This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

SphereSR: 360° Image Super-Resolution with Arbitrary Projection via Continuous Spherical Image Representation

Youngho Yoon, Inchul Chung, Lin Wang*, and Kuk-Jin Yoon
Visual Intelligence Lab., KAIST, Korea
{dudgh1732,inchul1221,wanglin,kjyoon}@kaist.ac.kr

Abstract

The 360° imaging has recently gained much attention; however, its angular resolution is relatively lower than that of a narrow field-of-view (FOV) perspective image as it is captured using a fisheye lens with the same sensor size. Therefore, it is beneficial to super-resolve a 360° image. Several attempts have been made, but mostly considered equirectangular projection (ERP) as one of the ways for 360° image representation despite the latitude-dependent distortions. In that case, as the output high-resolution (HR) image is always in the same ERP format as the low-resolution (LR) input, additional information loss may occur when transforming the HR image to other projection types. In this paper, we propose SphereSR, a novel framework to generate a continuous spherical image representation from an LR 360° image, with the goal of predicting the RGB values at given spherical coordinates for super-resolution with an arbitrary 360° image projection. Specifically, first we propose a feature extraction module that represents the spherical data based on an icosahedron and that efficiently extracts features on the spherical surface. We then propose a spherical local implicit image function (SLIIF) to predict RGB values at the spherical coordinates. As such, SphereSR flexibly reconstructs an HR image given an arbitrary projection type. Experiments on various benchmark datasets show that the proposed method significantly surpasses existing methods in terms of performance.

1. Introduction

The 360° imaging has recently gained much attention in many fields, including the AR/VR field. In general, raw 360° images are transformed into 2D planar representations while preserving the omnidirectional information, *e.g.*, equirectangular projection (ERP) and cube map projection (CP) to ensure compatibility with imaging pipelines. Omnidirectional images (ODIs)[1] are sometimes projected back onto a sphere or transformed with different types of projection and rendered for display in certain applications.

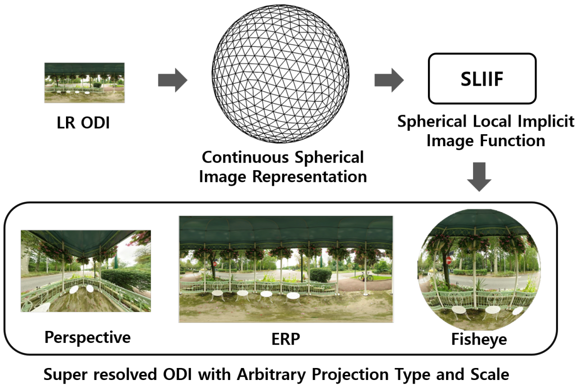
LR ODI
Continuous Spherical Image Representation
SLIIF
Spherical Local Implicit Image Function
Perspective
ERP
Fisheye
Super resolved ODI with Arbitrary Projection Type and Scale

Figure 1. Learning continuous spherical image representation. SphereSR leverages SLIIF to predict RGB values at given spherical coordinates for SR with arbitrary image projection.

However, the angular resolution of a 360° image tends to be lower than that of a narrow field-of-view (FOV) perspective image, as it is captured using a fisheye lens with an identical sensor size. Moreover, the 360° image quality can be degraded during a transformation between different image projection types. Therefore, it is imperative to super-resolve the low-resolution (LR) 360° image by considering various projections to provide high-level visual quality under diverse conditions. Early studies attempted to reconstruct high-resolution (HR) ODIs by interpolating the missing data between the LR image pixels [3, 5, 25].

Recently, deep learning (DL) has brought a significant performance boost to 2D single image super-resolution (SISR) [17, 37, 44]. These methods mostly explore super-resolving 2D LR image using high-capacity convolutional neural networks (CNNs) via, *e.g.*, residual connections [21], and learning algorithms including generative adversarial networks (GANs) [18, 40, 41]. However, directly using these methods for 360° images represented in 2D planar representations is less applicable as the pixel density and texture complexity vary across different positions in 2D planar representations of 360° images, as pointed in [10].

*Lin Wang is currently with HKUST.

[1] Throughout the paper, we use omnidirectional images and 360° images interchangeably.

Consequently, several attempts were made to address SR problems in relation to 360° imaging [10, 28, 36, 46]. In particular, 360-SS [28] proposes a GAN-based framework using the Pix2Pix pipeline [14]. however, it focuses only on the ERP format and does not fully consider the properties of 360° images. LAU-Net [10] introduces a method to identify ODI distortions on the latitude and upsample ODI pixels on segmented patches. However, this process leads to considerable disconnections along the patches. In a nutshell, existing methods for ODI SR ignore the projection process of 360° images in real applications and only take the ERP image as the LR input, producing the HR ERP output. Indeed, a 360° image can be flexibly converted into various projection types, as in real applications, the user specifies the projection type, direction, and FOV. Thus, it is vital to address the ERP distortion problems and strive to super-resolve an ODI image to an HR image with *an arbitrary projection type* rather than a fixed type.

In this paper, as shown in Fig. 1, we propose a novel framework, called SphereSR, with the goal of super-resolving an LR 360° image to an HR image *with an arbitrary projection type via continuous spherical image representation*. First, we propose a feature extraction module that represents spherical data based on icosahedron and efficiently extracts features on a spherical surface composed of uniform faces (Sec. 3.1). As such, we solve the ERP image distortion problem and resolve the pixel density difference according to the latitude. Second, we propose a spherical local implicit image function (SLIIF) that can predict RGB values at arbitrary coordinates on a sphere feature map, inspired by LIIF [7] (Sec. 3.2). SLIIF works on *triangular faces*, buttressed by position embedding based on normal plane polar coordinates to obtain relative coordinates on a sphere. Therefore, our method tackles pixel-misalignment issue when the image is projected onto another ODI projection. As a result, SphereSR can predict RGB values for any SR scale parameters. Additionally, to train SphereSR, we introduce a feature loss that measures the similarity between two projection types, leading to a considerable performance enhancement (Sec. 3.3). Extensive experiments on various benchmark datasets show that our method significantly surpasses existing methods.

