Efficient 3D Surface Super-resolution via Normal-based Multimodal Restoration

Miaohui Wang, Yunheng Liu, Wuyuan Xie, Boxin Shi, and Jianmin Jiang

Abstract—High-fidelity 3D surface is essential for vision tasks across various domains such as *medical imaging*, *cultural heritage preservation*, *quality inspection*, *virtual reality*, and *autonomous navigation*. However, the intricate nature of 3D data representations poses significant challenges in restoring diverse 3D surfaces while capturing fine-grained geometric details at a low cost. This paper introduces an efficient multimodal normal-based 3D surface super-resolution (*mn3DSSR*) framework, designed to address the challenges of microgeometry enhancement and computational overhead. Specifically, we have constructed one of the largest normal-based multimodal dataset, ensuring superior data quality and diversity through meticulous subjective selection. Furthermore, we explore a new two-branch multimodal alignment approach along with a multimodal split fusion module to mitigate computational complexity while improving restoration performances. To address the limitations associated with normal-based multimodal learning, we develop novel normal-induced loss functions that facilitate geometric consistency and improve feature alignment. Extensive experiments conducted on seven benchmark datasets across four different 3D data representations demonstrate that *mn3DSSR* consistently outperforms state-of-the-art super-resolution methods in terms of restoration accuracy with high computational efficiency.

Index Terms—Photometric stereo-based normal dataset, multimodal 3D surface super-resolution, microgeometry restoration

1 Introduction

High-fidelity 3D surfaces are essential for obtaining precise geometric and topological information [1], directly impacting both academic research and industrial applications. 3D surface super-resolution (3DSSR) has thus become a focal point of research across various domains [2], [3], with the potential to significantly enhance the performance of downstream vision tasks such as *object recognition*, *scene understanding*, *pose estimation*, and *quality inspection*. Similar to the advancements in 2D image super-resolution (2DISR) [4], [5], [6], where deep learning has significantly improved the restoration[1] of pixel-based details like texture and sharpness, 3DSSR provides the opportunity to recover fine-grained surface details and achieve micro-geometry accuracy from low-resolution 3D data representations. However, directly applying 2DISR methodologies to 3D surfaces is non-trivial due to the inherent complexities of *surface geometry*, *curvature*, *topological consistency*, and *lighting variations*, which are not adequately addressed in the 2D image domain.

The inherent complexity of 3D surface structures poses significant challenges for super-resolution, but deep learning techniques offer distinct advantages in 3DSSR modeling due to their powerful nonlinear representation capabilities. Existing 3DSSR methods largely rely on specific 3D data representations (*e.g.*, *point cloud*, *mesh*, *voxel*, *depth*, and *normal*), limiting the scalability, flexibility, and applicability of their convolutional network architectures. For example, point-based methods typically begin by obtaining topological information through point networks [7], followed by the use of upsampling modules to restore point positions, as illustrated in Fig. 1 (a). Mesh-based methods often employ surface subdivision to generate dense meshes, and then utilize graph networks [8] to enhance geometric details, as shown in Fig. 1 (b). Voxel-based methods commonly leverage 3D generative models [9], where sampling noise (or latent code) is mapped to high-resolution voxels, conditioned on corresponding low-resolution voxels, as illustrated in Fig. 1 (c). Depth-based methods usually interpolate the depth values before applying 2DISR networks to restore depth details [10], as depicted in Fig. 1 (d). Recently, we have developed normal-based approaches to enhance the restoration of high-resolution normal maps [11], followed by a surface-from-normal (SfN) module to reconstruct fine-grained 3D surfaces as depicted in Fig. 1 (e).

In this paper, we present a novel multimodal normal-based 3DSSR framework, as shown in Fig. 2. The proposed multimodal 3DSSR meets two primary challenges compared with the existing state-of-the-arts [19], [24], [25]: (1) effectively utilizing information from multiple modalities without overfitting, and (2) incorporating geometric constraints

- *This work was supported in part by the National Natural Science Foundation of China under Grant 62472290 and Grant 62372306, and in part by the Natural Science Foundation of Guangdong Province under Grant 2024A1515011972 and Grant 2023A1515011197. (Corresponding author: Wuyuan Xie)*
- *Miaohui Wang, Yunheng Liu, Wuyuan Xie, and Jianmin Jiang are with the Research Institute for Future Media Computing at the College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518000, China. (e-mail: 2210273085@email.szu.edu.cn, wuyuan.xie@gmail.com, jianmin.jiang@szu.edu.cn)*
- *Miaohui Wang is also with the State Key Laboratory of Radio Frequency Heterogeneous Integration (Shenzhen University), and College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518000, China. (e-mail: wang.miaohui@gmail.com)*
- *Boxin Shi is with State Key Laboratory of Multimedia Information Processing and National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing 100871, China. (e-mail: shiboxin@pku.edu.cn)*

1. In this paper, the term 'restoration' refers specifically to the recovery of fine-grained surface details from coarse 3D input representations, excluding broader degradation types such as image noise, blur, environmental interference, or compression artifacts.

TABLE 1: **Taxonomy of 3D surface super-resolution (3DSSR) frameworks**. Representative 2D and 3D methods are compared in terms of data representation, input modality, type of topology, time and space complexity, sensitivity to surface gradients, and robustness to noise. Notably, our method integrates three modalities with affordable complexity.

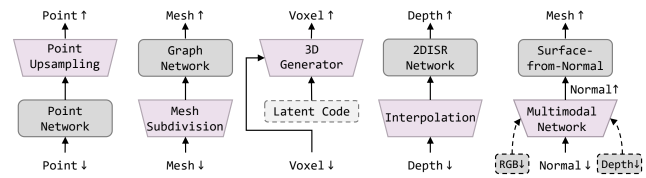
Algorithm	Type	Representation	Input modality	Topology	Complexity	Gradient	Robustness
Qian2021CVPR [12]	3D	point cloud	point cloud	n/a	high	✘	low
Feng2022CVPR [13]	3D	point cloud	point cloud	n/a	high	✔	low
He2023CVPR [7]	3D	point cloud	point cloud	n/a	high	✘	medium
Loop2008TOG [14]	3D	mesh	mesh	irregular	low	✘	medium
Liu2020TOG [8]	3D	mesh	mesh	irregular	high	✔	low
Xie2022TPAMI [15]	3D	voxel	voxel	regular 3D grid	high	✘	medium
Shim2023CVPR [16]	3D	voxel	voxel	regular 3D grid	high	✘	medium
Voynov2019ICCV [17]	2D	depth image	depth image, RGB image	regular 2D grid	low	✔	medium
Haefner2020TPAMI [18]	2D	depth image	depth image, RGB image	regular 2D grid	low	✔	medium
Deng2021TPAMI [19]	2D	depth image	depth image, RGB image	regular 2D grid	medium	✘	medium
Zhao2022CVPR [20]	2D	depth image	depth image, RGB image	regular 2D grid	low	✘	medium
Metzger2023CVPR [10]	2D	depth image	depth image, RGB image	regular 2D grid	medium	✘	medium
Wang2022TPAMI [21]	2D	depth image	binocular stereo images	regular 2D grid	low	✘	medium
Ju2024TCSVT [22]	2D	normal map	RGB images	regular 2D grid	medium	✔	medium
Xie2022CVPR [11]	2D	normal map	normal, depth, RGB images	regular 2D grid	high	✔	high
Xie2023IJCAI [23]	2D	normal map	normal, RGB images	regular 2D grid	high	✔	high
Ours	2D	normal map	normal, depth, RGB images	regular 2D grid	medium	✔	high

into model training to better capture 3D microgeometry structures. To address these challenges, our method incorporates three special components: multimodal pre-processing, texture-shape-based two-branch alignment, and multimodal split fusion. The pre-processing module is designed to transform raw inputs for better model training, the alignment module removes irrelevant data while preserving texture and shape information, and the fusion module optimizes the integration of multimodal features, significantly reducing model parameters and computational complexity. Additionally, we explore the geometry properties of normal maps and introduce curl-based and alignment-based loss functions that impose additional constraints and regularization on intermediate features, improving restoration performance and accelerating convergence.

To facilitate the development of normal-based deep 3DSSR models, we further establish a large-scale multimodal dataset, including three data modalities: normal maps, depth images, and multi-illumination RGB images. In summary, this paper makes four key contributions[2]:

- We propose an efficient multimodal normal-based 3D surface super-resolution (*mn3DSSR*) framework via: (i) introducing a multimodal Swin-Transformer alignment (MSTA) module that aligns texture and shape features across two network branches; (ii) developing a multimodal split fusion (MSF) module that dynamically integrates the two feature branches with the normal modality towards improvement of its restoration performances.
- To optimize the training of our *mn3DSSR*, we propose three new loss functions, where the first one distinguishes between foreground and background normal distortion calculations, the second leverages curl-based normal measurements to capture microgeometry structures, and the third enhances multimodal feature representation through shape-texture alignment. Hyper-parameter sensitivity experiments validate the effectiveness of these losses in the normal-based 3DSSR task.

2. The source code, normal-based multimodal dataset, and the collected 3DSSR and 2DISR methods are made publicly available at https://charwill.github.io/mn3dssr.html.



(a) Point-based (b) Mesh-based (c) Voxel-based (d) Depth-based (e) Normal-based

Fig. 1: **Comparison of 3DSSR architectures across 3D data representations**: (a) point cloud-based methods use point networks, (b) mesh-based use graph networks, (c) voxel-based adopt generative models, and (d) depth-based rely on 2DISR networks. In contrast, (e) our multimodal normal-based approach enables more effective texture–shape fusion.

- To obtain high-quality multimodal training data with fine-grained surface details and complex geometries, we propose to establish a dedicated dataset, one of the largest real-world normal-based datasets to date, consisting of 600 different 3D surfaces with rich details. Compared to existing benchmarks, our dataset outperforms them on average in terms of scale, resolution, quality, and diversity.
- Extensive experiments have been carried out to provide a comprehensive evaluation of our proposed against 24 representative 3DSSR and 2DISR methods on four normal-based benchmark datasets, and additional analysis has been extended to three additional 3D data representations–*point cloud*, *mesh*, and *depth* datasets. The experimental results demonstrate that our proposed method consistently outperforms cutting-edge approaches in terms of restoration accuracy with high computational efficiency.

Finally, the work described in this paper presents a significant extension of our earlier research [11], incorporating several key improvements, which can be highlighted as: (i) we have explored an efficient multimodal normal-based network architecture, featuring newly designed alignment and fusion modules to reduce computational overhead while improving restoration performance; (ii) we have devised new normal-induced loss functions that enable better model training; (iii) we have substantially expanded our multimodal dataset by increasing data scale, quality, and diversity through meticulous subjective selection; and (iv) we have conducted extensive experiments on four 3D data representations (*e.g.*, *point cloud-*, *mesh-*, *depth-*, and *normal-based* datasets), providing a comprehensive evaluation of the existing 3DSSRs and 2DISRs.

2 Related Work

In this section, we provide an overview of representative methods for 3DSSRs. Based on the type of 3D data representations, these methods can be roughly classified into 3D domain-based and 2D domain-based approaches, as provided in Table 1.

2.1 Surface Super-resolution in 3D Domain

3D domain-based studies have explored 3DSSR based on various 3D data representations, including *point clouds*, *mesh*, *voxel*, and other explicit 3D formats.

***Point Cloud*-based.** Point clouds represent 3D shapes as discrete sets of data points with *Cartesian* coordinates (x, y, z) which correspond to its position in a 3D space. Various strategies have been developed to improve the number of points, including point convolutional networks [26], graph models [12], and Transformers [27]. However, due to the sparsity and uneven density of point clouds, existing point cloud-based 3DSSRs are difficult to handle a large number of data points.

Recent advances, such as local neural implicit representations, address these issues by generating continuous interpolation points [13]. Additionally, combining point convolution modules with traditional techniques has shown promise in upsampling performance [7]. However, these methods still face limitations due to the need for intensive neighborhood look-ups, constrained by the absence of topological information.

***Mesh*-based.** Mesh is a widely-used 3D representation due to its ability to provide complete topological information, facilitating the computation of geometric features such as normal and curvature. Traditional methods [14], [28], [29] have been designed to improve mesh resolution by leveraging local connectivity patterns and have demonstrated maturity and efficiency. However, these handcrafted priors often struggle with diverse 3D objects and fail to accurately recover sharp geometric details.

The introduction of graph learning frameworks, particularly graph neural network [8], has enabled the recovery of surface details by leveraging neighborhood information in a mesh. Despite these advancements, mesh-based 3DSSRs face challenges due to their irregular topology, which necessitates the explicit storage and processing of adjacency information. Also, computational complexity makes it difficult to handle denser 3D meshes, limiting the scalability of these methods for high-resolution tasks.

***Voxel*-based.** Voxel representations divide 3D surfaces into a grid of volumetric pixels, with each voxel representing a small cube of space. The relationship to the object surface can be defined by a signed distance from the voxel center to the nearest point on the surface. Voxel representations have a regular structure, which makes it easy to apply convolutional networks.

Early voxel-based 3DSSRs [9], [15] have employed generative models to achieve conditional probability distributions. In addition to directly representing 3D shapes with voxels, some methods [16], [30], [31] have also leveraged implicit fields stored within voxels or other regular structures to capture finer geometric details. However, since memory consumption and computation complexity grow rapidly with the number of voxels, the output size of these methods is typically limited to small values (*e.g.*, 256^3).

2.2 Surface Super-resolution in 2D Domain

Although 3D-domain methods offer several advantages, they also face significant bottlenecks of storage and computation. In contrast, 2D domain-based methods can alleviate many of these challenges, and we mainly review 3DSSRs based on *depth* and *normal* data representations.

***Depth*-based.** Depth images significantly reduce the complexity of storing and processing 3D surfaces by encoding space positions in a regular 2D image format. This enables the application of well-established image super-resolution techniques to 3DSSR.

Due to the low-precision of depth images acquired by camera sensors, recent studies have primarily concentrated on using high-resolution RGB images to guide depth super-resolution (DSR). For instance, many DSRs [10], [20], [32], [33] have combined traditional image filtering with deep learning to enhance depth images. To further reduce texture interference while enhancing 3D surface details, depth images and corresponding RGB guidance can be processed separately, enabling better multimodal learning [19].

Depth-based 3DSSRs effectively recover position and contour information, but they struggle to restore finer geometric structures. To address this limitation, DSR techniques that better represent geometric details have been explored. For instance, the perceptual quality of depth images can be improved by incorporating normal maps or rendered images into loss functions [17]. Additionally, *photometric stereo* is combined with DSRs, where multiple high-resolution RGB images are used to recover normal maps and optimize depth image details [18]. While these approaches have employed normal maps to enhance surface quality, they primarily use rough normal information as an auxiliary input and lack further exploration to restore fine-grained 3D surface details.

***Normal*-based.** Normal maps encode rich 3D geometric information (*e.g.*, surface orientation, bump, and microgeometry) in the 2D domain and, compared to depth images, are better suited for representing small variations in surface details. Additionally, high-quality normal maps can be easily obtained through *photometric stereo* setups [34], [35].

In *photometric stereo*, multi-illumination RGB images capture pixel level 3D surface features, inspiring the exploration of joint super-resolution and surface reconstruction [22]. To further leverage multiple image modalities generated during surface acquisition, we have developed a Transformer-based multimodal 3DSSR scheme [11]. Later, we have employed a variational autoencoder (VAE) to model probability distributions in the latent space and improve modality alignment [23]. Our early methods provide a general multimodal framework for normal-based 3DSSR.

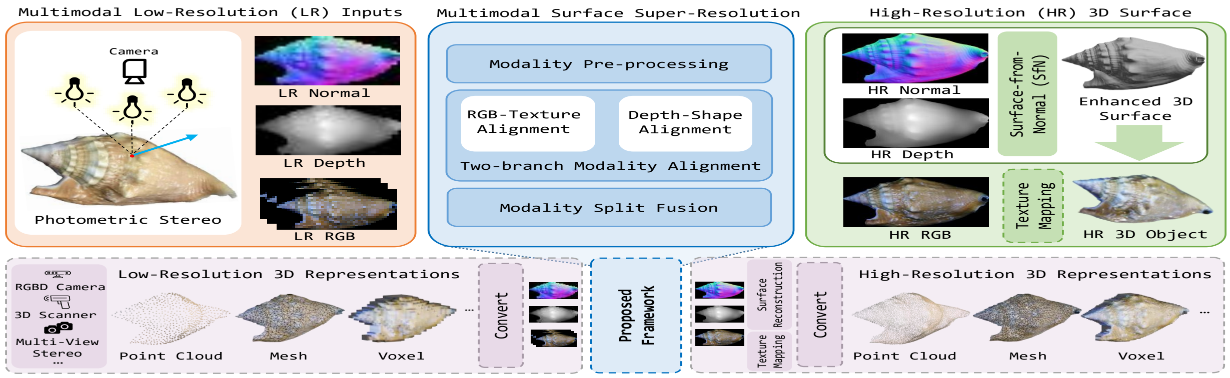


Fig. 2: **Illustration of the proposed framework for 3D surface super-resolution applications**. Low-resolution depth, normal, and RGB inputs are enhanced to recover high-resolution outputs. While our framework is trained on *photometric stereo* data, it generalizes to other 3D representations (*e.g.*, *point cloud*, *mesh*, *voxel*, and *depth*). *For detailed examination, please zoom in on the electronic version.*

In this paper, we refine it by proposing efficient network designs that significantly enhance multimodality utilization while reducing model complexity.

