Thesis for the Master of Science

Neural Implicit Surfaces for Large Scenes using Valid Region Sampling

Chaerin Min

Graduate School of Hanyang University

August 2023

Thesis for the Master of Science

Neural Implicit Surfaces for Large Scenes using Valid Region Sampling

Thesis Supervisor: Jongwoo Lim

A Thesis submitted to the graduate school of Hanyang University in partial fulfillment of the requirements for the degree of the Master of Science

Chaerin Min

August 2023

Department of Computer Science
Graduate School of Hanyang University

This thesis, written by Chaerin Min,
has been approved as a thesis for the Master of Science

August 2023

Committee Chairman: Tae Hyun Kim (Signature)

Committee member: Seungyong Lee (Signature)

Committee member: Jongwoo Lim (Signature)

Graduate School of Hanyang University

Table of Contents

ABSTRACT

Neural Implicit Surfaces for Large Scenes using Valid Region Sampling

Chaerin Min
Dept. of Computer Science
The Graduate School
Hanyang University

In this thesis, I propose the valid sample region approach on neural implicit surfaces for large scenes. Previous neural 3D reconstruction algorithms query the deep neural network based on uniformly sampled positions. Recent methods exploit the inverse CDF of the coarse network to adapt to the scene using a two-stage strategy. However, these mechanisms come at the cost of excessive reconstruction time and suffer from noisy outputs. Meanwhile, improved single-stage sampling strategies remain to be investigated. The proposed method progressively adapts the queries to the scene surfaces during optimization. In the discrete volume rendering process, this method introduces the sampling range proposal of possible surface existence. Moreover, with the observation that most areas of the inside-out scenes are empty spaces, the proposed formulation enables sample suppression for regions repeatedly diagnosed as non-surface. Also, since the algorithm adapts to the distance between the camera and the object, a memory reduction of 40% is made possible for the same reconstruction quality. Through experiments on both synthetic and real-world datasets, the proposed framework significantly outperforms the geometric results of the latest surface reconstruction approach by using the valid region sampling algorithms. The proposed method is up to four times faster, with comparison to the state-of-the-art surface reconstruction baseline. Furthermore, the proposed approach achieves substantially more reliable mesh

outcomes than the Instant-NGP, both from small-scale synthetic data and from challenging large scenes captured from the real world.

Chapter 1. Introduction

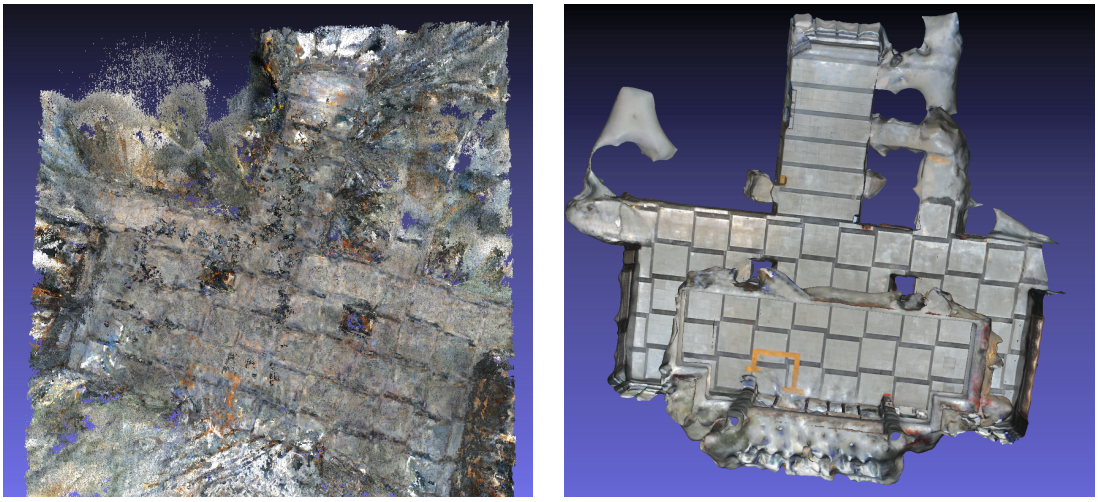
In the rapidly advancing field of artificial intelligence, computer vision is an essential component. Within this field, 3D reconstruction is a fundamental problem that is widely used in robotics, games, AR/VR, animation, and other applications. The capability to reconstruct 3D data from images is crucial for building metaverses and digital twins.

The faithful restoration of scenes of a size similar to that of the actual surrounding environment, as opposed to small objects, remains a challenge in the previous arts. Figure 1.1 displays different approaches of the scene reconstruction, extracting dense iso-surface through a traditional algorithm [1]. The state-of-the-art (a) [2] struggles to generate correct surfaces on geometry and exhibits apparent noise. While neural graphics primitives allow photo-realistic rendering at an interactive rate, a closer examination of geometry through the visualization of the level set alerts the limitations of current works in coupling its visual appeal with correct geometry in full.

Recent successes [3–5] have been reported in achieving more accurate geometry reconstruction. Several methods [6–8] improve on top of the impressive base models (a) [2] by the use of implicit surface representation, namely SDF. This includes the novel paradigm for creating distance fields [9, 10], the constraints using multi-view photo consistency [4], and the exploitation of monocular depth [6] to regularize surfaces. However, most of these methods hinge the original re-sampling strategy of [11] on their volume rendering stage, which in-

curs a linearly scaled number of model passes with respect to the size of the predefined space. To concentrate the samplings to a certain area, [9] proposes a sampling methodology that calculates the opacity error bound. It adapts the model to meaningful regions of the scene, resulting in accurate geometric reconstruction for unknown scenes. On the other hand, the exhaustive sampling method is impractical for large scenes as it requires up to five passes through the network to find the error bound, causing the model to be painfully slow. [3] offers functionality to adapt the model complexity by having separate auto-decoders for each level of the feature grid. However, this approach requires different processor for each level, limiting the practical resolution and making it unsuitable for large scenes.

The proposed method in this dissertation differs from [6, 9–14] in that I require only a single stage to adapt to the surface. With the motivation that the development of MVS techniques [15, 16] is capable of implying where the ray is likely to hit the nearer object, as the COLMAP [17] in DSNeRF [18], I encode the depth prior into the range proposal along the ray. With the range proposal boundary, I march the ray within the effective regions and integrate the influence of each network pass. The algorithm preserves the samples with high impact while diminishing the boilerplate samplings throughout the training. When the optimization progresses, I find the neural network ouputs to be a compelling guidance for caching certain regions in the space. This insight is made possible because the purpose of the scene reconstruction is to model a single spatial space. I cast the ray through a discretized and relatively coarse grid and read the value stored in the cache that the sample falls into, in order



(a) iNGP [2] (b) The proposed

Figure 1.1: Mesh results using Marching Cubes [1] on the Business dataset. Existing state-of-the-art neural radiance field [2] produces noisy geometry on large scenes when viewed with faces of zero-level set. I demonstrate that the proposed approach successfully regularize the geometry and find more accurate surface through sample proposal, surface caching, and regularizations

to decide whether to query another radiance and SDF from that voxel.

In this dissertation, I demonstrate that using depth prior on the volume accumulation can substantially improve reconstruction speed and geometry quality, at the same time, when modelling large scale environment. Additionally, I show that the spatial cache fields and the feature selection can further enable computational efficiency of 20 hours reduction and 40% decrease of memory used for fitting the network to the Business dataset. The experiment verifies the hypothesis that using the prior knowledge and the previous output allows effective surface positioning, which is different from the existing approaches [5, 6, 9, 10, 12, 19] that require iterative solution every time a ray is given.