In summary, the contributions of our paper are four-fold. (I) We propose a novel framework, called SphereSR, with the goal of super-resolving an LR 360° image to an HR image with an arbitrary projection type. (II) We propose a feature extraction module that represents spherical data based on an icosahedron and extracts features on a spherical surface. (III) We propose SLIIF, which predicts RGB values from the spherical coordinates. (IV) Our method achieves the significantly better performance in the extensive experiments.

2. Related Works

Omnidirectional Image SR and Enhancement. Early ODI SR methods [2, 4, 6, 15, 26] focused on assembling and optimizing multiple LR ODIs on spherical or hyperbolic surfaces. On the other hand, as the distortion in ODI arises due to the projection of the original spherical image onto a 2D planar image plane, recent research has focused on tackling and solving distortion in ODI using 2D convolution to achieve a qualitative result in the observation space, *i.e*., a spherical surface. Su *et al*. [36] and Zhou *et al*. [46] proposed evaluation methods for ODI weighted with the projected area on a spherical surface. Two other works [27, 30] adapted existing SISR models to ERP SR by fine-tuning or by adding a distortion map as an input to tackle different distortions. Ozcinar *et al*. [28] leveraged GAN to super-resolve an ODI by applying WS-SSIM [46]. Zhang *et al*. [45] also proposed the GAN-based framework employing multi-frequency structures to enhance the panoramic image quality up to the high-end camera quality. Liu *et al*. [22] focused on the 360° image SR utilizing single-frame and multi-frame joint learning and a loss function weighted differently along the latitude. Deng *et al*. [10] considered varying pixel density and texture complexity along latitude by proposing network allowing distinct up-scaling factors along the latitude bands. *Unlike the aforementioned methods, we propose to predict RGB values at the given spherical coordinates of an HR image with respect to an arbitrary project type from an LR 360° image.*

2D SISR with an Arbitrary Scale. Research on SISR with an arbitrary scale has been actively conducted. Lim *et al*. [21] first proposed a method that enables multiple scale factors over one network. MetaSR [13] achieves SR with the non-integer scale factors. However, both methods are limited to SR with the symmetric scales. Later on, Wang *et al*. [39] proposed a framework that enables asymmetric scale factors along horizontal and vertical axes. Moreover, SRWarp [34] generalizes SR toward an arbitrary image transformation. Although these methods are effective for 2D SISR with an arbitrary scale factor, they fail in that they are not directly applicable to 360° image SR due to the difference between the xy-coordinate (2D) and the spherical coordinates in ODI domains. *We overcome this challenge by proposing SphereSR, which leverages SLIIF to predict RGB values for arbitrary spherical coordinates.*

Continuous Image Representation. Research on implicit neural representation (INR) has been conducted to express 3D spaces, *e.g*., 3D reconstruction and novel view synthesis via continuous ways [23, 24, 32]. Since then, continuous image representation has been explored on the (x,y) coordinate. Some studies used networks to predict the RGB value of each pixel from the latent vector on (x,y) coordinate without a spatial convolution for 2D image generation [1, 33]. LIIF [7] proposes to bridge between discrete and continu-

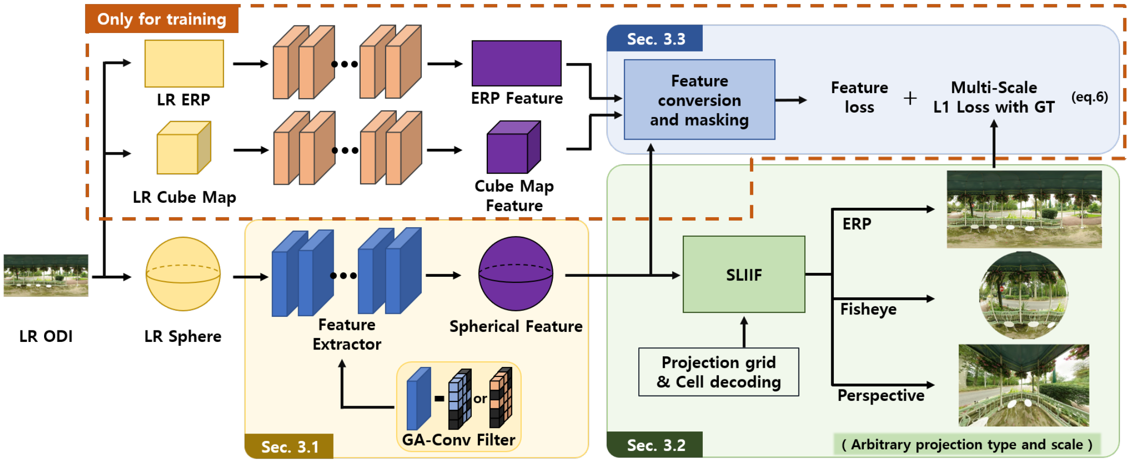

Figure 2. Overall framework of the proposed SphereSR.

