GiganticNVS: Gigapixel Large-scale Neural Rendering with Implicit Meta-deformed Manifold

Guangyu Wang, Jinzhi Zhang, Kai Zhang, Ruqi Huang and Lu Fang§

Abstract—The rapid advances of high-performance sensation empowered gigapixel-level imaging/videography for large-scale scenes, yet the abundant details in gigapixel images were rarely valued in 3d reconstruction solutions. Bridging the gap between the sensation capacity and that of reconstruction requires to attack the large-baseline challenge imposed by the large-scale scenes, while utilizing the high-resolution details provided by the gigapixel images. This paper introduces GiganticNVS for gigapixel large-scale novel view synthesis (NVS). Existing NVS methods suffer from excessively blurred artifacts and fail on the full exploitation of image resolution, due to their inefficacy of recovering a faithful underlying geometry and the dependence on dense observations to accurately interpolate radiance. Our key insight is that, a highly-expressive implicit field with view-consistency is critical for synthesizing high-fidelity details from large-baseline observations. In light of this, we propose meta-deformed manifold, where *meta* refers to the locally defined surface manifold whose geometry and appearance are embedded into high-dimensional latent space. Technically, meta can be decoded as neural fields using an MLP (i.e., implicit representation). Upon this novel representation, multi-view geometric correspondence can be effectively enforced with featuremetric deformation and the reflectance field can be learned purely on the surface. Experimental results verify that the proposed method outperforms state-of-the-art methods both quantitatively and qualitatively, not only on the standard datasets containing complex real-world scenes with large baseline angles, but also on the challenging gigapixel-level ultra-large-scale benchmarks.

Index Terms—Novel View Synthesis, Neural Scene Representation, Sparse-view, Large-baseline, Gigapixel Rendering

1 Introduction

Obtaining unprecedented levels of realism and scale of real-world scenes has long been pursued in computer vision and graphics. Establishing such a goal requires efforts from the fronts of both sensation techniques and reconstruction algorithms. Unfortunately, the development of the two fronts are yet to be synchronized, yielding a significant gap in-between. For instance, the recent advance of gigapixel-level sensation [1] [2] [3] has opened a multitude of opportunities for large-scale 3D reconstruction with ultra-high resolution [4], whereas the current state-of-the-art reconstruction methods often fall short of making full advantage of the gigapixel-level imaging output.

In this paper, we investigate the problem of novel view synthesis (NVS) from a given set of calibrated images, with a special emphasis on ultra-large-scale scenes and gigapixel-level inputs. View synthesis is a classical problem in computer vision [5] [6] [7] and being approached extensively in recent years by deep learning methods. While recent advances in deep image-based rendering [8] [9] [10] [11] [12] and implicit neural scene representations [13] [14] [15] [16] [17] have made great progress, the problem of synthesizing ultra-large-scale scenes remains unsolved. The technical challenges lie in: 1) the baseline angle between consecutive viewpoints becomes large, or equivalently, the sparsity of input views increases, given the same sensation cost, i.e., the same amount of views is used for the coverage of the scene. 2) the large-scale scene is full of rich information while existing image resolution is far from enough to reflect the fine-grained structures. Therefore, considering high-resolution imagery is essential for the realism of synthesis. Unfortunately, directly extending the scalablity of existing solutions cannot handle the challenging large-scale high-resolution scenes.

Examining the large baseline challenge, it can be interpreted that the sparsity of input views causes dramatic performance degradation of state-of-the-art view synthesis methods. We attribute this to their incapacity of incentivizing precise scene geometry with cross-view consistency, while geometry obviously has a core impact on synthesis quality. For example, deep image-based rendering methods [8] [10] [18] warp and aggregate features from nearby views based on fixed or soft geometry proxy. Another line of solution [9] [19] [20] [21] augments dense geometry with neural descriptors and applies differentiable rendering techniques. When only sparse views are provided, contents in consecutive views may differ significantly and these methods fail to enforce a faithful geometry, resulting in blurry artifacts and poor view-dependent appearance.

On the other hand, recently trending neural volumetric representations [13] [11] and neural implicit surfaces [14] [15] encode scene geometry using respectively volume density and signed distance field, while simultaneously learn the radiance field to render realistic view-dependent appearance. As pointed out in [22], an arbitrary incorrect geometry can be well explained by a careful choice of radiance. To disentangle the geometry and appearance,

Guangyu Wang, Jinzhi Zhang, Kai Zhang and Lu Fang are with Dept. of Electronic Engineering and Beijing National Research Center for Information Science and Technology at Tsinghua University. Guangyu Wang, Jinzhi Zhang and Ruqi Huang are with Shenzhen International Graduate School, Tsinghua University.

§ : *Correspondence Author (fanglu@tsinghua.edu.cn, http://luvision.net)*

Acknowledgement: This work is supported in part by Natural Science Foundation of China (NSFC) under contract No. 62125106, 61860206003 and 62088102, in part by Ministry of Science and Technology of China under contract No. 2021ZD0109901, in part by Beijing National Research Center for Information Science and Technology (BNRist) under Grant No. BNR2020RC01002.

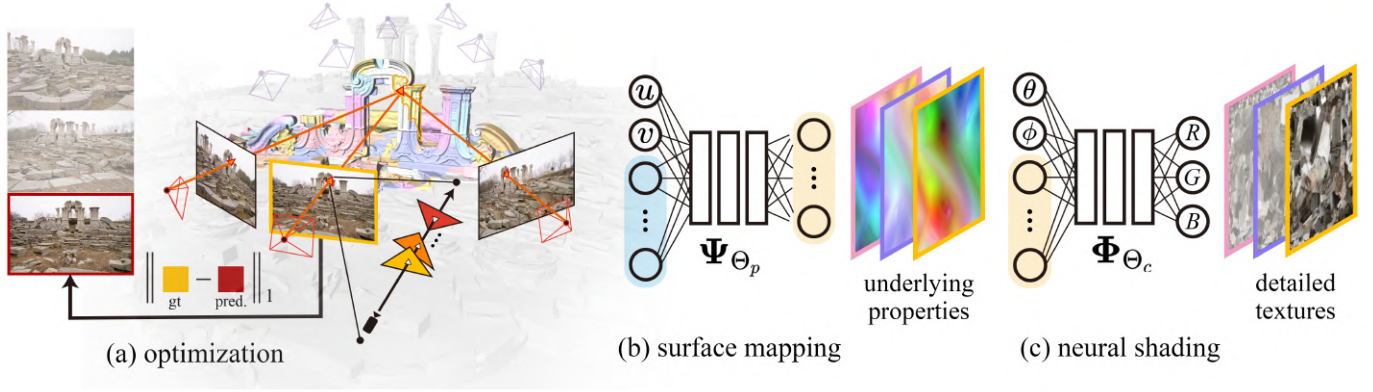


Fig. 1. An overview of the meta representation and neural rendering framework. Geometric and material properties are locally represented on a set of surface patches using an implicit mapping function modelled by multilayer perceptrons. Novel views can be synthesized by querying another shading function which implicitly accounts for illumination and reflectance. Optimization is done by matching the synthesized renderings with available observations through differentiable rendering.

the above methods rely on the inductive bias, i.e., enforcing the smoothness of the view-dependent function while keeping the expressivity of the spatial function by network structure design. Given dense input views as supervision, this will guarantee a reasonable geometry. However, when the input views become sparse but of ultra-high resolution, the implicit function is prone to overfitting and therefore, the inductive bias alone does not work well. In this case, the radiance field fails to generalize to unseen viewpoints and the underlying geometry severely distorts.

Aiming for bridging the gap between the sensation capacity and that of reconstruction, we introduce GiganticNVS for gigapixel large-scale novel view synthesis. To attack the large-baseline challenge imposed by the large-scale scenes, while utilizing the high-resolution details provided by the gigapixel inputs, a novel representation namely *meta-deformed manifold* is proposed. Here *meta* refers to the explicit surface patches augmented with high-dimensional latent vectors, encapsulating local information of both geometry and appearance. In contrast to previous neural scene representations, meta enables the learning of neural field on the well-defined surface manifold instead of in an unconstrained and redundant volume, thus benefiting from the inherent disentanglement for sparse observations and maintaining superior expressivity to represent a large-scale complex scene. Furthermore, detailed geometric correspondence can be implicitly reinforced with *meta deformation*, i.e. a featuremetric deformation with respect to latent vectors, which enables realistic synthesis of fine-grained contents.

Specifically, meta-deformed manifold is a novel implicit scene representation defined on a coarse triangulation, comprised of surface mapping, neural shading, and meta deformation. For each surface patch, the underlying properties such as surface normal and other BRDF parameters are continuously represented by a coordinate-based multi-layer perceptron network (MLP), taking as input a local latent vector. To make realistic colour predictions, the recovered surface properties are interpreted by another shading network which implicitly accounts for illumination and reflectance reasoning. Since precise geometry is crucial to boost the representational power of the implicit shading function, we further introduce meta deformation mechanism to re-map the initial geometry into an embedded latent space, where more detailed view correspondence can be implicitly incentivized. To render meta-deformed manifold into novel views, we leverage a differentiable rasterizer to sample the implicit field efficiently and smoothly. Notably, our meta-deformed manifold learns a set of implicit functions to continuously represent surface properties on mesh parameterizations, which can be easily deployed in any traditional graphics engines. Since the entire framework is fully differentiable, meta-deformed manifold can be optimized through back-propagation by comparing the synthesized renderings with given observations.

Our proposed method significantly outperforms state-of-the-art algorithms on various real-world challenging scenes given large-baseline observations, including GigaMVS [4], ETH3D [23], Tanks and Temples [24], and COLMAP [25] [26] dataset. Extensive experiments demonstrate that the proposed meta-deformed manifold is promising to enable gigantic novel view synthesis with unprecedented levels of realism and scale.

2 Related Work

2.1 Neural Scene Representations

Various scene representations for view synthesis have been developed and differ significantly in how the underlying scene geometry is used. Recent neural scene representations have triggered immense popularity in which deep neural networks are used as powerful tools to effectively represent scene geometry and appearance. Some methods leverage differentiable rendering to learn explicit geometry representation like voxels [27] [20], point clouds [9] [19], and meshes [21]. These approaches augment the explicitly defined geometry with learnable neural descriptors which are discretely defined on regular voxel grid, texture map, or irregular 3D points. A CNN-based neural renderer is followed to decode such descriptors into novel view RGB image. Although the inductive bias of CNNs prioritizes local interactions and enables highly detailed synthesis, the performance drastically drops when densely sampled imagery is not available, since these methods heavily rely on precise geometry and often fail to capture complex view dependence.

Implicit scene representations leverage coordinate-based MLP networks to continuously encode scene geometry, commonly in the form of signed distance fields [28] [29] [15] [16] or occupancy fields [30] [31]. Scene Representation Networks (SRNs) [32] continuously encode local scene properties by mapping coordinates to a feature representation and corresponding RGB colour. A differentiable ray marching algorithm is proposed to locate the ray's intersection with scene surface, in which the core is a differentiable rendering function implemented as a recurrent neural network. Niemeyer et al. [33] represents scene surface as an occupancy field and uses a numerical approach to locate the intersection after tracing pixel rays into 3D space, where the derivative is calculated exactly through implicit differentiation. Another implicit texture field is introduced which takes as inputs the intersection position and outputs the corresponding diffuse texture. However, these works lack the efficacy to represent detailed scene contents and are biased towards low frequency components, resulting in undesirable over-smoothness.