To enable effective data acquisition in large scenes, I adopt [20] to enhance the consistency between a batch of cameras installed on a same rig. I provide an

additional contribution by utilizing [15] for dense quality reconstruction of such data. Note that the models are evaluated using the data that is processed in the same manner. The proposed results demonstrate that surface reconstruction can be successfully achieved even in challenging environments where the light source is not conditioned and the space is uncontrolled. The main contributions of this dissertation are summarized as:

- Besides the urban reconstruction [21–24] using aerial photographs, I allow the accurate, dense, and detailed level set of geometry from real-world images of large scenes.

- The proposed ray casting method handles the inside-out views, which is challenging due to the less inferable surface location in the initial state, by the bounded sampling and regularization.

- I propose a novel method that exploits the priors regarding scene surfaces given omni-directional inputs, and this results in 41% higher performance regarding depth than iNGP and 56% acceleration of the reconstructing time with similar results on the real dataset, compared to MonoSDF [6].

Chapter 2. Related Works

2.1 Neural Fields for Implicit Surfaces

Classical methods express 3D shapes with explicit meshes and point clouds. Another approaches [25–27] focus on the signed distance fields, recently getting increasing attention for being able to handle complicated shapes efficiently and continuously. [28, 29] stores SDF values to encode the Euclidean space into the spatial data structure with mathematical functions. Recent approaches such as DeepSDF [30] and Occupancy Networks [31] attempt to approximate the SDF with the differentiable neural networks, mostly with MLPs. [32,33] allows the deep neural networks to represent the accurate geometry without specific restrictions on certain topology or resolution. Seminal papers [2, 11, 12, 34] achieve photo-realistic rendering, and they leverage the ray casting and density fields. [4, 8–10] aim to boost the accurate geometric reconstruction ability of the NeRF and replace the density with traditional signed distance field representation. Moreover, [6] presents surface guidance by adopting an off-the-shelf monocular depth and monocular normal from the pretrained Omnidata [35]. A similar approach can be found in [18] which improves the geometry by making point clouds using Structure-from-Motion. The proposed method extends this line of works, and I incorporate additional surface constraints so that I can avoid flickering and unsmooth surfaces.

2.2 Neural Fields for Large Scenes

Lately, researchers [7,21–23,36,37] consider large scene reconstruction from inside-out views by taking inspiration from [11]. [22] processes aerial photography and separates geometry from color, improving the quality of large scene reconstruction. [7] and [23] propose the combination of many NeRF modules to achieve high quality rendering of city-scale scenes. In addition, [7,21,23] release Waymo Block-NeRF dataset and also utilize Google Earth Studio data for the purpose of evaluating the large scene reconstruction of different algorithms. However, research on a large number of high-resolution images obtained directly from street-level views has been less extensively studied. [36] focuses on street renderings using LiDAR data, and it exhibits the visualization of renderings. [38] extends the possibility of reconstructing real-world scenes by introducing unbounded coordinate embedding. I differ from the previous techniques in that I improve the large-scene neural representation into quality dense iso-surface given the challenging street views.

2.3 Neural Fields with Adaptive Sampling

The neural scene reconstruction methods are limited in terms of long training times [39, 40]. It takes 12 hours for typical NeRF variants [6, 38] to overfit to an object given 479 512×512 images [11]. Since the process spends most of the time on neural auto-decoder, i.e., MLP, the training time generally grows proportional to the sampling rate. To mitigate this issue, iNGP [2] propose to store only the local information on its pre-located spatial nodes. Thus, it allows the model to reduce the MLP size and accelerates the rendering. Hash encoding

techniques [2, 3, 7], nonetheless, are still limited in memory efficiency, for the same size of feature grid encoding is required regardless of the scene complexity and visibility. The closest state-of-the-art solutions to the proposed method are [9] and [13,14] in the sense that they attempt to achieve more concentrated and thus more accurate sampling by reducing the search boundary. The iterative nature of [9, 10] and the involvement of additional networks [13, 14, 38] cause the models to introduce high computational costs. In constrast, the proposed method aims to filter the queries within the ray-surface intersections. To make the model suitable for reconstructing large scenes, I employ geometric prior and the model output to do the filtering while modulating the model complexity based on the scene visibility information.

Chapter 3. Methodology

In this thesis, I propose to bound the effective search range of both surfaces and surface grid features given a large-scale scene. In Section 3.1, I provide the background of neural implicit reconstruction. I then describe in Section 3.2 the data collection setting of an ego car view and discuss remaining challenges of neural surface fields. In Section 3.2.1, I introduce range boundary for efficient ray casting via depth guidance. The Section 3.2.2 presents sample culling by additionally using density cache grid. Furthermore, in Section 3.3, I suggest the selection for the grid encoding features, based on the distance to object boundaries. Lastly, Section 3.4 provides regularization to the implicit neural surfaces, followed by the proposed optimization functions described in Section 3.5.

3.1 Neural Implicit Surfaces

I represent the 3D scene as a SDF [41]. SDFs are signed distance functions $f : \mathbb{R}^3 \to \mathbb{R}$ to the closest surface $\mathcal{S} = \partial\mathcal{M}$ from a point $\mathbf{x} \in \mathbb{R}^3$, where volume $\mathcal{M} \subset \mathbb{R}^3$. The surface is represented by a level set of the $f(\mathbf{x})$ as follows:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}) = 0\}. \tag{1}$$

A neural SDF f_Θ approximates the SDF as a differentiable deep neural network [30]. In this dissertation, I formulate a scene with a SDF since SDF takes advantages as a compact form of formulating a 3D shape, of which the computational cost remains the same regardless of the scene size, given the

same size of the neural network.

I model the scene by encoding a position $\mathbf{x}$ into a feature vector. Then, I process this feature vector with a light-weighted Multi-Layer Perceptron (MLP) [42] to output the prediction of the SDF s and the radiance field c. I leverage the hash encoding [2] to exploit the geometric primitives and embed them in the space as

$$h(\mathbf{x}) = \left(\bigoplus_{i=1}^{3} \mathbf{x}_i \pi_i \right) \mod T \tag{2}$$

, where π_i is a large prime number and $\bigoplus$ represents bit-wise XOR operation. This hash encoding $h : \mathbb{R}^3 \to \mathbb{R}^F$ maps a position $\mathbf{x}$ to a feature of dimension F when the hash table size of a feature grid is restricted to $T \in \mathbb{N}$.

3.1.1 Volume Rendering of SDFs

In order to aggregate the occupancy along a ray so that I can determine rendered color C at a pixel, I transform the output sdf $s = f_\Theta(\mathbf{x})$ into corresponding density. For this purpose, I define the S-density function [10] using the logistic density distribution $\phi_\alpha(\mathbf{x}) = \alpha e^{-\alpha \mathbf{x}} / \left(1 + e^{-\alpha \mathbf{x}}\right)^2$. Note that α is a learnable parameter that allows the neural network to adapt the standard deviation of $\phi_\alpha(x)$ to each scene by adjusting $1/\alpha$ [10]. Given ray from the camera origin $\mathbf{o}$ and the direction $\mathbf{v}$, I denote the set of points along the ray as $\{\mathbf{p}(t) = \mathbf{o} + t\mathbf{v} | t \in \mathbb{R}^+\}$, where $\|\mathbf{v}\| = 1$. The density function of the coordinate $\mathbf{x}$, denoted σ, indicates the probability a infinitely small and thus volume-less point on a ray starting from $\mathbf{o}$, for each $t \geq 0$, falls inside an object,

$$\sigma(t) = \max\left(\frac{-\frac{d\Phi_\alpha}{dt}(f(\mathbf{p}(t)))}{\Phi_\alpha(f(\mathbf{p}(t))}, 0\right), \tag{3}$$

, where $\sigma(t)$ shall be proved [10] to attain a local maximum at a surface intersecting point at $\mathbf{p}(t^*)$, i.e. $f(\mathbf{p}(t^*)) = 0$ and to obtain larger contribution to the rendered value for the nearer points when $f(t)$ are competing on a ray.