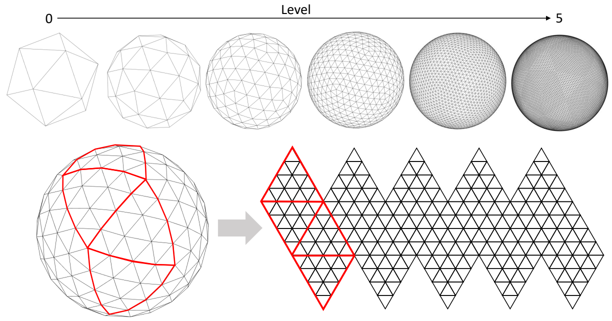

Figure 3. Subdivision process of an icosahedron. We define a pixel as the face of a subdivided icosahedron.

ous representation for images on the (x,y) coordinate. *We propose SLIIF, which enables continuous image representation on the unit sphere.*

CNNs for Spherical Images. Cohen *et al.* [8] proposed a CNN-based method on the sphere with structural characteristics called rotational equivariance. However, it requires Fourier transform for each step. Coors *et al.* [9] developed a CNN filter in light of spatial location on the sphere to solve the distortion problem of the ERP images. Su *et al.* [35] proposed a kernel transformer network that converts the pre-trained kernels on the perspective images into ODIs. SpherePHD [19] proposed a convolution kernel applicable to triangular pixels defined on the faces of the icosahedron. Zhang *et al.* [43] performed convolution using a hexagonal filter applicable to the vertices of icosahedron. In this work, we focus on ODI SR and propose SphereSR, which applies convolution to a spherical structure created through the subdivision of an icosahedron.

3. Method

Overview. As shown in Fig. 2, we propose a novel framework, SphereSR, the goal of which is to obtain continuous spherical image representation from a given icosahedron input. First, we introduce a feature extraction method for spherical images that efficiently extracts features from an image on an icosahedron (Sec. 3.1). Second, we propose the Spherical Local Implicit Image Function (SLIIF), which predicts RGB values through the extracted features in order to reconstruct an HR image flexibly with an arbitrary projection type(Sec. 3.2). Lastly, We propose a feature loss to obtain support from features of other projection types by utilizing the advantage of SLIIF, i.e., conversion to an arbitrary projection type(Sec. 3.3).

3.1. Feature Extraction for Spherical Images

Feature extraction is crucial yet challenging for spherical image SR, as we focus on very large scale factors, *e.g.*, ×16. In this situation, *it is imperative to tackle the memory overload issue while ensuring high SR performance*. Hence, the proposed SphereSR represents the spherical data based on an icosahedron and efficiently extracts features on a spherical surface composed of uniform faces. This is achieved by a new data structure on the icosahedron coupled with weight sharing between kernels of different directions.

Data structure. Inspired by the convolution of icosahedron data in SpherePHD [19], we propose a new spherical data structure. To implement the convolution operation, SpherePHD [19] uses the subdivision process of an icosahedron, described in Fig. 3, and creates a call-table containing the indices of N neighboring pixels for each pixel, subsequently using it to stack every neighboring pixel. Convolution is then performed with a kernel of size $[N+1, 1]$. However, this implementation is not memory-efficient, as it requires additional N channels for stacking the neighboring pixels for every convolution operation. To solve this problem, we propose a new data structure by which the convolution operation can be directly applied without stacking the neighbors in a call-table. As shown on the left side of Fig. 4, we rearrange the original data in the direction of the arrow while transforming the triangular pixels to rectangular pixels such that conventional 2D convolution can be applied.

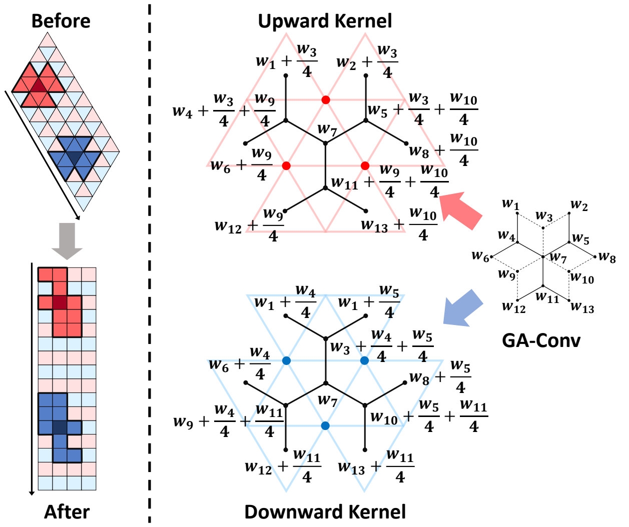

Figure 4. New kernel weight sharing. Left: proposed new data structure, Right: our kernel weight sharing scheme

Here, an upward kernel (red kernel) for each upward (△) aligned pixel is arranged in the odd-numbered rows, and a downward kernel (blue kernel) for each downward (▽) aligned pixel is arranged in the even-numbered rows. (More details can be found in the supplementary.)

Kernel Weight Sharing. While the memory overload issue can be resolved by the proposed data structure, it is still necessary to ensure high SR performance. SpherePHD [19] rotates each upward or downward kernel by 180° to obtain the same kernel shape. Therefore, it is possible to share weights of the up/downward kernels whose directions and shapes are symmetric to each other. However, as the direction of the kernel weight changes for adjacent pixels, high performance cannot be ensured if the characteristics of the texture according to the direction need to be identified.

To solve this problem, we introduce a kernel weight sharing scheme called GA-Conv which geometrically aligns up/downward directional kernels *without rotation*. As shown on the right side of Fig. 4, the pixel (face) combinations of two kernels, where up/downward kernels are applied, are shaped differently depending on the direction of the center pixel. However, if three vertices of the center pixel (denoted by the red and blue dots on the right side of Fig. 4) are included in the pixel combination as imaginary pixels, the shapes of two different up/downward pixel combinations can be made to be geometrically identical. To this end, rather than averaging and creating imaginary pixels, we distribute the kernel weight to nearby pixels. For the upward kernel, image pixel weights w_3, w_9, and w_{10} are distributed to the nearest four pixels, except the center pixel. The downward kernel weights w_4, w_5, and w_{11} are distributed in the same way. Details are presented on the right side of Fig. 4. In this way, the feature extraction module can be applied to any pixel without rotation.