Table 1 provides a general taxonomy of representative 3DSSRs. Our proposed framework leverages multimodal framework to enhance surface details, as shown in Fig. 2. Unlike existing 3D domain methods, our *mn3DSSR* primarily utilizes normal maps and maintains a regular topology, which ensures efficient computation and storage for dense 3D surfaces. In contrast to 2D domain methods, our *mn3DSSR* does not require additional high-resolution RGB images for guidance. By directly enhancing normal maps, our framework effectively preserves gradient information related to object surfaces, which is crucial for accurately characterizing microgeometry surface details. Additionally, our framework demonstrates strong resilience to noise, benefiting from the robustness of SfN reconstruction. Moreover, we have achieved computational efficiency comparable to that of most deep learning-based 2DISR methods.

3 Methodology

3.1 Problem Formulation

Let H, W, τ, and L denote the width, height, upsampling ratio, and the number of lighting directions in a *photometric stereo* setup. Our proposed *mn3DSSR* framework utilizes a low-resolution normal map $\mathbf{N}_{lr} \in \mathbb{R}^{\frac{H}{\tau} \times \frac{W}{\tau} \times 3}$ as the primary 3D surface representation. It leverages the depth modality $\mathbf{D}_{lr} \in \mathbb{R}^{\frac{H}{\tau} \times \frac{W}{\tau}}$ and the RGB modality $\mathbf{I}_{lr} \in \mathbb{R}^{\frac{H}{\tau} \times \frac{W}{\tau} \times 3 \times L}$ to generate a high-resolution normal map $\mathbf{N}_{sr} \in \mathbb{R}^{H \times W \times 3}$, which is then used to reconstruct a fine-grained 3D surface $\mathcal{S}_{\text{3D}}$. The overall process can be formulated as:

$$\mathcal{S}_{\text{3D}} = \mathcal{F}_{\text{SfN}}(\mathcal{F}_{\text{mn3DSSR}}(\mathbf{N}_{lr}, \mathbf{D}_{lr}, \mathbf{I}_{lr})), \tag{1}$$

where $\mathcal{F}_{\text{SfN}}$ represents a typical SfN approach, and $\mathcal{F}_{\text{mn3DSSR}}$ denotes our *mn3DSSR* model to generate $\mathbf{N}_{sr}$. It is important to note that the choice of the SfN method is not the focus of this paper, and any advanced SfN method [36], [37] can be used in Eq. (1). Consequently, 3DSSR can be further reduced to only perform the super-resolution operation on normal maps, akin to 2DISRs.

More specifically, $\mathcal{F}_{\text{mn3DSSR}}$ is further divided into three components: (1) we pre-process the three input modalities using $\mathcal{F}_{\text{process}}$ to extract the transformed features $\mathbf{N}^{t}_{lr}$, $\mathbf{N}^{s}_{lr}$, $\mathbf{D}'_{lr}$, and $\mathbf{I}'_{lr}$, respectively; (2) we then align the resulted low-resolution $\mathbf{I}'_{lr}$ and $\mathbf{D}'_{lr}$ to the normal domain using a two-branch alignment module $\mathcal{F}_{\text{align}}$. This delivers two outputs: $\mathbf{F}_{tn}$, representing the high-frequency texture normal feature, and $\mathbf{F}_{sn}$, representing the low-frequency shape normal feature; (3) finally, we fuse two alignment branches with the normal modality utilizing a fusion module $\mathcal{F}_{\text{fuse}}$, resulting in the final enhanced normal map $\mathbf{N}_{sr}$. These three components can be formulated as:

$$\begin{aligned} \mathbf{N}^{t}_{lr}, \mathbf{N}^{s}_{lr}, \mathbf{D}'_{lr}, \mathbf{I}'_{lr} &= \mathcal{F}_{\text{process}}(\mathbf{N}_{lr}, \mathbf{D}_{lr}, \mathbf{I}_{lr}), \\ \mathbf{F}_{tn}, \mathbf{F}_{sn} &= \mathcal{F}_{\text{align}}(\mathbf{N}^{t}_{lr}, \mathbf{N}^{s}_{lr}, \mathbf{D}'_{lr}, \mathbf{I}'_{lr}), \\ \mathbf{N}_{sr} &= \mathcal{F}_{\text{fuse}}(\mathbf{N}_{lr}, \mathbf{F}_{sn}, \mathbf{F}_{tn}). \end{aligned} \tag{2}$$

In *mn3DSSR*, our optimization objective is to minimize the normal pixel distance $\mathcal{L}_{\text{pix}}$ between the ground-truth normal $\mathbf{N}_{gt}$ and the enhanced normal $\mathbf{N}_{sr}$. Additionally, we introduce the curl normal loss to ensure model training more focused on microgeometry structures, including a curl-weighted normal loss $\mathcal{L}^{weight}_{\text{curl}}$ and a curl-regularized normal loss $\mathcal{L}^{regular}_{\text{curl}}$. Furthermore, we incorporate the multimodal alignment loss to provide auxiliary supervision from two alignment branches, including a RGB-texture loss $\mathcal{L}^{texture}_{\text{align}}$ and a depth-shape loss $\mathcal{L}^{shape}_{\text{align}}$. Consequently, the joint optimization objective is defined as:

$$\begin{aligned} \min_{\mathbf{N}_{sr}, \mathbf{F}_{tn}, \mathbf{F}_{sn}} &\{\mathcal{L}_{\text{pix}}(\mathbf{N}_{sr}, \mathbf{N}_{gt}) \\ &+ \lambda^{weight}_{\text{curl}} \times \mathcal{L}^{weight}_{\text{curl}}(\mathbf{N}_{sr}, \mathbf{N}_{gt}) + \lambda^{regular}_{\text{curl}} \times \mathcal{L}^{regular}_{\text{curl}}(\mathbf{N}_{sr}) \\ &+ \lambda_{\text{align}} \times (\mathcal{L}^{texture}_{\text{align}}(\mathbf{F}_{tn}, \mathbf{N}^{t}_{gt}) + \mathcal{L}^{shape}_{\text{align}}(\mathbf{F}_{sn}, \mathbf{D}_{gt}))\}, \end{aligned} \tag{3}$$

where $\lambda^{weight}_{\text{curl}}$, $\lambda^{regular}_{\text{curl}}$, and λ_{align} represent three scaling factors used to adjust the contributions of each loss term. We provide detailed explanations of the associated network modules and the loss function designs in the following. An overview of our *mn3DSSR* is illustrated in Fig. 3.

3.2 Multimodal Pre-processing Stage (MPS)

Essentially, MPS performs necessary feature transformation on the raw multimodal data, addressing two main issues: (1)

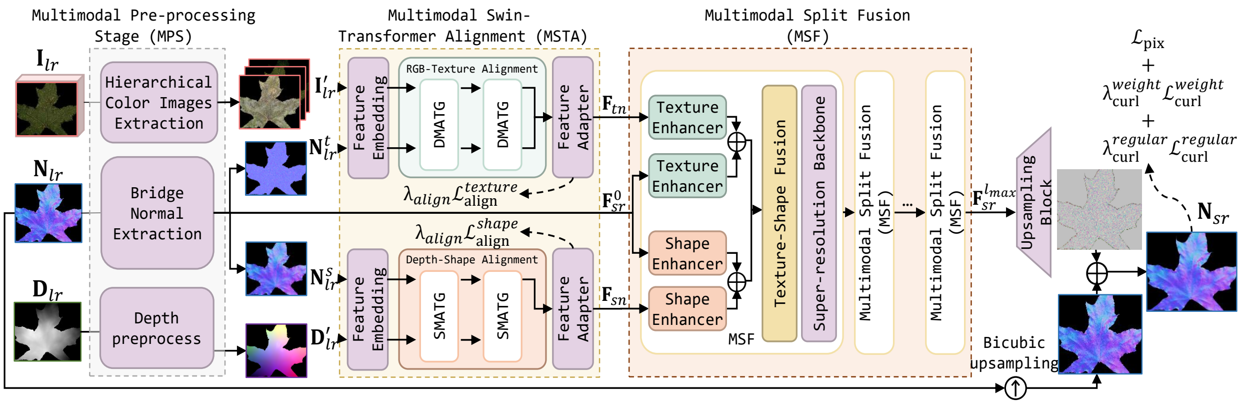


Fig. 3: **Pipeline of the proposed multimodal normal-based 3D surface super-resolution network (*mn3DSSR*).** The three input modalities are first processed by the multimodal pre-processing stage (MPS), followed by alignment via the multimodal Swin-Transformer alignment (MSTA) module. The aligned features are then fused by the multimodal split fusion (MSF) module to produce a high-resolution normal map for 3D surface reconstruction.

normalizing the input to minimize variations and distribution differences across modalities from various sources, and (2) constructing primary features for the alignment module, generating shape and texture features.

Normal Modality. As the ultimate target for 3DSSR, $\mathbf{N}_{lr}$ is used to guide the alignment of $\mathbf{I}_{lr}$ and $\mathbf{D}_{lr}$ to their respective shape and texture components. We employ the frequency separation [1] to obtain the high-frequency texture normal component $\mathbf{N}^t_{lr}$ and the low-frequency geometric shape component $\mathbf{N}^s_{lr}$. In other words, $\mathbf{N}_{lr} = \mathbf{N}^t_{lr} + \mathbf{N}^s_{lr}$.

Depth Modality To construct a 3D position encoding for the subsequent alignment module, we adopt the depth modality $\mathbf{D}_{lr}$. We normalize it by subtracting the mean and dividing by the standard deviation of depth values, resulting in $\mathbf{D}'_{lr} \in \mathbb{R}^{\frac{H}{\tau} \times \frac{W}{\tau} \times 3}$. This operation preserves the relative position information expressed by the depth while facilitating learning in subsequent networks.

RGB Modality. As our RGB modality includes multi-illumination images, three basic aggregation functions (*i.e.*, `max`, `min`, and `mean`) are used to maintain permutation invariance and obtain the brightest, darkest, and average intensity RGB images. These three types of RGB feature images retain important information, such as *specularities*, *shadows*, and *colors*, reflecting spatial variations in material and geometric characteristics on 3D surface $\mathcal{S}_{\text{3D}}$. To reduce domain differences, we normalize and concatenate them to obtain $\mathbf{I}'_{lr} \in \mathbb{R}^{\frac{H}{\tau} \times \frac{W}{\tau} \times 9}$, as shown in Fig. 4.

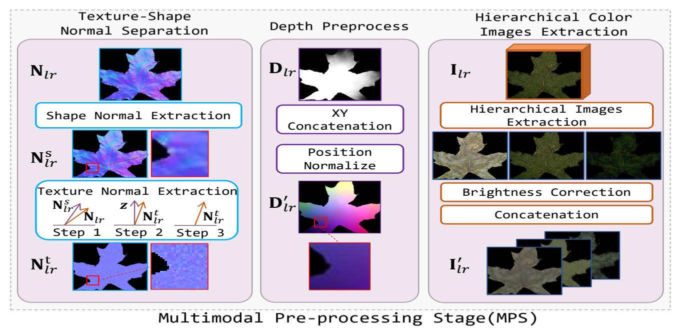


Fig. 4: **Multimodal Pre-processing Stage (MPS).** Three modalities are pre-processed to generate primary features for subsequent alignment and fusion.

3.3 Multimodal Swin-Transformer Alignment (MSTA)

To align the previously processed $\mathbf{I}'_{lr}$ and $\mathbf{D}'_{lr}$, we propose a new two-branch multimodal Swin-Transformer alignment (MSTA) module, consisting of a RGB-texture alignment branch and a depth-shape alignment branch. Specifically, our objective is primarily to enhance the texture normal information by aligning $\mathbf{I}'_{lr}$ with $\mathbf{N}^t_{lr}$ in the RGB-texture branch. Simultaneously, we inject the global 3D geometric information into the shape normal by aligning $\mathbf{D}'_{lr}$ with $\mathbf{N}^s_{lr}$ in the depth-shape branch. An overview of our proposed MSTA module is shown in Fig. 5 (a).

Both the texture and shape alignment branches are designed with three main components: (1) feature embedding, (2) Swin-Transformer alignment, and (3) feature adaptation. In the feature embedding, we utilize one 2-layer 3×3 convolution with the `ELU` activation function to project the inputs (*i.e.*, $\mathbf{N}^t_{lr}$, $\mathbf{N}^s_{lr}$, $\mathbf{D}'_{lr}$, and $\mathbf{I}'_{lr}$) into shallow features (*i.e.*, $\mathbf{F}_t$, $\mathbf{F}_s$, $\mathbf{F}_{dpt}$, and $\mathbf{F}_{rgb}$).

In the Swin-Transformer alignment, we draw inspiration from the complementary roles of cross-attention and self-attention in modeling modality interactions. We propose a mix attention Transformer (MAT). Our MAT module is tailored to integrate the functionalities of both the cross-attention [38] and the window self-attention [39]. The MAT layer is formulated as:

$$\begin{aligned}\mathcal{F}_{\text{mat}}(\mathbf{X},\mathbf{Y}) = ((1-\alpha) \times \texttt{softmax}(\frac{\mathbf{Y}\mathbf{W}^c_q(\mathbf{X}\mathbf{W}_k)^T}{\sqrt{d_k}} + \mathbf{B}) \\ + \alpha \times \texttt{softmax}(\frac{\mathbf{X}\mathbf{W}^s_q(\mathbf{X}\mathbf{W}_k)^T}{\sqrt{d_k}} + \mathbf{B}))\mathbf{X}\mathbf{W}_v,\end{aligned} \tag{4}$$

where $\mathbf{X}$ and $\mathbf{Y}$ represent two different multimodal features after window partition. $\mathbf{W}_k$, $\mathbf{W}_v$, $\mathbf{W}^c_q$, and $\mathbf{W}^s_q$ represent the projection matrices for key, value, cross-attention query and self-attention query, respectively. $\mathbf{B}$ denotes the relative position embedding, and d_k denotes the number of feature channels. $\alpha \in [0, 1]$ is a trainable scalar used to balance self-attention and cross-attention, which is initialized to 0.5. Finally, the feature adapter $\mathcal{F}_{\text{fa}}$ employs a combination of

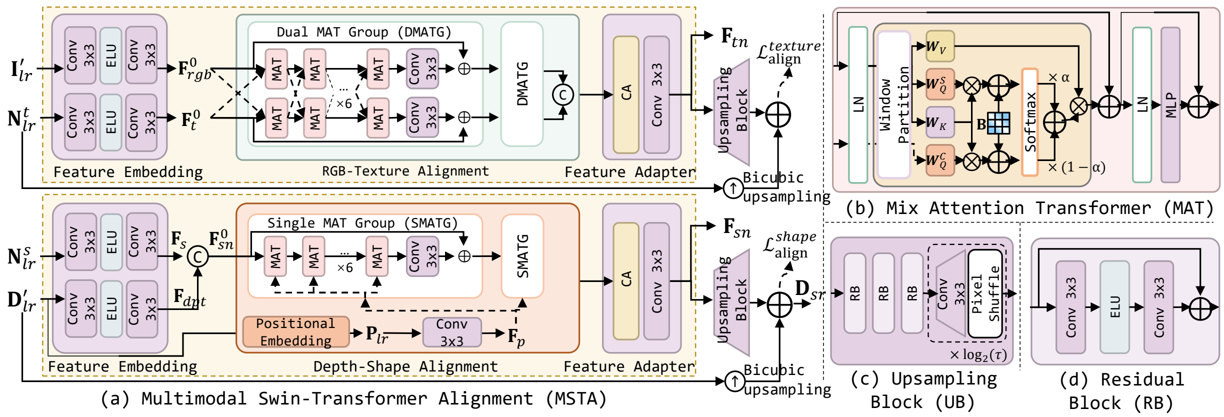


Fig. 5: **Multimodal Swin-Transformer Alignment (MSTA)**: (a) MSTA aligns RGB and depth features via an embedding–alignment–adaptation pipeline, (b) *Mix Attention Transformer* (MAT) extends Swin Transformer with cross and self-attention, (c) *Upsampling Block*, and (d) *Residual Block*.

channel attention (CA) and convolution layer to stabilize the aligned features, as shown in Fig. 5 (a).