Then, the transmittance function in the segment $[\mathbf{o}, \mathbf{p}(t)]$ is given by

$$T(t) = \exp\left(-\int_0^t \sigma(\mathbf{p}(u))du\right). \tag{4}$$

Finally, I theoretically integrate the color and density along the ray to obtain the rendered RGB value $\mathbf{C}(\mathbf{o}, \mathbf{v})$ as follows:

$$\mathbf{C}(\mathbf{o}, \mathbf{v}) = \int_0^{+\infty} T(t)\sigma(t)c(\mathbf{p}(t), \mathbf{v})dt. \tag{5}$$

3.2 Valid Region Sampling

Neural implicit fields mainly assume object-centric environment, densely reconstructing the geometry. However, this data acquisition system hardly apply to images captured by autonomous vehicles. In order to be consistent with the driving system, I propose a camera placement in a panoramic configuration, inspired by [43]. The ego vehicle obtains data surrounding itself in an inside-out manner, including feature-less walls and moving persons in many parts of the scene. This involves the real-world noises from camera effects such as vignetting, white balance, and auto-focusing, which I handle in Section 3.4. With the proposed close approximation to the robot driving scenario, the j-th ray of viewing direction $R_i\Pi_i^{-1}(\mathbf{u}_j)$, for the image pixel coordinate $\mathbf{u}_j$, given the projection function Π_i of the i-th camera for the given camera model,

are beamed outward from different views $P_i = [R_i, T_i], i = 1, ..., n$ without all of them having to point toward a certain concentrated target area. As a result, with the existing NeRF methods, optimizing a ego novel view synthesis becomes more challenging.

3.2.1 Sample Proposal

In this section, I aim to reduce the search space of the ray samples $\mathcal{T}$ given the inside-out multi-camera capture system. This is essential since the search range of $\mathcal{S}$ increases cubically to the scene resolution n, i.e. $O(n^3)$. The high cost with regards to the scene size become a substantial problem when it comes to driving scenarios because the discretized version of volume rendering scales the procedure time linearly to the sampling rate, as shown in Eq. 6. The numerical quadrature in Eq. 6 approximates the integral counterpart in Eq. 5 at the samples $\mathcal{T} = \{t_i\}_{i=1}^{m}, 0 = t_1 < t_2 < ... < t_m$, with the sample interval length $\triangle\, t$,

$$C(\mathbf{o}, \mathbf{v}) \approx \hat{C}_{\mathcal{T}}(\mathbf{o}, \mathbf{v}) = \sum_{i=1}^{m-1} T(t_i)\alpha(t_i)c_i(\mathbf{p}(t_i), \mathbf{v}) \tag{6}$$

, where $\alpha(t_i) = 1 - e^{-\sigma(t_i)\triangle t}$. $T(t)\sigma(t)$ is the approximated Probability Density Function [9] (PDF) of the contribution to the rendered output. Its quality depends on the approximation on the discrete samples $\mathcal{T}$ and their intervals. One solution to adaptively sample near the t where value the PDF is high is to invert the CDF, i.e. $\left(\int \sigma(t)T(t)dt\right)^{-1}$. Practically, I easily derive $O(t) = \int \sigma(t)T(t)dt = 1 - T(t)$. However, sampling with O^{-1} relies on the quality of the model density. Also, this sampling strategy consumes sample

queries on non-surface areas at least at the course sampling stage. [10] reported that they found four re-sampling stages in total were necessary to reach their geometric precision requirement. Another solution is to sample points using approximation error bound of opacity O [9]. However, this method requires additional queries to estimate the error bound itself, and I tested the impact of the number of queries on training time in Section 4.

I leverage the information that the camera installed on a robot rig can provide with. This is unique in that I assume the cameras are equipped on a driving robot, and thus have to be at a fixed configuration. The estimated depth [15] using the matching cost volume is used to bound the range of the approximation samples of ray casting. Given the equirectangular depth map, I map it into a fisheye image plane to let the depth field be suitable for the large FOV input I of this thesis, using inverse spherical sweeping [44].

Given the MVS depth field D^*, for each ray that is consistent with the image ray $\mathbf{p}$, the sample set $\mathcal{T}$ is computed based on the prior that the nearest obejct boundary to $\mathbf{o}$ is likely to be located near d^* for each pixel j:

$$\mathcal{T}_{dg} = \left\{ t | t \sim \mathcal{N}(d_j^*, k d_j^*) \right\}, \tag{7}$$

where k is a parameter to tune the variable variation of the Gaussian sample distribution $\mathcal{N}$. In order to consider that the large depth induces small disparity on the image plane I, I impose larger variance on t with larger depth. In scheme of Eq. 7, I assume the same number of samples on each ray.

However, Eq. 7 may overlook the empty space, which causes vulnerability in the early course of training. To alleviate this issue, I support the $\mathcal{T}_{dg}$ with

$\mathcal{T}_u$,

$$\mathcal{T}_u = \{t_i\}_{i=1}^{m_j}, t_i = (i-1)\frac{M_j}{n-1}, i \in [n]. \tag{8}$$

I subscript the pixel index j on M to highlight that I stop generating $\mathcal{T}_u$ at the Normalized Device Coordinates boundary , namely "NDC" [45]: $\mathcal{B} = [-1, 1]^3$, $\mathbf{x} \in \mathcal{B}$. By defining $\mathcal{B}$, I ensure the query points are within the grid coordinate allocated for the modelled scene, and I use the AABB-Intersection algorithm [46] between the ray and NDC to precisely figure out where to stop $\mathcal{T}_u$. Finally, the sample set $\mathcal{T}$ defines

$$\mathcal{T}_k(\mathbf{o}_i, \mathbf{v}_j, d_j^*) = \text{sort}(\mathcal{U}[\mathcal{T}_{dg}, \mathcal{T}_u]). \tag{9}$$

The proposed $\mathcal{T}$ works with a single drawing of $\{t_i\}$ on each $\mathbf{p}$. Eq. 5 can also be extended to the PDF of depth regarding $\{t_i\}$, where the proposed heuristic-driven sampling strategy produces a byproduct that works as a supervision.

$$D_{\mathcal{T}}(\mathbf{p}) = \sum_{i=1} \tau(t_i) t_i \quad \mathcal{L}_{depth} = \sum_{\mathbf{p} \in \mathcal{P}} \|\hat{D}_{\mathcal{T}}(\mathbf{p}) - \hat{D}^*(\mathbf{p})\|^2 \tag{10}$$

, where the derivative of O is $\tau(t_i) \triangleq \alpha(t_i)T(t_i)$. Inverse depth $\hat{D}(\cdot) = 1/D(\cdot)$ is used, which aligns with the disparity space.

3.2.2 Sample Culling

I describe the ray marching with sample culling procedure in Algorithm 1. The bitfield $\mathcal{V}$ is reused at the test phase to accelerate the rendering. When inference stage, additional stopping criteria is proposed. That is, the ray march-

ing runs until the sum of the densities reaches 1.

