Geometric Viewpoint Learning with Hyper-Rays and Harmonics Encoding

Zhixiang Min Juan Carlos Dibene Enrique Dunn
Stevens Institute of Technology

Abstract

Viewpoint is a fundamental modality that carries the interaction between observers and their environment. This paper proposes the first deep-learning framework for the viewpoint modality. The challenge in formulating learning frameworks for viewpoints resides in a suitable multimodal representation that links across the camera viewing space and 3D environment. Traditional approaches reduce the problem to image analysis instances, making them computationally expensive and not adequately modelling the intrinsic geometry and environmental context of 6DoF viewpoints. We improve these issues in two ways. 1) We propose a generalized viewpoint representation forgoing the analysis of photometric pixels in favor of encoded viewing ray embeddings attained from point cloud learning frameworks. 2) We propose a novel SE(3)-bijective 6D viewing ray, hyper-ray*, that addresses the DoF deficiency problem of using 5DoF viewing rays representing 6DoF viewpoints. We demonstrate our approach has both efficiency and accuracy superiority over existing methods in novel real-world environments.*

1. Introduction

Viewpoints play a critical role in a broad range of tasks in computer vision [15, 12, 33, 35], graphics [28, 10], robotics [29, 23, 34, 32] and HCI [20, 2]. Whether given as input or required as output, viewpoints embody complex dependencies among capture dynamics/constraints, observer preferences and task-specific goals. To enable AI systems to understand and characterize these complex dependencies in a data-driven manner, we propose a novel encoding scheme and learning framework for the viewpoint modality, similar to how images being encoded with CNNs. This is a first step towards research and application focusing on, for example cross-modal learning of text-and-view (i.e. find/describe views with texts), text-based robot navigation or behavior recognition from camera movements.

Viewpoint instances play the role of a geometric link between the environment and captured contents, with viewing rays conveying info on both capture-time camera pose and scene content observability. However, existing approaches [12, 15] focused on (rendered) viewpoint image analysis are handicapped in two important ways: First, the inherently local scope of individual photometric pixels obfuscates the geometric and environmental contexts that are crucial cues to characterize the viewpoint. Second, sampling the viewing space through image rendering is computationally demanding. Addressing both of these shortcomings requires novel data representations and compute frameworks.

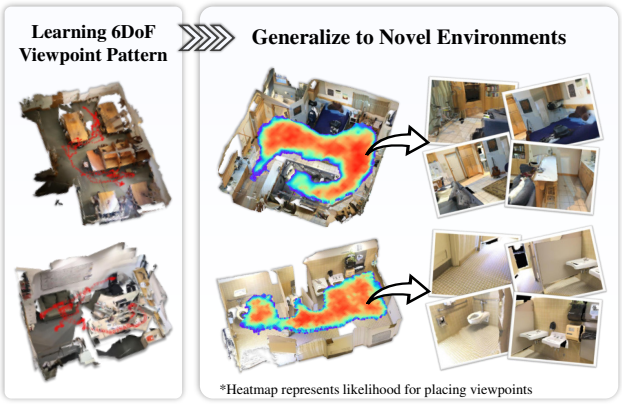


Figure 1: **Viewpoint Learning.** Examples showing our method learns to replicate 6DoF viewpoint capture patterns in novel environments.

Photometric pixels, which convey the signal carried by viewing rays, are widely used in existing viewpoint representations. However, the geometry of viewing rays (i.e. origin, length and direction) is usually ignored, even though such parameters govern the mapping from scene content to the observer. Importantly, viewing ray geometry uniquely describes the capture-time camera pose and observed scene geometry of a given viewpoint. Hence, we represent viewpoints as a collection of viewing rays, which are in turn encoded in terms of both their geometry and their carried signal. To encode viewing rays, we tightly couple the ray encoding process with point cloud learning frameworks

through our proposed Harmonics Ray Encoders (HREs). The HREs are spherical feature fields that can differentiably encode a generalized range-bearing feature of the viewing rays emanating from the observer to the scene elements. In this way, the ray geometry and the corresponding environmental content can be jointly encoded into latent embeddings with rich geometric and semantic context. Given that HREs are parameterized by a compact learnable set of coefficients, they are suitable as an extension for off-the-shelf point cloud learning frameworks.

Next, we address the heavy computational burden of viewpoint learning frameworks induce by the DoF deficiency problem of using viewing rays (5DoF) for representing viewpoints (6DoF). Due to the lower DoF of viewing rays, the methods need to spend large computation budgets on rendering an over-specified set of pixels/rays to uniquely characterize a viewpoint. This also results in a same viewing ray belonging to many different viewpoints, which lays heavy burden on the subsequent analyzer to inspect the interplay of the viewpoints' ray bundles for characterizing different viewpoints. Therefore, we propose a novel SE(3)-bijective 6D viewing rays, namely hyper-rays, where each hyper-ray can uniquely represent a 6DoF viewpoint. The introduction of hyper-rays removes the correlation among the viewing rays within a viewpoint, which greatly improves the efficiency by relieving the burden on the post-analysis stage as well as allowing sparser ray samplings.

Finally, we integrate our proposed ray representation and encoding mechanism into a hierarchical learning framework that first examines the panoramic environment to determine a location sanity score, followed by a 6DoF viewpoint sanity evaluation. By decoupling the location and viewpoint, we can quickly filter the search space by discarding the viewpoints at unlikely locations, as well as enhancing the viewpoint analysis with its panoramic environment. Since our method only needs a sparse point cloud as input, it can efficiently sample and analyze dense 6DoF viewpoint hypotheses of indoor environments within $\sim$10 seconds on a commodity GPU. We summarize our main technical contribution/insights as:

- We propose a new viewpoint representation using encoded viewing rays to endow rich geometric context for viewpoint learning.
- We propose harmonics ray encoders to bridge ray encoding with existing point cloud networks, endowing the embeddings with rich environmental context.
- We extend the representation dimension of viewing rays to enable unambiguous encoding of a $SE(3)$ camera pose in each ray's geometry.
- We propose an efficient inference workflow decouples location and orientation, enabling efficient dense 6DoF viewpoint sampling and analysis.