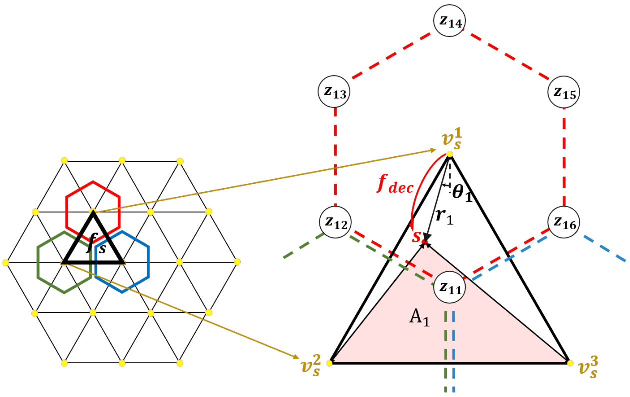

Figure 5. Spherical local implicit image function.

3.2. Spherical Local Implicit Image Function (SLIIF)

Overall Process of SLIIF. With the data efficiently represented, we now describe the method used to super-resolve ODIs efficiently on an arbitrary scale. Our main idea is to predict an RGB value for an arbitrary coordinate on the unit sphere S^2 using a feature map extracted by means of GA-Conv, as described in Sec. 3.1. Inspired by LIIF [7], we propose SLIIF, which learns an implicit image function on S^2 using icosahedral faces. SLIIF takes a spherical coordinate of the point on the unit sphere and its neighboring feature vectors as inputs and predicts the RGB value. It can be formulated as:

$$I(s) = f_{dec}(z, s), s \in S^2 \tag{1}$$

where f_{dec} is a decoding function shared with all icosahedral faces, s is the point on the unit sphere S^2, z represents a feature vector formed by concatenating the neighboring feature vectors of s, and $I(s)$ is the predicted RGB value of s.

For a pixel in an image that can be formed by any arbitrary projection from the unit sphere, there is a corresponding point s on unit sphere S^2.[2] The face containing s is denoted as f_s, and the three vertices surrounding f_s are denoted as v_s^1, v_s^2, and v_s^3 (see Fig. 5). The RGB values of s w.r.t. the coordinate system of the three vertices are first calculated and then ensembled based on the triangular areas A_1, A_2, and A_3 to obtain the final RGB value of point s. The RGB value of s w.r.t. each vertex v_s^j is calculated with the features of six faces containing the vertex and relative polar coordinate. The features of the six faces are concatenated clock-wise starting from f_s to preserve geometrical consistency. Here, we denote the concatenated features as z_j and the polar coordinate of s with respect to v_s^j as (r_j, θ_j). To better utilize the positional information, the (r_j, θ_j) values are encoded with $\gamma(p) = (\sin(2^0\pi p), \cos(2^0\pi p), ..., \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p))$

[2] The coordinate of s is computed by using the center point of a pixel.

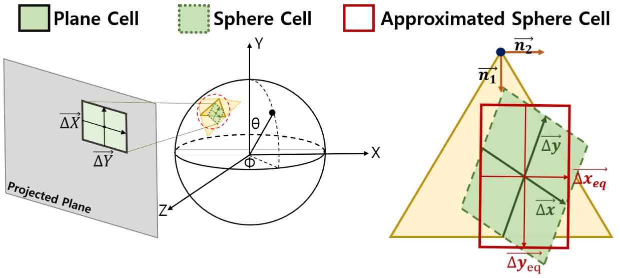
Plane Cell
Sphere Cell
Approximated Sphere Cell
Y
X
Z
θ
Φ
$\overrightarrow{\Delta X}$
$\overrightarrow{\Delta Y}$
Projected Plane
$\overrightarrow{n_2}$
$\overrightarrow{n_1}$
$\overrightarrow{\Delta y}$
$\overrightarrow{\Delta x_{eq}}$
$\overrightarrow{\Delta x}$
$\overrightarrow{\Delta y_{eq}}$

Figure 6. Sphere-oriented Cell Decoding.

to extend the dimension of the relative coordinate, as introduced in [24, 38]. As such, we can predict the RGB value of point $(\theta, \phi) \in S^2$, which can be formulated as follows:

$$I(\theta, \phi) = \sum_{j=1}^{3} \frac{A_j}{A} \cdot f_{dec}(z_j, [\gamma(r_j), \gamma(\theta_j)]) \tag{2}$$

When the pixel in the image corresponds to the vertex on S^2, we can still utilize the aforementioned procedure because any choice of neighboring vertices results in the same RGB value due to the triangle area-based weighting.

Sphere-oriented Cell Decoding. Through SLIIF, we can predict the RGB value for any point on S^2. That is, we can generate the desired HR image for any projection type by predicting the RGB value of each pixel.