3.3 Valid Feature Selection

Data captured from robot driving mostly covers a large scene, so the optimization takes extra time to converge. In this sense, I adopt the multi-resolution grid of geometric primitives [2,3] which allows independent encoding across different part of the 3D scenes. However, I found that the driving agent, by nature, moves through a route without visiting every part of the captured space equally. For instance, even if a robot took an image of a house alongside the street, it may only run through the line of the road. Still, the multi-resolution grid treats the near-distant trees and the house in the long distance with equal level of details. The motivation is to attenuate the grid feature by Nyquist [47] theorem. Objects away from the camera center are projected to fewer pixels in the image plane, where the computation of the objective function occurs. Therefore, the distant objects should be supersampled to meet the Nyquist frequency. This strategy, however, is not best suited for large scenes, and comes at the cost of already slow rendering process to describe delicate details of a window on a 10km-away building. I are inspired by the set of different discrete scales of the feature grid $\mathcal{Z}$. The feature vector of $L^{\max} \cdot F$ dimension represents all levels of downsampling scales. I select the level for each t, based on the footprint of the cone frustum intersecting the scene by the ray. This approach is known as *prefiltering* in digital signal processing and allows reducing the memory footprint of the hash table, corresponding to the feature grid. With the discrete level $l = 1, ..., L^{\max} \in \mathbb{N}$, the feature vector of each scale l is

computed and interpolated to be $\psi(\mathbf{x}; l, \mathcal{Z})$, $\psi : \mathbb{R}^3 \rightarrow \mathbb{R}^F$. Then, the input feature vector to the MLP is aggregated: $\mathbf{z}(\mathbf{x}; \mathcal{Z}) = \bigoplus_{l=1}^{L^{\max}} \psi(\mathbf{x}; l, \mathcal{Z}_\Theta)$, where $\bigoplus$ concatenates the features. To approximate the footprint of the cone frustum, the radius of the cone on the normalized image plane $\dot{r} = \mathbf{x}^{j+1} - \mathbf{x}^j$ of i-th view is multiplied by the sample t_k. Pixel $\mathbf{u}_i^j$ is back-projected into $\mathbf{x}^j = g(M_i^{-1}\mathbf{u}_i^j)$, where $g(\cdot)$ and M is the fisheye undistortion combined with normalization and the camera parameter matrix, respectively. The appropriate scale is characterized by t_i and the resolutions $N_{\min}$ and $N_{\max}$ of the multi-resolution grid $\mathcal{Z}$ is derived as follows:

$$\tilde{L} = \underset{l \in \mathcal{N}}{\operatorname{argmin}} \| \frac{2}{\lfloor N_{min} \cdot b^{l-1} \rfloor} - t_i \dot{r} \|, \tag{11}$$

$$b \triangleq \exp(\frac{lnN_{max} - lnN_{min}}{L^{\max} - 1}) \tag{12}$$

Then, I modulate $\psi_{1:L^{\max}}$ with the bell-shaped function:

$$\omega_L = k \cdot e^{-\frac{(\tilde{L}-L)^2}{2c^2}} \tag{13}$$

,where I take inspiration from Guided Stereo Matching [48]. The memory reduction is shifted from the post-training time to a precomputation phase; the maximum appropriate scale need only be computed once the train trajectory of the robot is given - the closest distance from the views and the position is all things to be considered. The SDF network f_Θ takes as input the modulated

feature vector $\tilde{\psi} = \tilde{\psi}_{1:L^{\max}}, \tilde{\psi}_L = \psi_L \circ \omega_L$:

$$f, \hat{\mathbf{z}} = f_\Theta(\gamma^{PE}(\mathbf{x}), \tilde{\mathbf{z}}(\mathbf{x}; \mathcal{Z}_\Theta), \mathbf{x}) \tag{14}$$

,where $\tilde{\mathbf{z}}(\mathbf{x}; \mathcal{Z}_\Theta) = \bigoplus_{l=1}^{L^{\max}} \tilde{\psi}(\mathbf{x}; l, \mathcal{Z}_\Theta)$ Additionally, I are inspired by Rahaman *et al.* [49], and use Positional Encoding from the Transformer [50] architecture, $\gamma^{\mathrm{PE}}(\mathbf{x}) = [\sin(\mathbf{x}), \cos(\mathbf{x}), ..., \sin(2^{J-1}\mathbf{x}), \cos(2^{J-1}\mathbf{x})]^\intercal$, to address higher frequency. The concatenation of the SDF features $\hat{\mathbf{z}}$, Spheircal Harmonics [44] encoding γ^{SH} for better integratation of view-dependent reflected radiance, normal $\mathbf{n}$, and position $\mathbf{x}$ is fed into the neural network c_Θ to synthesis the radiance field:

$$c = c_\theta(\mathbf{x}, \gamma^{SH}(\mathbf{v}), \hat{\mathbf{z}}, \mathbf{n}(f_\Theta(\mathbf{x}))) \tag{15}$$

, where I define $\mathbf{n}(\mathbf{x})$ in the subsequent section. The efficiency of the grid memory can be computed ahead of time, which I show the detailed explanation in Section 4.

3.4 Surface Regularization

Classical NeRF constraints [11, 12, 34] produce photo-realistic rendering, and the latest Neural SDFs [6, 9, 10] successfully make the $\tau(t_i)$ concentrated near $\mathcal{S}$ on a ray. However, when it comes to the structure between adjacent rays, conventional methods tend to have reported flickers and unsmooth surfaces. To address this issue, I explicitly regularize the model by deriving surface normal vectors, $\mathbf{n}$, during the training, from the predicted distance field f_Θ. By using the fact that the steepest direction vector of a SDF at a $\mathbf{x}$ is perpendicular to

the zero-level boundary of the SDF, I derive $\mathbf{n}(\mathbf{x})$ from

$$\mathbf{n}(\mathbf{x}) = \nabla_{\mathbf{x}} f_{\Theta}(\mathbf{x}). \tag{16}$$

Then, I provide two different geometric constraints using Eq. 16. First, I approximate the normal vector on the 2D image plane by accumulating the computed $\mathbf{n}(\mathbf{x})$ multiplied by the interval length and the probability density function [6]. In particular, I adopt the rectangle rule as follows,

$$N(\mathbf{p}) = \int_0^{\infty} \mathbf{n}(\mathbf{x}(t))\tau(t)dt \approx N_{\mathcal{T}}(\mathbf{o}, \mathbf{v}) = \sum_{i=1} \tau(t_i)\mathbf{n}(\mathbf{x}(t)). \tag{17}$$

The residual between the N and the pseudo GT normal can be used,

$$\mathcal{L}_{normal} = \sum_{\mathbf{p}\in\mathcal{R}} \|\hat{N}^*(\mathbf{p}) - \hat{N}(\mathbf{p})\|_1 + \|1 - \hat{N}^*(\mathbf{p})^{\intercal}\hat{N}(\mathbf{p})\|_1. \tag{18}$$

, where I normalize $N(\cdot)$, i.e. $\|\hat{N}\| = 1$. I calculate the pseudo GT normal from the given OmniMVS [15] depth field D^*. For pixel coordinates $\mathbf{u} \in \mathbb{N}^2$,

$$\hat{\mathbf{n}}^*(\mathbf{x}_j) = \frac{(\mathbf{x}_j - \mathbf{x}_k) \times (\mathbf{x}_j - \mathbf{x}_l)}{\|(\mathbf{x}_j - \mathbf{x}_k) \times (\mathbf{x}_j - \mathbf{x}_l)\|}, \tag{19}$$

where $\|\mathbf{u}_j - \mathbf{u}_k\| = 1, \|\mathbf{u}_j - \mathbf{u}_l\| = 1$ and $\mathbf{x}_j = (g((\mathbf{u}_j - \mathbf{c}_i)/f_i) * d_j)\mathbf{v}_i$. f_i is the focal length, c_i is the optical center, and $g(\cdot)$ indicates the mapping function between $\mathbf{r}_j = \|\mathbf{u}_j - \mathbf{c}_j\|_2$ and θ of a fisheye camera.