2. Related Works

Viewpoint Learning. The concept of viewpoint learning rises from the empirical observations that human share common pattern of viewpoint preference for 3D objects [5]. Many works focus on finding good features for evaluating human 2DoF viewpoint (i.e. azimuth+altitude) preferences of an object model. Early works measure the surface visibility [50] and mesh saliency [24] as viewpoint metrics. More recent works propose to use segmented-parts visibility [51], surface distinctness [25] and CNN-based image analysis [53, 22]. Hybrid works use a composition of geometric, appearance and semantic features [43, 28, 47, 18, 9] to evaluate a viewpoint in multiple aspects, where a summary can be found in [6]. Learning 6DoF viewpoint generalizes the scope of 2DoF object viewpoint into a complex traversable 3D environment. The extra 4DoF involves many more practical applications [20, 15, 12, 29] but also introduced many challenges due to the huge 6D pose space. To accommodate the extra DoF, works for indoor scenes [15, 46, 54] usually assume a known scene gravity and fix/restrict camera height, tilt and roll with prior knowledge to reduce the search space as well as keep a certain distance to surrounding objects. Viewpoint goodness is then rated by examining depth and semantics statistics of rendered view w.r.t. the prior data, e.g. Kyle *et al*. [12] proposed to build 3D histogram on camera frustums to record the depth map statistics of each semantic class respectively. The 6DoF works commonly analyze rendered image and depth structure, while disregards the capture time camera pose and its relationship to the environment. Only few works [15, 46, 54] weakly constraint the camera state with simple heuristics (e.g. height, tilt, and collisions). While discarded camera pose info may be partially recoverable by performing post-hoc analysis on the rendered content, doing so inherently compromises overall efficiency, robustness and accuracy of a viewpoint analysis system. Our work provides a general viewpoint representation which is aware of the full $SE(3)$ camera pose. Built on this, viewpoint patterns can be accurately learnt from data, obviating the need for designing heuristic priors on camera poses or image contents.

View Selection with Metrics. View selection applications with clear metrics such as minimizing reconstruction uncertainty [23, 55, 42, 36], maximizing coverage [30] or exploring [40, 29] usually do not learn viewpoint metrics from data. However, Sun *et al*. [48] recently shows that learning to approximate well-defined metrics (i.e. coverage) using neural networks enables fast optimization from a coarse scene geometric proxy. Suggesting the use of deep learning in view selection problems may reduce the demanding for scene fidelity and optimize the selection process.

Scene Representation Learning. 3D scene representation learning has been widely studied, where diverse network

architectures are designed for semantic scene understanding including MLP-based [37, 38], Point-convolution [27, 49, 52], GNN/Transformer [56, 14] and 3D-CNN [39, 46]. Given the close relationship between the viewpoints and its scene environment, this work tightly couples viewpoint learning with point cloud representation learning to endow our viewpoint representations with view-dependent rich environmental context extracted from the point cloud.

View-Dependent Encoding. In the graphics domain, image-based rendering (IBR) methods [26, 13, 44] model plenoptic functions to map viewing rays into photo-realistic appearance. Model-based renderings [45, 11, 41, 31] also model the view-dependent effects using BRDF, spherical harmonics or MLPs. While sharing similar ideas, we generalize the concept of rendering to viewpoint learning by aggregating view-dependent features from environmental content instead of rendering photo-realistic images.

Viewing Ray Representations. Many image-based rendering methods [26, 13, 44] have studied dimensionality reduction of generic 5D viewing rays. While this simplifies their plenoptic parameterization, it introduces degenaracies into the viewing space (i.e. camera pose space). Conversely, our method lifts the ray dimension to 6D, for a single ray to unambiguously represent camera poses in $SE(3)$, allowing efficient and effective viewpoint learning.

3. Method

Problem Formulation. To study viewpoint learning frameworks with minimal distractions, we focus on a simple and controlled setup that evaluates whether a method's inductive bias can faithfully capture viewpoint patterns underlying in training data and generalize to unseen environments. Specifically, we assume a dataset is created by an observer who samples viewpoints $\mathbf{T} \in SE(3)$ from an underlying distribution $\mathcal{D}(\mathbf{T} \mid \Pi)$, where Π is the scene 3D model. After learning from training data, we are interested in discriminating whether an arbitrary viewpoint $\mathbf{T}'$ in an novel environment Π' is sampled from $\mathcal{D}(\mathbf{T}' \mid \Pi')$ or an uniform pose distribution $\mathcal{U}(\mathbf{T}')$.

3.1. Viewpoint Representation

The conditional probability $\mathcal{D}(\mathbf{T} \mid \Pi)$ implies our interest in modelling viewpoints as the relationship between a camera pose and its environment. Accordingly, we represent viewpoints using their pencils of viewing rays emanating to the scene. Towards this end, we introduce our optic-ray and hyper-ray parameterizations.

Optic-Rays. Optic-rays [4] are 5DoF half-lines in 3D space, which can be parameterized as

$$\mathbf{L} \in \mathbb{L}^5 = \{\mathbf{o} \in \mathbb{R}^3, \mathbf{d} \in \mathbb{S}^2\} \tag{1}$$

where $\mathbf{o}$ is the 3-vector ray origin and $\mathbf{d}$ is ray direction on the 2-sphere $\mathbb{S}^2$. Compared with 6DoF camera poses in $SE(3)$, the optic rays only have 5DoF, which means a single optic ray is an ambiguous representation for camera pose. Mathematically, let $\mathbf{L}$ be a viewing ray from a viewpoint with pose $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ and passing through $[u, v, 1]$ in camera coordinates. From each optic-ray we can only determine a variety of camera poses satisfying the following

$$\alpha : \mathbb{L}^5 \times \mathbb{P}^2 \rightarrow \left\{ \mathbf{T} \;\middle|\; \mathbf{t} = \mathbf{o},\ \frac{\mathbf{R}[u, v, 1]^\top}{||\mathbf{R}[u, v, 1]^\top||} = \mathbf{d} \right\} \tag{2}$$

where $\mathbb{P}^2$ denotes camera coordinates in the projective plane. Eq.(2) implies a viewpoint's location is determined by a ray's origin, but the rotation is conditioned on the ray's image projection coordinates and under-determined as one DoF is unspecified. Co-located viewpoints with different orientation will share common optic-rays, requiring multiple ($\geq$2) optic-rays with known image projections for their disambiguation. As a result, depending on the companion rays, a same optic ray could both belong to a good viewpoint or bad viewpoint. This lays heavy burden on the post analyzer/classifier to inspect the interplay of the ray bundle to recover the full $SE(3)$ camera pose of the viewpoint, resulting in inefficient and inaccurate viewpoint modelling. However, as the ambiguity only exists in orientation, optic-rays are still good representations for viewpoint location.

Hyper-Rays. To unambiguously represent a viewpoint pose using a single viewing ray, we introduce a 6DoF hyper-ray representation parameterized as

$$\hat{\mathbf{L}} \in \mathbb{L}^6 = \{\hat{\mathbf{o}} \in \mathbb{R}^3, \mathbf{q} \in \mathbb{S}^3\} \tag{3}$$

The hyper-ray lifts the ray direction onto a 3-sphere as a unit quaternion $\mathbf{q} = (q_w, q_x, q_y, q_z)$, $\|\mathbf{q}\| = 1$. Geometrically, the extra DoF implicitly represents the roll-axis orientation of the optic-rays, which will be detailed in §3.3.2. Let $\mathcal{R}(\mathbf{q})$ be the rotation matrix form [17] of quaternion $\mathbf{q}$ as

$$\mathcal{R}(\mathbf{q}) = 2 \cdot \begin{bmatrix} q_w^2 + q_x^2 - 0.5 & q_x q_y - q_w q_z & q_w q_y + q_x q_z \\ q_w q_z + q_x q_y & q_w^2 + q_y^2 - 0.5 & q_y q_z - q_w q_x \\ q_x q_z - q_w q_y & q_w q_x + q_y q_z & q_w^2 + q_z^2 - 0.5 \end{bmatrix} \tag{4}$$

