Spherical Space Feature Decomposition for Guided Depth Map Super-Resolution

Zixiang Zhao\textsuperscript{1,2}, Jiangshe Zhang\textsuperscript{1*}, Xiang Gu\textsuperscript{1}, Chengli Tan\textsuperscript{1}, Shuang Xu\textsuperscript{3}, Yulun Zhang\textsuperscript{2}, Radu Timofte\textsuperscript{2,4}, Luc Van Gool\textsuperscript{2}

\textsuperscript{1}Xi’an Jiaotong University, \textsuperscript{2}Computer Vision Lab, ETH Zürich, \textsuperscript{3}Northwestern Polytechnical University, \textsuperscript{4}University of Würzburg

zixiangzhao@stu.xjtu.edu.cn, jszhang@mail.xjtu.edu.cn

Abstract

Guided depth map super-resolution (GDSR), as a hot topic in multi-modal image processing, aims to upsample low-resolution (LR) depth maps with additional information involved in high-resolution (HR) RGB images from the same scene. The critical step of this task is to effectively extract domain-shared and domain-private RGB/depth features. In addition, three detailed issues, namely blurry edges, noisy surfaces, and over-transferred RGB texture, need to be addressed. In this paper, we propose the Spherical Space feature Decomposition Network (SSDNet) to solve the above issues. To better model cross-modality features, Restormer block-based RGB/depth encoders are employed for extracting local-global features. Then, the extracted features are mapped to the spherical space to complete the separation of private features and the alignment of shared features. Shared features of RGB are fused with the depth features to complete the GDSR task. Subsequently, a spherical contrast refinement (SCR) module is proposed to further address the detail issues. Patches that are classified according to imperfect categories are input into the SCR module, where the patch features are pulled closer to the ground truth and pushed away from the corresponding imperfect samples in the spherical feature space via contrastive learning. Extensive experiments demonstrate that our method can achieve state-of-the-art results on four test datasets, as well as successfully generalize to real-world scenes. The code is available at https://github.com/ZhaoZixiang1228/GDSR-SSDNet.

1. Introduction

Depth maps, as images that measure the distance of scene points from the sensor, are widely used in autonomous driving [38, 50, 87–89], pose estimation [75, 83], virtual reality [41, 51, 79, 80], and scene understanding [24, 53, 65, 96].

*Corresponding author.

However, the depth maps produced by current consumer-level depth sensors, e.g., Time-of-Flight (ToF) and Kinect cameras, often have the disadvantages of low resolution and noise. These disadvantages are insufficient to meet the requirements of advanced computing vision tasks [52, 76, 93].

To obtain high-resolution depth maps, we naturally hope to accomplish depth map super-resolution (SR) by transferring mature SR models in the RGB image domain [13, 39, 86] to the depth SR task. However, a potential risk is that RGB SR models tend to focus on reconstructing high-frequency image information, such as details and texture. Conversely, for depth images, the objects’ depth information is often textureless and piecewise, and more sensitive to unclear edges and noise. Therefore, it is unreasonable to directly apply RGB SR models to the depth SR task. On the other hand, while acquiring the depth map, it is relatively easy to obtain RGB SR models to the depth SR task. On the other hand, while acquiring the depth map, it is relatively easy to obtain RGB SR models to the depth SR task.
depth maps [55]. Therefore, we hope to use the HR RGB images to provide edges and contour information missing in the depth map, and to fuse the multi-modal information to accomplish the LR depth image upsampling.

In the era of deep learning (DL), numerous methods have been utilized to learn the mapping between LR → HR depth maps. These methods succeeded to some extent in modeling the cross-modality features and reconstructing the contour and edge information. However, DL models which rely on natural priors will limit the model’s flexibility. Models learning the LR → HR mapping via data-driven methods are difficult to interpret due to the unclear working mechanism [90]. Thus, effectively extracting and distinguishing domain-shared and domain-private RGB/depth features is still a challenge. While microscopically, the obtained depth images are often plagued by three detail issues: blurry edges, noisy surfaces, and over-transferred RGB texture, all of which affect the display quality of depth maps.

In response to addressing the above challenges, we expect to limit the solution space by constraining the extracted features and further improve the modeling of the dependencies between different modalities. Based on our observation, RGB images and depth maps contain shared features, such as depth map discontinuities and edge features of RGB objects, which can be aligned in the feature space. In addition, unique private features, such as the distance information of the depth map and the texture of RGB object surface, should be separated conversely. Thus, in the feature space, domain-shared and domain-private features are expected to separate and align respectively, while the distance between HR features and imperfect features with the above-mentioned detail issues should also be pushed away.

The above goal can be divided into two steps. First, extracting features through an effective encoder, and then selecting an appropriate distance measure to complete the alignment and separation of features. Considering that CNN-based architectures limit feature extraction capabilities due to content-independent convolution kernels and the lack of global information, we employ the Restormer blocks [84], which has been proved to effectively extract features in the low-level vision domain, as the basic unit of our encoder. For the choice of distance measure, we first think of the Euclidean distance, such as $\ell_1$ or $\ell_2$ distance. However, cross-modality features often contain different scales and orders of magnitude, and the Euclidean distance is easily affected by the scale, which makes it difficult to achieve our goal. Recently, with the development of spherical DL models [4, 7, 21, 22], the spherical feature learning is well known by virtue of its advantages over the Euclidean feature learning in many applications, e.g., domain adaptation [21].


due to the distances between spherical features being regularized, the alignment and separation of features can be done more easily without losing the model’s representational capacity.