Second, note that, in Eq. 16, $\mathbf{n}$ is not guaranteed to be a unit vector. To regularize f_{Θ} to be geometrically SDF-like, I detour by forcing the normal vector, derived from f_{Θ}, to be of size 1. For this purpose, I adopt the Eikonal

function [51] as follows,

$$\mathcal{L}_{eik} = \sum_{\mathbf{x} \sim \mathcal{T} \cup \mathcal{X}} (\|\nabla_{\mathbf{x}} f_{\Theta}(\mathbf{x})\|_2 - 1)^2 \tag{20}$$

where I denote the set of uniform samples within the proposed $\mathcal{B}$ as $\mathcal{X}$. I additionally introduce the total variance term of $\mathbf{n}$:

$$\mathcal{L}_{nreg} = \sum_{\mathbf{x} \sim \mathcal{T} \cup \mathcal{X}} \|\mathbf{n}(\mathbf{x}) - \mathbf{n}(\tilde{\mathbf{x}})\|, \quad \tilde{\mathbf{x}} \sim N_{\mathbf{x}} \tag{21}$$

, and $\|\cdot\|_2$ denotes L-2 norm. The neighboring point $\tilde{\mathbf{x}}$ is sampled from the neighbor of $\mathbf{x}$, $N = \{\mathbf{x}_j | \|\mathbf{x}_j - \mathbf{x}_i\| < \rho, \mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^3\}$, of radius ρ. In addition, for the static assumption, I semantically segment moving objects by using the stable SOTA, Deeplab V3 [52], and remove the segmented pixels from the residual function defined in Eq. 23.

3.5 Optimization

The proposed objective function follows the original photometric loss proposed by NeRF [11]:

$$\mathcal{L}_{photo} = \sum_{\mathbf{p} \in \mathcal{P}} \|I^*_{\mathbf{p}} - I_{\mathcal{T}}(\mathbf{o}_{\mathbf{p}}, \mathbf{v}_{\mathbf{p}})\|_1 \tag{22}$$

, where $I \in \mathbb{R}^3$. On top of the color constraint, I combine the Eq. 10, 18, 20, and 21. Finally, the proposed overall objective function is as follows:

$$\mathcal{L} = \mathcal{L}_{photo} + \lambda_1 \mathcal{L}_{depth} + \lambda_2 \mathcal{L}_{normal} + \lambda_3 \mathcal{L}_{eik} + \lambda_4 \mathcal{L}_{nreg}. \tag{23}$$

Algorithm 1 Ray marching with sample culling at training

Input:

$\mathcal{P}$: set of rays

$\mathcal{B}$: NDC

i: frame index, j: pixel index

Require:

t_n: nearest t to sample

ϵ: exponential step factor

V: sample culling of a single resolution, $\mathcal{V}$: bitfield

Ω_j: set of densities on $\mathbf{p}_j$

Θ_0: warm-up trained SDF network parameter

σ_e: surface criteria, κ: bitfield tolerance, β: update step

Output:

updated bitfield $\mathcal{V}$

trained network parameter Θ

while *until convergence* **do**
  Sample a random ray $(\mathbf{o}_i, \mathbf{v}_j) \sim \mathcal{P}$
  $\mathbf{x}_0 \leftarrow \mathbf{o}_i + t_n \mathbf{v}_j$
  while $\mathbf{x}_k \in \mathcal{B}$ **do**
    if $\mathcal{V}^* = 1$ *and* $\mathbf{x}_k$ HITS $\mathcal{V}^*$ **then**
      $\sigma_k \leftarrow f_\Theta(\mathbf{x}_k; \mathcal{Z})$
      if $\sigma_k < \sigma_e$ **then**
        if $V^* > \kappa$ **then**
          $\mathcal{V}^* \leftarrow 0$
        else
          $V^* \leftarrow V^* + 1$
      else
        $V^* \leftarrow 0$
      $\Omega_j \leftarrow \sigma_k$
      $\mathbf{x}_k \leftarrow \mathbf{x}_k + (1+\epsilon) t \mathbf{v}_j$
    else
      $V^*, \mathbf{x}_{k+1} \leftarrow$ NEXTVOXEL$(\mathbf{p}_j)$
      $\mathbf{x}_k \leftarrow \mathbf{x}_{k+1}$
  $\hat{C}_{\Omega_j}, \hat{D}_{\Omega_j} \leftarrow$VOLUMERENDER$(\Omega_j, \Theta)$
  $\Theta \leftarrow \Theta - \beta \nabla_\Theta \mathcal{L}_{\text{total}}(\hat{C}, \hat{D})$

Return: $\mathcal{V}, \Theta$

Chapter 4. Experiments

4.1 Datasets

I trained the proposed model on Garage dataset, Aria dataset, and Business dataset. Garage is a synthetic dataset with BRDF. Using Blender, I created four 220 FOV cameras on a rig and 20 views for each camera. For Aria dataset, I operated 4 cameras on a helmet and captured data from an office named Aria. OmniSLAM [53] was used to estimate the poses, and OmniMVS predicted the depth. Aria dataset contains 1200 frames, but I selected 130 frames which covers a seminar room. Lastly, I operated the same system as Aria, inside Business building at Hanyang University, for which I named this dataset Business. The trajectory of Aria and Business includes loops for higher camera pose precision. Both of the real datasets contain challenging factors such as walking people, texture-less walls, and floors.

4.2 Implementation Details

I use PyTorch [54] and CUDA for implementing the proposed method. The proposed model fits to input images of size 1344×1080. Hyper-parameters pertaining to the multi-resolution hash grid follows [2, 6]. The NDC is created based on the maximum offset with regards to the average of the camera poses throughout all spatial dimensions. For synthetic datasets, I choose 0.1, 0.1, 0.1, 0.05 and for $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, respectively, while they are 0.05, 0.05, 0.1, 0.05 and for real-world datasets. For real datasets, I also apply decay on the λ weights as the training progresses.

4.3 Evaluations

4.3.1 Quantitative Comparisons

I compare the quantitative results against the State-of-the-art neural reconstruction models [2, 6, 12]. Among the variants that [6] proposed, I evaluate on the multi-resolution grid version of it since that is reported to show the best scores with regards to geometry quality. In Table 4.1, I adopt PSNR for rendering evalution, RMSE and MAE for depth maps, and angle error in degree for normal vectors in 2D. PSNR, Depth RMSE, and Normal angle error are computed on 50 8Ki batches and are averaged. Depth MAE is measured in the unit of images. I assume the normal vectors are directional vectors, i.e. unit vectors, and normalize them into size of 1. The normal vectors with sizes far from one largely appear in [2, 12] which lack surface regularizations. In this sense, I note the effectiveness of surface constraints. In Table 4.1, the proposedmethod significantly outperforms on real datasets while the result of this thesis degrades on a synthetic dataset. The proposed model particularly performs better at normal estimation since it introduces the sample proposal strategy and surface regularization.

In Table 4.2, I measure the time of reconstructing the Garage scene and demonstrate the time for an epoch. The proposed models accelerate by more than two times faster than MonoSDF [6] and three times than Mip-NeRF. iNGP records the fastest reconstruction time, but it fails to precisely reconstruct the geometry as shown in the qualitative result.

4.3.2 Qualitative Comparisons

I illustrate rendered images, depth maps, and meshes of Garage dataset in Figure 4.1. The images and depth maps are shown at 360 degree in horizontal and 180 degree in vertical to efficiently demonstrate the large scene reconstruction. Meshes are rendered with vertex color. The result from iNGP [2] produces messy surfaces with triangles flickering, since it fails to constrain samples around surfaces and does not provide regularized density fields due to the sole dependency on hash encoding. Mip-NeRF [12] successfully prevents aliasing using the Mip-map from graphics, and it adopts hierarchical sampling strategy from [11] without grid hash encoding. However, [12] still shows limited performance in surface representations and vertex colors due to the weight distribution in high frequency from naive samplings. Although [6] spends twelve times to improve upon [2], the object boundaries becomes comparably sharper. When using the proposed method, the vertex color has extra improvement while reducing the additional modelling time. Note that I encode the position with the hash encoding as [2], but the result of this paper is with more fidelity in terms of clean surfaces.