The mapping between the hyper-ray and camera pose is then straight-forward

$$\beta : \mathbb{L}^6 \rightarrow SE(3),\ \beta(\hat{\mathbf{L}}) = \{\mathbf{T} \mid \mathbf{t} = \hat{\mathbf{o}}, \mathbf{R} = \mathcal{R}(\mathbf{q})\} \tag{5}$$

Since the mapping from $\mathbf{q}$ to $\mathbf{R}$ is a double-cover (i.e. $\mathcal{R}(\pm\mathbf{q}) = \mathbf{R}$)[17], we fix the real part of quaternion q_w to be non-negative to make this mapping bijective. Hence, hyper-rays define an unambiguous ray representation that bijectively corresponds to the full $SE(3)$. With the extra DoF on ray direction, we avoid overlapping viewing rays between different viewpoints and the requirement of knowing ray image projections. In exchange for a mild computational

burden during rendering (described in §3.3.2), hyper-rays provide a distinguishable representation that eliminates the need to analyze the ray interplay, which reliefs burden on the downstream classifier.

3.2. Harmonics Ray Encoder (HRE)

We introduce the harmonics ray encoder (HRE) scheme to encode a viewpoint's pencil of viewing rays for downstream classification tasks. Applied harmonic analysis [3] expresses signals through the composition of individual oscillatory components such as sinusoidal functions. Based on this, HREs stack multiple individual sets of harmonics to form feature fields $\mathcal{F}(\cdot) \rightarrow \mathbb{R}^D$ to encode the range-bearing attributes of the rays.

Point Cloud Map. We first consider an input 3D scene model Π in point cloud format with P points as

$$\Pi = \left\{ \mathbf{x}_p, \mathbf{c}_p, \mathbf{n}_p \in \mathbb{R}^3 \; ; \; p = 1 \dots P \right\} \tag{6}$$

where $\mathbf{x}_p$ is the 3D coordinate, $\mathbf{n}_p$ is a unit-length normal vector, $\mathbf{c}_p$ is the color of point p. We use a PointNet[37] to process the point cloud map into per-point coefficients defining our harmonics feature fields as shown in Fig.3(a). Viewing rays emanating to the point p will be encoded with a feature from the its feature field in accordance to ray attributes at rendering time.

Direction Feature Field on $\mathbb{S}^2$ and $\mathbb{S}^3$. The ray direction of optic-rays $\mathbf{d}$ and hyper-rays $\mathbf{q}$ live on $\mathbb{S}^2$ and $\mathbb{S}^3$ respectively. Hence, we use spherical harmonics on $\mathbb{S}^2$ and $\mathbb{S}^3$ [3] to represent a feature field encoded by ray directions. We denote the directional feature fields of viewing rays incoming to point p respectively as $\mathcal{F}^p_{\mathbf{d}}$ and $\mathcal{F}^p_{\mathbf{q}}$ as

$$\mathcal{F}^p_{\mathbf{d}}(\mathbf{d}) = \sum_{l=0}^{H_2} \sum_{m=-l}^{l} \mathbf{a}^{[\mathbf{d}]}_{lm} Y_{lm}(\mathbf{d}) \tag{7}$$

$$\mathcal{F}^p_{\mathbf{q}}(\mathbf{q}) = \sum_{k=0}^{H_3} \sum_{l=0}^{k} \sum_{m=-l}^{l} \mathbf{a}^{[\mathbf{q}]}_{klm} Y_{klm}(\mathbf{q}) \tag{8}$$

where $Y.(\cdot)$ are harmonic polynomials, whose explicit formulas are included in the appendix. H_2 and H_3 are hyper-parameters of maximum harmonics degrees, and $\mathbf{a}^{[\mathbf{d}]}_{lm}, \mathbf{a}^{[\mathbf{q}]}_{klm} \in \mathbb{R}^D$ are PointNet predicted per-point coefficients that define the feature field.

Length Feature Field on $\mathbb{R}^1$. The remaining ray attribute is its origin $\mathbf{o}$. Given that HREs are conditioned on specific points and ray direction is explicitly encoded, we only need to encode a 1D ray length to uniquely imply the ray origin. Consider a given ray length γ, as a 1D parameter, it is natural to consider a Fourier series (i.e. a flattened circular harmonics) as

$$\mathcal{F}^p_{\gamma}(\gamma) = \mathbf{a}^{[\gamma]}_0 + \textstyle\sum_{n=1}^{H_1} \left(\mathbf{a}^{[\gamma]}_{2n+1} \, cos\big(\frac{2\pi\gamma}{\gamma_{max}} n\big) + \mathbf{a}^{[\gamma]}_{2n+2} \, sin\big(\frac{2\pi\gamma}{\gamma_{max}} n\big) \right) \tag{9}$$

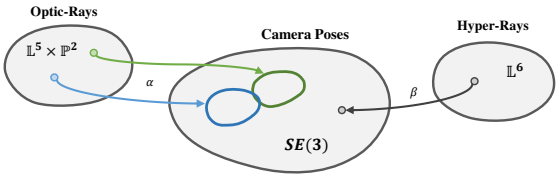


Figure 2: **Mappings from rays to camera poses.** The optic-rays map to a curve (1-DoF) in $SE(3)$ given an image coordinate. While our proposed hyper-rays can uniquely determine any camera poses in $SE(3)$ without the knowledge for image projection.

where γ_{max} is a pre-defined maximum length, H_1 the number of harmonics, and $\mathbf{a}^{[\gamma]}_n \in \mathbb{R}^D$ are PointNet predicted per-point coefficients defining the feature field. We set a slightly larger maximum distance γ_{max} to prevent the periodicity from mapping zero and max distance to the same feature.

3.3. Viewpoint Modelling

Leveraging our ray representation and HREs, we design an efficient two-stage workflow by decoupling location and orientation as shown in Fig.3(b,c). Our first stage uses optic-rays to efficiently analyze the location of a viewpoint, while the second stage takes into account the viewpoint orientation at selected locations through hyper-rays.

3.3.1 Location Branch.

We first determine a score for a given location $\mathbf{t}$ by examining its panoramic environment. Given our interest only on location, the simple optic-rays are enough to determine a unique 3DoF location without knowing image projection according to Eq.(2). Hence, we aggregate all optic-rays corresponding to $\mathbf{t}$ w.r.t. the point cloud as

$$\left\{ \mathbf{L}_{\mathbf{t}\rightarrow p} \;\middle|\; \mathbf{o}_{\mathbf{t}\rightarrow p} = \mathbf{t}, \, \mathbf{d}_{\mathbf{t}\rightarrow p} = \frac{\mathbf{x}_p - \mathbf{t}}{\|\mathbf{x}_p - \mathbf{t}\|} \; ; \; p = 1 \dots P \right\} \tag{10}$$

where we denote $\mathbf{L}_{\mathbf{t}\rightarrow p}$ as the optic-ray from location $\mathbf{t}$ to point p. We also denote the ray length as $\gamma_{\mathbf{t}\rightarrow p} = \|\mathbf{x}_p - \mathbf{t}\|$. We encode ray distance and directional attributes using Eq.(9) and (7), and we will have the final ray features from the sum of two features as

$$\mathbf{f}_{\mathbf{t}\rightarrow p} = \mathcal{F}^p_{\gamma}(\gamma_{\mathbf{t}\rightarrow p}) + \mathcal{F}^p_{\mathbf{d}}(\mathbf{d}_{\mathbf{t}\rightarrow p}) \tag{11}$$