NeRF [13] and follow-up works [34] [22] achieve quite appealing synthesis quality by representing scene geometry as continuous volumetric density field and modelling view-dependent appearance as emitted radiance field. Rendering is done by sampling multiple points along the camera ray and accumulate per-point colour using volume rendering integral. NSVF [35] proposes to define the implicit field on explicit voxel grids organized in a sparse voxel octree to make the function more aware of local properties and subsequently prune redundant voxels to accelerate rendering. Mip-NeRF [36] is another extension which represents scenes at continuously-valued scales and renders conical frustums instead of a single ray per pixel to effectively reduce aliasing and enhance details. Verbin et al. [37] further structures view-dependent radiance using reflected radiance and implicitly reasons spatially-varying BRDF properties, substantially improving the realism of specular reflections. However, these volumetric representations are only optimized on the volume integral of radiance field instead of enforcing the single intersection with the well-defined surface, leading to biased and inconsistent geometry. In this sense, the representational power of the radiance field severely suffers since the global function onerously fits the redundant volume space and a more complex view-dependent function is learned in compensation for the biased geometry. This also indicates that a sufficiently dense coverage of the scene and a large amount of samples along the rays are always required to achieve high quality rendering.

Other recently popular works learn neural implicit surfaces via differentiable sphere-tracing [14] [38] or volume rendering [39] [40] [41], enabling faithful surface reconstruction and view synthesis. Neural lumigraph rendering (NLR) [42] utilizes high-capacity representation with periodic activations to jointly optimize the implicit surface and radiance field, and enables real-time rendering with a rasterized renderer. However, finding the accurate surface intersection to boost the rendering network is typically hard since the geometry and appearance are entangled to a large extent. IDR [14] proposes disentanglement between geometry and appearance, yet the redundantly distributed signed distance field easily converges to over-smoothed surface predictions and subsequently hinders the expressivity of radiance function. Therefore, blur is inevitably induced in renderings and these methods are limited to relatively small-scale scenes with simple structures.

Arguably most related to our scene representation are deep surface light field (DSLF) [43] and surface radiance field (SurRF) [44]. DSLF [43] for the first time proposes to parameterize surface light field using a MLP network, which predicts the view-dependent residual colour pre-partitioned from the small-baseline image samples. SurRF [44] is a continuous neural representation for unsupervised dense multi-view stereo, comprised of both explicit surface deformation and implicit radiance field, defined on a dense set of triangle facets. Our method is similar to SurRF, as it also learns latent vectors on local surfaces jointly with the implicit fields. However, our representation is defined on parameterized surface patches rather than individual triangle facet, leading to superior continuity and compactness while maintaining comparable expressivity of local properties with conformal surface mapping. Besides, we propose to incentivize implicit correspondence in latent space and our shading mechanism is more capable of modelling complex view-dependence with high frequency details.

Given various limitations, current neural scene representations does not naturally lend themselves to large-scale real-world scenarios with sparse-view imagery. Instead, our meta-deformed manifold representation inherits the merits of both explicit geometry and implicit fields, by learning the implicit fields strictly on the explicitly defined surface manifold and effectively reinforcing multi-view correspondence to allow better allocation of network capacity. Our method can faithfully represent scenes with complex view-dependent details given only sparse observations.

2.2 Image-based Rendering

The computer vision and graphics community have long been interested in image-based rendering (IBR) techniques. There exists a natural tradeoff between 1) how many input images are needed and 2) how much is known about the scene geometry [5]. Classical works [45] [7] [46] focusing on light field and view interpolation requires ultra dense images with small baseline angles, since novel views are directly synthesized from input samples without geometric reasoning. Recently, deep learning methods have been incorporated into traditional view synthesis pipelines and achieves appealing performance. Methods like [47] [48] [49] [17] [12] leverage deep learning to tackle light field rendering. However, dense observations are still needed in these works, thus making it difficult to generalize to large-scale complex scenes. Another line of solution [50] [8] [10] [18]leverage estimated coarse geometry proxy to warp and aggregate source-view contents or deep features into novel views, enabling superior realism and flexibility. However, when only sparse observations are available, contents in consecutive views differ considerably and the coarse geometry can not lend itself to detailed optimization, resulting in blurry artifacts and degraded view-dependent appearance.

2.3 Sparse-view Rendering

Choi et al. [50] utilizes depth uncertainty and image priors to enable visually pleasing synthesis from two or three narrow-baseline inputs. Sun et al. [51] proposes Sparse-IBRNet, which fully exploits sparse observations to perform depth completion and subsequently aggregates source view features using Bi-ConvLSTM. For implicit scene representations, one approach for generalizing to the sparse-view setting is to leverage priors by pre-training radiance fields conditioned on multi-view features [52] [53] [54] [55] [11] [56]. PixelNeRF [52] utilizes 2D features extracted from source images, while MVSNeRF [54] warps image features to construct 3D cost volume which is later processed by 3D CNNs. However, these methods require laborious pre-training on multi-view dataset with various different scenes and the view synthesis quality is prone to degrade when testing on data that differs significantly. Therefore, several works address sparse-view rendering in a different way without pre-training by focusing more on the scene geometry. NeRS [57] learns an implicit representation of a closed surface that is diffeomorphic to a sphere, and on-surface BRDFs are learned to model view-dependent appearance in a physically grounded manner where view-dependent appearance is factorized into illumination, diffuse albedo, and specular shininess. This combination of a surface constraint and a factorized appearance allows NeRS to learn from sparse observations in the wild. However, strong assumptions about surface shape, material, and illumination are made, which can not extend to large-scale scenes under complex environmental lighting. RegNeRF [58] proposes to regularize the geometry and appearance of patches rendered from unobserved viewpoints. Annealing ray sampling space is also demonstrated to be useful in preventing divergent behavior at the start of optimization. Our method shares similar insights with RegNeRF in that the geometry is the major concern for sparse-view rendering. However, RegNeRF still fails to enforce a faithful geometry and suffers from excessively blurred artifacts due to the entangled nature and volume rendering integral, which can not be substantially addressed by regularization.

3 Meta Representation

As shown in Fig. 1, our pipeline takes a set of gigapixel-level images from sparse views as input, and trains a model synthesizing images from arbitrary unseen views. In particular, we propose meta representation, which enables a regular, explicit and interpretable geometric encoding in the network and allows efficient optimization through an auto-decoder framework (Fig. 1(b)). Furthermore, we formulate an image formation model tailored to the novel representation, which incorporates the intermediate geometry into the final rendering (Fig. 1(c)).

3.1 Definition

As mentioned in Section 1, the current state-of-the-art view synthesis methods typically represent geometry and appearance in an entangled manner, which in turn leads to dependence on dense input views for good approximations to both perspectives within a redundant volume space. To this end, it is critical to represent geometry and appearance in a disentangled way. We thus design a novel geometric representation for NVS namely *meta*, which enjoys the following properties:

- **Locality**: it encodes the scene geometry locally, alleviating the difficulty of representing large-scale complex scenes;
- **Regularity**: it admits a regular image-like structure, allowing simple feature operation;
- **Differentiability**: it fits naturally to modern gradient-based optimization techniques, offering efficient estimation of the underlying geometry.

We now give the detailed description of our design. First, as illustrated in Fig. 2, given a scene of interest $\mathcal{M}$, we first reconstruct the surface via multi-view stereo method and partition the mesh into a set of local surface patches $\{\mathcal{S}_i\}$. Then we conduct surface parametrization on each surface patch $\mathcal{S}$, i.e., finding a piecewise continuous function $f : \mathbb{R}^3 \mapsto \mathbb{R}^2$ that assigns 2D parameterized (u, v) coordinate for each 3D point on $\mathcal{S}$. Through the parametrization mapping f, we can encode various geometric attributes of $\mathcal{S}$ in terms of the constructed UV-coordinate systems (one for each patch), such as surface normal, displacement, albedo, roughness, metallic and so on. In this paper, we consider particularly surface normal and BRDF parameters, which are highly relevant to the shading process.

We now convert the above local parametrized geometry to a more compact neural representation, which naturally fits to an auto-decoder framework for the final optimization. For each surface patch $\mathcal{S}$, we denote by $\mathcal{A}_s$ the respective $UV-$coordinates, which is further normalized within the unit square. We define $\mathcal{Z}_s = \{\mathcal{Z}_s^{00}, \mathcal{Z}_s^{01}, \mathcal{Z}_s^{10}, \mathcal{Z}_s^{11}\}$ as a set of learnable high-dimensional latent vectors on the four corners of $\mathcal{A}_s$. For a local parametric coordinate $\hat{\boldsymbol{x}} = (u, v) \in \mathcal{A}_s$, we use a bi-linear interpolation function ψ to obtain the corresponding latent code $\boldsymbol{z}_s$:

$$\boldsymbol{z}_s(\hat{\boldsymbol{x}}) = \psi(\hat{\boldsymbol{x}}; \mathcal{Z}_s) \tag{1}$$

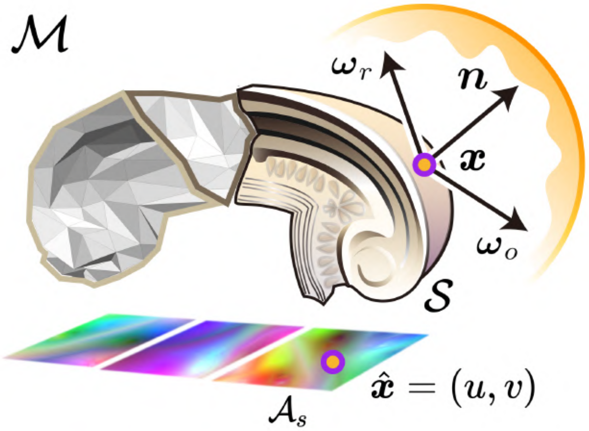


Fig. 2. Illustration of our meta representation. The scene $\mathcal{M}$ is composed of a set of local surface patches, where each surface patch $\mathcal{S}$ is parameterized as a local UV-map $\mathcal{A}_s$. Here, $\boldsymbol{x}$ represents the 3D coordinate of a point on surface patch $\mathcal{S}$, $\hat{\boldsymbol{x}}$ is the corresponding local parametric coordinate in the UV-map $\mathcal{A}_s$, and $\boldsymbol{n}$ is the respective normal vector. We denote by $\boldsymbol{\omega}_o$ the outgoing view direction and $\boldsymbol{\omega}_r$ the reflection of $\boldsymbol{\omega}_o$ about $\boldsymbol{n}$.

In particular, we emphasize that the regular structure of $\mathcal{A}_s$ enables the above simple operation, which is invalid under irregular representations such as point cloud and mesh.

Finally, we leverage coordinate-based multilayer perceptron neural network (MLP), $\boldsymbol{\Psi}_{\Theta_p}$, to decode the geometric attributes from $\boldsymbol{z}_s$:

$$\mathcal{P}(\hat{\boldsymbol{x}}, \mathcal{S}) = \boldsymbol{\Psi}_{\Theta_p}(\boldsymbol{z}_s) = \boldsymbol{\Psi}_{\Theta_p}(\psi(\hat{\boldsymbol{x}}; \mathcal{Z}_s)), \tag{2}$$

where $\mathcal{P}(\hat{\boldsymbol{x}}, \mathcal{S})$ is the selected geometric attributes on point $\hat{\boldsymbol{x}}$ in $\mathcal{A}_s$.

In other words, the geometry attributes at a point $\boldsymbol{x} \in \mathbb{R}^3$ are decoded from its parametrized coordinates $\hat{\boldsymbol{x}} = (u, v)$ and the associated latent code $\mathcal{Z}_s$. With the proposed meta representation, we can conduct the learning of radiance (or reflectance) strictly onto the explicitly defined manifold by optimizing the surface properties, leading to a better allocation of network capacity, thus stabilizing training and enabling expressive representations of the implicit function for detailed contents.

Representation Instantiation. One remaining detail yet to describe is how to construct the local surface patches $\{\mathcal{S}_i\}$ – in practice we use off-the-shelf multi-view stereo and surface parameterization techniques provided in Agisoft Metashape [59] without any data-specific fine-tuning. Due to the sparsity of input views, the reconstruction is not necessarily accurate. In Section 4.1, we propose a meta deformation module to robustify our representation with respect to the error-prone geometry initialization.