I conduct the experiments on real-world datasets in Figure 4.2 and 4.3. Ours successfully handles challenging parts, i.e. texture-less window in the left and walls, and improves upon [6] at areas with less ray intersection. The proposed model works significantly better for rendering and depth, particularly in real scenes. The floaters are reduced and the depth noise along with the color floaters is removed. The first row of the Figure 4.2 demonstrates depth and normal maps used for geometric supervision as [6]. The proposed proposed

method presents even higher quality geometry than the output of the guiding network. In Figure 4.3, I evaluate the baseline model and the proposed model on challenging condition of highly reflective and large scene. Though [2] minimizes the loss function more rapidly in the initial phase, it shows limited performance at the end, even if it is trained for as much time as the proposed method is trained. In Figure 1.1, the mesh generated by Marching Cubes [1] from the neural network is able to successfully represent the flat walls and the details on the honours board, located in the left side.

4.3.3 Ablation Study

I present Table 4.3 to highlight the effectiveness of individual components of the proposed model. I demonstrate the result on the Garage dataset. One of the core modules is the sample proposal, abbreviated as sample prop. Between the second and third row, the sample proposal favorably contributes to both the color and the geometry. The second row which is without sample proposal takes more than two hours for an epoch. On the other hand, the experiment using Sample Prop. reduces the training time into half. The model with the sample culling performs comparably to that without the sample culling, yet the sample culling accelerates the training by a factor of 4. This is shown between the second row and the fourth row. As demonstrated in row 1 and 4, the surface regularization is slower than the baseline. However, the Table 4.3 presents considerable improvement in geometric reconstruction quality of 19 degree, for the surface regularization method.

In addition, in the Figure 4.4, I show the qualitative comparison between

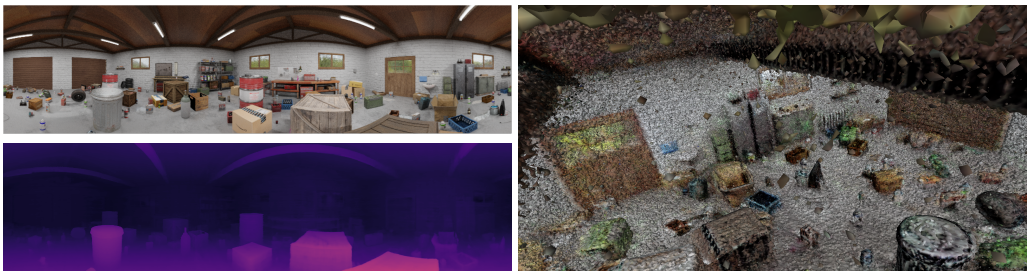
the base and the feature selection. The purpose of the feature selection is to reduce the memory footprint of the grid hash encoding. As the grid largely replaces the burden of the global MLP processor whose consumption is as little as 1.1 MB for this case, the memory issue occurs. For the $[2, 6]$, 1.3GB is consumed only for the grid hash buckets, and the same applies to the base model as I adopt the grid hyperparameters of [2]. By using the proposed feature selection, it is possible to reduce the hash memory consumption into 740MB. Note that I take the different sizes of the actual hash bucket of each 16 level into calculation, to be more precise. The quantitative result is 28.68dB, 10.1cm, and 6.97° for PSNR, Depth RMSE, and Normal angle error, respectively. As shown in Table 4.1, the numbers are on par with those of Ours. Also, by using the feature selection, I aim to meet the Nyquist [47] frequency at the less visible areas. Figure 4.4 effectively demonstrates this efficacy. The wall in a long distance, the upper right side, shows red bias at the base model, but the color of the same wall when I use the feature selection displays the correct color. This is because the selection of the grid features according to the visibility serves as the low-pass filter when the signal is projected on a relatively small number of pixels and thus causes aliasing.

		PSNR [dB]	Depth RMSE [m]	Depth MAE [m]	Normal Error [deg]
Garage	iNGP [2]	27.84	0.176	0.0864	27.5
	Mip-NeRF [12]	28.43	0.123	0.0492	32.2
	MonoSDF [6]	26.90	0.148	0.09363	**6.73**
	The proposed	**28.85**	**0.109**	**0.0489**	7.14
		PSNR [dB]	Depth RMSE [m]	Depth MAE [m]	Normal Error [deg]
Aria	iNGP [2]	27.85	0.304	0.226	31.9
	Mip-NeRF [12]	**30.61**	0.252	0.193	51.8
	MonoSDF [6]	28.34	0.196	0.0976	18.4
	The proposed	29.22	**0.150**	**0.0836**	**16.0**
		PSNR [dB]	Depth RMSE [m]	Depth MAE [m]	Normal Error [deg]
Business	iNGP [2]	21.56	0.847	0.549	46.3
	MonoSDF [6]	23.74	0.697	**0.309**	13.4
	The proposed	**23.95**	**0.626**	0.318	**12.0**

Table 4.1: Qualitative Evaluation on Garage, Tetra, and Business datasets

	Reconstruction Time [min]
iNGP [2]	5
Mip-NeRF [12]	98
MonoSDF [6]	61
The proposed	27

Table 4.2: Epoch train time on the Garage dataset.



(a) iNGP [2]

(b) Mip-NeRF [12]

(c) MonoSDF [6]

(d) The proposed

Figure 4.1: Qualitative comparisons on Garage dataset. Renderings, depth maps, and meshes are shown.

Sample Prop.	Sample Culling	Surface Reg.	PSNR [dB]	Depth Error [m]	Normal Error [deg]	Recon. Time [min]
✗	✓	✗	27.84	0.0864	27.5	**5**
✗	✗	✓	26.90	0.0936	6.73	61
✓	✗	✓	26.92	**0.0860**	**6.51**	27
✗	✓	✓	**28.24**	0.0934	7.76	14

Table 4.3: Ablation of Sample Proposal, Sample Culling, and Surface Regularization on Garage dataset. The Recon. Time is measured for an epoch of training on the Garage data, consisting of 80 frames.