The key difference between our MAT block and existing Transformer block is the mix attention layer, which allows for automatic trade-offs between cross-attention and self-attention during the learning process. This is achieved without significantly increasing the number of parameters, enabling the alignment module to model interactions between the two modalities more effectively and flexibly. Both the texture and shape alignment branches are constructed based on our MAT, albeit with slightly different designs. Detailed explanations of our two-branch alignments are provided in the following, as shown in Fig. 5 (b).

RGB-Texture Alignment. The RGB-texture branch primarily aligns $\mathbf{I}'_{lr}$ with $\mathbf{N}^t_{lr}$. While the extracted hierarchical RGB features $\mathbf{I}'_{lr}$ contain rich color and texture information, their high-frequency components can be complex, often containing noises or outliers due to *specularities* and *shadows*.

To mitigate these negative effects, we employ a symmetric dual MAT design to align RGB and normal texture features. This design facilitates adaptive information exchange between both modalities while refining their respective features. In this branch, we assemble six MAT blocks into a group, adding one 3×3 convolution and one residual connection at the end of each group to stabilize feature learning. We refer to it as a dual MAT group (DMATG) and use two DMATGs in the MSTA module. Specifically, DMATG is formulated as follows:

$$\{\mathbf{F}^k_{rgb}, \mathbf{F}^k_t\} = \begin{cases} \{\mathcal{F}_{\text{conv}}(\mathbf{I}'_{lr}), \mathcal{F}_{\text{conv}}(\mathbf{N}^t_{lr})\} & , k=0 \\ \{\texttt{Conv}(\mathbf{F}^{k-1}_{rgb}) + \mathbf{F}^{k-7}_{rgb}, \texttt{Conv}(\mathbf{F}^{k-1}_t) + \mathbf{F}^{k-7}_t\} & , k>0 \,\&\, k\%7=0 \\ \{\mathcal{F}_{\text{mat}}(\mathbf{F}^{k-1}_{rgb}, \mathbf{F}^{k-1}_t), \mathcal{F}_{\text{mat}}(\mathbf{F}^{k-1}_t, \mathbf{F}^{k-1}_{rgb})\} & , \text{otherwise}, \end{cases} \quad (5)$$

where $\mathbf{F}^k_{rgb}$ and $\mathbf{F}^k_t$ denote the k-th layer features produced by the corresponding RGB and normal modalities. `Conv` denotes one 3×3 convolutional layer, $\mathcal{F}_{\text{mat}}$ denotes a MAT layer, and $\mathcal{F}_{\text{conv}}(\cdot) = \texttt{Conv}(\texttt{ELU}(\texttt{Conv}(\cdot)))$ represents one 2-layer convolutional block with the `ELU` activation.

Finally, we concatenate the features $\mathbf{F}^{k_{\max}}_{rgb}$ and $\mathbf{F}^{k_{\max}}_t$ from the last layer into the subsequent feature adapter $\mathcal{F}_{\text{fa}}$ to obtain the aligned texture normal feature $\mathbf{F}_{tn}$ as:

$$\mathbf{F}_{tn} = \mathcal{F}_{\text{fa}}(\texttt{Concat}(\mathbf{F}^{k_{\max}}_{rgb}, \mathbf{F}^{k_{\max}}_t)), \quad (6)$$

where `Concat` represents the operation of concatenating features along the channel dimension, and the concatenated feature is $\mathbf{F}_{rgbt} = \texttt{Concat}(\mathbf{F}^{k_{max}}_{rgb}, \mathbf{F}^{k_{max}}_t)$. Our $\mathcal{F}_{\text{fa}}$ module consists of one CA block $\mathcal{F}_{\text{ca}}$ and one 3×3 convolution layer. Specifically, $\mathcal{F}_{\text{fa}}(\mathbf{F}_{rgbt}) = \texttt{Conv}(\mathcal{F}_{\text{ca}}(\mathbf{F}_{rgbt})) = \texttt{Conv}(\texttt{Sigmoid}(\texttt{Conv}_1(\texttt{ReLU}(\texttt{Conv}_1(\texttt{AvgPool}(\mathbf{F}_{rgbt}))))) \odot \mathbf{F}_{rgbt}))$, where $\texttt{Conv}_1$ denotes one 1×1 convolution and $\odot$ denotes an element-wise multiplication.

Depth-Shape Alignment. The depth-shape alignment focuses on aligning $\mathbf{D}'_{lr}$ with $\mathbf{N}^s_{lr}$. Given the strong correlation between depth and normal, this alignment branch is inherently simpler than the RGB-texture alignment. To control computational complexity, we concatenate the shallow depth feature $\mathbf{F}_{dpt} = \mathcal{F}_{\text{conv}}(\mathbf{D}'_{lr})$ and the shallow shape normal feature $\mathbf{F}_s = \mathcal{F}_{\text{conv}}(\mathbf{N}^s_{lr})$, and leverage MAT blocks to model their interactions.

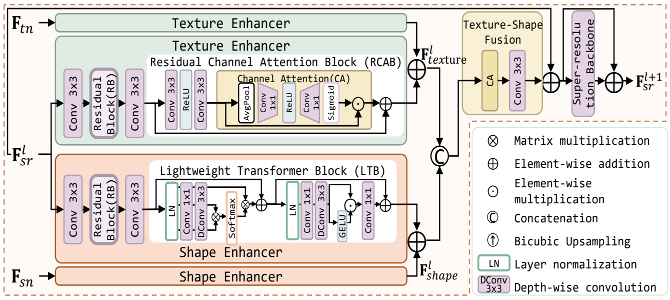
To enhance the representation of global geometry shape, we further inject 3D positional information into the MAT block. Specifically, we apply the sinusoidal position encoding [38] to $\mathbf{D}'_{lr}$ to obtain a 3D positional encoding $\mathbf{P}_{lr}$. This encoding normalizes the numerical range of 3D coordinates to the interval [0, 1], making them more suitable for training. We then extract the shallow position feature $\mathbf{F}_p$ from $\mathbf{P}_{lr}$ by one layer 3×3 convolution, $\mathbf{F}_p = \texttt{Conv}(\mathbf{P}_{lr})$. In this branch, we replace the dual MAT pair structure with a single MAT structure. We refer to it as the single MAT group (SMATG) and use two SMATGs in the MSTA module. The related computation process is formulated as:

$$\mathbf{F}^k_{sn} = \begin{cases} \texttt{Concat}(\mathcal{F}_{\text{conv}}(\mathbf{D}'_{lr}), \mathcal{F}_{\text{conv}}(\mathbf{N}^s_{lr})) & , k=0 \\ \texttt{Conv}(\mathbf{F}^{k-1}_{sn}) + \mathbf{F}^{k-7}_{sn} & , k>0 \,\&\, k\%7=0\,, \\ \mathcal{F}_{\text{mat}}(\mathbf{F}^{k-1}_{sn}, \mathbf{F}_p) & , \text{otherwise}, \end{cases} \quad (7)$$

where $\mathbf{F}^k_{sn}$ denotes latent features at the k-th layer that incorporate both depth and shape information. Finally, we apply the feature adapter to obtain the aligned shape normal feature $\mathbf{F}_{sn} = \mathcal{F}_{\text{fa}}(\mathbf{F}^{k_{\max}}_{sn})$.

3.4 Multimodal Split Fusion (MSF)

After processing the side-modality features in the MSTA module, $\mathbf{F}_{tn}$ and $\mathbf{F}_{sn}$ are further fused into the normal modality to assist in super-resolution feature extractions.



Multimodal Split Fusion (MSF)

Fig. 6: **Multimodal Split Fusion (MSF).** A texture and a shape enhancer decompose and refine features from $\mathbf{F}^{l}_{sr}$, which are then fused with enhanced $\mathbf{F}_{tn}$ and $\mathbf{F}_{sn}$. Subsequently, the fused texture and shape components (*i.e.*, $\mathbf{F}^{l}_{texture}$ and $\mathbf{F}^{l}_{shape}$) are recombined through the texture-shape fusion module.

In our previous method [11], a fusion module has been developed based on a spatial feature transform to modulate side-modalities. However, this approach often neglects dynamic changes in the main normal branch and struggles to adapt to the distinct characteristics of texture and shape normal features. To address this limitation, we propose a multimodal split fusion (MSF) module that integrates texture and shape normal features. The network architecture of our MSF is illustrated in Fig. 6.

As mentioned earlier, $\mathbf{F}_{tn}$ contains richer local texture information. Given that convolutional neural networks (CNNs) are particularly effective at extracting local spatial texture, we utilize a residual channel attention block (RCAB) as a texture enhancer $\mathcal{F}_{\text{texture}}$ to focus on extracting high-frequency local detail information from $\mathbf{F}_{tn}$ and the enhanced normal feature $\mathbf{F}_{sr}$. Subsequently, we perform the element-wise addition $\oplus$ to fuse the latent features. The texture enhancement is formulated as:

$$\mathbf{F}^{l}_{texture} = \mathcal{F}^{l}_{\text{texture}}(\mathbf{F}_{tn}) \oplus \mathcal{F}^{l}_{\text{texture}}(\mathbf{F}^{l}_{sr}), \tag{8}$$

where $\mathbf{F}^{l}_{texture}$ represents the enhanced texture feature in the l-th MSF block, and $\mathbf{F}^{0}_{sr} = \texttt{Conv}(\mathbf{N}_{lr})$. $\mathcal{F}^{l}_{\text{texture}}$ represents our texture enhancer, containing one residual block (RB) and one RCAB. The residual block is shown in Fig. 5 (d).

In contrast, since $\mathbf{F}_{sn}$ has more global shape information, we employ a lightweight Transformer block (LTB) as the shape enhancer $\mathcal{F}_{\text{shape}}$ to obtain the low-frequency global shape information from $\mathbf{F}_{sn}$ and $\mathbf{F}_{sr}$. We also perform the element-wise addition to aggregate the extracted features into $\mathbf{F}^{l}_{shape}$. The shape enhancement is formulated as:

$$\mathbf{F}^{l}_{shape} = \mathcal{F}^{l}_{\text{shape}}(\mathbf{F}_{sn}) \oplus \mathcal{F}^{l}_{\text{shape}}(\mathbf{F}^{l}_{sr}), \tag{9}$$

where $\mathcal{F}^{l}_{\text{shape}}$ denotes the l-th shape enhancer consisting of one RB and one LTB.

Once we obtain the fused feature components $\mathbf{F}^{l}_{texture}$ and $\mathbf{F}^{l}_{shape}$, we concatenate them and further apply a combination of the $\mathcal{F}_{\text{ca}}$ and $\texttt{Conv}$ layers in the texture-shape fusion module to generate a complete normal feature representation. The fused feature is then fed into a plug-and-play super-resolution backbone block $\mathcal{F}_{\text{sr}}$. This process can be formulated as:

$$\mathbf{F}^{l+1}_{sr} = \mathcal{F}_{\text{sr}}(\mathcal{F}_{\text{fa}}(\texttt{Concat}(\mathbf{F}^{l}_{texture}, \mathbf{F}^{l}_{shape})) + \mathbf{F}^{l}_{sr}). \tag{10}$$

For convenience, $\mathcal{F}_{\text{sr}}$ utilizes a recently proposed residual hybrid attention group [40], although it can be replaced with other advanced super-resolution backbones.

As shown in Fig. 3, our MSF module is repeated 12 times to fully fuse and leverage joint information from the cross-modality features. Finally, the resulting feature is processed through upsampling blocks as

$$\mathbf{N}_{sr} = \texttt{Norm}(\mathcal{F}_{\text{pub}}(\mathbf{F}^{l_{\max}}_{sr}) \oplus \mathcal{F}_{\text{bic}}(\mathbf{N}_{lr})), \tag{11}$$

where $\mathcal{F}_{\text{pub}}$ denotes the pixel-shuffle upsampling block, $\mathcal{F}_{\text{bic}}$ represents the *Bicubic* upsampling operation, and $\texttt{Norm}$ denotes the vector normalization operation, ensuring that the output normals remain unit vectors.

3.5 Loss Functions

To improve and optimize the training process of our *mn3DSSR*, we propose an integrated loss measurement framework, which consists of three loss items: (1) normal pixel loss $\mathcal{L}_{\text{pix}}$, (2) curl normal loss terms: $\mathcal{L}^{weight}_{\text{curl}}$ and $\mathcal{L}^{regular}_{\text{curl}}$, and (3) modality alignment loss terms: $\mathcal{L}^{texture}_{\text{align}}$ and $\mathcal{L}^{shape}_{\text{align}}$.

3.5.1 Normal Pixel Loss

In our previous work [11], we have employed a combination of $\ell 1$ loss and $\texttt{cosine}$ loss. However, we have identified certain limitations of these commonly used loss functions when applied to unit vectors in normal maps. For instance, $\ell 1$ loss exhibits a negative correlation with angular error beyond a certain threshold, which can affect the direction of gradient updates. Meanwhile, $\texttt{cosine}$ loss is functionally equivalent to mean squared error (MSE) for unit vectors, but it often results in slower convergence.

To address these issues, we propose to consider *normal angular error* (NAE), which shares properties with $\ell 1$ loss in measuring vector angular error. Specifically, the NAE loss satisfies the unit vector constraint in the foreground, and the MSE loss continues to be used in the background. This is primarily because the NAE loss becomes undefined or uninformative when computed on zero vectors in background regions. To address this issue, we introduce the normal pixel loss $\mathcal{L}_{\text{pix}}$, which applies distinct loss functions to foreground and background regions, respectively. In summary, our new normal pixel loss $\mathcal{L}_{\text{pix}}$ is defined as:

$$\begin{aligned}\mathcal{L}_{\text{pix}}(\mathbf{N}_{sr}, \mathbf{N}_{gt}) = &\frac{1}{HW}\sum_{p=1}^{HW} \mathbf{M}^{p} \times \underbrace{\texttt{arccos}(\gamma(\mathbf{N}^{p}_{sr} \cdot \mathbf{N}^{p}_{gt}))}_{\text{normal angular error}} \\ &+ \underbrace{\frac{1}{HW}\sum_{p=1}^{HW} (1 - \mathbf{M}^{p}) \times (\mathbf{N}^{p}_{sr} - \mathbf{N}^{p}_{gt})^{2}}_{\text{mean squared error}},\end{aligned} \tag{12}$$

where p denotes the pixel position, and $\cdot$ denotes the inner product. $\mathbf{M}$ represents the value of a binary object mask, with 1 indicating the foreground. $\mathbf{N}^{p}_{sr}$ and $\mathbf{N}^{p}_{gt}$ represent the unit vectors in $\mathbf{N}_{sr}$ and $\mathbf{N}_{gt}$, respectively. $\gamma(\cdot) = \texttt{min}(\texttt{max}(\cdot, \epsilon - 1), 1 - \epsilon)$ crops the input to the range $[\epsilon - 1, 1 - \epsilon]$ for the gradient of $\texttt{arccos}$ diverges at -1 and 1. ϵ denotes a small scalar, which is set to $\texttt{1e-5}$ in our implementation.

TABLE 2: **Effects of gradient kernel size on curl features**. Results are reported under the $\times 4$ setting on the *DiLiGenT*.

Kernel	PSNR	SSIM	MAE	MRDE	Time (ms)
2×2	28.7096	0.9287	3.7105	0.6932	**0.1795**
3×3	28.7131	**0.9290**	3.7079	**0.6770**	0.3745
5×5	28.7105	0.9283	3.7036	0.7073	0.3874
7×7	**28.7655**	0.9289	**3.6845**	0.7288	0.4491

3.5.2 Curl Normal Loss

In addition to the unit vector constraint, the local normals of a continuous surface usually needs to satisfy the integrability constraint in SfN. Non-integrable local normals indicate the presence of discontinuities or non-differentiable geometric details on 3D surface $\mathcal{S}_{\text{3D}}$. To consider these factors, we have designed a new normal curl function $\mathcal{F}_{\text{curl}}$ to measure the surface integrability of local normals.