Representation Continuity. Last but not least, it is critical to ensure the continuity of a patch-based representation. We achieve this by adding a regularization term in the training loss, which enforces the optimized normal on each patch to be close to the initial reconstructed surface normal. Note that this term encourages the optimized normals to be consistent along the patch boundaries. We defer the details to Section 4.2.

3.2 Shading

We have formulated the meta representation in Section 3.1 and proposed a surface mapping network $\boldsymbol{\Psi}_{\Theta_p}$ that predicts spatially-varying geometric attributes including diffuse colour $\boldsymbol{d} \in \mathbb{R}^3$, normal vector $\boldsymbol{n} \in \mathbb{R}^3$, specular albedo $\boldsymbol{b}_s \in \mathbb{R}^3$, together with a high-dimensional spatial feature $\boldsymbol{\kappa} \in \mathbb{R}^F$ encoding the material information, i.e.,

$$\mathcal{P} = (\boldsymbol{n}, \boldsymbol{d}, \boldsymbol{b}_s, \boldsymbol{\kappa}). \tag{3}$$

In this section, we derive a shading formula with respect to the geometric information in $\mathcal{P}$, as well as the ray, which reveals the interplay among the geometric attributes in rendering, and therefore guides our design of neural networks for mimicking the overall rendering procedure.

We assume the surface does not emit light and follows rendering equation [60] and to be given a $(\boldsymbol{x}, \boldsymbol{n})$, a shading point in $\mathbb{R}^3$ and the respective normal. Let $\mathcal{L}_i(\boldsymbol{x}, \boldsymbol{\omega}) \in \mathbb{R}^3$ be the illumination intensity at $\boldsymbol{x}$ along direction $\boldsymbol{\omega}$. We denote by $\boldsymbol{\omega}_o \in \mathbb{R}^3$ a unit vector point from $\boldsymbol{x}$ to the camera center and by $\boldsymbol{r}$ the ray direction opposite to $\boldsymbol{\omega}_o$, i.e., $\boldsymbol{r} = -\boldsymbol{\omega}_o$. The observed view-dependent colour $\mathcal{L}_o(\boldsymbol{x}, \boldsymbol{\omega}_o)$ is calculated as:

$$\mathcal{L}_o(\boldsymbol{x}, \boldsymbol{\omega}_o) = \int_\Omega \mathcal{L}_i(\boldsymbol{x}, \boldsymbol{\omega}) \mathcal{F}(\boldsymbol{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega})(\boldsymbol{\omega} \cdot \boldsymbol{n}) d\boldsymbol{\omega}, \tag{4}$$

where Ω is the upper hemisphere $\Omega = \{\boldsymbol{\omega} : \boldsymbol{\omega} \cdot \boldsymbol{n} > 0\}$, and $\mathcal{F}(\boldsymbol{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}_i)$ is the BRDF.

Note that the BRDF consists of a diffuse and a specular term:

$$\mathcal{F}(\boldsymbol{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}) = \mathcal{F}_d(\boldsymbol{x}) + \mathcal{F}_s(\boldsymbol{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega}). \tag{5}$$

We then decompose the left-hand side of Eqn. (4) into view-independent diffuse colour $\boldsymbol{d}$ and view-dependent specular reflectance $\boldsymbol{s}$:

$$\mathcal{L}_o(\boldsymbol{x}, \boldsymbol{\omega}_o) = \boldsymbol{d} + \boldsymbol{s}, \tag{6}$$

where

$$\begin{aligned} \boldsymbol{d} &= \textstyle\int_\Omega \mathcal{L}_i(\boldsymbol{x}, \boldsymbol{\omega}) \mathcal{F}_d(\boldsymbol{x})(\boldsymbol{\omega} \cdot \boldsymbol{n}) d\boldsymbol{\omega}, \\ \boldsymbol{s} &= \textstyle\int_\Omega \mathcal{L}_i(\boldsymbol{x}, \boldsymbol{\omega}) \mathcal{F}_s(\boldsymbol{x}, \boldsymbol{\omega}_o, \boldsymbol{\omega})(\boldsymbol{\omega} \cdot \boldsymbol{n}) d\boldsymbol{\omega}. \end{aligned} \tag{7}$$

Note that the diffuse colour $\boldsymbol{d}$, being view-independent, is already predicted in $\mathcal{P}$, thus it remains to estimate the view-dependent specular reflectance $\boldsymbol{s}$. Following the previous work [61] [62] [63] [37], we ignore self-occlusions and interreflections, neglect exposure variation and tone-mapping, and pre-integrate the illumination $\mathcal{L}_i(\boldsymbol{x}, \boldsymbol{\omega})$ to approximate as:

$$\boldsymbol{s} \approx \boldsymbol{b}_s \odot \mathcal{L}_s^*(\boldsymbol{x}, \boldsymbol{\omega}_r), \tag{8}$$

where $\boldsymbol{b}_s$ is the specular albedo and $\mathcal{L}_s^*(\boldsymbol{x}, \boldsymbol{\omega}_r)$ is the pre-integrated illumination for the specular component, depending on the mirrored view direction $\boldsymbol{\omega}_r$, which is the reflection of the view direction $\boldsymbol{\omega}_o$ about the surface normal $\boldsymbol{n}$, i.e., $\boldsymbol{\omega}_r = 2(\boldsymbol{\omega}_o \cdot \boldsymbol{n})\boldsymbol{n} - \boldsymbol{\omega}_o$.

Again, the specular albedo $\boldsymbol{b}_s$ is predicted in $\mathcal{P}$. To approximate the view-dependent term $\mathcal{L}_s^*(\boldsymbol{x}, \boldsymbol{\omega}_r)$, we design an illumination network $\widehat{\mathbf{L}}_{\Theta_l}$, which is an MLP consuming the mirrored view direction $\boldsymbol{\omega}_r$ and the spatial feature $\boldsymbol{\kappa}$ in place of shading point $\boldsymbol{x}$:

$$\mathcal{L}_s^* = \widehat{\mathbf{L}}_{\Theta_l}(\boldsymbol{\omega}_r, \boldsymbol{\kappa}), \text{where } \widehat{\mathbf{L}}_{\Theta_l} : \mathbb{R}^3 \times \mathbb{R}^F \mapsto \mathbb{R}^3 \tag{9}$$

Putting pieces together, the total colour originally defined in Eqn. (4) can be estimated based on neural network queries:

$$\mathcal{L}_o(\boldsymbol{x}, \boldsymbol{\omega}_o) = \boldsymbol{d} + \boldsymbol{b}_s \odot \widehat{\mathbf{L}}_{\Theta_l}(\boldsymbol{\omega}_r, \boldsymbol{\kappa}). \tag{10}$$

4 Neural Rendering Framework

4.1 Meta-deformed Manifold

The meta representation and the shading model introduced in Section 3 already constitute a view synthesis pipeline. Specifically, combining Eqn. (2), (3), (10), we can calculate the colour $\boldsymbol{c}$ of a parameterized point $\hat{\boldsymbol{x}}$ along ray direction $\boldsymbol{r}$ as:

$$\boldsymbol{c} = \boldsymbol{d} + \boldsymbol{b}_s \odot \widehat{\mathbf{L}}_{\Theta_l}(\boldsymbol{\omega}_r, \boldsymbol{\kappa}), \tag{11}$$

where

$$\begin{cases} (\boldsymbol{n}, \boldsymbol{d}, \boldsymbol{b}_s, \boldsymbol{\kappa}) = \boldsymbol{\Psi}_{\Theta_p}(\psi(\hat{\boldsymbol{x}}; \mathcal{Z}_s)), \\ \boldsymbol{\omega}_r = 2(\boldsymbol{\omega}_o \cdot \boldsymbol{n})\boldsymbol{n} - \boldsymbol{\omega}_o, \\ \boldsymbol{\omega}_o = -\boldsymbol{r}. \end{cases} \tag{12}$$

In other words, we can train a network using multi-view consistency across the input images and therefore obtain a view synthesis model.

However, as mentioned in Section 3.1, the naive implementation fails to recover the high-frequency details in gigapixel-level images (see also Fig. 7). We argue that this discrepancy comes from the inaccurate initial geometry.

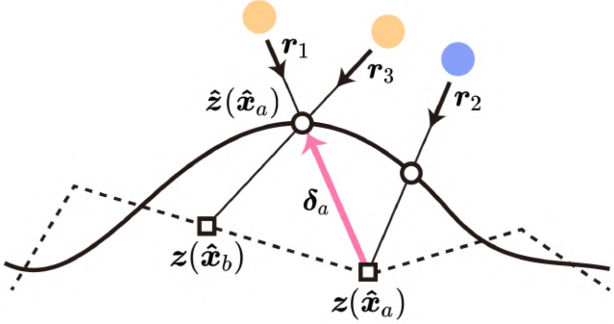

$\boldsymbol{r}_1$
$\boldsymbol{r}_3$
$\boldsymbol{r}_2$
$\hat{\boldsymbol{z}}(\hat{\boldsymbol{x}}_a)$
$\boldsymbol{\delta}_a$
$\boldsymbol{z}(\hat{\boldsymbol{x}}_b)$
$\boldsymbol{z}(\hat{\boldsymbol{x}}_a)$

Fig. 3. Illustration of the meta deformation mechanism. Here, $\hat{\boldsymbol{x}}_a$ and $\hat{\boldsymbol{x}}_b$ are the local parameterized coordinates of two shading points, $\boldsymbol{z}(\hat{\boldsymbol{x}}_a)$ and $\boldsymbol{z}(\hat{\boldsymbol{x}}_b)$ are the corresponding latent embedding. We denote by $\boldsymbol{r}_i$ the camera ray direction, $\boldsymbol{\delta}_a$ the predicted latent offset at $\hat{\boldsymbol{x}}_a$ along $\boldsymbol{r}_1$, and $\hat{\boldsymbol{z}}(\cdot)$ the meta-deformed embedding.

As shown in Fig. 3, we consider a toy example of one-dimension manifold: the solid curve represents the underlying geometry, and the dashed segments represents the inaccurate initial reconstruction. We first consider the rays $\boldsymbol{r}_1, \boldsymbol{r}_2$, which hit the curve at points of distinctive colours. Due to the inaccurate geometry estimation, they hit the same point on the reconstruction, therefore are represented by the same local parameterized point $\hat{\boldsymbol{x}}_a$. Thus, naively optimizing the meta representation results in contradicting labels, yellow and purple, at $\hat{\boldsymbol{x}}_a$. On the other hand, the rays $\boldsymbol{r}_1, \boldsymbol{r}_3$ hit the same point on the underlying curve and therefore are both yellow. However their intersections with respect to the dashed reconstruction are distinctive, leading to colour dispersion, i.e., the segment between $\hat{\boldsymbol{x}}_a, \hat{\boldsymbol{x}}_b$ are deemed to be yellow.