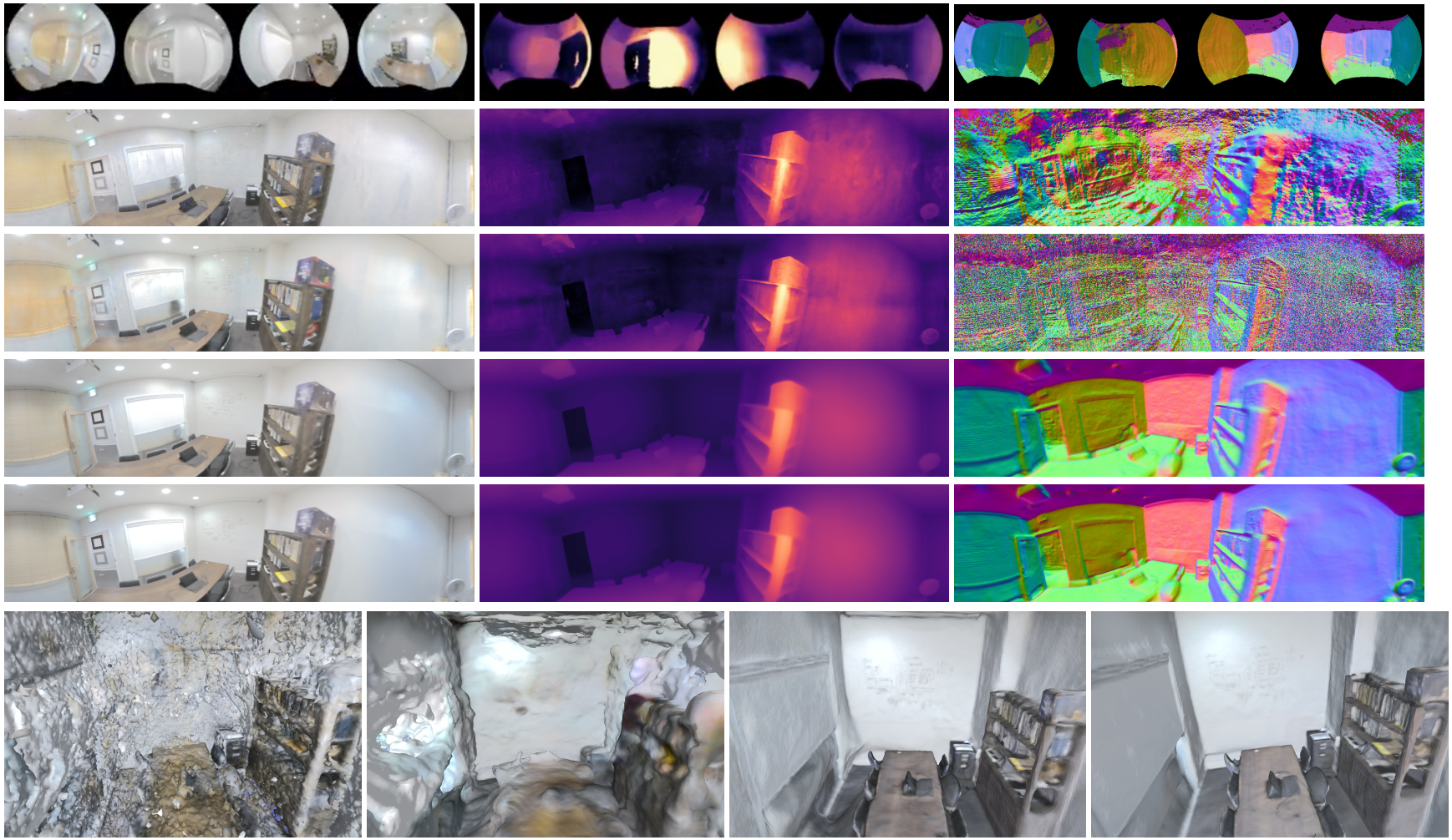


Figure 4.2: Qualitative comparisons on Aria dataset. Rendered images, depth maps, and normal maps are presented from left to right. From top to bottom, the results of the input data, iNGP [2], Mip-NeRF [12], MonoSDF [6], and the proposed method are illustrated. The last row illustrates the mesh results.

Figure 4.3: Qualitative comparisons on Business dataset from the same reconstruction time. From top to bottom, the results of the input data, iNGP [2], MonoSDF [6], and the proposed method are shown.

Figure 4.4: Qualitative comparisons on the Garage dataset. Above I show the base, below the feature selection is applied. While feature selection saves 40% of the grid encoding memory, the quality of the result remains the same.

Chapter 5. Conclusion

In this thesis, I present the neural radiance field for large-scale scene reconstruction. To this end, I employ the novel sample proposal and sample culling to constrain the queries to be around the surface. The state-of-the-art methods of neural implicit surfaces rely their approximated weight integration on the re-sampling strategy. However, the hierarchical approach necessarily includes queries from the nearest through the farthest, regardless of the optimization status. This induces weight accumulation involving unnecessary samplings, which leads to noise sensitivity and exacerbates the already slow training time for modelling a single scene. To alleviate this problem, I propose to exploit extra information from the widely-used depth supervision strategy. The proposed sample proposal approach uses geometric prior to bound the effective sampling boundary. Additionally, by confirming the regions that are not likely to include surfaces as the reconstruction progresses, the proposed model allows the volume caching which is consistent through multiple views. Still, the neural fields produce noisy SDF fields when naively applied, so I integrate the total variance term on the normal prediction derived from the network SDF. Note that I incorporate the spherical sweeping configuration to collectively process the data from omni-directions for large-scale scenes and evaluate the method with this same dataset on other approaches. Experiments show that the sample proposal and the sample culling accelerate the training from the state-of-the art methods. Still, the proposed method performs better at reconstruction quality. For instance, in the Aria dataset which is captured from a real

office, the depth estimation root mean square error drops 53% from the fastest baseline and 26% from the model with the closest approach as the proposed method. Since the proposed approach leverages extra prior from off-the-shelf depth and the model output, the reduced search space of samplings empirically proves to prevent messy surfaces in areas with high uncertainty. The 3D geometry represented both in image planes and meshes demonstrates that the proposed method provides additional improvement upon the baselines. To be specific, the proposed approach compares favorably to relevant state-of-the-art baselines at frequently occluded regions in the Aria seminar room and limited color representation in Garage mesh.

References

[1] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[2] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[3] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler, "Neural geometric level of detail: Real-time rendering with implicit 3d shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11358–11367, 2021.

[4] M. Oechsle, S. Peng, and A. Geiger, "Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5589–5599, 2021.

[5] L. Yariv, P. Hedman, C. Reiser, D. Verbin, P. P. Srinivasan, R. Szeliski, J. T. Barron, and B. Mildenhall, "Bakedsdf: Meshing neural sdfs for real-time view synthesis," *arXiv preprint arXiv:2302.14859*, 2023.

[6] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction," *arXiv preprint arXiv:2206.00665*, 2022.

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

[8] Q. Fu, Q. Xu, Y. S. Ong, and W. Tao, "Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3403–3416, 2022.

[9] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4805–4815, 2021.

[10] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *arXiv preprint arXiv:2106.10689*, 2021.

[11] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[12] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.

[13] A. Kurz, T. Neff, Z. Lv, M. Zollhöfer, and M. Steinberger, "Adanerf: Adaptive sampling for real-time rendering of neural radiance fields," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pp. 254–270, Springer, 2022.

[14] B. Attal, J.-B. Huang, C. Richardt, M. Zollhoefer, J. Kopf, M. O'Toole, and C. Kim, "Hyperreel: High-fidelity 6-dof video with ray-conditioned sampling," *arXiv preprint arXiv:2301.02238*, 2023.

[15] C. Won, J. Ryu, and J. Lim, "Omnimvs: End-to-end learning for omnidirectional stereo matching," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8987–8996, 2019.

[16] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 767–783, 2018.

[17] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

[18] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12882–12891, 2022.

[19] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7210–7219, 2021.

[20] C. Won, J. Ryu, and J. Lim, "Sweepnet: Wide-baseline omnidirectional depth estimation," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6073–6079, IEEE, 2019.

[21] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, "Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pp. 106–122, Springer, 2022.

[22] L. Xu, Y. Xiangli, S. Peng, X. Pan, N. Zhao, C. Theobalt, B. Dai, and D. Lin, "Grid-guided neural radiance fields for large urban scenes," *arXiv preprint arXiv:2303.14001*, 2023.

[23] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, "Citynerf: Building nerf at city scale," *arXiv preprint arXiv:2112.05504*, 2021.

[24] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12922–12931, 2022.

[25] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, pp. 1–11, 2013.

[26] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, 2011.

[27] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, p. 1, 2017.

[28] J. S. De Bonet and P. Viola, "Poxels: Probabilistic voxelized volume reconstruction," in *Proceedings of International Conference on Computer Vision (ICCV)*, vol. 2, p. 3, 1999.

[29] A. Broadhurst, T. W. Drummond, and R. Cipolla, "A probabilistic framework for space carving," in *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1, pp. 388–393, IEEE, 2001.