However, SLIIF provides the RGB value only for the center of the pixel and discards the information within the pixel area, except for the center value. To handle this, LIIF [7] defines the cell decoding value as the width and height of the query pixel of interest. Nonetheless, this definition cannot be directly applied to a sphere, as the corresponding region on the sphere does not have a rectangular shape and the direction of the reference vertex, where the RGB value is initially calculated, continually changes. Thus, we propose sphere-oriented cell decoding, a method that takes into account the relation between the pixel region on the projected output and the corresponding region on S^2. By adding the cell decoding value as input to SLIIF, we can fully utilize the information within the pixel area. As shown in Fig. 6, we aim to obtain the RGB value of a rectangular pixel on the projected plane. We call this rectangular pixel a plane cell, which can be expressed using the two vectors $\overrightarrow{\Delta X}, \overrightarrow{\Delta Y}$. The sphere cell, the corresponding area of the plane cell on the sphere, is located on the face on which the corresponding point of pixel center is located on the sphere. The sphere cell can also be expressed using the two vectors $\overrightarrow{\Delta x}, \overrightarrow{\Delta y}$. The relationship between $\overrightarrow{\Delta X}, \overrightarrow{\Delta Y}$ and $\overrightarrow{\Delta x}, \overrightarrow{\Delta y}$ depends on the projection type and location of the pixel center (refer to the supplementary for details).

For geometrical consistency of the orders of the concatenated features, the relative coordinate of s, and the cell decoding value among the pixels, we need to define new axis vectors, $\overrightarrow{n_1}$ and $\overrightarrow{n_2}$, invariant to the face orientation. The unit vector $\overrightarrow{n_1}$ is defined as a vector between the reference vertex and the face center, and the unit vector $\overrightarrow{n_2}$ is defined by the 90 degree counter-clockwise rotation of $\overrightarrow{n_1}$. To determine the height and width based on this coordinate system, we approximate the parallelogram sphere cell to an axis-aligned rectangle. The approximated sphere cell is defined as a rectangle which can be expressed using the two vectors $\overrightarrow{\Delta x_{eq}}, \overrightarrow{\Delta y_{eq}}$ with an area identical to that of the parallelogram sphere cell and with the largest intersection area within the parallelogram sphere cell. Based on the approximated rectangular sphere cell, we finally formulate the sphere-oriented cell decoding value as shown below:

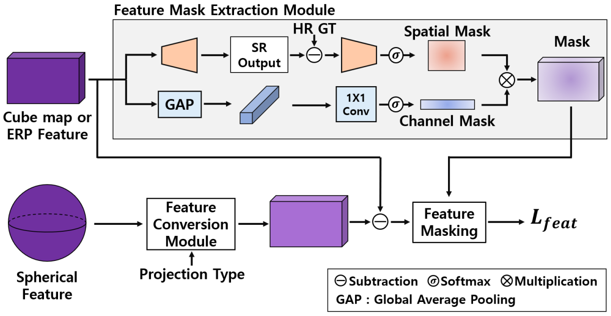

Figure 7. Proposed feature loss between the spherical and ERP features.

$$\begin{pmatrix} \overrightarrow{\Delta x} \\ \overrightarrow{\Delta y} \end{pmatrix} \approx \begin{pmatrix} \overrightarrow{\Delta x_{eq}} \\ \overrightarrow{\Delta y_{eq}} \end{pmatrix} = \begin{pmatrix} c_x \overrightarrow{n_1} \\ c_y \overrightarrow{n_2} \end{pmatrix} \tag{3}$$

$$\Rightarrow c = [c_x, c_y] = \left(\frac{|\overrightarrow{\Delta x_{eq}}|}{|\overrightarrow{n_1}|}, \frac{|\overrightarrow{\Delta y_{eq}}|}{|\overrightarrow{n_2}|} \right) \tag{4}$$

As a result, we can predict the RGB value $I(X, Y)$ for any point on the projected plane based on the following equation,

$$\begin{aligned} I(X, Y) &= I(\theta, \phi, c) \\ &= \sum_{j=1}^{3} \frac{A_j}{A} \cdot f_{dec}(z_j, [\gamma(r_j), \gamma(\theta_j)], [c_x, c_y]) \end{aligned} \tag{5}$$

3.3. Loss Function

We train the proposed framework using two loss terms. First, we use the multi-scale L1 loss. With the L1 loss defined on multiple scales, our framework can learn more about various relative coordinates and cell decoding values. Second, we design a feature loss module to measure the similarity between the features extracted from the sphere and other projection types.

As shown in fig. 7, we design a feature mask from the ERP or cube map feature. The spatial part of the mask is generated from the difference between the predicted SR

Table 1. ERP SR results on the ODI-SR and SUN 360 Panorama Dataset. **Bold** indicates the best results.

Scale	x8				x16			
Method	ODI-SR		SUN 360 Panorama		ODI-SR		SUN 360 Panorama	
	WS-PSNR	WS-SSIM	WS-PSNR	WS-SSIM	WS-PSNR	WS-SSIM	WS-PSNR	WS-SSIM
Bicubic	19.64	0.5908	19.72	0.5403	17.12	0.4332	17.56	0.4638
SRCNN [11]	20.08	0.6112	19.46	0.5701	18.08	0.4501	17.95	0.4684
VDSR [16]	20.61	0.6195	19.93	0.5953	18.24	0.4996	18.21	0.4867
LapSRN [17]	20.72	0.6214	20.05	0.5998	18.45	0.5161	18.46	0.5068
MemNet [37]	21.73	0.6284	21.08	0.6015	20.03	0.5411	19.88	0.5401
MSRN [20]	22.29	0.6315	21.34	0.6002	20.05	0.5416	19.87	0.5316
EDSR [21]	23.97	0.6417	22.46	0.6341	21.12	0.5698	21.06	0.5645
D-DBPN [12]	24.15	0.6573	23.70	0.6421	21.25	0.5714	21.08	0.5646
RCAN [44]	24.26	0.6628	23.88	0.6542	21.94	0.5824	21.74	0.5742
EBRN [29]	24.29	0.6656	23.89	0.6598	21.86	0.5809	21.78	0.5794
360-SS [28]	21.65	0.6417	21.48	0.6352	19.65	0.5431	19.62	0.5308
LAU-Net [10]	24.36	**0.6801**	24.02	0.6708	22.07	0.5901	21.82	0.5824
SphereSR(Ours)	**24.37**	0.6777	**24.17**	**0.6820**	**22.51**	**0.6370**	**21.95**	**0.6342**