To this end, we introduce meta-deformed manifold, which remaps the initial geometry into an embedded latent space to implicitly incentivize detailed correspondence. For instance, taking $\boldsymbol{r}_1$ and the corresponding parameterized point $\hat{\boldsymbol{x}}_a$, our goal is to find an offset $\boldsymbol{\delta}_a$ in the latent space, such that $\boldsymbol{z}(\hat{\boldsymbol{x}}_a)$, the latent code

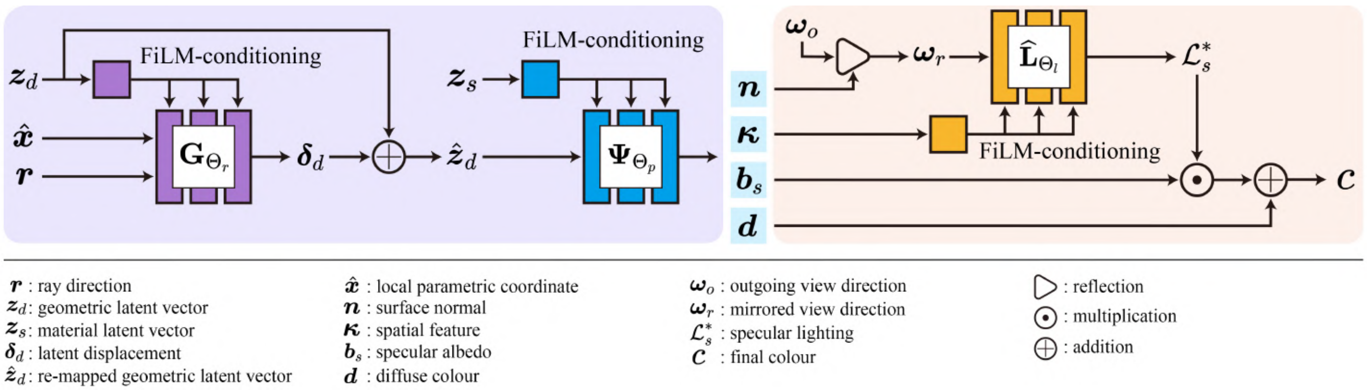


Fig. 4. The network architecture of our proposed framework. Featuremetric correspondence is reinforced by predicting the latent-space deformation through an MLP, whose inputs are the geometric latent vector, parametric coordinate and ray direction. Then, scene properties are continuously represented by a surface mapping MLP. Another illumination MLP is followed to implicitly accounts for lighting and reflectance reasoning.

of $\hat{x}_a$ is closer to the one representing the intersection between r_1 and the underlying curve. We denote by $\hat{z}(\hat{x}_a) = z(\hat{x}_a) + \delta_a$ the meta-deformed embedding with respect to r_1.

Formally, meta-deformed manifold is parameterized as a MLP network $\mathbf{G}_{\Theta_r}$ to compensate for the imperfect geometry by implicitly modelling the micro geometric displacement δ_d in the latent space. Similar to the definition of $\mathcal{Z}_s$ in Eqn. (1), each $UV-$coordinate system $\mathcal{A}_s$ for surface patch $\mathcal{S}$ is augmented with another geometric latent grid $\mathcal{Z}_d = \{\mathcal{Z}_d^{00}, \mathcal{Z}_d^{01}, \mathcal{Z}_d^{10}, \mathcal{Z}_d^{11}\}$. Given an $(u,v)-$coordinate $\hat{x} \in \mathcal{A}_s$ and a camera ray direction $r \in \mathbb{R}^3$, we

$$\begin{aligned} \delta_d &= \mathbf{G}_{\Theta_r}(\hat{x}, r, z_d), \\ \hat{z}_d &= z_d + \delta_d. \end{aligned} \tag{13}$$

Here $\psi(\cdot)$ denotes a bi-linear interpolation and $z_d = \psi(\hat{x}; \mathcal{Z}_d)$ is the interpolated latent code at $\hat{x}$.

Unlike traditional displacement mapping and SurRF [44] which represent the displacement as a 1D scalar along the normal direction or view direction, we propose to embed the per-ray geometry into high-dimensional latent space to perform an implicit deformation, which shows superior performance since the feature space offers more versatile reparameterizations, potentially allowing for more detailed featuremetric correspondence.

In the end, we augment the capacity of the surface mapping network Ψ_{Θ_p} originally defined in Eqn. (2) by the following modification:

$$\mathcal{P} = \Psi_{\Theta_p}(\hat{z}_d, z_s). \tag{14}$$

That is, the surface mapping network now takes two geometric latent vectors z_s and $\hat{z}_d$.

We finally remark that, though Eqn. (13) and (14) suggest the view-dependency of surface mapping Ψ_{Θ_p}, the optimization procedure actually drives the meta-deformed embedding $\hat{z}_d$ to be multi-view consistent, which shares similar insights with the view-dependent surface deformation in SurRF [44] and 3D reconstruction methods based on per-view depth map [26] [64] [65]. An intuition of this can be drawn from Fig. 3. Given the local parameterized point $\hat{x}_a$ on the inaccurate reconstruction (i.e. the dashed segments), the colour observations of $\hat{x}_a$ from multiple views (e.g. r_1 and r_2) are prone to severe distinction. Although the latent offset δ_a can be different when varying the ray direction from r_1 to r_2, the meta-deformed embedding $\hat{z}_d$ always converges to the latent location indicating the real underlying surface (i.e. the solid curve), which best explains different observations (e.g. r_1 and r_3) with cross-view colour consistency. Therefore, the surface mapping network Ψ_{Θ_p} defined in Eqn. (14) focuses on the modelling of geometric-related attributes of the underlying view-consistent surface represented by $\hat{z}_d$, maintaining the disentangled nature of our representation after optimization.

4.2 Network Design

Network Architecture: Equipped with the meta-deformed manifold representation, we now finalize our network design for novel view synthesis. As shown in Fig. 4, our network consists of two main blocks: The geometry block shaded in purple includes the meta deformation network $\mathbf{G}_{\Theta_r}$ and surface mapping network Ψ_{Θ_p}, which effectively enforces multi-view correspondence and decodes spatially-varying geometric attributes. The shading block in yellow performs in a physically-based manner by interpreting the decoded attributes and consuming the view direction to implicitly model the specular reflectance. Note that we use sinusoidal activations [66] and pass the latent vectors z_d, z_s, and spatial feature κ into neural networks through FiLM-conditioning [67] [68] [69], which feeds the latent vector not as part of the network input but rather as an affine transformation to the intermediate features, so as to enhance the capability of representing high-frequency details.

Loss Functions: We then formulate the loss function and optimization schemes used in the proposed neural network, which enable high-resolution and photo-realistic rendering.

Similar to other neural rendering frameworks, the L1 loss between the rendered and ground-truth pixel colour is used:

$$l_c = \sum_{k \in \mathcal{V}} \sum_{r \in \mathcal{R}_k} \left\| \hat{c}_k(r) - I_k(r) \right\|_1. \tag{15}$$

where $\mathcal{V}$ is the set of randomly selected camera views and $\mathcal{R}_k$ is the random batch of pixel rays selected from view k.

We further propose to guide the learning of geometric attributes by constraining the normal prediction of the surface mapping network Ψ_{Θ_p}, using the coarse normal computed from the input triangle mesh:

$$l_n = \sum_{k \in \mathcal{V}} \sum_{r \in \mathcal{R}_k} \left\| n_k(r) - N_k(r) \right\|_2^2. \tag{16}$$

where $n_k(r)$ denotes the normal prediction of the spatial point at which the pixel ray r traced from camera k intersects the nearest triangle, and $N_k(r)$ is the normal vector derived from the input geometry.

The final loss is composed of the colour loss and normal constraint:

$$l = l_c + \lambda_n l_n. \tag{17}$$

where λ_n denotes a scalar balance weight.

4.3 Image Formation Model

The task of view synthesis can be framed as learning a composed mapping $\mathbf{F}_\Theta$, formulated as:

$$\begin{aligned} &\mathbf{F}_\Theta : \mathbb{R}^{Z_d} \times \mathbb{R}^{Z_s} \times \mathbb{R}^3 \mapsto \mathbb{R}^3, \\ &\boldsymbol{c} = \mathbf{F}_\Theta(\psi(\hat{\boldsymbol{x}}; \mathcal{Z}_d), \psi(\hat{\boldsymbol{x}}; \mathcal{Z}_s), \boldsymbol{r}). \end{aligned} \tag{18}$$

The proposed meta-deformed manifold continuously represents the local scene properties on the surface. To render and optimize such a neural representation, a sampling mechanism is required. Similar to [44] [70], we rely on current differentiable rendering framework to rasterize the scene surface using z-buffer and aggregate nearby predictions to produce a smooth scene rendering.

In this way, view synthesis is achieved by tracing the pixel rays from the camera center and querying the neural networks on the intersected points. For each pixel ray $\boldsymbol{r}$ traced from camera k, the rendered colour $\hat{\boldsymbol{c}}_k(\boldsymbol{r})$ is computed as the mean value over a collection of rasterized colour predictions:

$$\hat{\boldsymbol{c}}_k(\boldsymbol{r}) = \frac{\sum_{i \in \mathcal{D}_k^r} \mathbf{F}_\Theta(\psi(\hat{\boldsymbol{x}}_i; \mathcal{Z}_d), \psi(\hat{\boldsymbol{x}}_i; \mathcal{Z}_s), \boldsymbol{r})}{\sum_{i \in \mathcal{D}_k^r} \mathbf{1}}, \tag{19}$$

where $\hat{\boldsymbol{x}}_i$ is the local parametric coordinate of the ray's intersection on mesh triangle indexed as i, $\mathcal{D}_k^r$ denotes the set of triangle indices selected from z-buffer. Note that only the first $\mathcal{N}_f$ triangles nearest to the camera center is selected and triangles that far from the first intersection in depth value are removed from the selection $\mathcal{D}_k^r$. We empirically find that the number of samples $\mathcal{N}_f$ only matters when the resolution of rendered image is much higher than that of the initial triangulation.

5 Experiment

In this section, we first provide the implementation details in Section 5.1, and then we demonstrate the experimental results on the Tanks and Temples [24] dataset (Section 5.2) and on the High-resolution Benchmarks (Section 5.3). Finally, to verify the necessity of each component in our pipeline, we conduct extensive ablation studies in Section 5.4. Overall, our method shows significant superiority, both qualitatively and quantitatively, on real-world challenging scenes given only sparse observations.

5.1 Implementation Details

In all the experiments, the meta deformation network $\mathbf{G}_{\Theta_r}$ consists of 4 fully-connected layers with 512 hidden units, the surface mapping network $\mathbf{\Psi}_{\Theta_p}$ consists of 7 fully-connected layers with 512 hidden units, while the illumination network $\widehat{\mathbf{L}}_{\Theta_l}$ is a smaller network containing 2 fully-connected layers with 256 hidden units. All the intermediate linear layers in these networks are accompanied with FiLM-conditioning [67] [68] [69] and sinusoidal activation function. The dimensionality of geometric latent codes $\boldsymbol{z}_s \in \mathbb{R}^{Z_s}, \boldsymbol{z}_d \in \mathbb{R}^{Z_d}$ is set to $Z_s = Z_d = 64$. The dimension of the spatial feature vector $\boldsymbol{\kappa} \in \mathbb{R}^F$ from the surface mapping network $\mathbf{\Psi}_{\Theta_p}$ is $F = 512$. We separately optimize our neural representation for each scene. At each training iteration, we first randomly select a set of camera views and then sample a random batch of pixel rays for each selected view. The number of randomly selected views $\|\mathcal{V}\|$ is set to 4 or 8 depending on the total number of training views available. The batch size of random pixel rays is $\|\mathcal{R}_k\| = 8192$. We use the Adam optimizer [71] with an initial learning rate of 2.0×10^{-4} decreased by a factor of 0.9 for every 100 epochs. Our algorithm typically converges after 5k training epochs. The sample number of rasterizer is $\mathcal{N}_f = 3$ for gigapixel rendering and $\mathcal{N}_f = 1$ otherwise. The scalar weight λ_n is empirically set to 1.0. For the geometry initialization, we utilize off-the-shelf multi-view stereo and surface parameterization techniques provided in Agisoft Metashape [59]. We use a single Nvidia RTX3090 GPU to train and evaluate our method.

