\frac{D}{d}))$. ■

D. Proof of Theorem 3

Proof. The occupancy updates involve ray-casting to determine the cells intersected with the rays, followed by updating their occupancy states on the map. We start by considering the worst-case scenario where all points are at the detection range R (as shown in Fig.9(b) in the primary article). In such cases, the number of cells traversed by rays can be approximated as $\mathcal{S}_{\texttt{rc}} = \boldsymbol{\mathcal{O}}(\frac{nR}{d})$, as derived in (3) and (5). The time complexity of map updates is $\boldsymbol{\mathcal{O}}(1)$ for a grid-based map and $\boldsymbol{\mathcal{O}}(\log(D/d))$ for an octree-based map. Therefore, the time complexity for occupancy updates on a grid-based map and an octree-based map is $\boldsymbol{\mathcal{O}}(\frac{nR}{d})$ and $\boldsymbol{\mathcal{O}}(\frac{nR}{d}\log(\frac{D}{d}))$, respectively. ■

E. Proof of Theorem 4

Proof. The query process in D-Map contains a query on the octree and a query on the hashing grid map whose time complexity is $\mathcal{O}(\log(D/d))$ and $\mathcal{O}(1)$, respectively. Therefore, the time complexity of the occupancy state query is easily obtained as $\mathcal{O}(\log(D/d))$. ■