According to the above analysis, we propose our Spherical Space feature Decomposition Network (SSDNet). The specific workflow is displayed in Fig. 2. Our contribution can be summarized in three-fold:

- We propose a spherical space feature decomposition framework to model the cross-modality features. The features extracted by the Restormer block are mapped to the spherical space for separation and alignment. This is the first time that the Transformer and spherical-space distance measure are applied to the GDSR task.
- The spherical contrast refinement module, cooperating with the imperfect patch classification and the corresponding contrastive learning branch, is proposed to address the possible detail issues in depth maps. This is also the first time that the contrastive learning technique has been used for the GDSR task.
- Experiments on four GDSR benchmarks and a real scene dataset demonstrate that our method can generate satisfactory HR depth maps in different scenarios and exhibits good generalization ability.

2. Related Work

In this section, we briefly introduce the GDSR task, and illustrate the Vision Transformer, spherical space DL and contrastive learning techniques utilized in SSDNet.

2.1. GDSR methods

Image super-resolution is a popular image processing and computer vision task with too many sub-categories, so we only discuss the GDSR task here. Conventional GDSR methods can be divided into local- [40, 43, 44, 46], global-[12, 17, 36, 48, 77] and learning-based [8, 19, 20, 73, 74] methods, etc. They extract cross-modality information by manually designed filters, optimizing equations, or sparse dictionary learning. With the rapid development of DL, GDSR is further promoted by the CNN-based methods [58, 60–62, 68, 81]. Generally, DL-based methods can be categorized in three groups, i.e., auto-encoder (AE) [23, 30, 35, 78, 90, 92], learnable filter [14, 31, 71], algorithm unfolding [10, 11, 54, 94] and unsupervised [9, 15] methods. AE-based methods learn cross-modality features via shared or private encoders and reconstruct HR depth maps with decoders. In the learnable filter group, the convolution kernels used to extract information are set to be context-dependent and spatially-variant, which improves the flexibility of the model. The algorithmic unfolding group establishes interpretable GDSR models by building a bridge between traditional optimization functions and deep learning. Unsupervised methods change the training paradigm that requires paired LR/HR images, thus solving the difficulty of obtaining paired training data.
2.2. Spherical deep learning

The works [4, 6, 7] introduce the geometrical neural network or the spherical convolution neural networks for analyzing the spherical signals that are rotation invariant. Gu et al. [22] propose the spherical multi-layer perceptron for aligning feature distributions of different domains in spherical feature space for domain adaptation. Our idea of spherical space feature decomposition is inspired by Gu et al. [22] that shows the advantages of spherical feature learning over the Euclidean feature learning for domain adaptation in the “high-level” image classification task. Different from [22], we tackle the “low-level” GDSR task by aligning/separating the shared/private features of paired RGB image and depth map, rather than aligning the feature distributions as in [22]. We further propose the spherical contrast refinement for tackling the issues of blurry edges, noisy surfaces, and over-transferred RGB texture for GDSR.

2.3. Transformer in vision

Transformer, proposed by Vaswani et al. [64], has now become a popular technology in computer vision. It has achieved dominance in classification [16, 42, 63], object detection [2, 95], segmentation [66, 91], etc. Simultaneously, it has also made progress in low-level vision [1, 18, 33, 67]. Chen et al. [5] proposed IPT based on the standard Transformer and multi-task learning. Liang et al. [37] proposed SwinIR whose core module inherits from Swin Transformer [42]. Recently, Restormer [84] improves the transformer block by the gated-Dconv network and multi-Dconv head transposed attention. By applying self-attention in the feature dimension, the ability of local-global-representation learning is maintained while being more friendly to high-resolution input images.

2.4. Contrastive Learning

Contrastive learning, which aims to address the scarcity of source and target paired data and prioritize self-supervised representation learning [25, 28]. The primary objective is to bring samples close to their positive counterparts and push them away from negative ones within high-dimensional manifolds, and it seeks to increase the distance between samples belonging to different classes. Recently, contrastive learning has gained significant attention in computer vision, particularly in high-level tasks such as object detection [72], video segmentation [3], and object tracking [82]. It has also found applications in low-level tasks such as super-resolution [85], image translation [49], and image dehazing [70].

2.5. Comparison with existing approaches

The methods most related to our approach are feature-decomposition DL methods [10, 90], which propose the idea that cross-modality features contain the common and private information. In contrast, our method, for the first time, exploits the Transformer model’s powerful ability to model globally dependent features. Meanwhile, the distance measure in spherical space is proved to decompose the cross-modality features more effectively than Euclidean distances. Additionally, instead of only focusing on optimizing the $\ell_2$ loss between the reconstructed image and ground truth, we give classification-based fine-tuning for image patches under different imperfect conditions, thus further improving the performance of GDSR.