5.2 Tanks and Temples Dataset

We compare against the current state-of-the-art methods, including Mip-NeRF [36], RegNeRF [58], NPBG [9], and IBRNet [11], on seven real-world complex scenes from the intermediate set of the Tanks and Temples dataset [24]. For each scene, a set of images are selected as training views or source images, and a disjoint set of target views are held out for evaluation. Following [72], we further select a small proportion of all available training views by consecutively sampling a view from every sparsity $= n$ camera index, i.e., $\{1, n+1, 2n+1, ...\}$. Two different sparse-view settings – sparsity $= 3, 5$ – are investigated and the corresponding quantitative results are reported in Table 1. For scene-specific methods including Mip-NeRF [36], RegNeRF [58], and our method, per-scene optimization is performed. For NPBG [9], we fine-tune the pretrained CNN-renderer but optimize the neural point descriptor from scratch on each scene. For IBRNet [11], we separately perform fine-tuning on each scene based on the pretrained model. All methods are quantitatively assessed with respect to the widely used PSNR/SSIM (higher is better), and LPIPS [73] (lower is better). Note that we run multi-view stereo strictly using the same sparsity to obtain the input geometry for NPBG and our method. We apply the same foreground mask obtained from dense multi-view stereo before evaluating the metrics, as done in [35] [56]. Obviously, our pipeline demonstrates top performance compared to all baseline methods under sparse observations. Notably, NeRF-based methods like [36] [58] obviously fail when only sparse observations are available.

The corresponding qualitative results when sparsity is 3 are shown in Fig. 5. Note that NPBG [9] often fails to reproduce view-dependent colour with high-fidelity, such as the shadow in Francis (scene 2, Fig. 5 (a)) and Lighthouse (scene 4, Fig. 5 (a)), and the specularity in Horse (scene 3, Fig. 5 (a)) and Train (scene 7, Fig. 5 (a)). The results of IBRNet [11] in Fig. 5 (b) exhibit blurry artifacts and lack detailed contents due to the underconstrained geometry and sparse inputs. Whereas our method is able to capture fine-grained geometric details with superior realism in view-dependence, as shown in Fig. 5 (c). Close-ups reveal our meta-deformed manifold leads to more faithful reconstruction of high-frequency details, such as the fine-grained texture on the base of the sculpture in Francis (scene 2, Fig. 5 (c)), and the delicate wrinkles, sharp characters and edges appear in M60 (scene 5, Fig. 5 (c)). Moreover, our representation enables superior realism in view-dependent appearance, especially for regions where complex specular reflections occur, e.g. the specularity on Horse (scene 3, Fig. 5 (c)), and the roof of Lighthouse (scene 4, Fig. 5 (c)).

TABLE 1
Quantitative results of state-of-the-art view synthesis algorithms on the Tanks and Temples [24] dataset. Our method, meta-deformed manifold, outperforms state-of-the-art algorithms in all evaluation metrics.

Sparsity	Scene	Mip-NeRF [36]			RegNeRF [58]			NPBG [9]			IBRNet [11]			**Ours**		
		PSNR ↑	SSIM ↑	LPIPS↓	PSNR ↑	SSIM ↑	LPIPS↓	PSNR ↑	SSIM ↑	LPIPS↓	PSNR ↑	SSIM ↑	LPIPS↓	PSNR ↑	SSIM ↑	LPIPS↓
3	Family	20.57	0.799	0.235	21.03	0.802	0.244	24.76	0.818	0.093	28.03	0.909	0.111	**29.95**	**0.926**	**0.068**
	Francis	25.35	0.899	0.103	25.33	0.890	0.125	28.19	0.901	0.054	31.41	0.944	0.065	**31.98**	**0.952**	**0.044**
	Horse	23.13	0.929	0.066	23.00	0.922	0.088	27.18	0.933	0.041	31.07	0.967	0.034	**31.91**	**0.969**	**0.029**
	M60	16.57	0.664	0.413	16.64	0.670	0.424	21.18	0.735	0.208	23.36	0.830	0.195	**25.96**	**0.863**	**0.159**
	Lighthouse	17.21	0.658	0.425	17.43	0.647	0.412	18.75	0.658	0.214	20.29	0.704	0.276	**22.71**	**0.792**	**0.175**
	Panther	19.46	0.698	0.431	18.91	0.707	0.438	23.72	0.764	0.193	25.04	0.835	0.185	**27.71**	**0.879**	**0.168**
	Train	16.61	0.661	0.470	16.31	0.648	0.455	19.94	0.711	0.181	21.63	0.777	0.209	**23.07**	**0.836**	**0.145**
	Mean	19.84	0.758	0.306	19.80	0.755	0.312	23.39	0.789	0.141	25.83	0.852	0.154	**27.61**	**0.888**	**0.112**
5	Family	17.35	0.773	0.265	18.09	0.782	0.266	24.35	0.812	**0.116**	23.54	0.857	0.192	**25.02**	**0.872**	0.118
	Francis	22.23	0.881	0.128	21.72	0.874	0.139	27.66	0.898	0.066	25.57	0.908	0.115	**30.23**	**0.935**	**0.059**
	Horse	21.40	0.916	0.085	21.64	0.917	0.088	25.74	0.922	0.048	22.87	0.927	0.076	**30.32**	**0.960**	**0.041**
	M60	14.36	0.631	0.427	14.50	0.646	0.433	20.23	0.702	0.287	16.75	0.702	0.389	**22.77**	**0.803**	**0.202**
	Lighthouse	14.81	0.615	0.412	15.52	0.637	0.425	17.87	0.643	0.285	16.88	0.655	0.359	**20.74**	**0.736**	**0.227**
	Panther	16.55	0.665	0.465	16.92	0.684	0.453	22.53	0.743	0.277	18.96	0.738	0.385	**25.00**	**0.820**	**0.208**
	Train	14.96	0.634	0.460	14.77	0.621	0.444	19.21	0.689	0.244	16.00	0.637	0.432	**21.31**	**0.788**	**0.192**
	Mean	17.38	0.730	0.320	17.59	0.737	0.321	22.51	0.773	0.189	20.08	0.775	0.278	**25.06**	**0.844**	**0.150**

5.3 High-resolution Benchmark

We now show experimental results on more challenging scenarios, i.e. ultra-high-resolution view synthesis of large-scale complex scenes given only sparse observations. We first experiment on three challenging scenes with approximately 6K image resolution, including South Building provided by COLMAP [25] [26], and Courtyard and Facade from ETH3D [23] dataset. Furthermore, we perform evaluation on five challenging scenes from GigaMVS [4] dataset with gigapixel-level resolution, including Great Fountain, Haiyan Hall, Weilun Building, Museum and Library. We use the same way as described in section 5.2 to train Mip-NeRF [36], RegNeRF [58], NPBG [9], IBRNet [11], and our method. We additionally compare with SVS [10], a scene-agnostic method pretrained on the Tanks and Temples [24] dataset.

Quantitative results are reported in Table 2 and the qualitative visual comparisons are shown in Fig 6. and Fig. 8. Our method significantly outperforms prior state-of-the-art methods in various real-world challenging scenes, both quantitatively and qualitatively. As shown in Table 2, Mip-NeRF [36] and RegNeRF [58] produce severely blurry and distorted synthesis in all cases, probably due to their poorly estimated geometry with significant errors entangled in the radiance function, failing to generalize to large-scale 360-degree unbounded scenes with large-baseline inputs. SVS [10] only succeeds when the target contents are well observed in nearby source views, while generates excessive blur or cracked artifacts when baseline angles between selected source views are sufficiently large, as shown in Fig. 6 (a) and Fig. 8 (a). IBRNet [11] similarly suffers from ghosting or blurry artifacts and often fails to reproduce high-frequency details, since significant errors are induced into the entangled volumetric geometry. On the contrary, NPBG [9] excels at handling large-baseline imagery, and reasonable synthesis can always be achieved in all challenging scenarios. However, due to its heavy dependence on the input point cloud, the synthesis resolution of fine details is quite limited and noisy artifacts are often induced. In addition, NPBG also struggles to reconstruct complex view-dependent effects. Different from prior works, our method enables robust, high-resolution view synthesis with fine-grained details and photo-realistic specularity, e.g. the high-resolution structures of bricks in South Building (scene 1, Fig. 6), Courtyard (scene 2, Fig. 6), and Facade (scene 3, Fig. 6), the complicated textures of the grass and stones in Great Fountain (scene 1, Fig. 8) and Haiyan Hall (scene 2, Fig. 8), and also the high-fidelity specular reflections that appear on the glass of South Building and Courtyard.

In general, synthesizing fine-grained appearance details and reproducing complex view dependence is quite challenging, particularly with large-baseline sparse observations. Meta-deformed manifold successfully tackles this issue by learning implicit fields on local surface manifolds and incentivizing featuremetric correspondence to enhance geometric details, thus effectively boosting the expressivity of reflectance model and leading to prominently detailed renderings with realistic colour.

5.4 Ablation Studies

In this section, ablation experiments are provided to analyze and validate the efficacy of several key components in our meta-deformed manifold representation.

Meta Deformation. We first study the effect of meta deformation $\mathbf{G}_{\Theta_r}$ via ablations. Experiments are performed on 6 scenes with complex details, including Family, M60, Lighthouse, Panther, Train in Tanks and Temples [24] dataset, and Great Fountain in GigaMVS [4] dataset. The input sparsity here is set to 3 and other experiment settings are strictly the same as described in Section 5.2. As shown in Table 3, with meta deformation, the quantitative metrics of PSNR and SSIM constantly improves and the LPIPS perceptual loss drops correspondingly. A visual inspection of the efficacy of meta deformation is provided in Fig. 7. Notably, an absence of meta deformation results in over-smoothed results lack of high-frequency details. The reason is that only with precise geometric correspondence can the network capacity be well allocated. Otherwise, the observed colour for any imprecise off-surface points is indeed the colour from nearby surfaces, along different ray directions (see Fig. 3). In such cases, the shading network has to onerously fit a much more complex view-dependent function rather than the real underlying BRDF to compensate for the imprecise geometry.

Reflectance Modelling. Fig. 9 shows a qualitative validation on the separation of diffuse and specular colour by our meta-deformed manifold. Experiments are performed on Horse of the Tanks and Temples dataset and Haiyan Hall from the GigaMVS dataset. Horse is comprised of highly specular surface under complex environmental illumination, whereas Haiyan Hall is