Spherical Voronoi pooling with optic-rays. Naively aggregating all rays in Eq.(10) as the location representation leads to bias on viewing directions having denser points. Hence, we propose spherical Voronoi pooling to equalize ray density along each viewing direction. As shown in Fig.3(b), we create a 2-sphere Voronoi diagram at location $\mathbf{t}$, where we use spherical Fibonacci sampling [16] to evenly sample points $\left\{ \mathbf{d}_c \;\middle|\; c = 1 \dots V_2 \right\}$ on $\mathbb{S}^2$ as the center

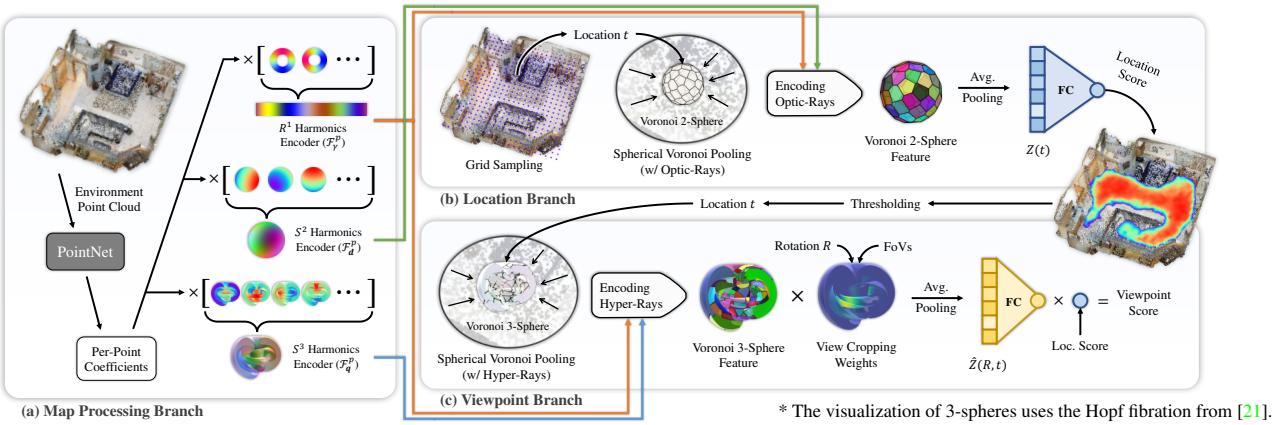


Figure 3: **Workflow.** a) The scene point cloud is processed with PointNet to assign per-point HREs. b) The location branch estimates a score for each location by analyzing its panoramic optic-rays. c) For selected locations, the viewpoint branch caches panoramic hyper-rays in a Voronoi 3-sphere feature, and the viewpoint features are cropped from the 3-sphere for determining the viewpoint score.

of each Voronoi cell. We aggregate weighted ray features and store them in each cell, hence we denote the sphere as a Voronoi sphere feature. We average all cell features together to form the feature descriptor of location $\mathbf{t}$ as

$$\mathbf{Z}(\mathbf{t}) = \frac{1}{V_2}\sum_{c=1}^{V_2}\sum_{p=1}^{P}\frac{w_{c\leftarrow p}}{\sum_{p=1}^{P} w_{c\leftarrow p}}\mathbf{f}_{\mathbf{t}\rightarrow p} \tag{12}$$

where $w_{c\leftarrow p}$ is a visibility weight of ray $\mathbf{L}_{\mathbf{t}\rightarrow p}$ to cell c. To determine its value, we conduct a simple efficient visibility test for each cell as

$$w_{c\leftarrow p} = \begin{cases} 1 & \text{if } \; p = \underset{p^*\in\Omega_c}{argmin}\, \gamma_{\mathbf{t}\rightarrow p^*} \\ 0 & otherwise \end{cases} \tag{13}$$
$$where \quad \Omega_c = \left\{ p \;\middle|\; c = \underset{c^*}{argmax}\, \mathbf{d}_{c^*}\cdot\mathbf{d}_{\mathbf{t}\rightarrow p} \right\}$$

Ω_c is a set of all points whose ray intersects the cell c. The formula implies that only the ray with shortest length within the cell is considered visible. Finally, we map the location $\mathbf{t}$'s feature descriptor $\mathbf{Z}(\mathbf{t})$ to a score value range $[0, 1]$ as

$$S_{loc}(\mathbf{t}) = \Phi_{loc}\big(\mathbf{Z}(\mathbf{t})\big) \tag{14}$$

where $\Phi(\cdot)$ is a linear classifier with sigmoid activation.

3.3.2 Viewpoint Branch.

Next, we score a viewpoint pose $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ by aggregating hyper-rays corresponding to the viewpoint. As hyper-rays are not yet associated with geometry entities in 3D space, we define an auxiliary mapping that "collapses" one of its dimension to yield an optic-rays used for rendering as

$$aux : \mathbb{L}^6 \rightarrow \mathbb{L}^5,\; aux(\hat{\mathbf{L}}) = \left\{ \mathbf{L} \;\middle|\; \mathbf{o} = \hat{\mathbf{o}},\; \mathbf{d} = \mathcal{R}_3(\mathbf{q}) \right\} \tag{15}$$

where $aux(\cdot)$ gives the auxiliary optic-ray of a hyper-ray and $\mathcal{R}_3(\mathbf{q})$ denotes the third column of $\mathcal{R}(\mathbf{q})$ as in Eq.(4). We also denote the inverse of Eq.(15) as lifting the optic-ray to a variety of hyper-ray as

$$lift : \mathbb{L}^5 \rightarrow \mathbb{L}^6,\; lift(\mathbf{L}) = \left\{ \hat{\mathbf{L}} \;\middle|\; aux(\hat{\mathbf{L}}) = \mathbf{L}_{\mathbf{t}\rightarrow p} \right\} \tag{16}$$

The $aux(\cdot)$ and $lift(\cdot)$ build the geometric relationship between optic-rays and hyper-rays, where the auxiliary optic-ray gives the direction of the optical axis of the hyper-ray's corresponding viewpoint. This choice guarantees the hyper-ray gathers the "wanted" scene content (i.e. observable from its corresponding viewpoint) from its auxiliary optic-ray geometry. On the other hand, lifting an optic-ray can be seen as implicitly assigning an extra dimension to represent the roll-axis orientation of the optic rays. Since the $lift(\mathbf{L})$ corresponds to all viewpoints having $\mathbf{L}$ as their optical axis, whose roll angles comes from the extra lifted dimension.