ODI and the HR ground truth. The channel part of the mask is generated from the features via channel-wise global average pooling. In this way, we obtain a feature mask emphasizing the relevant parts with high accuracy. Also, the spherical features are converted to the shapes of other projection types via the SLIIF feature conversion module. Finally, the converted features are subtracted and masked to formulate the feature loss L_{feat}. The total loss is as follows:

$$Loss = \frac{1}{N}\sum_{j=1}^{N} \parallel I_j^{est} - I_j^{gt} \parallel_1 + \lambda L_{feat} \qquad (6)$$

4. Experiments

4.1. Dataset and Implementation

We train and test SphereSR using the ODI-SR dataset [10] and the SUN360 panorama dataset [42]. For training, 750 out of 800 ODI-SR training images are used and the remaining 50 images are used for validation. For testing, we use 100 images from the ODI-SR test dataset and another 100 images from the SUN360 panorama dataset. The resolution of an HR ODI is 1024×2048, and training is performed for the scales of ×8 and ×16. As shown in Fig. 3, SphereSR takes an image on an icosahedron as input converted from LR ODIs, and the icosahedron subdivision levels for the scales of ×8 and ×16 are set to 5 and 6, respectively. (More details are in the supplementary.)

4.2. Evaluation on ERP

We use the ODI-SR and SUN360 Panorama datasets for an evaluation. We compare SphereSR with 9 models for 2D SISR, including SRCNN [11], VDSR [16], LapSRN [17], MemNet [37], MSRN [20], EDSR [21], D-DBPN [12], RCAN [44], EBRN [29] and 2 models for ODI-SR, i.e., 360-SS [28] and LAU-Net [10]. We use WS-PSNR [46] and WS-SSIM [46] as evaluation metrics.

Quantitative results. Table 1 shows the results of quantitative comparisons of ×8 and × 16 SR on the ODI-SR and the SUN 360 panorama datasets. As shown here, SphereSR outperforms all of the other methods on both datasets, except in the ×8 SR case on the ODI-SR dataset, where SphereSR shows performance comparable to that of LAU-Net. However, for ×16 SR, SphereSR shows better performance compared to LAU-Net in WS-PSNR and WS-SSIM on both the ODI-SR and the SUN360 panorama datasets.

Qualitative comparison. Figure 8 shows the results of a visual comparison of ×8 SR images on the ODI-SR dataset. As shown here, SphereSR reconstructs clear textures and more accurate structures, while the other methods compared in this case are affected by the problems of blurred edges or distorted structures. From this visual comparison, we can conclude that SphereSR produces textures of repeated patterns more accurately than ERP networks.

4.3. SR for Other Projection Types

In this section, we verify whether the proposed SphereSR, trained using the ERP images on the ODI-SR dataset, can perform well for any projection type. First, we conduct an experiment involving conversion to a FOV 90° perspective image with a size 512×512. We then conduct another experiment on the conversion to a FOV 180° fisheye image with a size 1024×1024. In addition, we use circular fisheye projection, one of several types of fisheye projections. For a comparison with other SR models, we use bicubic interpolation to convert to the desired projection type. The ERP GT image is also interpolated using the bicubic method to the desired projection type for a performance evaluation. We use PSNR and SSIM as evaluation metrics. Note that we select five random directions, generate a projection output suitable for the corresponding direction, and calculate the mean value for PSNR and SSIM.

Perspective Image. Table 2 shows the quantitative results for perspective image SR. SphereSR again achieves the best performance on both the ODI-SR and the SUN360 datasets. LAU-Net [10] achieves PSNR values of 26.39dB

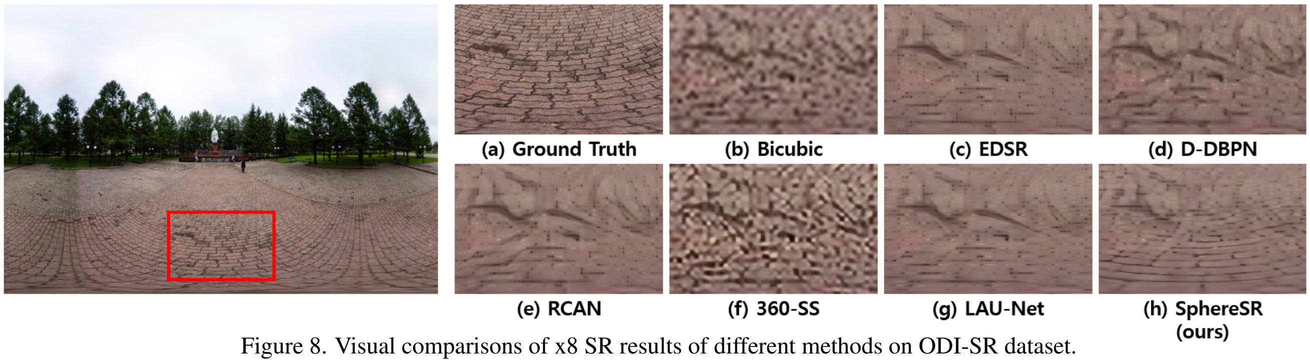

Figure 8. Visual comparisons of x8 SR results of different methods on ODI-SR dataset.