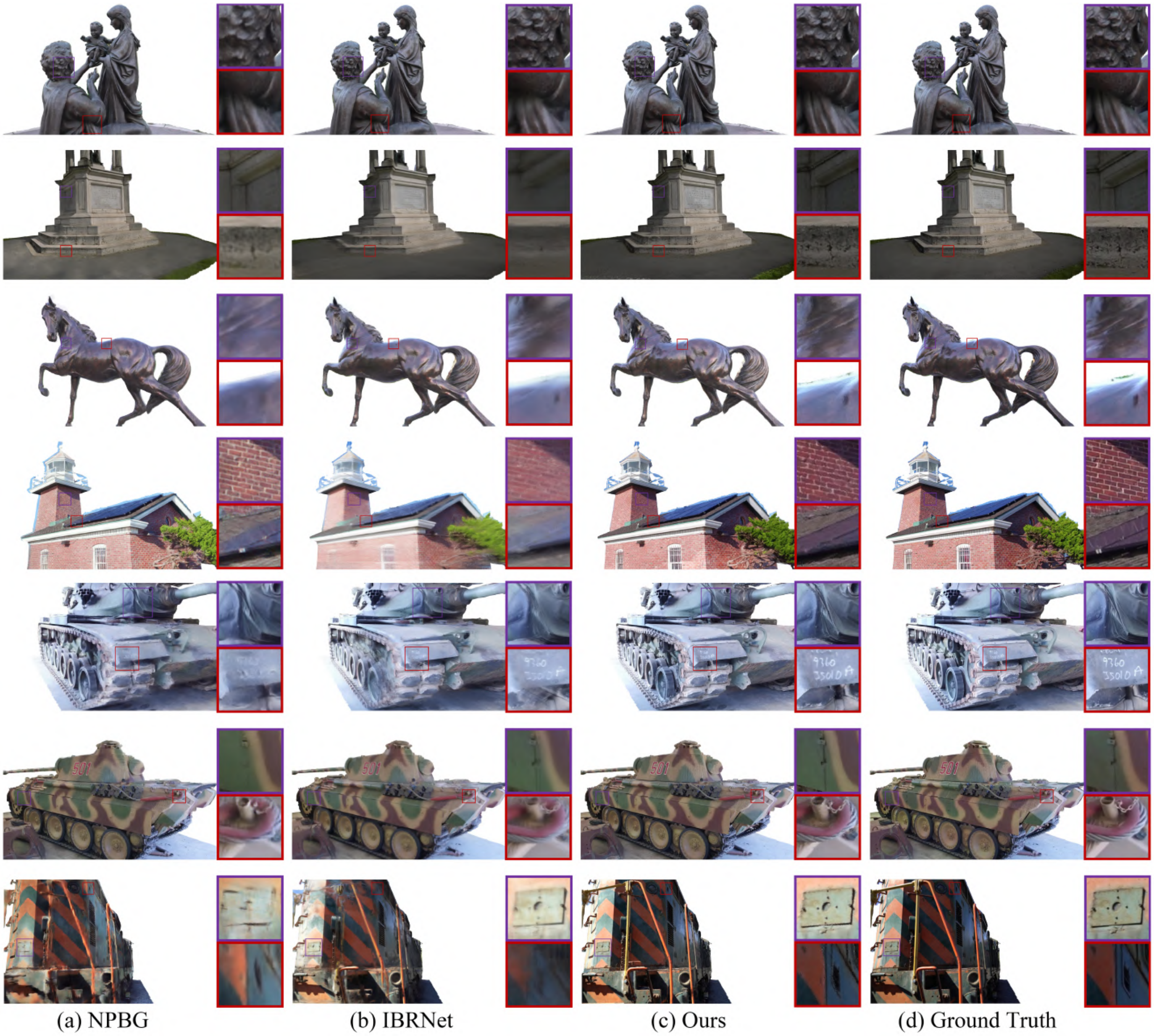
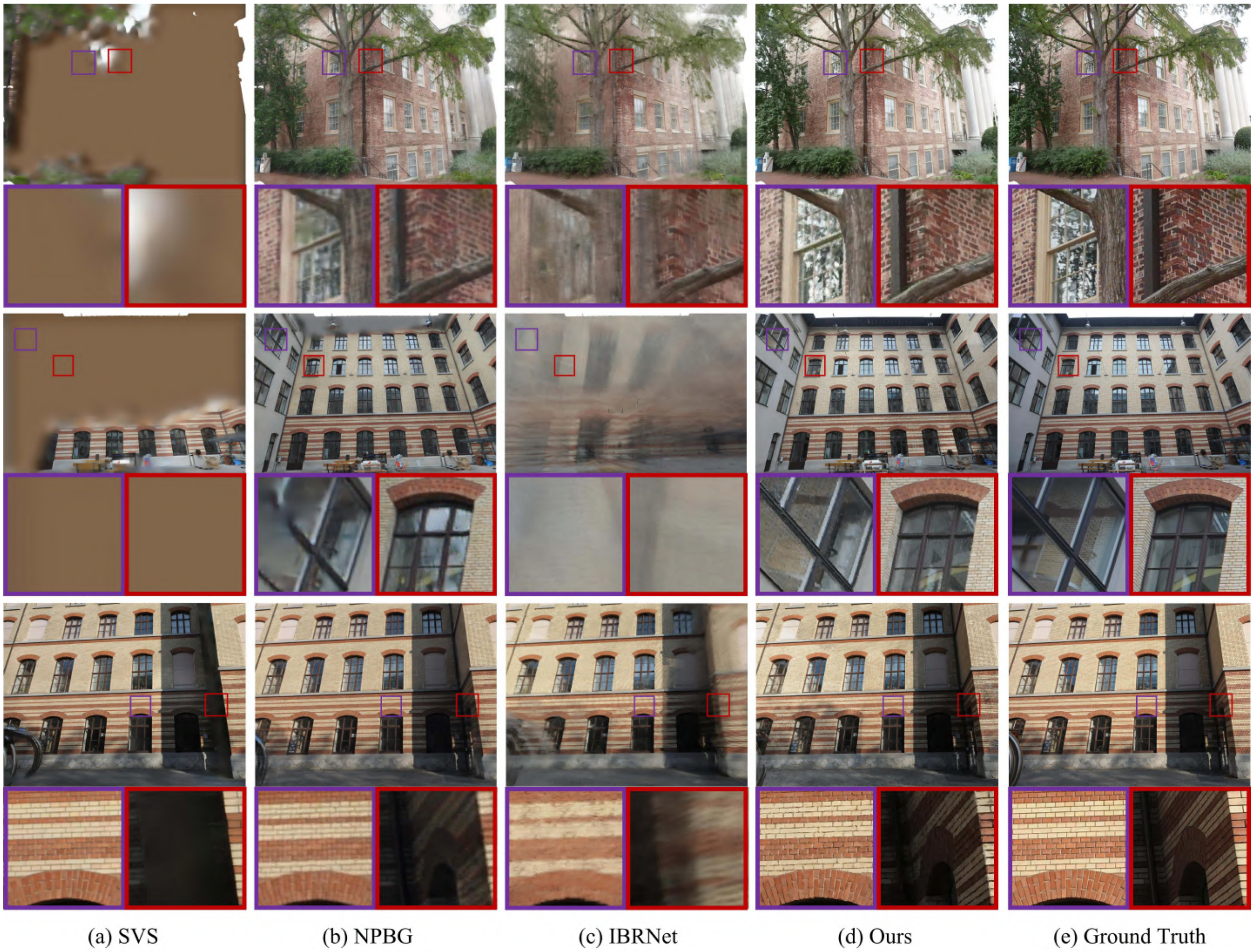


Fig. 5. Visual comparisons on the intermediate set of Tanks and Temples dataset [24] when input sparsity is 3. Our method captures fine-grained details with superior realism in view-dependent appearance, compared with NPBG [9] and IBRNet [11].

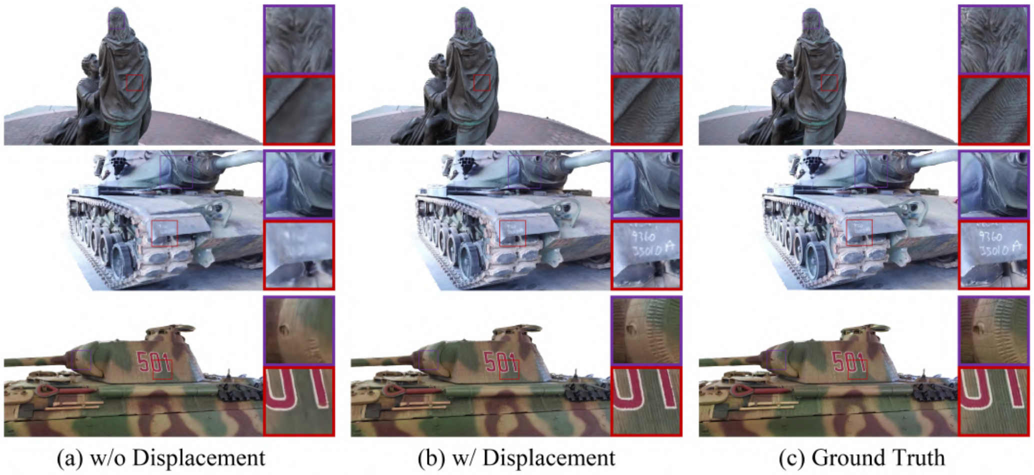
TABLE 2
Quantitative results on large-scale challenging scenes selected from COLMAP [25] [26], ETH3D [23], and GigaMVS [4] dataset. Our meta-deformed surface achieves state-of-the-art performance among all prior works given sparse-view high-resolution imagery.

Scene	Mip-NeRF [36]			Reg-NeRF [58]			SVS [10]			NPBG [9]			IBRNet [11]			**Ours**		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM ↑	LPIPS↓	PSNR ↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Great Fountain	21.81	0.587	0.428	15.40	0.457	0.579	21.21	0.623	0.287	18.94	0.516	0.250	21.72	0.631	**0.223**	**23.47**	**0.669**	0.246
Haiyan Hall	20.92	0.528	0.487	20.51	0.499	0.554	17.98	0.541	0.392	18.19	0.463	0.371	19.46	0.562	0.323	**21.09**	**0.577**	**0.312**
Library	18.74	0.602	0.491	17.52	0.593	0.528	22.72	0.817	**0.188**	18.72	0.785	0.213	23.82	0.809	0.235	**26.69**	**0.824**	0.229
Weilun Building	20.98	0.816	0.205	20.27	0.780	0.343	22.61	0.851	0.135	22.33	0.850	**0.124**	23.46	0.843	0.155	**27.41**	**0.881**	0.126
Museum	20.13	0.569	0.476	19.57	0.543	0.482	13.33	0.608	0.505	20.69	0.746	0.182	24.31	**0.845**	0.193	**25.37**	0.843	**0.181**
Courtyard	15.56	0.588	0.614	14.94	0.573	0.688	13.74	0.605	0.758	17.47	0.574	0.354	13.40	0.572	0.731	**19.88**	**0.700**	**0.228**
Facade	17.15	0.475	0.418	17.10	0.464	0.493	17.46	0.612	0.279	15.85	0.408	0.369	17.05	0.484	0.344	**19.15**	**0.664**	**0.221**
South Building	11.88	0.419	0.749	16.24	0.464	0.703	16.49	0.545	0.463	19.74	0.511	0.321	18.74	0.588	0.393	**20.40**	**0.626**	**0.211**



(a) SVS (b) NPBG (c) IBRNet (d) Ours (e) Ground Truth

Fig. 6. Novel view synthesis of South Building, Courtyard, and Facade from COLMAP [25] [26] and ETH3D [23] dataset, compared with SVS [10], NPBG [9], IBRNet [11]. Our method synthesizes much more detailed and realistic renderings with complex view-dependent appearance.



(a) w/o Displacement (b) w/ Displacement (c) Ground Truth

Fig. 7. Qualitative comparisons of ablations on meta deformation. Meta-deformed manifold significantly improves high-frequency details by capturing fine-grained featuremetric deformation.