Spherical Voronoi pooling with hyper-rays. Consider the set of hyper-rays associated with location $\mathbf{t}$ emanating to every point cloud element p as

$$\left\{\, lift(\mathbf{L}_{\mathbf{t}\rightarrow p})\;;\; p = 1 \ldots P \,\right\} \tag{17}$$

where each optic-ray is lifted to a continuous variety of hyper-rays corresponding to all possible viewpoints looking at point p. Our goal is to evenly sample hyper-rays from this continuous variety along each viewing orientation in $SO(3)$. Since evenly sampling on a 3-sphere is equivalent to evenly sampling rotations in $SO(3)$ [17, 1], we propose a spherical Voronoi pooling on the hyper-rays analogous to the one on optic-rays in §3.3.1. This mechanisms will enable us to deal with a continuous variety instead a discrete set of rays. As shown in Fig.3(c), we first create a Voronoi 3-sphere feature at location $\mathbf{t}$ using super-Fibonacci spiral [1] which samples V_3 evenly distributed

points $\{\mathbf{q}_c \,|\, c = 1 \dots V_3\}$ on $\mathbb{S}^3$ as the center of each Voronoi cell. We define the feature of cell c as

$$\hat{\mathbf{Z}}_c(\mathbf{t}) = \sum_{p=1}^{P} \frac{w_{c \leftarrow p}}{\sum_{p=1}^{P} w_{c \leftarrow p}} \big(\mathcal{F}_{\gamma}^{p}(\gamma_{\mathbf{t} \to p}) + \mathcal{F}_{\mathbf{q}}^{p}(\mathbf{q}_{\mathbf{t} \to p}^{c})\big) \quad (18)$$

where $\mathbf{q}_{\mathbf{t} \to p}^{c}$ is a hyper-ray direction that satisfies

$$\begin{aligned} \mathbf{q}_{\mathbf{t} \to p}^{c} &= \underset{\mathbf{q}^*}{argmin}\big(\angle(\mathbf{q}^*)\big) \circ \mathbf{q}_c \\ s.t. \quad & \mathcal{R}_3(\mathbf{q}^* \circ \mathbf{q}_c) = \mathbf{d}_{\mathbf{t} \to p} \end{aligned} \quad (19)$$

where $\circ$ denotes the quaternion product and $\angle(\cdot)$ gives the rotation degree of the quaternion. The formula implies $\mathbf{q}^*$ to be the smallest quaternion rotation that aligns $\mathcal{R}_3(\mathbf{q}_c)$ to $\mathbf{d}_{\mathbf{t} \to p}$, which guarantees $\mathbf{q}_{\mathbf{t} \to p}^{c}$ belongs to the variety in Eq.(17). The $\mathbf{q}^*$ can be found in a closed form as

$$\begin{aligned} \mathbf{q}^* &= \big[q_w^*, q_x^*, q_y^*, q_z^*\big] = \Big[cos\big(\frac{|\mathbf{v}|}{2}\big),\ sin\big(\frac{|\mathbf{v}|}{2}\big)\frac{\mathbf{v}}{|\mathbf{v}|}\Big] \\ where \quad \mathbf{v} &= \frac{\mathcal{R}_3(\mathbf{q}_c) \times \mathbf{d}_{\mathbf{t} \to p}}{\big|\mathcal{R}_3(\mathbf{q}_c) \times \mathbf{d}_{\mathbf{t} \to p}\big|} \cdot cos^{-1}\big(\mathcal{R}_3(\mathbf{q}_c) \cdot \mathbf{d}_{\mathbf{t} \to p}\big) \end{aligned} \quad (20)$$

where $\times$ denotes the cross product. The point-to-cell visibility $w_{c \leftarrow p}$ in Eq.(18) is then estimated using Eq.(13) with a modified $\hat{\Omega}_c$ as

$$\hat{\Omega}_c = \big\{p \;\big|\; c = \underset{c^*}{argmax}\, \mathcal{R}_3(\mathbf{q}_{c^*}) \cdot \mathbf{d}_{\mathbf{t} \to p}\big\} \quad (21)$$

View Cropping. With the Voronoi 3-sphere feature for location $\mathbf{t}$, we define a view cropping operation to extract a feature descriptor for a given camera rotation $\mathbf{R}$. The view cropping on a Voronoi 3-sphere feature is an efficient weighted averaging of cell features, written as

$$\hat{\mathbf{Z}}(\mathbf{R}, \mathbf{t}) = \frac{\sum_{c=1}^{V_3} w_c(\mathbf{R}) \hat{\mathbf{Z}}_c(\mathbf{t})}{\sum_{c=1}^{V_3} w_c(\mathbf{R})} \quad (22)$$

where $w_c(\mathbf{R}) \in [0, 1]$ is the weight of cell c. It is determined by the difference between cell direction $\mathbf{q}_c$ and query rotation $\mathbf{R}$ in a Gaussian kernel manner on Euler angles as

$$\begin{aligned} w_c(\mathbf{R}) &= exp(-\mathbf{u}^T\, diag(\sigma_\phi^2, \sigma_\theta^2, \sigma_\psi^2)^{-1}\, \mathbf{u}) \\ where \quad \mathbf{u} &= \big[\angle(\phi, \phi_c), \angle(\theta, \theta_c), \angle(\psi, \psi_c)\big]^T \end{aligned} \quad (23)$$

where (ϕ, θ, ψ) and $(\phi_c, \theta_c, \psi_c)$ are yaw, pitch and roll angles of $\mathbf{R}$ and $\mathcal{R}(\mathbf{q}_c)$, and $\mathbf{u}$ defines a vector angular distance between the query rotation and cell orientation. We let the view cropping sigmas $(\sigma_\phi, \sigma_\theta)$ linearly depend on viewpoint camera horizontal/vertical FoVs (η_h, η_v) with a hyper-parameter λ controlling the receptive field as

$$[\sigma_\phi, \sigma_\theta, \sigma_\psi] = \lambda \cdot [\eta_h, \eta_v, \eta_r] \quad (24)$$

where η_r is a pre-defined virtual roll-axis FoV. This weighting mechanism rates the visibility between the cells and the viewpoint considering the actual camera FoVs, where the cells-of-interest are assigned with high weight to contribute more in the viewpoint feature. Next, we map the viewpoint $(\mathbf{R}, \mathbf{t})$'s feature descriptor $\hat{\mathbf{Z}}(\mathbf{R}, \mathbf{t})$ to a score value range $[0, 1]$ using as a linear classifier

$$S_{view}(\mathbf{R}, \mathbf{t}) = \Phi\big(\hat{\mathbf{Z}}(\mathbf{R}, \mathbf{t})\big) \quad (25)$$

We finally rate a viewpoint based on a composition of location and view score

$$S_{final}(\mathbf{R}, \mathbf{t}) = S_{loc}(\mathbf{t}) \cdot S_{view}(\mathbf{R}, \mathbf{t}) \quad (26)$$