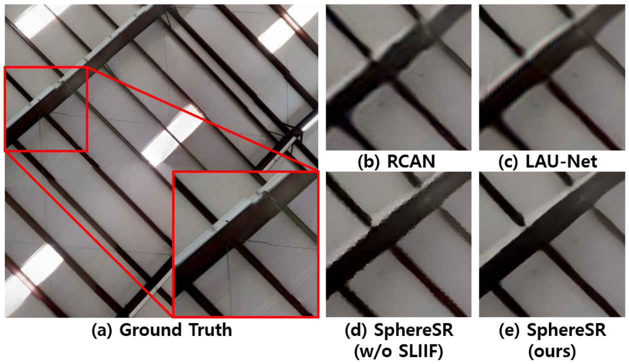

Figure 9. Visual comparison for x8 SR of perspective images on ODI-SR dataset.

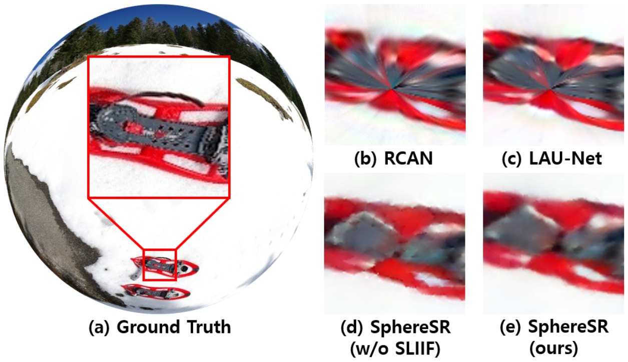

Figure 10. Visual comparison for x8 SR of fisheye images on ODI-SR dataset.

and 24.33dB on the corresponding datasets. In contrast, our method significantly surpasses LAU-Net and achieves the highest PSNR values of 26.76dB and 24.46dB on the two respective both datasets. In addition, when removing the SLIIF component in SphereSR, the corresponding PSNR values drop by 0.1dB and 0.14dB. In Fig. 9, we show a visual comparison between SphereSR with SLIIF, SphereSR without SLIIF, RCAN [44] and LAU-Net [10]. As shown in the figure, SphereSR reconstructs clear straight lines and textures better than other the RCAN(b) and LAU-Net(c) methods. Also, as a comparison of the usage of SLIIF(d,e), triangle-shaped artifacts are created when SLIIF is not used(d), but it can be confirmed that clear straight lines are created when SLIIF is used(e).

Fisheye. Table 2 shows the quantitative results for fisheye image SR. It can be seen that SphereSR has the highest performance in terms of the PSNR and SSIM values on the ODI-SR and SUN360 panorama datasets. Among the methods for 2D SISR, RCAN achieves the second-highest PSNR value of 24.40dB on the ODI-SR dataset. On the SUN360 panorama dataset, LAU-Net achieves the second-highest PSNR of 24.97dB. Our method results in the highest PSNR and SSIM values, showing the best SR performance. In Fig. 10, we show a visual comparison between SphereSR with SLIIF, SphereSR without SLIIF, RCAN [44] and LAU-Net [10]. Specifically, we crop the area to view the SR results at the south pole. As shown in the figure, RCAN(b) and LAU-Net(c) generate inappropriate textures with several lines rushing to the south pole. On the other hand, SphereSR(w/o SLIIF)(d) and SphereSR(w/ SLIIF)(e) do not encounter such a problem. Moreover, In the case of (e), it eliminates the triangle-shaped artifact generated in (d).

4.4. Ablation Study and Analysis

In this section, we study the effectiveness of each of our proposed modules, *e.g.*, GA-Conv, SLIIF, and feature loss. In addition, we validate the memory load during CNN operation using the proposed data structure and using SpherePHD [19].

GA-Conv. We compare the results of Models 1 and 3 in Table 3 when adding or removing GA-Conv. GA-Conv is used in the feature extraction module to obtain the feature vector to be used in SLIIF. If GA-Conv is not used, the kernel weight sharing proposed by SpherePHD [19], which gives 180 degrees rotation per kernel, is used. Table 3 shows that using GA-Conv improves the PSNR score by 0.14dB and 0.12dB on the ODI-SR and the SUN360 Panorama datasets, respectively, for ×8 SR.

SLIIF. SphereSR uses SLIIF to present SR results for the ERP projection type through the feature vectors presented on the sphere. Identical to an earlier method [31], we implement a pixel-shuffle algorithm capable of performing SR on an icosahedron without SLIIF for a performance comparison. When using the pixel-shuffle step, the last feature map was subdivided by the scale factor multiple, after which the

Table 2. Perspective and fisheye SR results on the ODI-SR and SUN 360 Panorama Dataset. **Bold** indicates the best results.

Projection Type	Perspective				Fisheye			
FOV	90				180			
Method	ODI-SR		SUN 360 Panorama		ODI-SR		SUN 360 Panorama	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	25.40	0.6858	23.49	0.6516	23.27	0.7117	22.75	0.7157
SRCNN [11](+Bicubic)	26.04	0.7005	23.98	0.6654	23.92	0.7246	23.47	0.7295
EDSR [21](+Bicubic)	26.53	0.7192	24.91	0.6916	24.21	0.7323	23.98	0.7452
D-DBPN [12](+Bicubic)	26.59	0.7139	24.63	0.6836	24.39	0.7308	24.02	0.7401
RCAN [44](+Bicubic)	26.70	0.7191	24.81	0.6901	24.40	0.7348	24.08	0.7452
360-SS [28](+Bicubic)	23.28	0.6528	21.95	0.6205	22.00	0.6957	21.61	0.6962
LAU-Net [10](+Bicubic)	26.39	0.7197	24.72	0.6943	24.33	0.7346	24.97	0.7727
SphereSR(w/o SLIIF)(+Bicubic)	26.66	0.7176	24.83	0.6930	24.32	0.7345	25.00	0.7477
SphereSR(Ours)	**26.76**	**0.7208**	**24.97**	**0.6962**	**24.46**	**0.7393**	**25.14**	**0.7780**

Table 3. Ablation studies on ERP SR on ODI-SR and SUN360 Panorama Dataset for both ×8 and ×16 SR.