To define our normal curl function, we consider a discrete scalar field $\mathbf{G}_{grid}$ defined on a regular 2D grid, and use a finite difference to approximate the differential operator: $\nabla \mathbf{G}_{grid} = (\nabla_x \mathbf{G}_{grid}, \nabla_y \mathbf{G}_{grid})$. In the experiments, we approximate the differential operators along x- and y-axis as convolutions with the kernels $[0,0,0;-1,0,1;0,0,0]$ and $[0,1,0;0,0,0;0,-1,0]$, respectively. As shown in Table 2, we compare the 2×2, 3×3, 5×5, and 7×7 kernel configurations, where the 3×3 kernel achieves the best MRDE and the second-lowest computational latency (on NVIDIA A100@40G). Moreover, the 2×2 kernel represents a composite structure of $[-1,1]$ for x-axis and $[1,-1]^{\text{T}}$ for y-axis. Larger kernels tend to blur curl features, reducing angular error due to smoothing local noise but slightly increasing MRDE by compromising global accuracy at discontinuities. Overall, the subtle MAE and MRDE variations indicate that the proposed curl feature is robust to kernel size.

With this approximation, we define the discrete curl operator between two inputs $(\mathbf{G}^1_{grid}, \mathbf{G}^2_{grid})$ as:

$$Curl(\mathbf{G}^1_{grid}, \mathbf{G}^2_{grid}) = \nabla_x \mathbf{G}^2_{grid} - \nabla_y \mathbf{G}^1_{grid}. \quad (13)$$

According to Stokes' theorem [41], the curl value of a gradient field obtained from a differentiable surface is zero everywhere. It can be proven that $Curl(\nabla \mathbf{G}_{grid}) \equiv 0$ based on the associativity of convolution operations. However, under a single-view projection, $\mathbf{D}_{lr}$ often contains discontinuous or non-differentiable regions, which cause inconsistencies between the gradients obtained from $\mathbf{N}_{lr}$ and those obtained by finite difference from $\mathbf{D}_{lr}$. In other words, this may result in a non-zero curl field, $\frac{\mathbf{N}_x}{\mathbf{N}_z} \neq \nabla_x \mathbf{D}_{lr}$ or $\frac{\mathbf{N}_y}{\mathbf{N}_z} \neq \nabla_y \mathbf{D}_{lr}$, where $\mathbf{N}_{x,y,z}$ represents three different components of $\mathbf{N}_{lr}$. Therefore, the curl field computed from the normal map can reflect sharp geometric details.

Inspired by this observation, we define a handcrafted curl function as

$$\mathcal{F}_{\text{curl}}(\mathbf{N}_{lr}) = \left| \mathtt{tanh}(Curl(\frac{\mathbf{N}_x}{\mathbf{N}_z}, \frac{\mathbf{N}_y}{\mathbf{N}_z})) \right| \odot \mathbf{M}, \quad (14)$$

where $|\cdot|$ denotes the absolute operation. $\mathbf{M}$ enables the curl feature to focus on the foreground. tanh is applied to compress the range of curl values.

Curl-weighted Normal Loss. To enhance the recovery of geometric details, we propose a curl-weighted normal loss $\mathcal{L}^{weight}_{\text{curl}}$. Specifically, we use the curl feature to weight the normal angular error, and $\mathcal{L}^{weight}_{\text{curl}}$ is defined as:

$$\mathcal{L}^{weight}_{\text{curl}}(\mathbf{N}_{sr}, \mathbf{N}_{gt}) = \sum_{p=1}^{HW} \frac{\mathcal{F}^p_{curl}(\mathbf{N}_{gt})}{||\mathcal{F}_{curl}(\mathbf{N}_{gt})||_1} \mathtt{arccos}(\gamma(\mathbf{N}^p_{sr} \cdot \mathbf{N}^p_{gt})). \quad (15)$$

Similar to other differential loss functions [42], [43], $\mathcal{L}^{weight}_{\text{curl}}$ can accelerate convergence and improve the restoration accuracy. We use the same loss type for the background but with different weights, which allows us to adjust the strength of the curl weighting via $\lambda^{weight}_{\text{curl}}$ in Eq. (3).

Curl-regularized Normal Loss. To mitigate the impact of noise and outliers introduced by *photometric stereo* setups, we further propose a curl-regularized loss $\mathcal{L}^{regular}_{\text{curl}}$, aimed at ensuring the consistency of local normals. As previously mentioned, the curl feature should be zero for smooth surfaces in the gradient field. Incorporating this loss helps to eliminate inconsistencies between local normals, thereby enhancing the overall restoration quality. Based on the curl feature, $\mathcal{L}^{regular}_{\text{curl}}$ is formulated as:

$$\mathcal{L}^{regular}_{\text{curl}}(\mathbf{N}_{sr}) = \frac{1}{HW} \|\mathcal{F}_{curl}(\mathbf{N}_{sr})\|_1. \quad (16)$$

3.5.3 Modality Alignment Loss

To emphasize the importance of fine-grained normal details for the texture alignment module, we upsample the aligned texture normal feature $\mathbf{F}_{tn}$ to obtain enhanced textures. Subsequently, we use the ground-truth texture normal map $\mathbf{N}^t_{gt}$ for supervision learning. As a result, the RGB-texture alignment loss $\mathcal{L}^{texture}_{\text{align}}$ is defined as:

$$\mathcal{L}^{texture}_{\text{align}}(\mathbf{F}_{tn}, \mathbf{N}^t_{gt}) = \|\mathcal{F}^{texture}_{\text{up}}(\mathbf{F}_{tn}) \oplus \mathcal{F}_{\text{bic}}(\mathbf{N}^t_{lr}) - \mathbf{N}^t_{gt}\|_1, \quad (17)$$

where $\mathcal{F}^{texture}_{\text{up}}$ represents a texture upsampling block composed of three RBs and several pixel-shuffle upsampling layers, as shown in Fig. 5 (c). The exact number of upsampling layers depends on the ratio τ, where one more upsampling layer is added when the value of τ doubles.

Similarly, we upsample the shape feature $\mathbf{F}_{sn}$ and use the ground-truth depth $\mathbf{D}_{gt}$ for supervision learning. This is tailored to assist the shape alignment module in emphasizing the significance of overall geometry features. The depth-shape alignment loss $\mathcal{L}^{shape}_{\text{align}}$ is defined as:

$$\mathcal{L}^{shape}_{\text{align}}(\mathbf{F}_{sn}, \mathbf{D}_{gt}) = ||\mathcal{F}^{shape}_{\text{up}}(\mathbf{F}_{sn}) \oplus \mathcal{F}_{\text{bic}}(\mathbf{D}'_{lr}) - \mathbf{D}_{gt}||_1, \quad (18)$$

where $\mathcal{F}^{shape}_{\text{up}}$ denotes a shape upsampling block that has the same network architecture as $\mathcal{F}^{texture}_{\text{up}}$. Meanwhile, we obtain an enhanced depth image $\mathbf{D}_{sr} = \mathcal{F}^{shape}_{\text{up}}(\mathbf{F}_{sn}) \oplus \mathcal{F}_{\text{bic}}(\mathbf{D}'_{lr})$.

3.6 Normal-based Multimodal Dataset

Several normal-based datasets have been established [44], [45], [46]. Nonetheless, we continue to face the following fundamental challenges in training our *mn3DSSR* model: (i) a notable scarcity of diverse 3D surface shape datasets, particularly within open-source repositories; (ii) significant difficulties in obtaining high-quality normal maps and corresponding multimodal data that represent fine-grained surface details and complex geometries; and (iii) the limited scale of existing high-quality normal datasets, which is typically insufficient in size to support the training of robust deep super-resolution models.

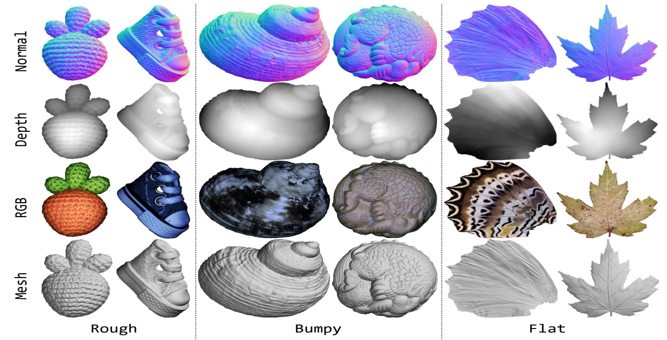
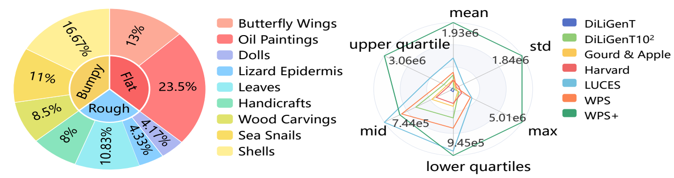


Fig. 7: **Typical examples from our multimodal dataset.** This dataset contains a diverse range of 3D objects, spanning from complex natural objects to intricate man-made items. The modalities shown from top to bottom are: *normal*, *depth*, *RGB*, and the corresponding mesh surface.



(a) Object Proportion (b) Vertex Distribution

Fig. 8: **Statistics of diverse and complex geometries in our *WPS+*.** (a) *Object Proportion*: Inner pie segments show concavity levels; outer segments show object class distribution. (b) *Vertex Distribution*: Comparison of normal-based datasets by vertex count, including mean, standard deviation, quartiles, and maximum.

To address these challenges, we propose to establish a dedicated and large-scale normal-based multimodal dataset acquired using *photometric stereo* setups. The essential motivation lies in the fact that *photometric stereo* is typically more accurate for computing surface normals than other methods based on single image prediction, such as shape-from-shading or deep learning models [34]. Fig. 7 shows typical samples from our dataset. The normal modality $\mathbf{N}_{gt}$ provides fine-grained geometry information. Meanwhile, the depth modality $\mathbf{D}_{gt}$ provides a continuous 3D surface constraint. In contrast, the RGB modality $\mathbf{I}_{gt}$ contains 18 images captured under diverse calibrated lightings, which represents complex texture and material features that provide rich visual cues for multimodal surface processing.

3.6.1 Dataset Improvement

We have initially established a normal-based multimodal dataset called *wonderful photometric stereo* (*WPS*) [11] to support the training of deep 3DSSR models. However, *WPS* suffers from several limitations. First, it captures RGB images using a camera sensor without applying *Gamma* correction, resulting in inaccuracies when acquiring surface normal maps. Second, it uses a single least squares-based *Lambertian* method [47] for synthesizing normal maps, which limits the overall quality of 3D surface reconstructions. Third, it contains only 400 objects, offering a foundation for surface shape analysis but lacking diversity and scale. Lastly, outdated data capture and processing techniques limit its ability to generate a high-quality 3DSSR dataset.

To address these shortcomings, we make six major improvements and construct one of the largest normal-based multimodal datasets, namely *WPS+*: (1) re-capturing low-quality samples to enhance overall data quality; (2) applying *Gamma* correction before obtaining surface normal maps to resolve inaccuracies; (3) using three different *photometric stereo*-based methods [48], [49], and [50] to significantly improve the quality of 3D surface reconstructions; (4) involving three professionals who spent over 1,000 hours carefully evaluating and selecting the best 3D reconstruction results to ensure high-quality samples; (5) expanding the dataset to 600 objects, offering better representation of diverse surface shapes; and (6) scaling the dataset with larger magnifications to create a comprehensive super-resolution dataset, including $\times 2$, $\times 4$, and $\times 8$ sampling settings.

To illustrate the overall statistics of diverse and complex geometries in our *WPS+*, we have visualized the sample types and vertices. For example, Fig. 8 (a) depicts the proportion of samples across various bump degrees and object types. As demonstrated, our dataset covers a wide range of surface features, such as *rough*, *bump*, and *flat*. Fig. 8 (b) shows the distribution of vertex counts within *WPS+* (*i.e.*, green line). After removing invalid background vertices, the average number of vertices is 1.93×10^6, and the maximum reaches 5.01×10^6. Since *photometric stereo*-based methods generate 3D surfaces with the same resolution as the resulted normal maps, each sample generally has high-resolution. Notably, our *WPS+* achieves the highest vertex counts among these datasets.

3.6.2 Dataset Quality Comparison

To demonstrate the dataset quality, we have conducted quantitative comparison among seven popular normal-based datasets, including *DiLiGenT* [44], *Gourd & Apple* [51], *Havard* [52], *LUCES* [45], *DiLiGenT*10^2 [53], *WPS* [11], and our newly constructed *WPS+*.

Specifically, we have evaluated these normal-based datasets across four key metrics: number of shape, resolution, entropy, and BRISQUE.

- 'Shape' represents the number of unique 3D surfaces contained in each dataset, demonstrating the richness of object shapes in each dataset.
- 'Pixel' shows the average and standard deviation of the number of normal pixels, which corresponds to the average normal resolution in each dataset.
- 'Entropy' [54] measures the uniformity of the normal pixel value distribution, which is used to quantify the complexity and diversity of content details.
- 'BRISQUE' [55] is a commonly used no-reference visual quality assessment metric, which is employed to quantify blurriness and noise in each dataset.

In Table 3, these quantitative metrics collectively demonstrate the advantages of our *WPS+* in terms of scale, resolution, diversity, and quality in comparison with mainstream normal-based benchmark datasets: (1) *WPS+* contains the largest number of unique 3D object shapes among all the evaluated datasets. (2) *WPS+* has the largest average number of normal pixels, indicating the highest resolution.

TABLE 3: **Dataset quality comparison of normal-based datasets.** Best and second-best results are shown in **bold** and <u>underlined</u>, respectively. Our WPS+ provides high-quality normal maps with greater complexity and scale.

Dataset	Shape (↑)	Pixel (↑)	Entropy (↑)	BRISQUE (↓)
Gourd&Apple	3	0.33(±0.06)Mpx	2.89(±0.79)	49.36(±1.41)
Harvard	7	0.39(±0.16)Mpx	**4.45**(±0.32)	**29.46**(±7.45)
DiLiGenT10[2]	10	0.52(±0.01)Mpx	3.38(±0.62)	46.16(±4.44)
DiLiGenT	10	0.06(±0.02)Mpx	2.47(±0.46)	48.42(±4.15)
LUCES	14	<u>1.35</u>(±0.44)Mpx	3.56(±0.66)	43.47(±5.35)
WPS	<u>400</u>	0.52(±0.18)Mpx	3.57(±0.66)	40.10(±7.41)
WPS+	**600**	**2.34**(±1.78)Mpx	<u>3.66</u>(±0.78)	<u>35.20</u>(±6.63)

(3) While not the highest, *WPS+* exhibits the second-highest diversity in image content and visual quality. As a result, the comprehensive nature of *WPS+* enables more robust and reliable evaluations of normal-based 3DSSR methods.

4 Experimental Validations

In this section, we discuss the implementation details of our *mn3DSSR*. We also compare our method against representative 3DSSR models on popular normal-based datasets and demonstrate various aspects of our contributions via a detailed ablation study. Additionally, we provide complexity analysis and evaluate performance across other 3D data representations (*i.e.*, *point cloud*, *mesh*, and *depth*).

4.1 Experimental Protocols

Normal-based Datasets. To prepare a comprehensive evaluation, we randomly divide our *WPS+* dataset into training, validation and testing sets, comprising 520, 30, and 50 samples, respectively. Both the training and testing sets are downsampled using the *Bicubic* method by factors of $\times\frac{1}{2}$, $\times\frac{1}{4}$, and $\times\frac{1}{8}$ to generate low-resolution inputs (*i.e.*, $\mathbf{N}_{lr}$, $\mathbf{D}_{lr}$, and $\mathbf{I}_{lr}$). We have trained all learning-based methods (excluding point cloud- and mesh-based) on the training set of our *WPS+* and tested all methods on the testing set.

In addition, we have conducted the **cross-dataset validation** on three small normal-based datasets: *DiLiGenT*, *Havard*, and *LUCES*, details of which are summarized in Table 4 together with *WPS+*. As seen, these datasets exhibit distinct characteristics that reflect the generalization capabilities of 3DSSR methods.

Evaluation Metrics. To comprehensively evaluate the overall performance of 3DSSR methods, we have adapted four metrics from the 2D image and 3D surface domains. In the 2D image domain, we employ the classic peak signal-to-noise ratio (PSNR) to measure normal pixel-level accuracy. Additionally, we use the structural similarity index measure (SSIM) to evaluate the structure similarity.

In the 3D surface domain, we utilize mean angular error (MAE) [50], [56], a widely used metric for normal-based 3D data representation, which converts the average per-pixel normal error into the angular value. We apply MAE to assess the reconstruction errors for the enhanced normal maps, as it is sensitive to local details. Furthermore, we adopt mean relative depth error (MRDE) [10], [57], another commonly used quality metric for depth-based 3D data representation, which evaluates the accuracy of the resulting 3D surface by comparing normalized depth images and is particularly sensitive to overall shape accuracy.

TABLE 4: **Characteristics of four normal-based datasets.** These datasets are selected to evaluate 3DSSR methods, based on their distinct features such as resolution, material diversity, and surface complexity.