3.4. Implementation

Training. Our framework's learnable parts are *i)* the PointNet used to learn an HRE for each point cloud element, *ii)* a pair of linear classifiers for mapping location and viewpoint features to a scalar score. We use binary cross entropy (BCE) losses on in-sample (i.e positive) and out-of-sample (i.e. negative) viewpoints from the training data. We first consider the location loss

$$\mathcal{L}_{loc} = -\sum_i log\big(S_{loc}(\mathbf{t}_i^+)\big) - \sum_j log\big(1 - S_{loc}(\mathbf{t}_j^-)\big) \quad (27)$$

where $\mathbf{t}^+$ denotes a location from the training viewpoints and $\mathbf{t}^-$ denotes a location randomly sampled within the scene boundary. Similarly, we consider the viewpoint loss

$$\mathcal{L}_{view} = -\sum_i log\big(S_{view}(\mathbf{R}_i^+, \mathbf{t}_i^+)\big) - \sum_j log\big(1 - S_{view}(\mathbf{R}_i^-, \mathbf{t}_j^+)\big) \quad (28)$$

where $\mathbf{R}_i^+$ denotes the rotation of training viewpoint w.r.t. location $\mathbf{t}_i^+$, and $\mathbf{R}^-$ is a randomly sampled rotation. Due to our decoupled inference, we only sample negative viewpoints with positive locations. This also leads to a faster training since randomly sampled 6DoF viewpoints contain a large number of uninformative poor samples.

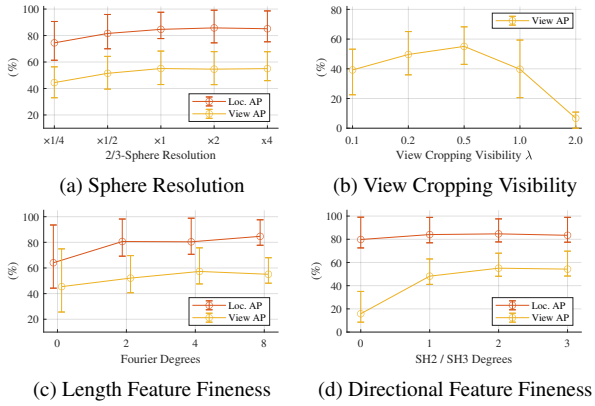
Traversal. Our inference performs a traversal on the $SE(3)$ space to score every viewpoint. For efficiency, we perform hierarchical scheme which decouples the location and orientation. We first grid sample 3D locations and estimate their location scores. For those score above a threshold, we estimate their Voronoi 3-sphere feature and crop viewpoint features for uniformly sampled rotations from super-fibonacci spiral [1]. Finally, we apply non-maximum-suppression and threshold the viewpoint scores to extract locally optimal viewpoints. Our implementation has an inference speed at ∼8KHz for rendering a 2-sphere feature and 90Hz for rendering a 3-sphere feature on a GTX1080Ti with a customized pytorch acceleration kernel. The remaining computation (e.g. linear classifier, view cropping, NMS) is negligible compare to the rendering. It takes ∼10 seconds to sample a scene from ScanNet with a sampling grid

	Location (<0.5m)			View (<0.5m & <30°)				View + GT Loc. (<0.5m & <30°)			
Method	Prec.	Recall	AP	Prec.	Recall	AP	FID	Prec.	Recall	AP	FID
GT H.R.P.	-	-	27.15	-	-	10.70	-	-	-	56.73	-
Adrian *et al.* [43]	48.87	68.10	53.87	3.81	22.79	3.36	230.88	14.44	62.43	15.24	199.49
+ fix gravity	-	-	-	17.76	54.53	19.52	169.31	63.91	67.24	68.10	134.86
Kyle *et al.* [12]	64.54	73.62	71.89	19.24	46.33	20.93	187.63	64.62	65.28	62.60	158.52
+ fix gravity	-	-	-	20.91	48.11	22.99	180.92	67.26	74.82	70.41	148.41
Ours	**82.53**	**95.07**	**84.67**	**52.79**	**79.92**	**55.08**	**142.14**	**70.93**	**92.63**	**72.41**	**123.78**

Table 1: **Viewpoint selection accuracy on ScanNet.** We respectively study the selection accuracy of the locations, viewpoints and viewpoints under GT locations. The accuracy metrics were estimated with a tolerance threshold shown in the first line. The precision, recall and FID are reported at a threshold of highest F1 score.

$\mathbb{R}^1$ Length Enc.	-	✓	-	-	✓	✓	-	✓
$\mathbb{S}^2$ Direction Enc.	-	-	✓	-	✓	-	✓	✓
$\mathbb{S}^3$ Direction Enc.	-	-	-	✓	-	✓	✓	✓
Loc. AP	50.16	76.93	79.26	50.30	80.88	79.79	64.06	82.53
View AP	3.65	15.22	7.95	29.58	15.78	53.96	45.44	52.79

Table 2: **Ablation Study on ScanNet.** We study the effect of each ray encoding components in the HREs.



(a) Sphere Resolution (b) View Cropping Visibility

(c) Length Feature Fineness (d) Directional Feature Fineness

Figure 4: **Analysis over hyper-parameters.** The error bar indicates the first and third quartile AP on ScanNet scenes.

of 0.2m and 4096 orientations for every location above the threshold of 0.8.

Detailed Configurations. For training, we use a batch size of 4 and train for 100K steps using Adam optimizer. We sample 4 inlier locations with 32 randomly sampled locations per batch for training location loss in Eq.(27), and 4 inlier viewpoints with each inlier viewpoint replaced with 8 randomly sampled rotation for training view loss in Eq.(28). We augment the point cloud with random rotations. Training only takes 6 hours on a single GTX1080Ti. We attached a table for hyper-parameter in appendix.

4. Experiments

4.1. ScanNet Dataset.

Setups. We use ScanNet[7] to test our method on indoor environments. ScanNet contains 1513 scans of 707 different scenes reconstructed using RGB-D SLAM [8]. Viewpoints are captured by humans guided by a software indicating image featurefulness[7]. We sample locations within the scene boundary into a 0.2 meter 3D grid and only filter those under a small threshold (0.01) to retain as many samples as possible for evaluation. For selected locations, we sample 4096 evenly distributed rotations using [1].

Baselines. We implement the viewpoint metrics used in [12] and [43], which statistically model the semantic labels and depth distribution of rendered images from ScanNet. We also include a random baseline (GT H.R.P.) as a reference for environment scale, where we randomly select viewpoints with average GT height, roll and pitch of each scene. We detail their implementations in the appendix.

Quantitative Study. In Table.1, we evaluate location and viewpoint selection precision per each scan separately and report mean precision, recall, AP and FID [19]. Precision denotes the percentage of selected viewpoints localized within a distance threshold from any GT viewpoint. Conversely, recall denotes the percentage of GT viewpoint being retrieved by any selected viewpoints. The FID score [19] indicates the appearance similarity and distribution between the rendered image (from mesh) of GT and selected viewpoints. For baseline methods, to compensate their orientation agnostic modeling, we align the roll-axis to the gravity direction and denote as "+fix gravity". The precision, recall and FID are all reported at a global threshold across scenes that gives the highest F1 score. Our method shows large advantage for all metrics, especially for the most challenging $SE(3)$ (i.e. 6DoF) viewpoint selection.