[30] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.

[31] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460–4470, 2019.

[32] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5939–5948, 2019.

[33] T. Davies, D. Nowrouzezahrai, and A. Jacobson, "Overfit neural networks as a compact shape representation," *arXiv preprint arXiv:2009.09808*, vol. 2, 2020.

[34] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5501–5510, 2022.

[35] A. Eftekhar, A. Sax, J. Malik, and A. Zamir, "Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10786–10796, 2021.

[36] Z. Xie, J. Zhang, W. Li, F. Zhang, and L. Zhang, "S-nerf: Neural radiance fields for street views," *arXiv preprint arXiv:2303.00749*, 2023.

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

[38] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022.

[39] K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, "Nerf: Neural radiance field in 3d vision, a comprehensive review," *arXiv preprint arXiv:2210.00379*, 2022.

[40] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," in *Computer Graphics Forum*, vol. 41, pp. 641–676, Wiley Online Library, 2022.

[41] S. Osher, R. Fedkiw, S. Osher, and R. Fedkiw, "Signed distance functions," *Level set methods and dynamic implicit surfaces*, pp. 17–22, 2003.

[42] J. L. McClelland, D. E. Rumelhart, P. R. Group, *et al.*, *Parallel distributed processing*, vol. 2. MIT press Cambridge, MA, 1986.

[43] C. Won, J. Ryu, and J. Lim, "End-to-end learning for omnidirectional stereo matching with uncertainty prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 3850–3862, 2020.

[44] C. Müller, *Spherical harmonics*, vol. 17. Springer, 2006.

[45] J. F. Blinn, "A trip down the graphics pipeline: pixel coordinates," *IEEE Computer Graphics and Applications*, vol. 11, no. 4, pp. 81–85, 1991.

[46] P. Cai, C. Indhumathi, Y. Cai, J. Zheng, Y. Gong, T. S. Lim, and P. Wong, "Collision detection using axis aligned bounding boxes," *Simulations, Serious Games and Their Applications*, pp. 1–14, 2014.

[47] H. Landau, "Sampling, data transmission, and the nyquist rate," *Proceedings of the IEEE*, vol. 55, no. 10, pp. 1701–1706, 1967.

[48] M. Poggi, D. Pallotti, F. Tosi, and S. Mattoccia, "Guided stereo matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 979–988, 2019.

[49] G. Rainer, A. Ghosh, W. Jakob, and T. Weyrich, "Unified neural encoding of btfs," in *Computer Graphics Forum*, vol. 39, pp. 167–178, Wiley Online Library, 2020.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[51] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2492–2502, 2020.

[52] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[53] C. Won, H. Seok, Z. Cui, M. Pollefeys, and J. Lim, "Omnislam: Omnidirectional localization and dense mapping for wide-baseline multi-camera systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 559–566, IEEE, 2020.

[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

국문 요지

본 학위 청구 논문은 큰 장면에서의 신경망 암시적 표면을 위한 유효한 샘플 범위 분석을 제안한다. 기존의 신경망 3차원 복원 알고리즘은 학습 진행과 사전 지식에 상관 없이 균일하게 샘플링한 위치에 일관적으로 기반하여 인공 신경망 쿼리를 진행한다. 심화된 방법으로서 기존 접근법들은 coarse 네트워크에서 얻은 가중치의 누적 밀도 함수의 역함수를 이용해 2-stage 또는 그보다 많은 반복문을 이용해 장면에 적응적인 샘플링을 시도한다. 하지만 이러한 방법은 표면이 아닌 곳에서의 불필요한 쿼리를 필수적으로 많이 생성하여 장면의 모델링 시간을 늘리고 볼륨 렌더링에서 잡음이 개입되기 쉽게 한다. 1-stage로 유효 범위를 제안 및 제한하는 샘플링 방법은 신경망 radiance장 연구에서 아직까지 심도 있게 다뤄지지 않았다. 본 논문은 기존에 광범위하게 사용되고 있는 깊이 가이드 프레임워크에서 추가 비용 없이 얻을 수 있는 기하학적 사전 지식을 이용하는 한편 최적화가 진행됨에 따라 물체 표면의 위치에 대해 추출할 수 있는 정보를 이용하여 보증된 유효 쿼리 범위를 좁혀나간다. 먼저 볼륨 렌더링의 이산화된 근사치 계산에서 실제 쿼리를 하기 전 지도학습의 부산물로 발생하는 깊이 정보를 이용해 표면이 있을 만한 곳에 1차원 범위 제안을 제시한다. 또한 반복적으로 비 표면 지역으로 투표된 이산 지역을 여러 뷰에서의 결과를 학습 과정 동안 통합하여 만든 3차원 격자에 캐싱하여 추후의 샘플링을 효과적으로 억제한다. 이는 장면에서 대부분의 위치는 빈 공간이나 물체 내부로, 모델링의 대상이 되는 물체 표면은 공간의 적은 부분을 차지한다는 휴리스틱에 기반한 것이다. 본 논문에서 제안하는 방법은 특징 그리드 인코딩 시 카메라 중심으로부터 샘플까지의 거리에 따라 불필요하게 높은 주파수를 가지는 것으로 계산된 해상도의 피처의 신호 세기를 낮추기도 하는데 이는 같은 복원 성능에서 40 퍼센트의 메모리 절감 효과를 가져올 수 있다. 합성 데이터셋과 실제로 촬영된

데이터셋에서 시행한 실험에서 기하학적 성능은 제안하는 표면 적응적 유효 샘플 영역 기법을 사용할 시 유의미하게 향상되었다. 최신 장면 복원 접근 방법에 비해서도 디자인 선택에 따라 최대 4배 빠른 학습 시간을 가졌다. 뿐만 아니라 제안하는 모델을 사용하여 합성 데이터셋과 실제 세계의 큰 규모 장면에서 촬영한 어려운 데이터셋 모두에서 뚜렷한 질적 차이를 가지고 iNGP에 비해 더 복원 대상의 기하와 색깔에 가까운 메쉬를 생성할 수 있었다.

Declaration of Ethical Conduct in Research

I, as a graduate student of Hanyang University, hereby declare that I have abided by the following Code of Research Ethics while writing this dissertation thesis, during my degree program.

"First, I have strived to be honest in my conduct, to produce valid and reliable research conforming with the guidance of my thesis supervisor, and I affirm that my thesis contains honest, fair and reasonable conclusions based on my own careful research under the guidance of my thesis supervisor.

Second, I have not committed any acts that may discredit or damage the credibility of my research. These include, but are not limited to : falsification, distortion of research findings or plagiarism.

Third, I need to go through with Copykiller Program(Internet-based Plagiarism-prevention service) before submitting a thesis."

JUNE 02, 2023

Degree : Master

Department : DEPARTMENT OF COMPUTER SCIENCE

Thesis Supervisor : Jongwoo Lim

Name : MIN CHAERIN (Signature)

연구 윤리 서약서

본인은 한양대학교 대학원생으로서 이 학위논문 작성 과정에서 다음과 같이 연구 윤리의 기본 원칙을 준수하였음을 서약합니다.

첫째, 지도교수의 지도를 받아 정직하고 엄정한 연구를 수행하여 학위논문을 작성한다.

둘째, 논문 작성시 위조, 변조, 표절 등 학문적 진실성을 훼손하는 어떤 연구 부정행위도 하지 않는다.

셋째, 논문 작성시 논문유사도 검증시스템 "카피킬러"등을 거쳐야 한다.

2023년06월02일

학위명 : 석사

학과 : 컴퓨터·소프트웨어학과

지도교수 : 임종우

성명 : 민채린 (서명)

한 양 대 학 교 대 학 원 장 귀 하