Dataset	Samples	Characteristics
DiLiGenT	10	low-resolution, multiple materials, smooth surface, rich geometric shape.
Harvard	7	medium-resolution, relatively simple materials, moderate surface complexity, random texture details.
LUCES	14	high-resolution, multiple materials, complex artificial objects.
WPS+	600	high-resolution, multiple materials, complex natural objects.

4.2 Implementation Details

Our framework is implemented using *PyTorch*. For the model hyper-parameters, we employ two DMATGs and two SMATGs in the RGB-texture and depth-shape branches, where each group consists of six dual or single MAT blocks in the MSTA module. The maximum value of k is set to $k_{\max}$=14, and the maximum value of l is set to $l_{\max}$=12 in the MSF module. The number of feature channels is set to d_k=64. We train our *mn3DSSR* model for 1,000 epochs with a batch size of 12, employing the *Adam* optimizer with default parameters (β_1 = 0.9 and β_2 = 0.999). During training, we randomly crop the low-resolution version of $\mathbf{I}_{lr}$, $\mathbf{N}_{lr}$, and $\mathbf{D}_{lr}$ to 64×64 pixels and correspondingly crop the high-resolution images to $64\tau\times64\tau$ pixels. To avoid encountering empty content, we ensure that the cropped region contains at least 1% valid pixels.

Meanwhile, we apply simple data augmentation techniques, including random rotations (90°, 180°, and 270°) and horizontal flipping. It is important to note that for the calculation of the normal curl $\mathcal{F}_{\text{curl}}$, rotating the normal directions is necessary to maintain consistency with the corresponding normal map $\mathbf{N}_{lr}$.

Finally, we employ bilateral normal integration (BiNI) [37] as $\mathcal{F}_{\text{SfN}}$ in Eq. (1), for which the main justifications are highlighted as follows: (1) The implementation of BiNI is optimized with CUDA and supports the inputs of normal and depth maps, which may further accelerates the SfN reconstruction when a depth map is available. (2) Additionally, it is one of the most advanced SfN methods, capable of constraining the continuity of the reconstructed surface and ensuring stable optimization. It is noted that the depth ground-truths of the *photometric stereo* datasets are obtained using BiNI, while those for the *point cloud* and *mesh* datasets are derived directly from the original 3D meshes.

4.3 Comparison Settings

We have compared our *mn3DSSR* with several representative 3DSSR methods, categorizing them into six groups: (1) point cloud-based methods (denoted by 'Point'), (2) mesh-based methods (denoted by 'Mesh'), (3) voxel-based methods (denoted by 'Voxel'), (4) depth-based methods (denoted by 'Depth'), (5) normal-based methods (denoted by 'Normal'), and (6) other methods (denoted by 'Other'). To ensure a fair evaluation, all 2D-based learning methods (*e.g.*, 'Depth', 'Normal', and 'Other') are trained from scratch using the official settings and the same number of training iterations as *mn3DSSR*.

TABLE 5: **Quantitative comparison results on four normal-based datasets.** Multimodal methods with multiple input images are marked by '†'. '↑' means the higher the better, while '↓' means the lower the better. The best results are highlighted in **bold** for each dataset in each column.

Scale	Type	Method	*DiLiGenT* Dataset	*LUCES* Dataset	*Harvard* Dataset	*WPS+* Dataset
			PSNR↑/SSIM↑/MAE↓/MRDE↓	PSNR↑/SSIM↑/MAE↓/MRDE↓	PSNR↑/SSIM↑/MAE↓/MRDE↓	PSNR↑/SSIM↑/MAE↓/MRDE↓
×2	Points	*Qian2021CVPR* [12]	21.5205/0.7537/16.6481/10.4802	25.3958/0.8131/10.8209/4.6622	27.0114/0.8771/9.1227/4.0856	26.5784/0.7741/9.9124/4.5185
		Feng2022CVPR [13]	24.8409/0.8573/8.9281/5.1319	28.4936/0.8423/7.2500/3.6336	29.8471/0.8935/5.4839/2.4614	28.8998/0.7816/7.9061/4.0450
		He2023CVPR [7]	25.2609/0.8439/8.9925/4.7844	28.4786/0.8556/6.7594/4.4457	31.3664/0.9142/4.5513/1.5815	27.6150/0.8090/8.7612/3.9435
	Mesh	*Loop2008TOG* [14]	23.6890/0.8562/13.7712/6.0359	26.7048/0.9309/8.8377/3.2275	25.9211/0.9325/11.2417/2.6708	26.0932/0.8337/10.2676/4.6271
		Liu2020TOG [8]	21.0356/0.7771/14.1434/14.0244	29.2878/0.9373/5.5632/4.0861	26.8023/0.9107/6.1911/9.5300	27.3981/0.8169/8.2536/5.7965
	Voxel	*Shim2023CVPR* [16]	28.0317/0.8694/7.6986/2.3947	26.6993/0.8528/6.8165/2.1461	27.1100/0.8738/7.1167/1.8951	27.3601/0.7661/5.3871/1.1561
	Depth	*Voynov2019ICCV*† [17]	25.7812/0.8645/8.7205/2.4052	26.2306/0.9198/9.6347/0.4487	30.3008/0.9406/5.4318/1.6237	25.8256/0.8496/9.9071/0.6447
		Deng2021TPAMI† [19]	25.3162/0.8253/10.1997/1.8397	27.7277/0.8776/5.9519/0.3938	27.2236/0.8890/6.6401/1.4593	28.0729/0.8198/6.4739/0.6859
		Zhao2022CVPR† [20]	24.5019/0.8199/11.1954/1.1523	24.4583/0.8387/7.7851/0.3780	22.8610/0.8600/9.2004/1.6434	25.6361/0.7638/7.8814/0.6589
		Metzger2023CVPR† [10]	26.9412/0.8469/7.8396/1.2309	28.6380/0.8581/5.5886/0.4200	27.7262/0.8629/6.3207/1.4351	29.3251/0.8342/5.8326/0.6335
	Normal	*Ju2022IJCV*† [43]	23.1065/0.8319/14.7224/5.8263	20.0419/0.8247/19.6390/10.4813	31.0314/0.9608/6.1599/3.3774	26.8583/0.8367/10.1528/4.6685
		Xie2022CVPR† [11]	29.5686/0.9539/2.7782/0.6202	36.7395/0.9869/0.9747/0.3446	39.3625/0.9874/0.9221/0.4195	38.2878/0.9585/1.9099/0.5155
		Xie2023IJCAI† [23]	29.5122/0.9556/2.6582/0.5192	37.0027/0.9886/1.1021/0.3016	39.8172/0.9883/0.9033/0.3898	38.7200/0.9591/1.9046/0.4033
		Ours†	**30.0964/0.9653/2.2177/0.4792**	**37.8117/0.9904/0.9106/0.2014**	**39.9253/0.9894/0.9015/0.2817**	**39.1803/0.9603/1.8625/0.2623**
	Other	*Dong2016TPAMI* [4]	28.7417/0.9490/3.1792/0.8159	36.0270/0.9850/1.1318/5.1583	38.3080/0.9864/1.0522/1.3647	38.0379/0.9572/1.9538/7.5054
		Zhang2018ECCV [58]	29.2855/0.9536/2.8439/0.5010	36.4213/0.9866/0.9747/0.3087	38.8861/0.9868/0.9471/0.3454	38.4605/0.9589/1.8959/0.4184
		Zhang2021TPAMI [5]	29.1753/0.9518/2.9009/0.5328	36.2576/0.9862/1.0141/0.3245	38.5769/0.9864/0.9569/0.3665	38.3391/0.9588/1.9096/0.4911
		Chen2021CVPR [59]	26.2368/0.9212/4.3759/0.7457	33.5741/0.9791/1.4287/0.3557	33.9233/0.9758/1.4672/0.3258	36.4797/0.9542/2.0941/0.4106
		Ma2022TPAMI [42]	25.7202/0.9114/5.8661/2.1216	33.0455/0.9766/2.8939/2.1873	32.8075/0.9716/2.8809/2.7745	32.8602/0.9222/3.9467/3.8088
		Saharia2023TPAMI [6]	25.2859/0.8968/5.7538/1.5450	32.1990/0.9626/2.6178/1.5501	32.4225/0.9610/2.3954/1.5745	29.5368/0.7763/6.2832/1.7206
		Zamir2023TPAMI [60]	28.3680/0.9459/3.2054/1.0574	35.6948/0.9846/1.0785/0.3968	36.8507/0.9841/1.0428/0.4628	38.3083/0.9579/1.9308/0.5581
		Chen2023CVPR [40]	29.9392/0.9576/2.7236/0.4848	37.2532/0.9881/0.9942/0.2536	39.4886/0.9878/1.0232/0.3013	38.8492/0.9599/1.9038/0.3251
		Li2019CVPR† [24]	29.9861/0.9592/2.7348/0.5248	36.5975/0.9868/1.0311/0.4327	39.6884/0.9889/0.9567/0.4356	38.8153/0.9592/1.9113/0.5371
		Deng2021TIP† [61]	27.2352/0.9346/3.7392/0.6016	34.5740/0.9826/1.2210/0.4908	34.4011/0.9802/1.2171/0.3861	36.6442/0.9567/2.0492/0.3820
		Georgescu2023WACV† [25]	28.7764/0.9461/3.2204/0.4913	35.4533/0.9835/1.1103/0.2901	39.5173/0.9886/0.9032/0.2868	38.1612/0.9574/1.9345/0.3913
×4	Points	*Qian2021CVPR* [12]	19.4619/0.6735/22.7640/12.4302	23.9298/0.8271/12.3992/5.2151	23.8567/0.8335/12.9725/7.7488	23.7155/0.7129/12.9399/5.0828
		Feng2022CVPR [13]	22.6214/0.7716/13.6446/8.7235	27.1583/0.8423/7.9873/3.5863	27.5179/0.8863/7.0508/4.7806	26.7534/0.7734/9.0821/2.6533
		He2023CVPR [7]	23.8644/0.8089/10.5820/6.5163	28.8689/0.8678/6.3430/2.8543	29.9116/0.9159/5.0185/2.3387	26.0900/0.7732/9.4049/2.8089
	Mesh	*Loop2008TOG* [14]	21.3994/0.7698/17.1752/10.6798	25.8076/0.8930/8.8072/4.2971	24.8308/0.8984/12.1518/6.4080	25.4176/0.7944/10.7641/3.6463
		Liu2020TOG [8]	20.0684/0.6989/16.8268/14.3957	22.6429/0.4944/15.4329/2.3094	24.2402/0.6128/11.3384/10.0990	27.1892/0.8024/8.0731/4.4842
	Voxel	*Shim2023CVPR* [16]	26.4087/0.8244/9.1314/2.5167	24.3759/0.7831/9.0512/3.1556	26.6625/0.8505/8.7521/2.9069	25.1194/0.7209/7.5280/1.6833
	Depth	*Voynov2019ICCV*† [17]	25.2444/0.8456/9.4351/2.5317	26.0950/0.9076/9.8388/0.5438	30.2157/0.9340/5.5844/2.6285	25.9591/0.8398/9.9270/0.8415
		Deng2021TPAMI† [19]	24.3490/0.7959/12.0763/1.8249	25.9906/0.7993/7.8407/0.4624	26.6179/0.8571/7.5854/2.5280	25.4388/0.7217/8.9398/0.7081
		Zhao2022CVPR† [20]	26.0160/0.8242/9.4762/0.7470	27.8537/0.8389/6.3721/0.4172	27.8165/0.8760/6.4804/2.4598	27.7418/0.7554/7.0776/0.6842
		Metzger2023CVPR† [10]	24.2559/0.7208/11.5112/0.7523	26.0775/0.7384/8.1720/0.4604	25.9317/0.7668/8.1994/2.4324	27.4071/0.7265/7.5291/0.6494
	Normal	*Ju2022IJCV*† [43]	23.6945/0.8169/14.3491/5.3658	20.5209/0.8193/21.1528/10.9605	31.7100/0.9563/5.9421/3.3935	25.8775/0.7927/11.1832/4.5518
		Xie2022CVPR† [11]	27.8224/0.9156/4.4862/1.1880	33.9617/0.9614/2.0019/0.4984	35.8632/0.9726/1.4867/0.5728	34.8750/0.9042/3.0366/0.6762
		Xie2023IJCAI† [23]	27.1040/0.9193/4.1405/0.9460	34.3098/0.9670/2.0200/0.4215	36.0767/0.9741/1.4306/0.4462	35.1734/0.9104/3.0038/0.5727
		Ours†	**28.7131/0.9290/3.7079/0.6770**	**35.7181/0.9723/1.6818/0.3813**	**36.2563/0.9784/1.2877/0.4050**	**35.5205/0.9120/2.8670/0.5234**
	Other	*Dong2016TPAMI* [4]	25.1860/0.8861/5.8101/1.4976	32.4057/0.9508/2.4634/7.0355	33.0354/0.9645/1.8742/2.7786	33.3222/0.8964/3.2684/9.4412
		Zhang2018ECCV [58]	25.8061/0.9001/5.2445/0.8978	32.9987/0.9594/2.0921/0.4615	34.0208/0.9684/1.6499/0.4875	34.3417/0.9031/3.0691/0.6060
		Zhang2021TPAMI [5]	25.1623/0.8925/5.6062/0.8816	32.9208/0.9594/2.0792/0.4516	33.5749/0.9678/1.6468/0.4804	33.7713/0.9006/3.1508/0.7009
		Chen2021CVPR [59]	23.9179/0.8684/7.1466/1.2134	31.2852/0.9519/2.6992/0.8033	31.3168/0.9577/2.4145/1.0713	33.2212/0.8938/3.6183/1.7222
		Ma2022TPAMI [42]	25.1041/0.8533/8.5745/3.3527	30.9190/0.9213/4.9812/2.7102	31.9418/0.9470/4.1858/4.3094	31.0938/0.8342/5.2829/3.1220
		Saharia2023TPAMI [6]	25.3546/0.8377/8.5486/4.8608	31.5897/0.9057/5.0092/3.3019	32.2972/0.9231/4.6888/3.6931	30.5059/0.7965/5.8525/3.6009
		Zamir2023TPAMI [60]	26.3604/0.9009/5.1851/1.3666	33.7176/0.9583/2.1165/0.4545	35.2186/0.9700/1.5926/0.5277	34.8140/0.9024/3.0810/0.5602
		Chen2023CVPR [40]	26.8292/0.9119/4.7237/0.8484	34.1865/0.9661/1.9916/0.4920	35.2483/0.9725/1.5995/0.4812	34.7780/0.9075/3.0185/0.6914
		Li2019CVPR† [24]	26.7922/0.9086/4.8960/0.8713	33.5816/0.9598/2.0735/0.4642	35.0161/0.9708/1.5974/0.4894	34.6881/0.9042/3.0344/0.5569
		Deng2021TIP† [61]	26.0174/0.9057/5.1768/0.8069	33.1432/0.9613/2.1834/0.4992	34.2250/0.9707/1.6862/0.4755	34.3588/0.9069/3.2309/0.5932
		Georgescu2023WACV† [25]	26.1868/0.8946/5.6397/1.1730	32.4522/0.9516/2.5479/0.6621	35.1024/0.9685/1.8641/0.8663	34.1142/0.9001/3.2489/1.0604
×8	Points	*Qian2021CVPR* [12]	16.9014/0.6000/31.2461/13.3988	22.0753/0.8089/16.3038/7.7652	21.5358/0.7961/17.2581/9.9125	19.0508/0.5775/26.3291/11.0666
		Feng2022CVPR [13]	19.9236/0.6903/20.8992/11.3735	24.4549/0.8214/11.0594/6.4862	23.5167/0.8311/12.5423/8.8582	25.1123/0.7555/11.5991/1.2533
		He2023CVPR [7]	21.0649/0.7110/16.3457/9.0166	25.8029/0.8399/8.5403/5.7959	26.6067/0.8719/8.0658/5.5851	24.9251/0.7468/10.5306/3.9372
	Mesh	*Loop2008TOG* [14]	19.1102/0.6879/22.6620/12.3683	24.1792/0.8472/10.9330/7.3818	23.1042/0.8544/14.2458/9.5637	24.1412/0.7534/12.1212/6.0607
		Liu2020TOG [8]	18.4508/0.6713/22.2178/14.5972	24.2564/0.6024/11.5519/6.2898	23.6396/0.7544/10.1586/12.7474	24.8277/0.6448/11.0250/5.5614
	Voxel	*Shim2023CVPR* [16]	20.1724/0.6133/19.7176/4.2958	21.2187/0.7022/16.6011/3.4408	21.3996/0.7190/15.3591/3.5065	22.0704/0.6650/13.1685/2.6224
	Depth	*Voynov2019ICCV*† [17]	24.3711/0.8029/11.3918/2.6537	25.6484/0.8696/10.6182/1.2231	29.4428/0.9092/6.2227/3.5769	20.3142/0.4840/17.6697/0.9027
		Deng2021TPAMI† [19]	22.4909/0.7512/14.9900/2.5119	23.6757/0.6695/11.6895/0.4948	25.8947/0.8439/8.6121/3.6068	23.8492/0.6218/10.4165/0.8207
		Zhao2022CVPR† [20]	22.8280/0.7262/14.2287/1.9137	25.7689/0.8209/8.0283/0.4200	26.0626/0.8546/8.0135/3.5380	26.1740/0.7253/8.2435/0.8027
		Metzger2023CVPR† [10]	22.1654/0.6631/14.5418/1.9036	25.3585/0.7777/8.4643/0.4183	25.0190/0.7899/8.9110/3.4506	26.7641/0.7190/8.0038/0.7867
	Normal	*Ju2022IJCV*† [43]	20.0871/0.7158/21.6305/6.5890	19.6450/0.7796/23.4022/11.4969	28.7485/0.9105/8.2786/3.8103	25.9611/0.7780/10.9498/4.4978
		Xie2022CVPR† [11]	25.0843/0.8393/8.0426/2.3772	31.5689/0.9115/3.7768/0.6222	33.2322/0.9480/2.4416/0.5620	31.9214/0.8464/4.3476/0.7362
		Xie2023IJCAI† [23]	25.2703/0.8423/8.0434/1.9133	31.4897/0.9172/3.7653/0.5306	33.5175/0.9542/2.2767/0.4802	32.1371/0.8522/4.1615/0.6376
		Ours†	**25.7237/0.8625/7.1961/1.8227**	**32.2302/0.9221/3.4651/0.3957**	**34.0295/0.9558/2.1642/0.3903**	**32.6014/0.8607/3.9955/0.4627**
	Other	*Dong2016TPAMI* [4]	22.3600/0.7986/9.8243/3.5007	29.0747/0.8911/4.4773/9.0862	30.5524/0.9286/3.0934/4.4478	30.2180/0.8261/4.8554/10.4912
		Zhang2018ECCV [58]	23.0081/0.8139/9.0414/2.0897	30.0082/0.9099/3.9646/0.5352	31.0623/0.9434/2.6928/0.5913	31.1540/0.8467/4.3909/0.6581
		Zhang2021TPAMI [5]	22.6106/0.8113/9.3422/2.0341	29.7349/0.9068/4.0889/0.5316	30.6502/0.9404/2.8075/0.5147	30.9765/0.8447/4.4534/0.6886
		Chen2021CVPR [59]	22.5603/0.8109/9.8191/1.9706	29.5980/0.9066/4.0774/0.4327	30.2652/0.9377/2.9600/0.4582	30.8966/0.8420/4.5104/0.7379
		Ma2022TPAMI [42]	21.2590/0.7354/16.1192/8.4633	25.8630/0.8311/9.8113/4.9456	26.5519/0.8875/8.6186/7.0718	26.2264/0.7426/9.4538/7.1149
		Saharia2023TPAMI [6]	24.6408/0.7985/9.9772/3.4295	30.4030/0.8507/5.8307/1.6300	32.3771/0.9060/3.9114/1.6186	30.1325/0.7771/6.0019/1.6160
		Zamir2023TPAMI [60]	23.9001/0.8220/8.8270/2.8057	30.3411/0.9048/4.0399/0.6615	32.0807/0.9406/2.7081/0.6791	31.6386/0.8393/4.6032/1.5153
		Chen2023CVPR [40]	23.9123/0.8337/8.5006/1.9669	30.8567/0.9145/3.7527/0.5374	32.3350/0.9493/2.4624/0.5574	31.7032/0.8526/4.2188/0.5343
		Li2019CVPR† [24]	23.4429/0.8209/9.0796/1.9885	29.8536/0.9058/4.0870/0.5756	30.1759/0.9386/2.9066/0.5516	31.2784/0.8472/4.3786/0.5535
		Deng2021TIP† [61]	24.0620/0.8318/8.3685/2.3549	30.5195/0.9073/3.9345/0.6036	30.1044/0.9404/2.8382/0.6794	31.6936/0.8463/4.3476/0.6529
		Georgescu2023WACV† [25]	22.7995/0.8023/9.8721/2.3170	29.0951/0.8958/4.5168/0.9210	30.9109/0.9352/2.9789/0.8344	30.7917/0.8381/4.6318/0.9310