Qualitative Study. Fig.5 compares viewpoint selection results to baselines. Our method captures more fine-grained location distribution as depicted in the heatmap. Histogram plots shows our selection captures realistic viewpoint orientations, whereas baseline methods all exhibit large drift in height, pitch and roll. More results in appendix.

Ablation Study. Table.2 shows the ablation study over the viewing-ray encoding components. We disable an encoding component by setting its HRE degree to zero, yielding

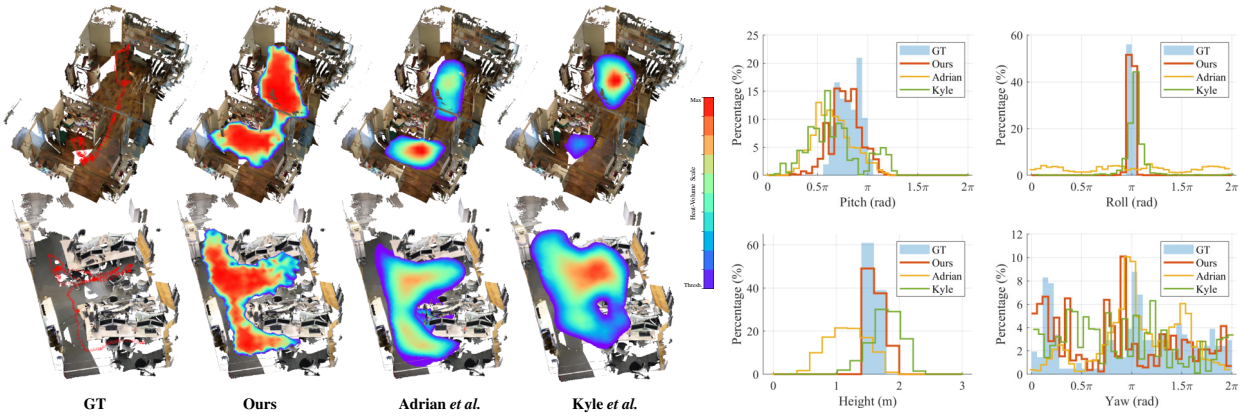


Figure 5: **Results on ScanNet.** The visualized heat-maps are 3D heat-volumes clipped to the height of maximal accumulated score. The right plots are histograms comparing the camera pose distribution of selected viewpoints v.s. GT viewpoints in the scene of first row.

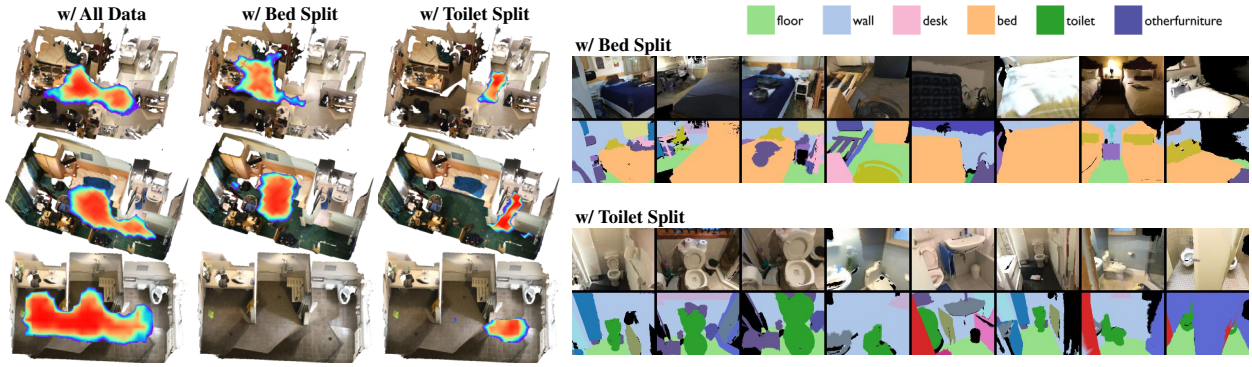


Figure 6: **Results with semantically divergent training splits.** The left shows inferred location score with models trained using all/bed/toilet splits respectively. The right shows selected viewpoints, rendered with RGB and GT semantic labels.

a constant feature field. Missing either length and direction encoding for optic-rays slightly drops the location AP, while missing both them corrupts performance. By disabling the $\mathbb{S}^3$ direction encoding, we degenerate the hyper-rays to optic-rays, which gives significantly lower viewpoint AP (15.78% vs 52.79%), while location AP was not affected. Fig.4 shows a performance study over hyper-parameters. Our method is robust to the choice of harmonics degrees and Voronoi sphere resolution. View cropping visibility λ needs to be chosen carefully, since too large values will homogenize features of different viewpoints while too small values have too narrow receptive field.

Learning the explicit semantic purpose. While the original capture patterns from the ScanNet dataset explore the entire scene for reconstruction, we further test our model's ability to capture more explicit semantic purpose underlying viewpoint patterns. We create semantic splits of the ScanNet dataset by only keeping the captures observing a manually selected object class (e.g. bed). In this case, we attach GT semantic labels to the point cloud as inputs, where its performance gain is studied in the appendix.

We show qualitative results in Fig.6, where we test on two semantic object classes (i.e. bed and toilet). The left figure compares the location score (visualized as a heatmap) of our model trained on all data, on the "bed" split and "toilet" split, respectively. Our method correctly assigns high score to the locations that are reasonable to observe the objects of selected class. The figures on the right are randomly selected viewpoints of both models trained on semantic splits. The resulting images correctly contain the object of interest. The result shows the effectiveness of our data-driven viewpoint learning at capturing the observer semantic purposes.

5. Conclusions, Limitations and Prospects

In this work, we propose a fundamental set of tools for viewpoint learning, which includes a powerful viewpoint geometric abstraction, an encoding scheme that extends existing point cloud networks into viewpoint learning net-

works, and an efficient decoupled SE(3) traversal scheme enabling high-density viewpoint analysis. As a result, our method can efficiently capture viewpoint patterns of appropriate semantic purposes, realistic camera poses and sensible geometric context for novel environments. This paves the way for cross-modal integration with language models such as CLIP to learn a semantically meaningful metric space, which further enables exciting future research directions include cross-modal learning of text-and-view (i.e. find/describe views with texts), and diverse applications spanning from robotics (e.g. text-based robot navigation) to AR/VR (e.g. behavior recognition from camera movements). Per limitations, our method models the geometric properties of viewpoint preferences, not focusing on aesthetic appearance properties, which may require the use of a high-fidelity mesh processing front-end and can be considered as future work.

Acknowledgement

Work sponsored by the Defense Advanced Research Projects Agency contract HR00112220003, contents do not necessarily reflect the position of the U.S. Government, and no official endorsement should be inferred.