3. Method

In this section, we first present the specific architecture of our SSDNet. Then, comprehensive descriptions are provided for each module. Finally, the loss function and training details will also be addressed.

3.1. Overview

We can define the input LR depth map, RGB image and the HR depth map as $D_{LR} \in \mathbb{R}^{H \times W}$, $D_{HR} \in \mathbb{R}^{H \times W}$ and $R \in \mathbb{R}^{H \times W \times 3}$, where $\{H, W\}$ and $\{h, w\}$ are the height and width of input RGB and depth images, respectively. For our SSDNet network, it consists of five modules, namely, the Encoders for RGB and Depth images, the Decoders for RGB and Depth images, and the Spherical Contrast Refinement (SCR) module, which is denoted as $E_R(\cdot), E_D(\cdot), D_R(\cdot), D_D(\cdot)$, and $S(\cdot)$, respectively.

In general, as shown in Fig. 2, SSDNet utilizes $E_R$ and $E_D$ to extract features, which are then projected onto the spherical space for feature separation and alignment. Then, the depth features and shared RGB features are input into $D_R$ to obtain reconstructed depth maps, and RGB features are fed into $E_R$ to get reconstructed RGB images. Finally, the reconstructed depth maps are subsequently provided to the SCR module to refine details and output the final $D_{HR}$.

The basic unit we employed for feature extraction and image restoration is the Restormer block [84]. The reason we opted for the Restormer block in $E_R$, $E_D$, $D_R$ and $D_D$ is because it allows for the extraction of global features from high-resolution input images by utilizing self-attention across feature dimensions [84]. This approach enables the extraction of cross-modality shallow features without significantly increasing computational requirements. For details on the Restormer block architecture, please refer to the supplementary material or the original paper [84].

3.1.1 Spherical space transform

We first define the mapping based on Riemannian geometry between Euclidean Space feature maps and Spherical Space feature maps, i.e., $\mathcal{G}(\cdot)$ and $\mathcal{E}(\cdot)$, which are employed in the feature decomposition. A schematic diagram for the
mappings is shown in Fig. 2d. The distance measure on the spherical space is also defined.

Definition 1 (Spherical exponential mapping). Given a vector \( v \) in \( d \)-dimensional Euclidean space, we define the \((d+1)\)-dimensional vector \( \tilde{v} \) by \( \tilde{v} = (v, r) \) where \( r \) is a hyperparameter named radius. Then the spherical exponential transform [69] \( \exp_N : T_N S^{d+1}_{r} \rightarrow S^{d+1}_{r} \) is defined as

\[
\exp_{N}(v) = N \cos \theta + \tilde{v} \frac{\sin \theta}{\theta},
\]

where \( S^{d+1}_{r} = \{ x \in \mathbb{R}^{d+1} : \| x \| = r \} \subset \mathbb{R}^{d+1} \) is the \( d \)-dimensional spherical space, \( N = (0, \cdots, 0, r) \in S^{d+1}_{r} \) is the north pole, \( \theta = \| \tilde{v} \| \), and \( T_N S^{d+1}_{r} = \{ (v, r) : v \in \mathbb{R}^{d} \} \) is the tangent space of \( S^{d+1}_{r} \). Further, the spherical exponential mapping \( EXP : \mathbb{R}^{h \times w \times d} \rightarrow \mathbb{R}^{N \times w \times (d+1)} \) for an Euclidean feature map \( \Phi \) is defined as

\[
E X P (\Phi)[i, j, :] = \exp_N (\Phi[i, j, :])
\]

where \( \Phi[i, j, :] \) is the feature in location \((i, j)\).

Definition 2 (Spherical logarithmic mapping). Given the spherical feature \( x \in S^{d+1}_{r} \) with \( \| x \| = r \), we define the spherical logarithmic transform [69] \( \log_N : S^{d+1}_{r} \rightarrow T_N S^{d+1}_{r} \) by

\[
\log_N(x) = \frac{\psi}{\sin \psi} (x - N \cos \psi),
\]

where \( N = (0, \cdots, 0, r) \in S^{d+1}_{r} \) is the north pole, \( \psi = \arccos(N^T x / r^2) \). Further, the spherical logarithmic mapping \( LOG : \mathbb{R}^{h^N \times w^N \times (d+1)} \rightarrow \mathbb{R}^{h^N \times w^N \times d} \) for a spherical feature map \( \Phi \) is defined by

\[
LOG(\Phi)[i, j, :] = \mathcal{H}(\log_N(\Phi[i, j, :])),
\]

where \( \mathcal{H} : T_N S^{d+1}_{r} \rightarrow \mathbb{R}^d \) is defined by \( \mathcal{H}((v, r)) = v \).