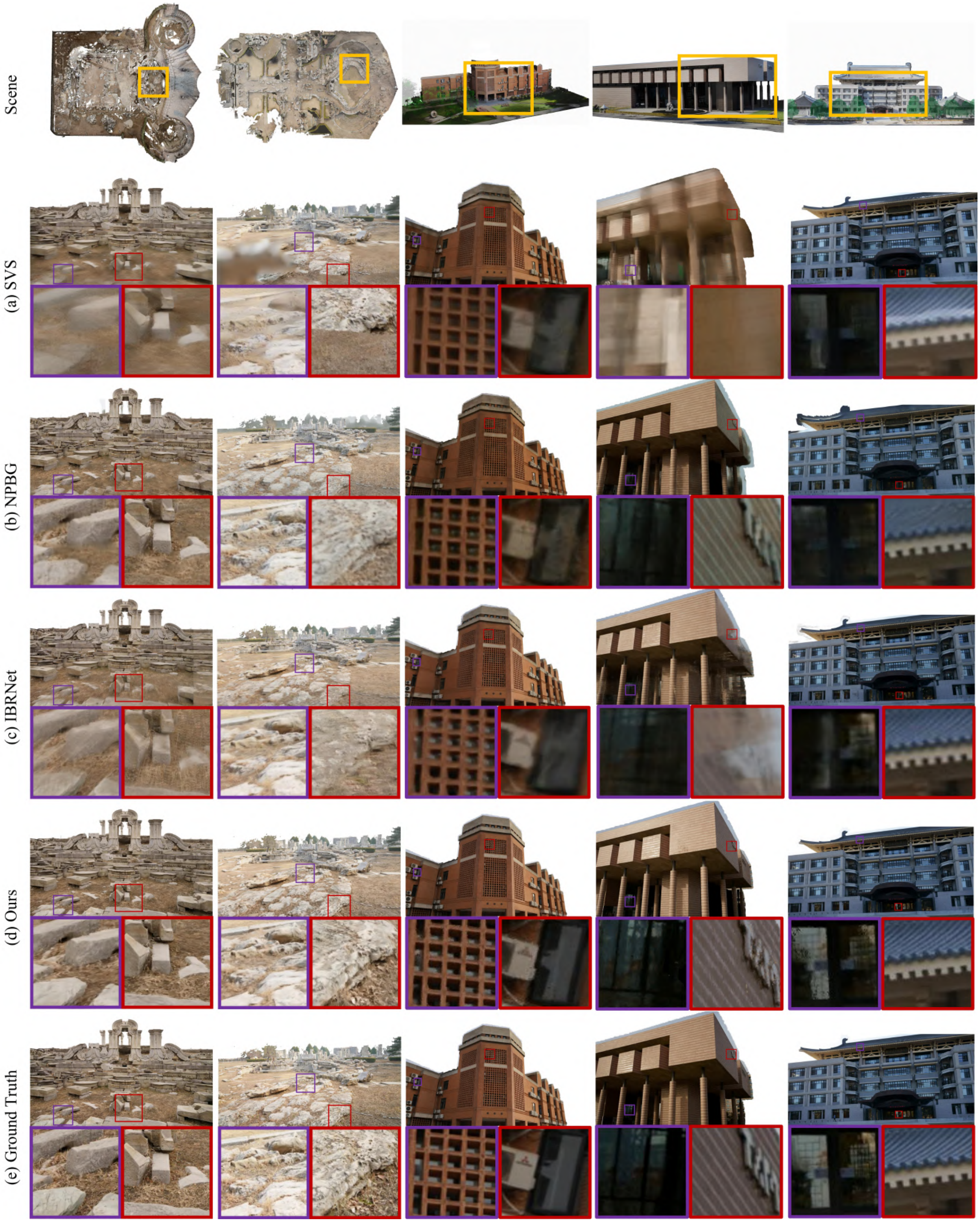


Fig. 8. Novel view synthesis of Great Fountain, Haiyan Hall, Weilun Building, Museum, and Library from GigaMVS [4] dataset, compared with SVS [10], NPBG [9], IBRNet [11]. Our method synthesizes detailed structures with high-fidelity.

TABLE 3
Ablations on meta deformation.

Scene	w/o displacement			w/ displacement		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Family	29.58	0.924	0.096	**29.95**	**0.926**	**0.068**
M60	25.61	0.849	0.191	**25.96**	**0.863**	**0.159**
Lighthouse	22.53	0.761	0.239	**22.71**	**0.792**	**0.175**
Panther	27.49	0.866	0.199	**27.71**	**0.879**	**0.168**
Train	22.75	0.805	0.206	**23.07**	**0.836**	**0.145**
Great Fountain	23.15	0.643	0.320	**23.47**	**0.669**	**0.246**

mainly made up of diffuse components. Recall that our final colour prediction $\boldsymbol{c}$ is decomposed into a view-independent diffuse component $\boldsymbol{d}$ and a view-dependent specular component $\boldsymbol{s}$, i.e. $\boldsymbol{c} = \boldsymbol{d} + \boldsymbol{s}$. Our implicit shading formulation approximates the rendering equation in Eqn. (4), where the diffuse colour $\boldsymbol{d}$ is a multiplication of the spatially-varying diffuse albedo and a diffuse lighting term: $\boldsymbol{d} = \mathcal{F}_d(\boldsymbol{x}) \odot \int_\Omega \mathcal{L}_i(\boldsymbol{x}, \boldsymbol{\omega})(\boldsymbol{\omega} \cdot \boldsymbol{n}) d\boldsymbol{\omega}$. Therefore, the appearance shown in Fig. 9 (b) is an overall reflection of both diffuse albedo and the incident energy measured by the interaction between illumination and surface normal. Note that we only rely on the inductive bias of neural networks to seek a reasonable but not exactly correct decomposition, where the ultimate goal is to boost the representational power by forcing the surface mapping network to learn the highly complex spatially-varying surface properties, while making the smaller illumination network to focus on the modelling of a generally smoother function with respect to view direction.

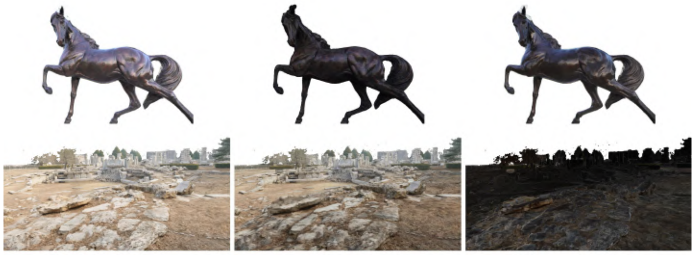

(a) Final Colour (b) Diffuse Colour (c) Specular Colour

Fig. 9. Decomposition of (a) final rendered colour into (b) diffuse colour and (c) specular colour. Our method separates diffuse and specular components reasonably for real-world scenes under complex environmental illumination.

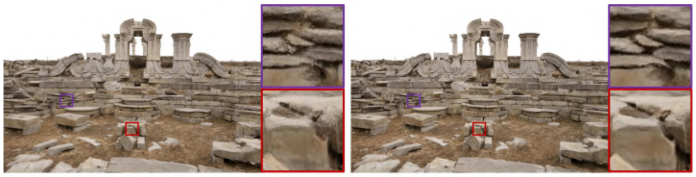

(a) Fixed Normal (b) Normal Guidance

Fig. 10. Effect of normal guidance. Using fixed surface normal hinders the learning of reflectance and consequently induces noisy artifacts.

Normal Guidance. We now provide ablations on the normal guidance in the optimization process. An alternative baseline here is implemented by using fixed surface normal to reflect the outgoing view direction, which is then used as input for the illumination network $\widehat{\mathbf{L}}_{\Theta_l}$. The difference is that the surface normal is directly estimated from the input geometry and fixed during training, instead of being predicted by the surface mapping network $\boldsymbol{\Psi}_{\Theta_p}$. As shown in Fig. 10, using fixed coarse normal as input to the illumination network induces noisy artifacts in the rendering results. Intuitively, the imprecise geometry recovered from multi-view stereo and surface reconstruction techniques should only be considered as a coarse proxy rather than a reliable ground truth, and the goal is to guide the learning of surface mapping by weak supervision and regularize the patch-based representation for better continuity.

Robustness to the Initialization. To measure the dependence of our representation on the initial geometry, we perform ablation experiments by varying the resolution of input geometry. In all of these experiments, we only focus on the effect of input geometry, so a fixed set of input imagery is used for optimizing our neural representation, which is the same as Section 5.2 when sparsity is 3. Two different ways are investigated to affect the input geometry, one is to vary the input sparsity when performing multi-view stereo, the other is to down-sample the surface reconstruction using mesh decimation [74]. The experiments are evaluated on Panther from the Tanks and Temples [24] dataset.

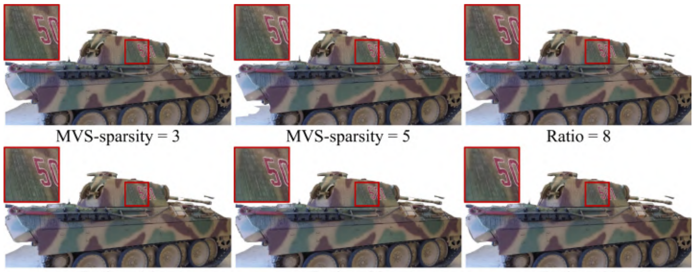

MVS-sparsity = 3 MVS-sparsity = 5 Ratio = 8
Ratio = 16 Ratio = 32 Ratio = 64

Fig. 11. Qualitative comparisons with respect to the resolution of input geometry.

Table 4 shows the quantitative results under different resolutions of input geometry, where MVS-sparsity denotes the sparsity of input images when performing multi-view stereo to obtain the initial geometry, and Ratio denotes the down-sample rate of mesh decimation. The experimental setup for the first line is the same as that in Section 5.2 when sparsity is 3. Corresponding qualitative comparisons are shown in Fig. 11. When MVS-sparsity is increased from 3 to 5, the reconstructed mesh maintains completeness while the precision severely suffers. In this case, the performance of our method remains stable and fine-grained details could still be faithfully synthesized, since the meta deformation mechanism effectively reinforces correspondence and implicitly compensates for the degraded geometry. For the mesh decimation ratio of 8 and 16, our method still works well and enables detailed renderings. Whereas the detailed appearance is smoothed out for the extreme decimation ratio of 64, the prediction is still reasonable in low-frequency contents. In the future work, we would consider optimizing the vertex position and topology of the initial triangulation so as to achieve a better coordination with meta deformation. In conclusion, our method generally shows good robustness with respect to the resolutions of input geometry, which demonstrates the compactness and efficacy of our scene representation.

Effect of Latent Dimensionality. Since meta-deformed manifold continuously represent surface properties using locally conditioned latent vectors, it gains inherent advantages in terms of compactness. In this part, we provide further demonstration on the compactness of our representation by reducing the dimension of latent vectors. The experiments are performed on Great Fountain of the GigaMVS dataset, and the quantitative and qualitative evaluations are shown in Table 5 and Fig. 12, respectively. The

TABLE 4
Quantitative results with respect to different geometric resolutions.

MVS-sparsity	Ratio	No. of Faces	Mem. of Mesh	PSNR↑	SSIM↑	LPIPS↓
3	-	1568524	187MB	**27.71**	**0.879**	0.168
5	-	1548804	184MB	27.59	0.871	0.168
3	8	196064	22MB	27.66	0.878	**0.157**
3	16	98032	11MB	27.63	0.873	0.169
3	32	49016	6MB	27.54	0.870	0.176
3	64	24508	3MB	27.43	0.872	0.179

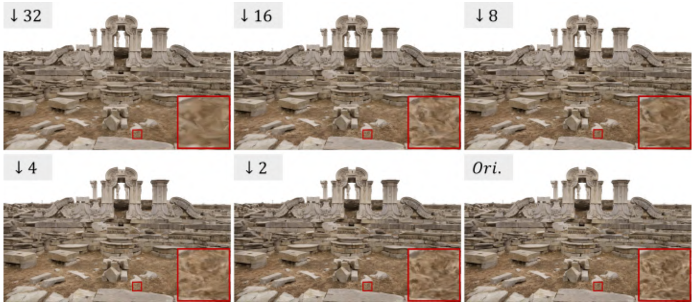


Fig. 12. Qualitative evaluations under different compress ratios for the dimensionality of latent vectors.

Ratio in Table 5 denotes the reduction rate of dimension for both of the geometric latent vector z_d and the textural latent vector z_s. Despite the reduction on latent dimension, neither the visual quality nor evaluation metrics drop significantly.

TABLE 5
Quantitative evaluations with respect to different latent dimensionality.

Ratio	PSNR↑	SSIM↑	LPIPS↓
1	23.47	0.669	0.246
2	23.38	0.662	0.270
4	23.40	0.654	0.284
8	23.24	0.648	0.290
16	22.91	0.638	0.299
32	22.74	0.613	0.340

TABLE 6
Comparison of time and memory cost with competing methods.

Method		Time Efficiency		GPU Mem.
		Per-scene Optimization	Rendering	
Scene-specific Optimization	Ours	26.63 h	3.04 s	16.4 GB
	MipNeRF [36]	6.03 h	55.10 s	21.8 GB
	RegNeRF [58]	4.65 h	117.85 s	21.8 GB
Scene-agnostic Optimization	SVS [10]	-	4.85 s	10.6 GB
	IBRNet [11]	5.45 h	314.53 s	5.6 GB
	NPBG [9]	**0.19 h**	**0.36 s**	**3.7 GB**

5.5 Limitation

Though our method demonstrates state-of-the-art view synthesis performance, we identify the following limitations: time efficiency, memory cost, and deterioration for extreme sparse-view scenarios.