References

[1] Marc Alexa. Super-fibonacci spirals: Fast, low-discrepancy sampling of so (3). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8291–8300, 2022. 5, 6, 7

[2] Carlos Andújar, P Vázquez, and Marta Fairén. Way-finder: Guided tours through complex walkthrough models. In *Computer Graphics Forum*, volume 23, pages 499–508. Wiley Online Library, 2004. 1

[3] Sheldon Axler, Paul Bourdon, and Ramey Wade. *Harmonic function theory*, volume 137. Springer Science & Business Media, 2013. 4

[4] James R Bergen and Edward H Adelson. The plenoptic function and the elements of early vision. *Computational models of visual processing*, 1:8, 1991. 3

[5] Volker Blanz, Michael J Tarr, and Heinrich H Bülthoff. What object attributes determine canonical views? *Perception*, 28(5):575–599, 1999. 2

[6] Xavier Bonaventura, Miquel Feixas, Mateu Sbert, Lewis Chuang, and Christian Wallraven. A survey of viewpoint selection methods for polygonal models. *Entropy*, 20(5):370, 2018. 2

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 7

[8] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 7

[9] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Studying aesthetics in photographic images using a computational approach. In *European conference on computer vision*, pages 288–301. Springer, 2006. 2

[10] Ritendra Datta and James Z Wang. Acquine: aesthetic quality inference engine-real-time automatic rating of photo aesthetics. In *Proceedings of the international conference on Multimedia information retrieval*, pages 421–424, 2010. 1

[11] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 3

[12] Kyle Genova, Manolis Savva, Angel X Chang, and Thomas Funkhouser. Learning where to look: Data-driven viewpoint set selection for 3d scenes. *arXiv preprint arXiv:1704.02393*, 2017. 1, 2, 7

[13] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996. 3

[14] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 3

[15] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2

[16] JH Hannay and JF Nye. Fibonacci numerical integration on a sphere. *Journal of Physics A: Mathematical and General*, 37(48):11591, 2004. 4

[17] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013. 3, 5

[18] Jingwu He, Linbo Wang, Wenzhe Zhou, Hongjie Zhang, Xiufen Cui, and Yanwen Guo. Viewpoint assessment and recommendation for photographing architectures. *IEEE transactions on visualization and computer graphics*, 25(8):2636–2649, 2018. 2

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[20] Jacek Jankowski and Martin Hachet. Advances in interaction with 3d environments. In *Computer Graphics Forum*, volume 34, pages 152–190. Wiley Online Library, 2015. 1, 2

[21] Niles Johnson. A visualization of the hopf fibration. https://nilesjohnson.net/hopf.html. 5

[22] Seong-heum Kim, Yu-Wing Tai, Joon-Young Lee, Jaesik Park, and In So Kweon. Category-specific salient view selection via deep convolutional neural networks. In *Computer*

Graphics Forum, volume 36, pages 313–328. Wiley Online Library, 2017. 2

[23] Michael Krainin, Brian Curless, and Dieter Fox. Autonomous generation of complete 3d object models using next best view manipulation planning. In *2011 IEEE international conference on robotics and automation*, pages 5031–5037. IEEE, 2011. 1, 2

[24] Chang Ha Lee, Amitabh Varshney, and David W Jacobs. Mesh saliency. In *ACM SIGGRAPH 2005 Papers*, pages 659–666. 2005. 2

[25] George Leifman, Elizabeth Shtrom, and Ayellet Tal. Surface regions of interest for viewpoint selection. *IEEE transactions on pattern analysis and machine intelligence*, 38(12):2544–2556, 2016. 2

[26] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 3

[27] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31, 2018. 3

[28] Tianqiang Liu, Jim McCann, Wilmot Li, and Thomas Funkhouser. Composition-aware scene optimization for product images. In *Computer Graphics Forum*, volume 34, pages 13–24. Wiley Online Library, 2015. 1, 2

[29] Mehdi Maboudi, MohammadReza Homaei, Soohwan Song, Shirin Malihi, Mohammad Saadatseresht, and Markus Gerke. A review on viewpoints and path-planning for uav-based 3d reconstruction. *arXiv preprint arXiv:2205.03716*, 2022. 1, 2

[30] Aaron Mavrinac and Xiang Chen. Modeling coverage in camera networks: A survey. *International journal of computer vision*, 101(1):205–226, 2013. 2

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[32] Zhixiang Min and Enrique Dunn. Voldor+ slam: For the times when feature-based or direct methods are not good enough. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13813–13819. IEEE, 2021. 1

[33] Zhixiang Min, Naji Khosravan, Zachary Bessinger, Manjunath Narayana, Sing Bing Kang, Enrique Dunn, and Ivaylo Boyadzhiev. Laser: Latent space rendering for 2d visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11122–11131, 2022. 1

[34] Zhixiang Min, Yiding Yang, and Enrique Dunn. Voldor: Visual odometry from log-logistic dense optical flow residuals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4898–4909, 2020. 1

[35] Zhixiang Min, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Enrique Dunn, and Manmohan Chandraker. Neurocs: Neural nocs supervision for monocular 3d object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21404–21414, 2023. 1

[36] Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. *arXiv preprint arXiv:2209.08546*, 2022. 2

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3, 4

[38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3

[39] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017. 3

[40] Mike Roberts, Debadeepta Dey, Anh Truong, Sudipta Sinha, Shital Shah, Ashish Kapoor, Pat Hanrahan, and Neel Joshi. Submodular trajectory optimization for aerial 3d scanning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5324–5333, 2017. 2

[41] Christophe Schlick. An inexpensive brdf model for physically-based rendering. In *Computer graphics forum*, volume 13, pages 233–246. Wiley Online Library, 1994. 3

[42] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European conference on computer vision*, pages 501–518. Springer, 2016. 2

[43] Adrian Secord, Jingwan Lu, Adam Finkelstein, Manish Singh, and Andrew Nealen. Perceptual models of viewpoint preference. *ACM Transactions on Graphics (TOG)*, 30(5):1–12, 2011. 2, 7

[44] Heung-Yeung Shum and Li-Wei He. Rendering with concentric mosaics. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 299–306, 1999. 3

[45] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, 2002. 3

[46] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017. 2, 3

[47] Rung-De Su, Zhe-Yo Liao, Li-Chi Chen, Ai-Ling Tung, and Yu-Shuen Wang. Imitating popular photos to select views for an indoor scene. In *Computer Graphics Forum*, volume 38, pages 141–148. Wiley Online Library, 2019. 2

[48] Yifan Sun, Qixing Huang, Dun-Yu Hsiao, Li Guan, and Gang Hua. Learning view selection for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14464–14473, 2021. 2

[49] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for

point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 3

[50] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Wolfgang Heidrich. Viewpoint selection using viewpoint entropy. In *VMV*, volume 1, pages 273–280. Citeseer, 2001. 2

[51] Wencheng Wang and Tianhao Gao. Constructing canonical regions for fast and effective view selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4114–4122, 2016. 2

[52] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 3

[53] Changhe Yang, Yanda Li, Can Liu, and Xiaoru Yuan. Deep learning-based viewpoint recommendation in volume visualization. *Journal of visualization*, 22(5):991–1003, 2019. 2

[54] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5287–5295, 2017. 2

[55] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1517, 2014. 2

[56] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. Dgcn: Diversified recommendation with graph convolutional networks. In *Proceedings of the Web Conference 2021*, pages 401–412, 2021. 3