Definition 3 (Spherical space distance). Given two spherical feature maps \( \Phi_1, \Phi_2 \in \mathbb{R}^{N \times w^N \times (d+1)} \) with \( \| \Phi_1[i, j, :] \| = \| \Phi_2[i, j, :] \| = 1 \) and \( \| \Phi_1[i, j, :] \| = \| \Phi_2[i, j, :] \| = 1 \), the spherical distance between \( \Phi_1 \) and \( \Phi_2 \) is defined as

\[
\text{Distance}(\Phi_1, \Phi_2) = \sum_{i, j} \frac{1}{\mathcal{H}(\log_N(\Phi_1[i, j, :]))^2}
\]

where \( \mathcal{H} : T_N S^{d+1}_{r} \rightarrow \mathbb{R}^d \) is defined by \( \mathcal{H}((v, r)) = v \).
\[ \| \Phi_2[i, j, :] \| = r \text{ for any } i, j, \text{ the spherical space distance of } \Phi_1 \text{ and } \Phi_2 \text{ is defined as} \]

\[
S \{ \Phi_1, \Phi_2 \} = \sum_{i=1}^{k'} \sum_{j=1}^{w'} \frac{1}{\sqrt{2}} \Phi_1[i, j, :] \Phi_2[i, j, :]. \tag{5}
\]

### 3.1.2 Encoder

We use the feature extraction of \( D_{LR} \) as an example to explain the procedure, and \( R \) can be carried out similarly to \( D_{LR} \) by changing the subscripts from \( D \) to \( R \). First, a \( 3 \times 3 \) convolution is used to obtain shallow features embedding \( \Phi_D^{(p)} \). The main body of feature extraction consists of a cascade of \( P \) Restormer blocks, and we denote the \( p \)-th Restormer block in \( \mathcal{E}_D \) as \( \mathcal{R}_D^{(p)} \), where \( p = 1, \cdots, P \). The input of each \( \mathcal{R}_D^{(p)} \) is represented by \( \Phi_D^{(p)} \).

At the \( p \)-th step of feature extraction, \( \Phi_D^{(p-1)} \) passes through \( \mathcal{R}_D^{(p)} \) to obtain the preliminary extracted feature \( \Phi_D^{(p)} \). According to the analysis for motivation in Sec. 1, after the multi-head self-attention mechanism, \( \Phi_D^{(p)} \) should contain shared features in some channels and private features in other channels. Thus, we assume that features in the first \( \text{dim}_1 \) channels are shared and represent cross-modality information, while features in the latter \( \text{dim}_2 \) channels are private and represent the characteristics of their own modality. To achieve feature separation and alignment, \( \Phi_D^{(p)} \) is mapped to the spherical space using the spherical exponential mapping \( \mathcal{E}X\mathcal{P}() \), and then recovered in the feature domain by the spherical logarithmic mapping \( \mathcal{L}\mathcal{O}G() \) after calculating the feature decomposition loss. Finally, the recovered features are re-concatenated along the channel dimension to obtain \( \Phi_D^{(p+1)} \), which will be input to the next \( \mathcal{R}_D^{(p+1)} \). The feature decomposition loss will be illustrated in subsequent sections.

The total feature extraction process of \( p \)-th step is:

\[
\begin{align*}
\Phi_D^{(p)} &= \mathcal{R}_D^{(p)}(\Phi_D^{(p-1)}) \tag{6a} \\
\hat{\Phi}_{D,\text{align}}^{(p)} &= \Phi_D^{(p)}[0: \text{dim}_1/2], \quad \hat{\Phi}_{D,\text{sep}}^{(p)} = \Phi_D^{(p)}[\text{dim}_1/2: \text{dim}] \tag{6b} \\
\Phi_{D,\text{align}}^{(p)} &= \mathcal{L}\mathcal{O}G(\mathcal{E}X\mathcal{P}(\hat{\Phi}_{D,\text{align}}^{(p)})) \tag{6c} \\
\Phi_{D,\text{sep}}^{(p)} &= \mathcal{L}\mathcal{O}G(\mathcal{E}X\mathcal{P}(\hat{\Phi}_{D,\text{sep}}^{(p)})) \tag{6d} \\
\Phi_D^{(p)} &= \text{Cat}(\hat{\Phi}_{D,\text{align}}^{(p)}, \hat{\Phi}_{D,\text{sep}}^{(p)}) \tag{6e}
\end{align*}
\]

where \( \{\hat{\Phi}_{D,\text{align}}^{(p)}, \hat{\Phi}_{D,\text{sep}}^{(p)}\} \) and \( \{\Phi_{D,\text{align}}^{(p)}, \Phi_{D,\text{sep}}^{(p)}\} \) are the aligned and separated features after/before calculating the feature decomposition loss, and \( \text{Cat}(\cdot, \cdot) \) is the channel concatenation operator.

Eventually, the entire encoder feature extraction can be regarded as:

\[
\Phi_D = \mathcal{E}_D(D_{LR}) , \quad \Phi_R = \mathcal{E}_R(R), \tag{7}
\]

where \( \Phi_D \) and \( \Phi_R \) are abbreviated forms of \( \Phi_D^{(p)} \) and \( \Phi_D^{(p)} \). Then \( \Phi_D \) and \( \Phi_R \) will be used as the input to the decoder.