- For point cloud-based methods, we select three learning-based methods: *Qian2021CVPR* [12], *Feng2022CVPR* [13], and *He2023CVPR* [7]. In the experiment, we have sampled 3D objects into point clouds while ensuring that the overall number of points is comparable to that of the normal or depth pixels. We first apply these three methods and then reconstruct the upsampled point clouds into a mesh surface. Finally, we render the reconstructed meshes back into depth and normal maps for evaluation.
- For mesh-based methods, we select a widely used classic mesh subdivision method *Loop2008TOG* [14] and a recent learning-based method *Liu2020TOG* [8]. It is important to note that *Liu2020TOG* requires closed 3D meshes. Therefore, we utilize the extrude operation in the *PyVista* to create watertight test meshes.
- For voxel-based methods, we select a diffusion-based generative method *Shim2023CVPR* [16]. We have fine-tuned the super-resolution stage based on a pre-trained model and utilized the *PySDF* library to convert the test data into voxel-shaped signed distance fields (SDF). Due

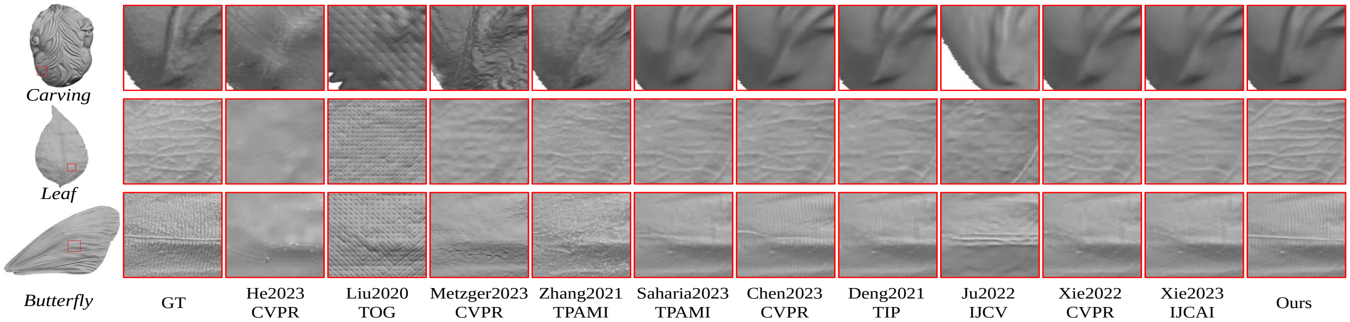


Fig. 9: **Visual comparison for the ×8 setting on *WPS+* dataset**. 'GT' means the ground-truth surface, as highlighted in the red box. It can be observed that the micro geometric structures on the *Carving*, the leaf stems on the *Leaf*, and the natural strip-like texture on the *Butterfly* are better restored by the proposed method.

to substantial storage and computational requirements, we crop all 3D surfaces into 64×64 patches during both fine-tuning and testing. Finally, SDF-based voxels are reconstructed and reassembled into 3D meshes.

- For depth-based methods, we choose three cutting-edge DSR techniques: *Voynov2019ICCV* [17], *Zhao2022CVPR* [20] and *Metzger2023CVPR* [10]. These methods require high-resolution RGB images, which is incompatible with our setting. To ensure a fair comparison, we upsample the low-resolution RGB images using a recent 2DISR [40]. When calculating the evaluation metrics, we estimate normal maps from the enhanced depth images through finite difference and vector normalization.
- For normal-based methods, we compare our *mn3DSSR* with multimodal normal-based methods *Xie2023IJCAI* [23] and *Xie2022CVPR* [11]. Considering the close relationship between our proposed framework and *photometric stereo*-based surface reconstruction, we have also extended *Ju2022IJCV* [43] to our setting for comparison, combining their super-resolution method [22].
- For other methods, considering that 2DISR methods can also be adapted to our scenario by replacing RGB images with normal maps, we have included comparisons with representative 2DISRs. We further classify them into unimodal and multimodal categories:
  1) For unimodal 2DISRs, we have selected the first convolution-based method *Dong2016TPAMI* [4], channel attention-based method *Zhang2018ECCV* [58], dense connection-based method *Zhang2021TPAMI* [5], GAN-based method *Ma2022TPAMI* [42], diffusion-based method *Saharia2023TPAMI* [6], attention mechanism-based networks *Chen2021CVPR* [59], *Zamir2023TPAMI* [60], and the current leading method *Chen2023CVPR* [40].
  2) For multimodal 2DISRs, we choose *Li2019CVPR* [24], *Deng2021TIP* [61], and *Georgescu2023WACV* [25] to represent hybrid fusion methods across different domains. Since these methods primarily accept two input modalities, we use the normal map as the main modality and the brightest RGB images as the auxiliary modality to the 3DSSR task.

4.4 Performance Comparisons

4.4.1 Quantitative Performance

To thoroughly demonstrate the performance comparison between our *mn3DSSR* and 24 representative methods, we have conducted experiments on four normal-based benchmark datasets. As shown in Table 5, point cloud-based methods [7], [12], [13] and mesh-based methods [8], [14] generally yield poorer results compared to normal-based methods [11], [23] and depth-based methods [17], [20]. This discrepancy can primarily be attributed to the sparsity and irregularity of point cloud and mesh representations, which hinder the learning of intricate geometric features. Voxel-based method [16] also produces suboptimal results, which can be attributed to its restricted resolution, hindering the effective modeling of long-range interactions.

Depth-based methods [10], [17], [20] typically achieve promising performance, highlighting the advantages of utilizing 2D data representations. However, despite leveraging RGB information as guidance, depth-based methods still face challenges in improving super-resolution performance in terms of four quality metrics. For normal-based methods, *Ju2022IJCV* [43] effectively recovers certain geometric details from a substantial number of low-resolution multi-illumination RGB images. Nonetheless, due to the inherent complexities of *photometric stereo*-based surface reconstructions, the resulting normal maps often exhibit significant discrepancies from the ground-truth. In contrast, *Xie2022CVPR* [11] and *Xie2023IJCAI* [23] utilize multimodal information to enhance the effectiveness of 3DSSR, achieving superior performance compared to existing methods. Meanwhile, our proposed *mn3DSSR* achieves state-of-the-art results across various scaling ratios.

4.4.2 Qualitative Performance.

To demonstrate visual performance, we have provided the 3DSSR results on four normal-based datasets. Fig. 9 shows typical enhanced examples from the *WPS+* dataset. As observed, our *mn3DSSR* is capable of effectively recovering complex textures while being less susceptible to noise introduced during *photometric stereo* acquisition. In contrast, other methods struggle to recover sharp geometric structures (*e.g.*, *He2023CVPR*) or produce distorted details (*e.g.*, *Liu2020TOG*, *Metzger2023CVPR*, and *Zhang2021TPAMI*).

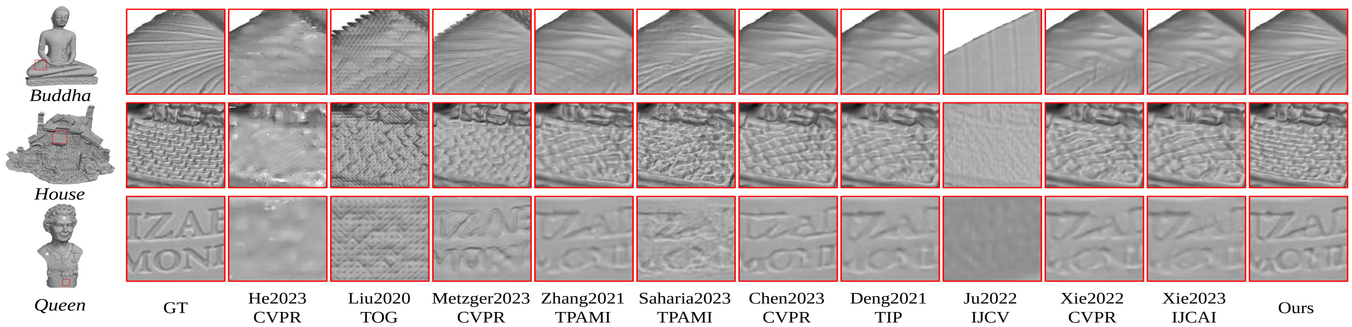


Fig. 10: **Visual comparison for the $\times 8$ setting on the *LUCES* dataset**. 'GT' means the ground-truth surface, as highlighted in the red box. It can be observed that fine-grained details like the intricate patterns on the *Buddha*, the tiles on the *House*, and the carved text on the *Queen* are better restored by the proposed method.

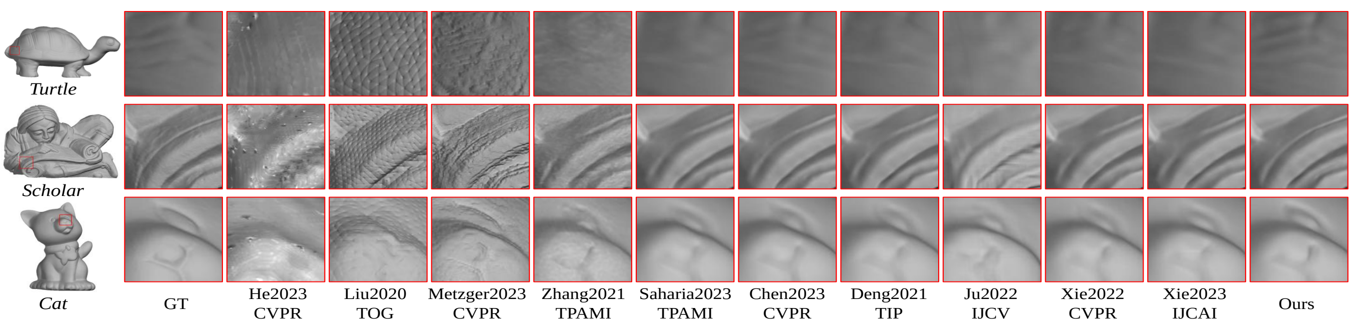


Fig. 11: **Visual comparison for the $\times 8$ setting on the *Harvard* dataset**. 'GT' means the ground-truth surface, as highlighted in the red box. It can be observed that surface textures on the *Turtle*, engraved lines on the *Scholar*, and eyes of the *Cat* are better recovered by the proposed method.

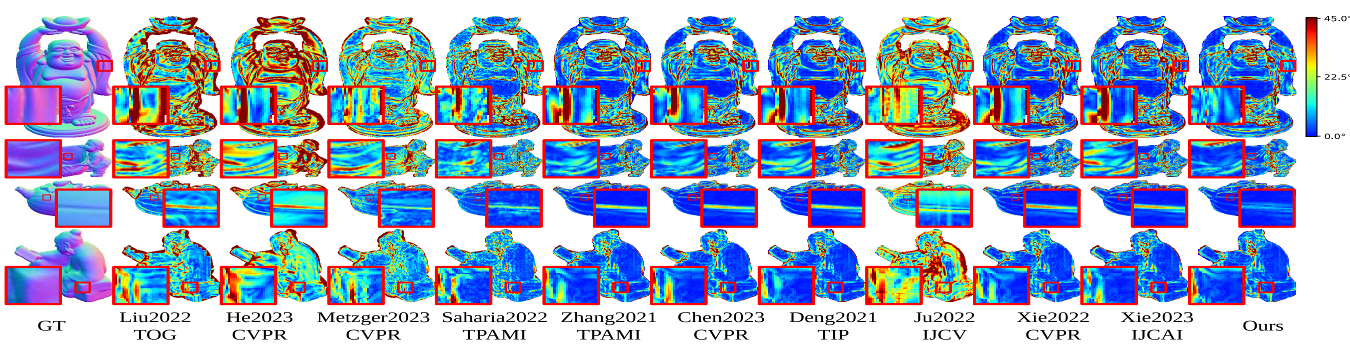


Fig. 12: **Visual comparison for $\times 8$ setting on the *DiLiGenT* dataset**. 'GT' denotes the ground-truth normal. Heatmaps show MAE in the range $(0°, 45°)$ for better comparison. Due to limited geometric complexity in *DiLiGenT*, we highlight four challenging samples: *buddha*, *harvest*, *pot1*, and *reading*. Our method better preserves edge details with lower errors.