We first report comparisons on time and memory cost of the competing methods in Table 6. The experiments are conducted on the scene Panther from the Tanks and Temples dataset. For per-scene optimization, we follow the same experimental setup as Section 5.2 with sparsity being 3, and the optimization is halted once the performance is saturated. The rendering efficiency and memory cost are measured by evaluating on a single 1080×2048 image. All the methods are evaluated using the official implementation.

Following the neural field approaches like [13] [36] [58], we exhaustively queries the implicit functions and requires per-scene optimization to faithfully represent scene-specific information, which puts our method at disadvantage on time efficiency when compared to methods based on the CNN architecture and large-scale pretraining [9] [8] [10] [19]. Noting that our method is naively implemented in PyTorch, we plan to incorporate the recent acceleration techniques [75] [76] to improve the efficiency. On the other hand, compared with volume rendering based implicit methods, our method demonstrates superior rendering efficiency, since the meta-deformed manifold is defined on explicit surface patches which can be efficiently rasterized, so that only very few points (1 or 3) are evaluated by the implicit function for a pixel ray. Regarding memory occupation, our method consumes lots of memory due to the usage of large MLP networks and learnable high-dimensional latent vectors, which may be alleviated by pruning techniques [77].

Finally, our method does not generalize to extreme sparse-view settings where a complete initial mesh can not be obtained by off-the-shelf multi-view stereo and surface reconstruction methods. Incorporating depth completion [51] or MVS methods robust to the sparse-MVS setting [78] [72] into our pipeline can be an interesting direction for the future work, which potentially tackles more extreme large-baseline sparse-view settings, e.g. synthesis from as few as 2 or 3 views with limited overlap. The background content is not recovered in our method, which is considered to be quite difficult when only given wide-baseline sparse observations.

6 Conclusion

In this paper, we propose a novel representation, dubbed meta-deformed manifold, to synthesize novel views of real-world complex scenes from sparse, gigapixel-level observations. By continuously representing informative properties on local surface manifolds and incentivizing featuremetric multi-view correspondence, meta-deformed manifold fully exploits the expressivity of neural implicit functions to represent the scene. Extensive experiments demonstrate the state-of-the-art performance of meta-deformed manifold not only on standard benchmarks containing complex real-world scenes, but also on more challenging large-scale, gigapixel-level scenarios.

References

[1] X. Yuan, M. Ji, J. Wu, D. J. Brady, Q. Dai, and L. Fang, "A modular hierarchical array camera," *Light: Science & Applications*, vol. 10, no. 1, pp. 1–9, 2021.

[2] J. Zhang, T. Zhu, A. Zhang, X. Yuan, Z. Wang, S. Beetschen, L. Xu, X. Lin, Q. Dai, and L. Fang, "Multiscale-vr: multiscale gigapixel 3d panoramic videography for virtual reality," in *2020 IEEE international conference on computational photography (ICCP)*. IEEE, 2020, pp. 1–12.

[3] X. Wang, X. Zhang, Y. Zhu, Y. Guo, X. Yuan, L. Xiang, Z. Wang, G. Ding, D. Brady, Q. Dai *et al.*, "Panda: A gigapixel-level human-centric video dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3268–3278.

[4] J. Zhang, J. Zhang, S. Mao, M. Ji, G. Wang, Z. Chen, T. Zhang, X. Yuan, Q. Dai, and L. Fang, "Gigamvs: A benchmark for ultra-large-scale gigapixel-level 3d reconstruction," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 01, pp. 1–1, 2021.

[5] H. Shum and S. B. Kang, "Review of image-based rendering techniques," in *Visual Communications and Image Processing*, 2000.

[6] S. E. Chen and L. R. Williams, "View interpolation for image synthesis," *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993.

[7] M. Levoy and P. Hanrahan, "Light field rendering," *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996.

[8] G. Riegler and V. Koltun, "Free view synthesis," in *European Conference on Computer Vision*. Springer, 2020, pp. 623–640.

[9] K.-A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. Lempitsky, "Neural point-based graphics," in *European Conference on Computer Vision*. Springer, 2020, pp. 696–712.

[10] G. Riegler and V. Koltun, "Stable view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 216–12 225.

[11] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser, "Ibrnet: Learning multi-view image-based rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4690–4699.

[12] M. Suhail, C. Esteves, L. Sigal, and A. Makadia, "Light field neural rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8269–8279.

[13] 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," *arXiv preprint arXiv:2003.08934*, 2020.

[14] 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, 2020.

[15] J. Zhang, Y. Yao, and L. Quan, "Learning signed distance field for multi-view surface reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6525–6534.

[16] J. Zhang, Y. Yao, S. Li, T. Fang, D. McKinnon, Y. Tsin, and L. Quan, "Critical regularizations for neural surface reconstruction in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6270–6279.

[17] B. Attal, J.-B. Huang, M. Zollhöfer, J. Kopf, and C. Kim, "Learning neural light fields with ray-space embedding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 819–19 829.

[18] Y. Shi, H. Li, and X. Yu, "Self-supervised visibility learning for novel view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9675–9684.

[19] D. Rückert, L. Franke, and M. Stamminger, "Adop: Approximate differentiable one-pixel point rendering," *arXiv preprint arXiv:2110.06635*, 2021.

[20] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer, "Deepvoxels: Learning persistent 3d feature embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2437–2446.

[21] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[22] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[23] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3260–3269.

[24] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[25] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[27] S. Lombardi, T. Simon, J. M. Saragih, G. Schwartz, A. M. Lehrmann, and Y. Sheikh, "Neural volumes," *ACM Transactions on Graphics (TOG)*, vol. 38, pp. 1 – 14, 2019.

[28] C. Jiang, A. Sud, A. Makadia, J. Huang, M. Nießner, T. Funkhouser *et al.*, "Local implicit grid representations for 3d scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6001–6010.

[29] 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*, 2019, pp. 165–174.

[30] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4460–4470.

[31] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 523–540.

[32] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," in *Advances in Neural Information Processing Systems*, 2019, pp. 1121–1132.

[33] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3504–3515.

[34] 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," *arXiv preprint arXiv:2008.02268*, 2020.

[35] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[36] 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*, 2021, pp. 5855–5864.

[37] D. Verbin, P. Hedman, B. Mildenhall, T. E. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-nerf: Structured view-dependent appearance for neural radiance fields," *ArXiv*, vol. abs/2112.03907, 2021.

[38] A. Bergman, P. Kellnhofer, and G. Wetzstein, "Fast training of neural lumigraph representations using meta learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 172–186, 2021.

[39] 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*, 2021, pp. 5589–5599.

[40] 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.

[41] 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," in *NeurIPS*, 2021.

[42] P. Kellnhofer, L. C. Jebe, A. Jones, R. Spicer, K. Pulli, and G. Wetzstein, "Neural lumigraph rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4287–4297.

[43] A. Chen, M. Wu, Y. Zhang, N. Li, J. Lu, S. Gao, and J. Yu, "Deep surface light fields," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 1, no. 1, pp. 1–17, 2018.

[44] J. Zhang, M. Ji, G. Wang, X. Zhiwei, S. Wang, and L. Fang, "Surrf: Unsupervised multi-view stereopsis by learning surface radiance field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[45] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 43–54.

[46] A. Davis, M. Levoy, and F. Durand, "Unstructured light fields," in *Computer Graphics Forum*, vol. 31, no. 2pt1. Wiley Online Library, 2012, pp. 305–314.

[47] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–10, 2016.

[48] G. Wu, Y. Liu, L. Fang, and T. Chai, "Revisiting light field rendering with deep anti-aliasing neural network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[49] V. Sitzmann, S. Rezchikov, B. Freeman, J. Tenenbaum, and F. Durand, "Light field networks: Neural scene representations with single-evaluation rendering," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19 313–19 325, 2021.

[50] I. Choi, O. Gallo, A. Troccoli, M. H. Kim, and J. Kautz, "Extreme view synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7781–7790.

[51] Y. Sun, S. Zhou, R. Cheng, W. Tan, B. Yan, and L. Fu, "Learning robust image-based rendering on sparse scene geometry via depth completion,"

in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7813–7823.

[52] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.

[53] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll, "Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7911–7920.

[54] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 124–14 133.

[55] A. Trevithick and B. Yang, "Grf: Learning a general radiance field for 3d representation and rendering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 182–15 192.

[56] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, and U. Neumann, "Point-nerf: Point-based neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5438–5448.

[57] J. Zhang, G. Yang, S. Tulsiani, and D. Ramanan, "Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 835–29 847, 2021.

[58] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. Sajjadi, A. Geiger, and N. Radwan, "Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5480–5490.

[59] Agisoft LLC, "Agisoft metashape." [Online]. Available: https://www.agisoft.com

[60] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986, pp. 143–150.

[61] B. Karis, "Real shading in unreal engine 4 by," 2013.

[62] J. Kautz, P.-P. Vázquez, W. Heidrich, and H.-P. Seidel, "Unified approach to prefiltered environment maps," in *Rendering Techniques*, 2000.

[63] M. Boss, V. Jampani, R. Braun, C. Liu, J. T. Barron, and H. P. A. Lensch, "Neural-pil: Neural pre-integrated lighting for reflectance decomposition," in *NeurIPS*, 2021.

[64] 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)*, 2018, pp. 767–783.

[65] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, "Recurrent mvsnet for high-resolution multi-view stereo depth inference," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5525–5534.

[66] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. NeurIPS*, 2020.

[67] V. Dumoulin, E. Perez, N. Schucher, F. Strub, H. de Vries, A. C. Courville, and Y. Bengio, "Feature-wise transformations," *Distill*, 2018.

[68] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, "Film: Visual reasoning with a general conditioning layer," in *AAAI*, 2018.

[69] E. Chan, M. Monteiro, P. Kellnhofer, J. Wu, and G. Wetzstein, "pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5795–5805, 2021.

[70] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7708–7717.

[71] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[72] M. Ji, J. Zhang, Q. Dai, and L. Fang, "Surfacenet+: An end-to-end 3d neural network for very sparse multi-view stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4078–4093, 2020.

[73] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[74] M. Garland and P. S. Heckbert, "Simplifying surfaces with color and texture using quadric error metrics," in *Proceedings Visualization'98 (Cat. No. 98CB36276)*. IEEE, 1998, pp. 263–269.

[75] 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.

[76] Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt, and L. Liu, "Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction," *arXiv preprint arXiv:2212.05231*, 2022.

[77] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" *Proceedings of machine learning and systems*, vol. 2, pp. 129–146, 2020.

[78] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2009.

Guangyu Wang is currently a Ph.D student in Shenzhen International Graduate School, Tsinghua University. He received B.E. from University of Electronic Science and Technology of China in 2021. His research interests include 3D reconstruction and neural rendering.

Jinzhi Zhang is currently a Ph.D student in Shenzhen International Graduate School, Tsinghua University. He received B.E. from Huazhong University of Science and Technology in 2019. His research interest is 3D vision.

Kai Zhang is currently an undergraduate student in Department of Electronic Engineering, Tsinghua University. His research interest is 3D vision.

Ruqi Huang is currently an Assistant Professor in Shenzhen International Graduate School, Tsinghua University. He obtained his doctoral degree from University of Paris-Saclay in 2016. His research interests are geometry processing and 3D computer vision.

Lu Fang is currently an Associate Professor in Tsinghua University. She received Ph.D from the Hong Kong Univ. of Science and Technology in 2011, and B.E. from Univ. of Science and Technology of China in 2007. Her research interests include computational imaging and visual intelligence. Dr. Fang is currently IEEE Senior Member, Associate Editor of IEEE TIP.