### 3.1.3 Decoder

According to Sec. 3.1.2, we obtained features \( \Phi_D \) and \( \Phi_R \) each containing shared information across \( \text{dim}_1 \) channels and private information across the other \( \text{dim}_2 \) channels. The features that we consider helpful for the HR depth reconstruction task are the full depth features \( \Phi_D \) and the shared RGB features \( \Phi_{R,\text{sep}} \). Therefore, we concatenate \( \Phi_D \) and \( \Phi_{R,\text{sep}} \) in channel dimension and input them to \( \mathcal{D}_D \) to obtain the reconstructed depth image \( \hat{D}_{HR} \), and input \( \Phi_R \) to \( \mathcal{D}_R \) to obtain the reconstructed RGB image \( \hat{R} \), which is:

\[
\hat{D}_{HR} = \mathcal{D}_D(\text{Cat}(\Phi_D, \Phi_R[0: \text{dim}_2/2])), \quad \hat{R} = \mathcal{D}_R(\Phi_R). \tag{8}
\]

### 3.1.4 Spherical Contrast Refinement module

After decoding, we have obtained a preliminary estimation of the HR depth map \( \hat{D}_{HR} \). However, it potentially has some minor issues, such as blurry edges, noisy surfaces and over-transferred RGB texture. Hence, in the Spherical Contrast Refinement (SCR) module, we target the imperfect issues in \( \hat{D}_{HR} \) by contrastive learning, and the diagram is shown in Fig. 2c.

**Defect patches classifier.** Firstly, we artificially synthesize the “imperfect image dataset” using the training set. For example, for an \( m \times m \) image patch from the ground truth in the training set, we can add random noise to make the patch noisy, apply Gaussian blur to it blurry, and add the same-location RGB image to make it texture over-transferred. Patches that are not processed can be regarded as perfect. In this way, we obtain a dataset with labels “noisy”, “blurry”, “texture over-transferred” and “perfect” which can be used to train a defect patches classifier (DPC) based on ResNet34 [26].

**Positive and negative samples.** After obtaining the decoder output \( \hat{D}_{HR} \) in Eq. (8), we randomly crop it into an \( m \times m \) patch \( \hat{D}_{HR}^{\text{pat}} \), and input it into the well-trained DPC to obtain the imperfect type of \( \hat{D}_{HR}^{\text{pat}} \). After getting the imperfect label, similar to the operation for making the “imperfect image dataset”, we can transform the corresponding ground truth \( D_{HR} \) into an imperfect depth map \( \hat{D}_{HR} \). Then, we randomly crop \( N \) \( m \times m \) patches from \( \hat{D}_{HR} \) that are different from the position of \( \hat{D}_{HR}^{\text{pat}} \) and the corresponding imperfect patches \( \hat{D}_{HR}^{\text{pat}} \), named negative samples, are generated. Meanwhile, we refer to \( \hat{D}_{HR}^{\text{pat}} \) as the anchor and the same-location ground truth \( D_{HR}^{\text{pat}} \) as the positive sample. The \( k \)-th positive, anchor and negative samples are represented by \( \mu_k^+, \mu_k \), and \( \mu_k^- \), respectively.

**Spherical contrast refinement.** We input \( \mu_k^+, \mu_k \), and \( \mu_k^- \) into \( \mathcal{E}_D \) and get the contrastive refinement loss \( \mathcal{L}_{\text{SCR}} \):

\[
\mathcal{L}_{\text{SCR}} = \sum_{k=1}^{K} \sum_{n=1}^{N} \frac{S}{\mathcal{L}_D(\mu_k), \mathcal{E}_D(\mu_k^+)} - \frac{S}{\mathcal{L}_D(\mu_k), \mathcal{E}_D(\mu_k^-)} \tag{9}
\]
where $S\{\cdot, \cdot\}$ is the spherical space distance in Definition 3, and $\mu_k^+$ is the $n$th negative sample in $\mu_k^-$. Through gradient descent, we minimize the distance between $\mu_k^+$ and $\mu_k^-$ features and maximize that between $\mu_k^+$ and $\mu_k^-$ features. This process fine-tunes $E_D$. However, since the increased training cost for the SCR module, we incorporate it into the regular network training process every few iterations to strike a balance between training efficiency and effectiveness.

### 3.2. Training loss

The training loss in this paper comprises several components: the depth map reconstruction loss $L_{\text{pixel}}^D$, the RGB reconstruction loss $L_{\text{pixel}}^R$, the feature decomposition loss $L_{\text{dec}}$, and the spherical contrastive refinement loss $L_{\text{SCR}}$. We describe each loss separately next.

$L_{\text{pixel}}^D$ ensures that the estimated depth map $\hat{D}_{HR}$ output by our SSDNet is close to the ground truth depth map $D_{HR}$. $L_{\text{pixel}}^R$ ensures that the output $\hat{R}$ is close to the input $R$. Although theoretically unrelated to the depth map super-resolution task, this loss item is used to guarantee that the semantic information from the RGB image is involved in the shared RGB features, rather than simply generating random noise that is approximate to the depth features to meet the feature decomposition. Specifically,

$$L_{\text{pixel}}^D = \sum_{k=1}^{K} \| \hat{D}^{(k)}_{HR} - D_{HR}^{(k)} \|_2^2, \quad L_{\text{pixel}}^R = \sum_{k=1}^{K} \| \hat{R}^{(k)} - R^{(k)} \|_2^2.$$  