Fig. 10 depicts the super-resolution results on the *LUCES* dataset, which contains more artificial objects and precise ground-truth normals. It can be seen that the proposed *mn3DSSR* successfully restores regular textures, whereas other methods produce erroneous patterns. Among the compared methods, *Ju2022IJCV* is difficult to recover basic geometric shapes due to the influence of non-Lambertian materials and near-field lighting effects. Other methods (*e.g.*, *Chen2023CVPR* and *Deng2021TIP*) are affected by aliasing effects and tend to produce erroneous patterns.

Fig. 11 shows the super-resolution results on the *Harvard* dataset. It can be observed that for the fine incised lines on these enhanced samples, our *mn3DSSR* achieves sharper and more accurate restorations compared to other methods. In contrast, existing methods either produce subtle non-existent textures (*e.g.*, *Liu2020TOG* and *Metzger2023CVPR*) or overly smooth results (*e.g.*, *Saharia2023TPAMI* and *Chen2023CVPR*).

Fig. 12 presents error maps for a detailed visualization on the *DiLiGenT* dataset. Although the *DiLiGenT* dataset has a much smaller sample size and smoother surfaces, our *mn3DSSR* still restores more accurate geometric structures compared to other methods, verifying its strong generalization capability. Compared to our method, most of the compared methods exhibit large overall restoration errors. For example, *Xie2022CVPR* and *Xie2023IJCAI* achieve better

TABLE 6: **Ablation study of normal, RGB, and depth modalities**. The '-' symbol indicates that the corresponding modality is replaced with normal maps. Four quality metrics are obtained under the ×4 setting on the *DiLiGenT*.

Normal	RGB	Depth	PSNR	SSIM	MAE	MRDE
✓	-	-	27.1924	0.9122	4.5054	0.9274
✓	✓	-	28.0503	0.9205	3.9200	0.8703
✓	-	✓	27.8570	0.9167	4.1053	0.9188
✓	✓	✓	**28.7131**	**0.9290**	**3.7079**	**0.6770**

results but still incur large errors in geometric details.

In summary, these presented visual results demonstrate the superiority of our *mn3DSSR* for normal-based 3DSSR tasks. The ability of our method to recover intricate textures, mitigate acquisition noise, and maintain accurate geometric structures across four normal-based datasets highlights its effectiveness and robustness.

4.5 Ablation Study

In this section, we evaluate both the individual and combined contributions of different modalities, as well as the specially designed network modules in our *mn3DSSR*.

4.5.1 Modality Ablation

To thoroughly demonstrate the rationale behind our modality selection, we have conducted the experiments on various combinations of modalities on *DiLiGenT* under the ×4 setting. In the experiments, we replace the additional modality with the normal modality, without altering the network architecture, the results of which are listed in Table 6. As seen, the results indicate that the modalities chosen in our *mn3DSSR* are both reasonable and effective, with the best performance achieved when all three modalities are utilized.

Notably, using RGB alone can yield better performance than using depth alone, as reflected in the angular and relative depth error metrics. This is primarily because RGB information tends to correlate more strongly with surface normals, as lighting variations in multi-illumination images encode rich geometric cues directly influenced by surface orientation. In contrast, the depth modality often provides less fine-grained detail for accurately inferring normals, particularly when it is low-resolution or noisy.

4.5.2 Module Ablation

To verify the effectiveness of the MPS, MSTA, and MSF modules, we have further conducted additional experiments with the same settings as described above. Five independent experiments have been carried out as shown in Table 7, where the selected (not selected) modules are represented by the check-mark symbol '✓' ('-').

To validate the effectiveness of MPS, we have replaced the brightest and darkest RGB images with the average one. For the normal modality, we have omitted the frequency separation process (*e.g.*, $\mathbf{N}_{lr} = \mathbf{N}_{lr}^{t} + \mathbf{N}_{lr}^{s}$), while for the depth modality, we have removed the normalization and position encoding processes. To evaluate the effectiveness of MSTA and MSF, we have substituted them with simple concatenation and stacking of RBs, ensuring that the model size is comparable to the original. As shown in Table 6, the best results are obtained when all three modules are utilized.

TABLE 7: **Ablation study of MPS, MSTA, and MSF Modules**. The '-' symbol indicates that the corresponding module is replaced with residual blocks having a similar number of model parameters on the ×4 *DiLiGenT*.

MPS	MSTA	MSF	PSNR	SSIM	MAE	MRDE
-	-	-	27.9204	0.9205	4.0013	0.8990
✓	-	-	27.9833	0.9206	3.9141	0.8500
✓	✓	-	28.4257	0.9248	3.8667	0.7015
✓	-	✓	28.3062	0.9237	3.9025	0.7949
✓	✓	✓	**28.7131**	**0.9290**	**3.7079**	**0.6770**

TABLE 8: **Performance of the subpixel-translation on the super-resolution of surface normals**. Two quality metrics are obtained under the ×4 setting on the *DiLiGenT*.

Translation Level	*Chen2023CVPR* [40]		*Ours*	
	MAE	MAE$_{trans}$	MAE	MAE$_{trans}$
0 pixel	4.7237	0.0000	3.7079	0.0000
1/4 pixel	4.7798	3.0283	3.7409	2.4353
2/4 pixel	4.9338	3.9015	3.7311	3.0993
3/4 pixel	4.7358	3.1646	3.7309	2.4146
4/4 pixel	4.8729	0.6961	3.7468	0.4083

4.6 Subpixel-translation Analysis

To evaluate subpixel-translation consistency, we have conducted experiments under the ×4 setting on the *DiLiGenT*, as shown in Table 8. We adopt the MAE$_{trans}$ to measure the angle difference between the outputs before and after subpixel-translation, where a smaller value indicates greater consistency. As seen, integer-pixel translations preserve the aliasing pattern in the input normal maps, while fractional translations (*e.g.*, 1/2-pixel shifts) result in the largest discrepancies due to altered aliasing. Nonetheless, our method consistently yields lower MAE and MAE$_{trans}$ values than the recent 2DISR method *Chen2023CVPR*, demonstrating improved robustness to subpixel misalignments.

4.7 Hyper-parameter Analysis

To demonstrate the impact of loss function weights, we have employed a greedy approach to select the optimal values for different loss terms by iteratively searching at certain intervals. Specifically, we first fix the weight of the normal pixel loss at 1 and then search for the optimal weights for $\lambda_{\text{curl}}^{weight}$, $\lambda_{\text{curl}}^{regular}$, and λ_{align}. After determining the optimal value for $\lambda_{\text{curl}}^{weight}$, we repeat the process for $\lambda_{\text{curl}}^{regular}$ as well as for λ_{align}. The overall results of three hyper-parameters are provided as illustrated in Fig. 13. In addition, we have the following observations based on our extensive experimental results:

- $\lambda_{\text{curl}}^{weight}$ accelerates the convergence speed, excessively large weights may lead to unstable training.
- $\lambda_{\text{curl}}^{regular}$ imposes certain constraints on the restoration noise in the output normals, but large weights can result in overly smooth super-resolution results.
- λ_{align} is beneficial for model training, but it may compete with the primary super-resolution task, leading to performance degradation when assigned high weights.

Therefore, we choose $\lambda_{\text{curl}}^{weight} = 0.25$, $\lambda_{\text{curl}}^{regular} = 0.1$, and $\lambda_{\text{align}} = 0.5$ to train our *mn3DSSR*.

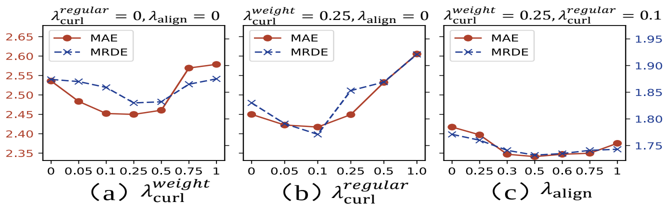


Fig. 13: **Effects of the hyper-parameter sensitivity for** $\lambda_{\text{curl}}^{weight}$, $\lambda_{\text{curl}}^{regular}$, and λ_{align}. The MAE and MRDE results are computed under the $\times 4$ setting on our *WPS+* dataset.

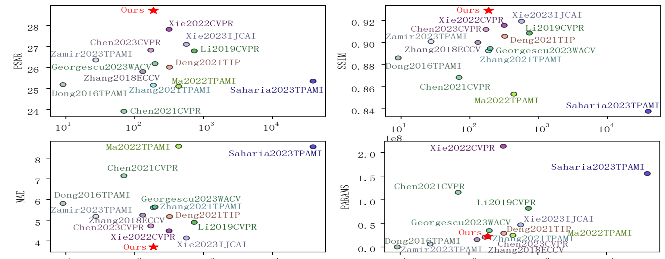


Fig. 14: **Complexity comparison of model parameters and computations.** The x-axis denotes the multiplication-addition operation (Mult-Adds), while the y-axis represents PSNR, SSIM, MAE, and PARAMS, respectively. Results are computed under the $\times 4$ setting on the *DiLiGenT*.

TABLE 9: Effects of the `cosine` loss, the $\ell 1$ loss, and the proposed normal pixel loss $\mathcal{L}_{\text{pix}}$. Four quality metrics are obtained under the $\times 4$ setting on the *DiLiGenT*.

loss	PSNR	SSIM	MAE	MRDE
`cosine`	28.2097	0.9229	3.9826	0.8261
$\ell 1$	28.1144	0.9221	3.9524	0.8001
$\mathcal{L}_{\text{pix}}$	28.3299	0.9280	3.7661	0.7399
$\mathcal{L}_{\text{pix}}+\mathcal{L}_{\text{curl}}^{*}+\mathcal{L}_{\text{align}}^{*}$	28.7131	0.9290	3.7079	0.6770

In addition, we have also conducted additional ablation studies to evaluate the performance gains of the proposed normal pixel loss $\mathcal{L}_{\text{pix}}$ compared to traditional $\ell 1$ and `cosine` losses. As seen in Table 9, it shows that using only $\mathcal{L}_{\text{pix}}$ yields consistently better performance than using either the $\ell 1$ or `cosine` loss alone.

4.8 Complexity Analysis

To illustrate the model size and computational complexity, we have further carried out additional experiments comparing our *mn3DSSR* with other approaches. Since different models have significant differences in network architecture, to maintain comparison fairness, we primarily evaluate the storage and computational costs with representative unimodal and multimodal super-resolution methods. Specifically, we calculate the parameter count (PARAMS) and the number of multiplication and addition operations (Mult-Adds) with the *torchinfo* library to evaluate the space and time complexity.

As shown in Fig. 14, our *mn3DSSR* demonstrates acceptable computational overhead when compared to recent advanced multimodal 3DSSR methods. Notably, our method significantly reduces the computational complexity compared to *Xie2022CVPR*. While it may not have the fewest model parameters compared to other 2DISR models, our *mn3DSSR* achieves one of the best results, effectively balancing computational overhead and super-resolution performance.

4.9 Validation of Other 3D Data Representations

As illustrated in Fig. 2, our *mn3DSSR* can be applied to a wide range of 3D representations by data conversion. To validate its scalability, we have further conducted comparisons across *point cloud*, *mesh*, and *depth* datasets.

4.9.1 Evaluation on Point Cloud Dataset

For point cloud representation, we have performed experiments following the validation method suggested by [13] on the *Sketchfab2* dataset [13]. To convert point cloud data into multimodal images suitable for *mn3DSSR*, we use point splatting to synthesize the required normal, depth, and RGB images from 64 viewpoints for each sample. It is noted that since *Sketchfab2* does not provide RGB information, we use the default diffuse white material to render the RGB image.

To conduct a unified evaluation, we convert the results of all comparison methods into meshes and render them as multi-view images for quantitative metric calculations. Specifically, for the proposed *mn3DSSR*, we use the *SuperNormal* [62] to reconstruct a complete mesh from the enhanced multi-view normal maps. For other methods, we employ the neural kernel surface reconstruction (NKSR) [63] to generate an enhanced mesh. From Fig. 15 and Table. 10, it can be observed that our *mn3DSSR* effectively restores continuous and smooth surfaces from sparse point clouds, while other point cloud-based methods struggle to recover finer geometric structures due to their inability to utilize normal information. It is worth noting that due to the limited number of *Sketchfab2*, we fine-tune *mn3DSSR* on *Google Scanned Objects* dataset using the same training samples and process meshes using the same method as described in Section 4.9.2, but resample them into point clouds during data preparation.

4.9.2 Evaluation on Mesh Dataset

For mesh representation, we have validated the generalization of our *mn3DSSR* on the real captured mesh dataset *Google Scanned Objects (GSO)* [64], containing high-quality texture maps. Specifically, we randomly select 500 samples for fine-tuning and 64 samples for testing, and synthesize the required normal, depth, and other modalities through multi-view rasterization.

During testing, we utilize the *SuperNormal* [62] to reconstruct a complete mesh from the enhanced normal maps. Next, we render the reconstructed mesh into normal and depth maps from six directions (*i.e.*, *front*, *back*, *left*, *right*, *up*, and *down*) to calculate the evaluation metrics. As shown in Fig. 15 and Table 10, our *mn3DSSR* leverages additional RGB information to recover better geometric details.

4.9.3 Evaluation on Depth Dataset

For depth representation, we have performed additional experiments on the *Digital Image Media Laboratory* (DIML)

TABLE 10: **Quantitative comparison results on three different 3D data representations.** The average results of *mn3DSSR* are provided for three different 3D datasets. "↑" means the higher the better, while "↓" means the lower the better. The best results are highlighted in **bold** for each dataset in each column.

Method	×2	×4	×8
	Point Cloud Dataset: ***Sketchfab***		
	PSNR↑/SSIM↑/MAE↓/MRDE↓	PSNR↑/SSIM↑/MAE↓/MRDE↓	PSNR↑/SSIM↑/MAE↓/MRDE↓
Feng2022CVPR [13]	27.1797/0.8922/9.1853/2.3569	27.2716/0.8947/8.8525/2.0369	26.9247/0.8881/9.2709/1.9502
He2023CVPR [7]	28.6677/0.9187/7.2417/1.4316	28.3503/0.9132/7.4939/1.5623	26.0556/0.8683/10.4705/1.9522
Ours	**29.3290/0.9338/6.5276/1.2350**	**28.5316/0.9188/7.3908/1.3059**	**27.5911/0.9079/8.2345/1.4904**
	Mesh Dataset: ***Google Scanned Objects*** **(GSO)**		
	PSNR↑/SSIM↑/MAE↓/MRDE↓	PSNR↑/SSIM↑/MAE↓/MRDE↓	PSNR↑/SSIM↑/MAE↓/MRDE↓
Loop2008TOG [14]	28.3602/0.8815/6.7123/**1.3101**	26.1830/0.8461/8.6325/1.9182	23.9059/0.8050/11.6647/2.7964
Liu2020TOG [8]	27.9929/0.8855/6.6182/1.3164	26.7629/0.8529/7.7574/2.2238	24.6799/0.8060/10.5905/2.8801
Ours	**29.0197/0.8930/6.1005**/1.4506	**27.6110/0.8680/7.2453/1.8963**	**26.0434/0.8378/8.9210/2.4913**
	Depth Dataset: ***Digital Image Media Laboratory*** **(DIML)**		
	PSNR↑/SSIM↑/MAE↓/MRDE↓	PSNR↑/SSIM↑/MAE↓/MRDE↓	PSNR↑/SSIM↑/MAE↓/MRDE↓
Zhao2022CVPR [20]	26.2097/0.7817/6.9070/0.1487	22.7714/0.5900/11.1449/0.1670	19.9095/0.4890/15.3025/0.4378
Metzger2023CVPR [10]	29.2894/0.9033/4.2398/**0.1351**	26.8029/0.7740/6.8725/**0.1457**	23.1542/0.6599/8.8073/0.3157
Ours	**31.0789/0.9322/3.1226**/0.1363	**27.0288/0.7886/6.1761**/0.1534	**24.4032/0.6793/8.3741/0.2642**

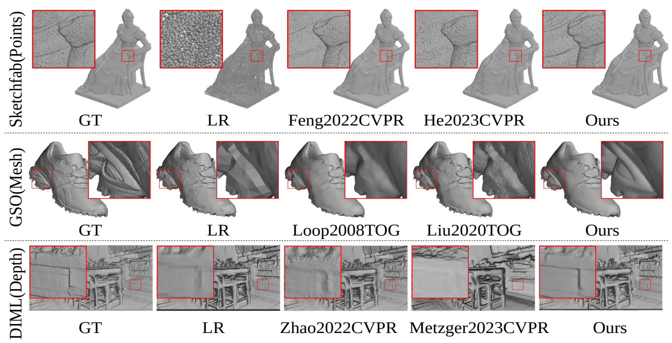


Fig. 15: **Visual comparison on** ***point cloud, mesh, depth*** **datasets**. 'GT' refers to the ground-truth surface, as highlighted in the red box. Fine-grained 3D surfaces are obtained under the ×8 setting on the *Sketchfab*, *GSO*, and *DIML*.

dataset [65], which is captured using a Time-of-Flight (ToF) RGB-D camera (*Kinect V2*). *DIML* encompasses a wide range of realistic scenes with relatively high-precision depth images and aligned RGB images.