Regarding the feature decomposition loss $L_{\text{dec}}$, we utilize the spherical space distance to enhance the similarity between shared features while reducing the similarity between separated features. We define the specific structure of the feature decomposition loss $L_{\text{dec}}$ as follows:

$$L_{\text{dec}} = L_{\text{align}} - (1 - L_{\text{sepn}})^2,$$  

where

$$L_{\text{sepn}} = \sum_{p=1}^{P} S\{\Phi_{D,(p)}, \Phi_{R,(p)}\}, \quad L_{\text{align}} = \sum_{p=1}^{P} S\{\Phi_{D,(p)}^{\text{align}}, \Phi_{R,(p)}^{\text{align}}\}.$$  

Unlike $\ell_2$ distance, the spherical distance captures relative differences without being influenced by scale. For 1D feature maps with three consecutive pixels: $f_1 = [0.4, 0.5, 0.6]$, $f_2 = [4, 5, 6]$, $f_3 = [0.6, 0.5, 0.4]$, $f_1$ and $f_2$ exhibit similar pixel-wise increase structures and potentially similar extracted features, while $f_3$ differs. Thus, $f_1, f_2$ should be closer in distance and aligned, while $f_1, f_3$ should be separated. However, $\ell_2(f_1, f_2) = 7.90$, $\ell_2(f_1, f_3) = 0.28$ while $S\{f_1, f_2\} = 7.7 \times 10^{-8}$, $S\{f_1, f_3\} = 0.02$. Therefore, the spherical distance is a better choice.

When optimizing $L_{\text{dec}}$, since the value range of spherical distance $S\{\cdot, \cdot\}$ is $[0, 2]$, we have $L_{\text{align}} \rightarrow 0$, $L_{\text{sepn}} \rightarrow 1$. According to Definition 3, the derivation gives:

$$C\{\Phi_{D,(p)}^{\text{align}}, \Phi_{R,(p)}^{\text{align}}\} \rightarrow 1 \quad \text{and} \quad C\{\Phi_{D,(p)}^{\text{sepn}}, \Phi_{R,(p)}^{\text{sepn}}\} \rightarrow 0,$$

where $C$ is the cosine similarity (the last term of Eq. (5)). When...
\[ C = 1, \text{vectors have an angle of 0, indicating higher similarity.} \]

If \( C = 0 \), the vectors are orthogonal, signifying no correlation. By forcing \( C = 1 \) or \( C = 0 \) for aligned/private features mapped to spherical space, we ensure the enhancement/reduction of similarity, which achieves our goal of feature decomposition.

By combining Eqs. (9) to (11), we obtain the total loss used for training, which can be expressed as follows:

\[ \mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pixel}}^{D} + \alpha_{1}\mathcal{L}_{\text{pixel}}^{R} + \alpha_{2}\mathcal{L}_{\text{dec}} + \alpha_{3}\mathcal{L}_{\text{SCR}}. \] (13)

Note that \( \alpha_{3} = 0 \) if the current epoch does not require the spherical contrast refinement.

4. Experiment

This section will conduct a comprehensive set of experiments that aim to showcase the effectiveness of our model in addressing the GDSR task. Moreover, we will also provide evidence that substantiates the soundness of our SSDNet.

4.1. Setup

Datasets. Our experiments follow the protocol established in [27, 31, 90]. To evaluate our model, we employ four popular benchmarks: Middlebury [29, 56], Lu [45], NYU v2 [57], and RGBDD [27]. Our training and validation sets consist of the first 1000 images of NYU v2 dataset [57], divided into a 9:1 ratio. The last 449 pairs in NYU v2 are utilized as the test dataset. Middlebury [29, 56] (30 pairs), Lu [45] (6 pairs), and RGBDD [27] (405 pairs) are also used as test sets to evaluate the depth map SR ability across different scenes and objects. To furthermore demonstrate the generalization ability of our model to unknown scenarios, we incorporate the real-world branch of the RGBDD dataset, which comprises 2215/405 pairs of RGB-D images for training/test sets, respectively. Lastly, we use the root-mean-square error (RMSE) metric to evaluate the super-resolution effect of our model.

Implementation details. In our experiments, we apply bicu-
Table 3: Results of time-consuming comparison for generating a 640 × 480 HR depth map.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Middlebury [29, 56]</th>
<th>NYU V2 [57]</th>
<th>Lu [45]</th>
<th>RGBDD [27]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>×4</td>
<td>×8</td>
<td>×16</td>
<td>×4</td>
</tr>
<tr>
<td>PAC [59]</td>
<td>1.32</td>
<td>2.62</td>
<td>4.58</td>
<td>1.20</td>
</tr>
<tr>
<td>CUNet [11]</td>
<td>1.10</td>
<td>2.17</td>
<td>4.33</td>
<td>0.91</td>
</tr>
<tr>
<td>DKN [31]</td>
<td>1.23</td>
<td>2.12</td>
<td>4.24</td>
<td>0.96</td>
</tr>
<tr>
<td>FDKN [31]</td>
<td>1.08</td>
<td>2.17</td>
<td>4.50</td>
<td>0.82</td>
</tr>
<tr>
<td>FDSR [27]</td>
<td>1.13</td>
<td>2.08</td>
<td>4.39</td>
<td>1.29</td>
</tr>
<tr>
<td>GraphSR [8]</td>
<td>1.11</td>
<td>2.12</td>
<td>4.43</td>
<td>0.92</td>
</tr>
<tr>
<td>DCTNet [90]</td>
<td>1.10</td>
<td>2.05</td>
<td>4.19</td>
<td>0.88</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>1.02</strong></td>
<td><strong>1.91</strong></td>
<td><strong>4.02</strong></td>
<td><strong>1.60</strong></td>
</tr>
</tbody>
</table>

Table 1: Quantitative comparisons among the SOTA methods and our SSDNet in test datasets. The best and second best values are highlighted by bold and underline, respectively.

<table>
<thead>
<tr>
<th>Methods</th>
<th>RMSE</th>
<th>Methods</th>
<th>RMSE</th>
<th>Methods</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVLRM [47]</td>
<td>8.05</td>
<td>DJF [34]</td>
<td>7.90</td>
<td>DJFR [35]</td>
<td>8.01</td>
</tr>
<tr>
<td>DKN [31]</td>
<td>7.38</td>
<td>FDKN [31]</td>
<td>7.50</td>
<td>FDSR [27]</td>
<td>7.50</td>
</tr>
<tr>
<td>DCTNet [90]</td>
<td>7.32</td>
<td>SSDNet</td>
<td>7.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDSR* [27]</td>
<td>5.49</td>
<td>DCTNet* [90]</td>
<td>5.43</td>
<td>SSDNet*</td>
<td><strong>5.38</strong></td>
</tr>
</tbody>
</table>

Table 2: Quantitative comparisons among the SOTA methods and SSDNet on real-world branch of RGBDD. The best and second best RMSEs are highlighted by bold and underline. Model* means that the model has been fine-tuned on the real-world training data.

<table>
<thead>
<tr>
<th>Methods</th>
<th>PAC</th>
<th>CUN</th>
<th>DKN</th>
<th>FDKN</th>
<th>FDSR</th>
<th>GSR</th>
<th>DCT</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>0.20</td>
<td>0.23</td>
<td>0.21</td>
<td>0.04</td>
<td>0.01</td>
<td>0.92</td>
<td>0.08</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Table 3: Results of time-consuming comparison for generating a 640 × 480 HR depth map.

4.2. Empirical validation experiment

**Impact of network hyper-parameters.** In our proposed SSDNet, the number of Restormer Blocks $P$, and number of dimensions in Restormer $C$, and the number of negative samples in SCR $N$ are crucial factors in enhancing the super-resolution performance. We conduct experiments on the NYU validation set to study the impact of different combinations of $P, C, N$. Initially, we fix any two parameters at $C = 64$, $P = 4$, and $N = 8$, respectively, and evaluate the prediction quality of the unfixed parameter at $P = 3, 4, 5, 6$, $C = 32, 64, 96, 128$, and $N = 2, 4, 8, 16$. The results are summarized in Fig. 3. We observe that the performance is limited when $P < 4$, and redundant parameter increases do not yield corresponding improvements when $P > 4$. Similarly, increasing $C$ to more than 64 does not produce significant benefits but instead increased the training cost and computational expense. For the SCR module, the increase in training burden does not match the improvement in effectiveness when $N$ exceeds 8. Therefore, to ensure a balance between performance and computational cost, we choose $P = 4, C = 64$, $N = 8$ for subsequent experiments.

**Visualization of feature decomposition.** Aligned and separated features are visualized in Fig. 4. Aligned features extract shared properties (edges/contours) from the depth/RGB image pair. Separated features capture modality-specific details (textures in RGB images, smooth depth distance in depth maps). The visualization aligns with our motivation.

4.3. Comparison with SOTA methods

This section aims to evaluate the performance of our SSDNet on several popular benchmarks in Sec. 4.1, and compare our results with the state-of-the-art methods, including PAC [59], CUNet [11], DKN [31], FDKN [31], FDSR [27], GraphSR [8], and DCTNet [90].

**Qualitative Comparison.** We present comparisons of the error maps with ground truth depth maps among SOTA methods in Figs. 5 and 6. Intuitively, the depth predictions of SSDNet exhibit lower prediction errors and are closer to the ground truth images, especially in terms of perception and restoration for the edges and contours. Further visual comparisons are shown in the supplementary material.

**Quantitative results for traditional testsets.** The quantitative results with scaling factors of 4, 8, and 16 on four test sets are shown in Tab. 1. Compared with existing methods that achieve good results on certain datasets or super-resolution factors, our SSDNet produces satisfactory predic-
We conduct all testing without any additional fine-tuning, ensuring that all testing is conducted without any additional fine-tuning. The number of Restormer blocks is increased to ensure that the model has a better super-resolution performance than previous methods.

**Quantitative results for real-world branch.** In addition, following [27, 90], we use the pre-trained model to directly test on the real-world branch of the RGBDD dataset to explore the generalization ability in real-world scenarios. We conduct all testing without any additional fine-tuning, and the quantitative results are presented in Tab. 2. Furthermore, we also perform targeted training and testing on this dataset. Regardless of whether fine-tuning or not, SSDNet outperforms previous methods in RMSE, highlighting its powerful generalization ability in handling unknown scenes.