In our experiments, we invert the depth values and obtain the normal modality through finite difference and vector normalization. For the RGB modality, we replace the brightest, darkest, and average RGB inputs with the same RGB image. We fine-tune the proposed model on 800 scenes randomly selected from the DIML training set and conducted testing on 256 scenes from the testing set. To ensure a fair comparison, we use upsampled RGB images as guidance for all depth-based methods.

We compare our *mn3DSSR* with two advanced depth-based methods, with the results presented in Fig. 15 and Table 10. Compared to other depth-based methods, our model exhibits greater accuracy in local geometric details and promising overall quality in terms of PSNR, SSIM, and MAE on normal maps as well as MRDE on depth images.

5 Conclusion and Future Work

In this paper, we have presented an efficient multimodal normal-based framework for 3D surface super-resolution (*mn3DSSR*). Compared with the existing state of the arts, our contributions can be highlighted in three aspects, including: (i) we have constructed one of the largest normal-based multimodal dataset, providing a useful resource for future research; (ii) we have developed a novel two-branch multimodal alignment module and a multimodal split fusion module, enabling more efficient and accurate utilization of texture and shape features; (iii) we have explored new curl- and alignment-based loss functions, further improving the model capability to capture fine-grained details and align multimodal features. Extensive experiments compared to 24 super-resolution methods across four different 3D data representations validate the superiority of our proposed *mn3DSSR*.

While our framework demonstrates significant improvements in 3D surface super-resolution, several avenues remain for future exploration. First, *mn3DSSR* primarily enhances 3D surfaces for a single view. A new dataset will be investigated to record panorama object surface in a single normal map, which could avoid occlusions and lead to more robust restoration. Second, we see potential in incorporating large-scale and general-purpose models to further enhance the super-resolution performance. For example, advanced large language models (LLMs) could provide richer multimodal information, improving both texture and shape feature extraction. Finally, extending the *mn3DSSR* framework to address other low-level 3D vision tasks is another intriguing research direction. For instance, adapting it to 3D surface denoising or inpainting could enhance its versatility and applicability to a broader range of real-world scenarios.

References

[1] W. Xie, M. Wang, D. Lin, B. Shi, and J. Jiang, "Surface Geometry Processing: An Efficient Normal-Based Detail Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13 749–13 765, 2023. 1, 5

[2] M. Pesavento, M. Volino, and A. Hilton, "Super-resolution 3D Human Shape from a Single Low-Resolution Image," in *Springer European Conference on Computer Vision (ECCV)*, 2022, pp. 447–464. 1

[3] F. Mortazavi and M. Saadatseresht, "High Resolution Surface Reconstruction of Cultural Heritage Objects Using Shape from Polarization Method," *arXiv preprint arXiv:2406.15121*, 2024. 1

[4] C. Dong, C. C. Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016. 1, 11, 12

[5] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual Dense Network for Image Restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2480–2495, 2021. 1, 11, 12

[6] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image Super-Resolution via Iterative Refinement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4713–4726, 2023. 1, 11, 12

[7] Y. He, D. Tang, Y. Zhang, X. Xue, and Y. Fu, "Grad-PU: Arbitrary-Scale Point Cloud Upsampling via Gradient Descent with Learned Distance Functions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 5354–5363. 1, 2, 3, 11, 12, 16

[8] H.-T. D. Liu, V. G. Kim, S. Chaudhuri, N. Aigerman, and A. Jacobson, "Neural subdivision," *ACM Transactions on Graphics*, vol. 39, no. 2, pp. 10–16, 2020. 1, 2, 3, 11, 12, 16

[9] Z. Chen, V. G. Kim, M. Fisher, N. Aigerman, H. Zhang, and S. Chaudhuri, "Decor-GAN: 3D Shape Detailization by Conditional Refinement," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 740–15 749. 1, 3

[10] N. Metzger, R. C. Daudt, and K. Schindler, "Guided Depth Super-Resolution by Deep Anisotropic Diffusion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 18 237–18 246. 1, 2, 3, 10, 11, 12, 16

[11] W. Xie, T. Huang, and M. Wang, "MNSRNet: Multimodal Transformer Network for 3D Surface Super-Resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 703–12 712. 1, 2, 3, 7, 9, 11, 12

[12] G. Qian, A. Abualshour, G. Li, A. Thabet, and B. Ghanem, "Pu-GCN: Point Cloud Upsampling Using Graph Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 683–11 692. 2, 3, 11, 12

[13] W. Feng, J. Li, H. Cai, X. Luo, and J. Zhang, "Neural Points: Point Cloud Representation with Neural Fields for Arbitrary Upsampling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 18 633–18 642. 2, 3, 11, 12, 15, 16

[14] C. Loop and S. Schaefer, "Approximating Catmull-Clark Subdivision Surfaces with Bicubic Patches," *ACM Transactions on Graphics*, vol. 27, no. 1, pp. 1–11, 2008. 2, 3, 11, 12, 16

[15] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. N. Wu, "Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2468–2484, 2022. 2, 3

[16] J. Shim, C. Kang, and K. Joo, "Diffusion-Based Signed Distance Fields for 3D Shape Generation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 20 887–20 897. 2, 3, 11, 12

[17] O. Voynov, A. Artemov, V. Egiazarian, A. Notchenko, G. Bobrovskikh, E. Burnaev, and D. Zorin, "Perceptual Deep Depth Super-Resolution," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 5653–5663. 2, 3, 11, 12

[18] B. Haefner, S. Peng, A. Verma, Y. Quéau, and D. Cremers, "Photometric Depth Super-Resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2453–2464, 2020. 2, 3

[19] X. Deng and P. L. Dragotti, "Deep Convolutional Neural Network for Multi-Modal Image Restoration and Fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3333–3348, 2021. 1, 2, 3, 11

[20] Z. Zhao, J. Zhang, S. Xu, Z. Lin, and H. Pfister, "Discrete Cosine Transform Network for Guided Depth Map Super-Resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5697–5707. 2, 3, 11, 12, 16

[21] L. Wang, Y. Guo, Y. Wang, Z. Liang, Z. Lin, J. Yang, and W. An, "Parallax Attention for Unsupervised Stereo Correspondence Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2108–2125, 2022. 2

[22] Y. Ju, M. Jian, C. Wang, C. Zhang, J. Dong, and K.-M. Lam, "Estimating High-resolution Surface Normals via Low-resolution Photometric Stereo Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 4, pp. 2512–2524, 2024. 2, 3, 12

[23] W. Xie, T. Huang, and M. Wang, "3D Surface Super-resolution from Enhanced 2D Normal Images: A Multimodal-driven Variational AutoEncoder Approach," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023, pp. 1578–1586. 2, 3, 11, 12

[24] Y. Li, V. Tsiminaki, R. Timofte, M. Pollefeys, and L. V. Gool, "3D appearance Super-Resolution with Deep Learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9671–9680. 1, 11, 12

[25] M.-I. Georgescu, R. T. Ionescu, A.-I. Miron, O. Savencu, N.-C. Ristea, N. Verga, and F. S. Khan, "Multimodal Multi-Head Convolutional Attention with Various Kernel Sizes for Medical Image Super-Resolution," in *IEEE Winter Conference on Applications of Computer Vision (WCACV)*, 2023, pp. 2195–2205. 1, 11, 12

[26] L. Yu, X. Li, C. Fu, D. Cohen-Or, and P. Heng, "Pu-net: Point cloud upsampling network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2790–2799. 3

[27] S. Qiu, S. Anwar, and N. Barnes, "Pu-Transformer: Point Cloud Upsampling Transformer," in *Asian Conference on Computer Vision (ACCV)*, 2022, pp. 2475–2493. 3

[28] E. Catmull and J. Clark, "Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes," *Elsevier Computer-Aided Design*, vol. 10, no. 6, pp. 350–355, 1978. 3

[29] C. Loop, "Smooth Subdivision Surfaces Based on Triangles," Ph.D. dissertation, The University of Utah, January 1987. 3

[30] T. Shen, J. Gao, K. Yin, M.-Y. Liu, and S. Fidler, "Deep Marching Tetrahedra: A Hybrid Representation for High-Resolution 3D Shape Synthesis," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, no. 1, pp. 6087–6101, 2021. 3

[31] X.-Y. Zheng, H. Pan, P.-S. Wang, X. Tong, Y. Liu, and H.-Y. Shum, "Locally Attentional SDF Diffusion for Controllable 3D Shape Generation," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–13, 2023. 3

[32] S. Gu, S. Guo, W. Zuo, Y. Chen, R. Timofte, L. Van Gool, and L. Zhang, "Learned Dynamic Guidance for Depth Image Reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2437–2452, 2020. 3

[33] X. Ye, B. Sun, Z. Wang, J. Yang, R. Xu, H. Li, and B. Li, "PMBANet: Progressive Multi-Branch Aggregation Network for Scene Depth Super-Resolution," *IEEE Transactions on Image Processing*, vol. 29, pp. 7427–7442, 2020. 3

[34] Y. Ju, K.-M. Lam, W. Xie, H. Zhou, J. Dong, and B. Shi, "Deep Learning Methods for Calibrated Photometric Stereo and Beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 11, pp. 7154–7172, 2024. 3, 9

[35] S. Ikehata and Y. Asano, "SpectraM-PS: Spectrally Multiplexed Photometric Stereo under Unknown Spectral Composition," in *Springer European Conference on Computer Vision (ECCV)*, 2024, pp. 1–18. 3

[36] W. Xie, Y. Zhang, C. C. Wang, and R. C.-K. Chung, "Surface-from-gradients: An Approach Based on Discrete Geometry Processing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2195–2202. 4

[37] X. Cao, H. Santo, B. Shi, F. Okura, and Y. Matsushita, "Bilateral Normal Integration," in *Springer European Conference on Computer Vision (ECCV)*, 2022, pp. 552–567. 4, 10

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008. 5, 6

[39] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin-Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *IEEE International Conference on Computer Vision (ICCV)*, 2021, pp. 10 012–10 022. 5

[40] X. Chen, X. Wang, J. Zhou, Y. Qiao, and C. Dong, "Activating More Pixels in Image Super-Resolution Transformer," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 22 367–22 377. 7, 11, 12, 14

[41] R. Courant, F. John, A. A. Blank, and A. Solomon, *Introduction to Calculus and Analysis*. Springer, 1965, vol. 1. 8

[42] C. Ma, Y. Rao, J. Lu, and J. Zhou, "Structure-Preserving Image Super-Resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7898–7911, 2022. 8, 11, 12

[43] Y. Ju, B. Shi, M. Jian, L. Qi, J. Dong, and K.-M. Lam, "NormAttention-PSN: A High-frequency Region Enhanced Photometric Stereo Network with Normalized Attention," *Springer International Journal of Computer Vision*, vol. 130, no. 12, pp. 3014–3034, 2022. 8, 11, 12

[44] B. Shi, Z. Wu, Z. Mo, D. Duan, S.-K. Yeung, and P. Tan, "A Benchmark Dataset and Evaluation for Non-lambertian and Uncalibrated Photometric Stereo," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3707–3716. 8, 9

[45] R. Mecca, F. Logothetis, I. Budvytis, and R. Cipolla, "LUCES: A dataset for near-field point light source photometric stereo,"

Computing Research Repository (CoRR), vol. abs/2104.13135, 2021. 8, 9
[46] H. Guo, J. Ren, F. Wang, B. Shi, M. Ren, and Y. Matsushita, "DiLiGenRT: A Photometric Stereo Dataset with Quantified Roughness and Translucency," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 11 810–11 820. 8
[47] R. J. Woodham, "Photometric Method for Determining Surface Orientation from Multiple Images," *SPIE Optical engineering*, vol. 19, no. 1, pp. 139–144, 1980. 9
[48] K. H. Cheng and A. Kumar, "Revisiting Outlier Rejection Approach for Non-Lambertian Photometric Stereo," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1544–1555, 2019. 9
[49] G. Chen, K. Han, and K.-Y. K. Wong, "PS-FCN: A Flexible Learning Framework for Photometric Stereo," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 3–18. 9
[50] S. Ikehata, "Scalable, Detailed and Mask-Free Universal Photometric Stereo," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 13 198–13 207. 9, 10
[51] N. Alldrin, T. Zickler, and D. Kriegman, "Photometric Stereo with Non-paraMetric and Spatially-varying Reflectance," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8. 9
[52] Y. Xiong, A. Chakrabarti, R. Basri, S. J. Gortler, D. W. Jacobs, and T. Zickler, "From Shading to Local Shape," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 67–79, 2014. 9
[53] J. Ren, F. Wang, J. Zhang, Q. Zheng, M. Ren, and B. Shi, "Diligent102: A Photometric Stereo Benchmark Dataset with Controlled Shape and Material Variation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 581–12 590. 9
[54] X. Chen, Q. Zhang, M. Lin, G. Yang, and C. He, "No-Reference Color Image Quality Assessment: From Entropy to Perceptual Quality," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, pp. 1–14, 2019. 9
[55] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012. 9
[56] W. Xie, M. Wang, X. Qi, and L. Zhang, "3D Surface Detail Enhancement from a Single Normal Map," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2325–2333. 10
[57] S. Kumar, Y. Dai, and H. Li, "Superpixel soup: Monocular dense 3d reconstruction of a complex dynamic scene," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1705–1717, 2021. 10
[58] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image Super-Resolution Using very Deep Residual Channel Attention Networks," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301. 11, 12
[59] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained Image Processing Transformer," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 299–12 310. 11, 12
[60] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Learning Enriched Features for Fast Image Restoration and Enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 1934–1948, 2023. 11, 12
[61] X. Deng, Y. Zhang, M. Xu, S. Gu, and Y. Duan, "Deep Coupled FeedBack Network for Joint Exposure Fusion and Image Super-Resolution," *IEEE Transactions on Image Processing*, vol. 30, no. 1, pp. 3098–3112, 2021. 11, 12
[62] X. Cao and T. Taketomi, "SuperNormal: Neural Surface Reconstruction via Multi-View Normal Integration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 20 581–20 590. 15
[63] J. Huang, Z. Gojcic, M. Atzmon, O. Litany, S. Fidler, and F. Williams, "Neural Kernel Surface Reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 4369–4379. 15
[64] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2553–2560. 15
[65] D. I. M. Lab, "DIML RGBD," Accessed: Aug. 01, 2023, http://diml.yonsei.ac.kr/DIML_rgbd_dataset/. 16

Miaohui Wang received the Ph.D. degree from the Department of Electronic Engineering, The Chinese University of Hong Kong (CUHK), Hong Kong, China. Currently, he is a tenured Associate Professor at the College of Electronics and Information Engineering, Shenzhen University (SZU), China. He was the recipient of the *Best Thesis Award* from the Ministry of Education of Shanghai City and Fudan University (FDU), respectively. He received the *Outstanding Reviewer Award* from *IEEE International Conference on Multimedia & Expo* (2021). He serves as an Associate Editor for *IEEE SIGNAL PROCESSING LETTERS*.

Yunheng Liu is currently perusing the M.E. degree from Shenzhen University, Shenzhen, P. R. China. His research interests include 3D signal processing, super-resolution, and quality evaluation.

Wuyuan Xie received the Ph.D. degree from the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong (CUHK) in 2016, the M.S. degree from the South China University of Technology University (SCUT) in 2009, and the B.S degree from the Central South University (CSU) in 2006, respectively. She is currently an Associate Professor of the Research Institute for Future Media Computing, Shenzhen University.

Boxin Shi received the BE degree from the Beijing University of Posts and Telecommunications, in 2007, the ME degree from Peking University, in 2010, and the PhD degree from the University of Tokyo, in 2013. He is currently a Boya Young fellow associate professor (with tenure) and research professor with Peking University, where he leads the Camera Intelligence Lab. He is an Associate Editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence*/*International Journal of Computer Vision* and an area chair of CVPR/ICCV/ECCV.

Jianmin Jiang received the PhD degree from the University of Nottingham, U.K., in 1994. From 1997 to 2001, he was a professor of computing with the University of Glamorgan, Wales, U.K. In 2002, he joined the University of Bradford, U.K., as a chair professor of Digital Media, and the director of Digital Media and Systems Research Institute. He was with the University of Surrey, U.K., as a professor during 2010–2014 and a distinguished professor (1000-plan) with Tianjin University, China, during 2010–2013. He is a Chartered Engineer, Fellow of IEE, Fellow of RSA, member of EPSRC College, U.K., and EU FP-6/7 evaluator.