**Parameter & running time comparison.** We demonstrate the relationship between the number of model learnable parameters vs. RMSE on the RGBDD dataset for ×4, ×8, and ×16 and Middlebury for ×4 in Fig. 1. Our model exhibits a clear advantage over existing methods with relatively fewer parameters. The time-consuming comparison is presented in Tab. 3. They both demonstrate the efficiency of our method for the GDSR task, showing the potential for developing lightweight networks and practical applications in the future.

### 4.4. Ablation Studies

In this section, we validate the design rationality of our SSDNet through ablation experiments based on the RGBDD testset, and present the results in Tab. 4.

**SSDNet architecture.** In Exp. I, we share the encoders and decoders for RGB and depth images, i.e., a unified encoder and decoder are used instead of \( \{E_D, E_R\} \) and \( \{D_D, D_R\} \). The number of Restormer blocks is increased to ensure that the learnable parameter number is comparable.

**Spherical feature decomposition.** In Exp. II, we eliminate the entire feature decomposition module, i.e., \( \mathcal{L}_{dec} \) in Eq. (11) will not be employed. In Exp. III, we change the distance measure for \( \mathcal{L}_{sepn} \) and \( \mathcal{L}_{align} \) in Eq. (12) from spherical space distance to Euclidean distance, i.e., \( \ell_2 \)-loss.

**Reconstruction RGB.** In Exp. IV, We eliminate \( \mathcal{L}_{pixel}^R \) in Eq. (10) and explore the change for feature extraction capability without the restriction of reconstructing RGB.

**Spherical contrast refinement.** In Exp. V, we remove the SCR module and use \( \hat{D}_{HR} \) in Eq. (8) as the final output of our model.

**Analysis.** Tab. 4 shows that changing settings leads to performance degradation, validating our model design. Exp. I: Unified encoder/decoder ignores modality-specific features, hindering efficient cross-modal feature extraction and causing the largest degradation. Exp. II: The absence of \( \mathcal{L}_{dec} \) prevents decomposing the shared/private features. Exp. III: \( \ell_2 \) distance partially decomposes features but is less effective than \( S \). Exp. IV: \( \mathcal{L}_{pixel}^R \) preserves semantic information in RGB and avoids excessive adaptation of feature separation/alignment. Exp. V: Removal of SCR leads to mentioned detailed issues and performance degradation.

### 5. Conclusion

We propose a Spherical Space feature Decomposition network (SSDNet) for guided depth map super-resolution, where a Restormer-based encoder is used to extract the global features of the inputs, and the intermediate features will be mapped to the spherical space to complete the separation/alignment of shared/private features, respectively. Finally, a Restormer-based decoder is to reconstruct the HR depth map. The spherical contrast refinement module is then employed to address the possible detail issues, e.g., blurry edges, noisy surfaces and over-transferred RGB texture. The satisfactory output results and the lightweight size demonstrate the superiority of our approach.

### Acknowledgement

This work has been supported by the National Key Research and Development Program of China under grant 2018AAA0102201, the National Natural Science Foundation of China under Grant 61976174 and 12201497, Shaanxi Fundamental Science Research Project for Mathematics and Physics under Grant 22JQ033, the Fundamental Research Funds for the Central Universities under Grant D5000220060, and partly supported by the Alexander von Humboldt Foundation.

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Encoder&amp;Decoder</th>
<th>Form of ( \mathcal{L}_{dec} )</th>
<th>( \mathcal{L}_{pixel}^R )</th>
<th>SCR Module</th>
<th>( \times 4 )</th>
<th>( \times 8 )</th>
<th>( \times 16 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp. I</td>
<td>Shared</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1.23</td>
<td>2.10</td>
<td>3.49</td>
</tr>
<tr>
<td>Exp. II</td>
<td>Private</td>
<td>×</td>
<td>√</td>
<td>✓</td>
<td>1.17</td>
<td>1.94</td>
<td>3.27</td>
</tr>
<tr>
<td>Exp. III</td>
<td>Private</td>
<td>✓</td>
<td>√</td>
<td>✓</td>
<td>1.15</td>
<td>1.83</td>
<td>3.06</td>
</tr>
<tr>
<td>Exp. IV</td>
<td>Private</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>1.16</td>
<td>1.89</td>
<td>3.29</td>
</tr>
<tr>
<td>Exp. V</td>
<td>Private</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>1.18</td>
<td>1.87</td>
<td>3.36</td>
</tr>
<tr>
<td>Ours</td>
<td>Private</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td><strong>1.04</strong></td>
<td><strong>1.72</strong></td>
<td><strong>2.92</strong></td>
</tr>
</tbody>
</table>

Table 4: Results of ablation experiments on the RGBDD test set. **Bold** indicates the best score in terms of RMSE.
References


[80] Mingde Yao, Dongliang He, Xin Li, Zhihong Pan, and Zhiwei Xiong. Bidirectional translation between uhd-hdr and hd-sdr videos. *IEEE Transactions on Multimedia*, 2023. 1