Scale	Component			x8				x16			
Model	GA-Conv	SLIIF	Feature loss	ODI-SR		SUN 360 Panorama		ODI-SR		SUN 360 Panorama	
				WS-PSNR	WS-SSIM	WS-PSNR	WS-SSIM	WS-PSNR	WS-SSIM	WS-PSNR	WS-SSIM
1	x	✓	x	24.20	0.6688	23.98	0.6719	22.44	0.6341	21.92	0.6318
2	✓	x	x	24.31	0.6731	24.07	0.6749	22.47	0.6335	21.92	0.6294
3	✓	✓	x	24.34	0.6765	24.10	0.6816	22.47	0.6364	21.93	0.6336
4	✓	✓	✓	24.37	0.6777	24.17	0.6820	22.51	0.6370	21.95	0.6342

Table 4. Comparisons of activation memory between SpherePHD and the proposed data structure. The network architectures of SpherePhD and ours have the same number of convolution layers (16) and hidden feature dimension (32).

Level	4	5	6	7
SpherePHD(MB)	660	1896	6714	26032
New Data Structure(MB)	**374**	**724**	**2138**	**7450**

final ERP output was derived by means of bicubic interpolation. As shown in Models 2 and 3 in Table 3, using SLIIF for continuous image presentation achieves higher performance (24.34dB vs. 24.31dB) than the pixel-shuffle method of subdivision of the icosahedron for ×8 SR.

Feature loss. We propose a feature loss that measures the feature similarity of the crucial areas through feature masking using features generated from other projection types. To confirm the effectiveness of feature loss, we compare the SR performance when adding and removing this loss. Models 3 and 4 in Table 3 show the ablation results. The results of the performance comparison indicate a performance improvement for all metrics in the ×8 and ×16 SR cases.

Data Representation Efficiency. In Sec. 3.1, we point out that the CNN implementation of SpherePHD is not efficient for SR. We thus propose a new data structure to tackle this problem. To ascertain the efficiency of the new data structure, we implement a simple CNN model and then conduct an experiment to compare the activation memory. The CNN model is a simple structure in which the convolution layers are stacked; the number of convolution layer is set to 16 and the hidden feature dimension is set to 32. Table. 4 shows the experiments from input level 4 to input level 7. As indicated in the table, the new data structure in GA-Conv has a much lower activation memory level. In addition, it is found that as the input level is increased, the ratio of using new data structure memory to SpherePHD decreases. Based on this, the proposed data structure is shown to be more efficient in terms of memory compared to SpherePHD. Moreover, the efficiency increases as the input resolution is increased.

5. Conclusion

In this paper, we proposed a novel framework, SphereSR, which generates a continuous spherical image representation from an LR 360° image. SphereSR predicts the RGB values at the given spherical coordinates of an HR image corresponding to an arbitrary project type. First, we proposed geometry-aligned convolution to represent spherical data efficiently, after which we proposed SLIIF to extract RGB values from the spherical coordinates. As such, SphereSR flexibly reconstructed an HR image with an arbitrary projection type and SR scale factors. Experiments on various benchmark datasets demonstrated that our method significantly surpasses existing methods.

Limitation and Future Work. We focused on finding an efficient data structure and kernel weight sharing method to extract meaningful features with the ODI input based on GA-Conv (Sec. 3.1). Future studies will therefore need to improve the network architecture using the properties of ODIs compared to perspective images, and then we can achieve better SR results via SLIIF.

Acknowledgement. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2022R1A2B5B03002636).

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[2] Zafer Arican and Pascal Frossard. L1 regularized super-resolution from unregistered omnidirectional images. pages 829–832, 2009. 2

[3] Zafer Arican and Pascal Frossard. Joint registration and super-resolution with omnidirectional images. *IEEE Transactions on Image Processing*, 20(11):3151–3162, 2011. 1

[4] Zafer Arican and Pascal Frossard. Joint registration and super-resolution with omnidirectional images. volume 20, pages 3151–3162, 2011. 2

[5] Luigi Bagnato, Yannick Boursier, Pascal Frossard, and Pierre Vandergheynst. Plenoptic based super-resolution for omnidirectional image sequences. In *2010 IEEE International Conference on Image Processing*, pages 2829–2832. IEEE, 2010. 1

[6] Luigi Bagnato, Yannick Boursier, Pascal Frossard, and Pierre Vandergheynst. Plenoptic based super-resolution for omnidirectional image sequences. pages 2829–2832, 2010. 2

[7] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021. 2, 4, 5

[8] T.S. Cohen, M. Geiger, J. Khler, and M. Welling. Spherical cnns. In *International Conference on Learning Representations*, pages 4302–4311, 2018. 3

[9] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018. 3

[10] Xin Deng, Hao Wang, Mai Xu, Yichen Guo, Yuhang Song, and Li Yang. Lau-net: Latitude adaptive upscaling network for omnidirectional image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9189–9198, 2021. 1, 2, 6, 7, 8

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 38, pages 295–307, 2016. 6, 8

[12] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6, 8

[13] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[15] Hiroshi Kawasaki, Katsushi Ikeuchi, and Masao Sakauchi. Super-resolution omnidirectional camera images using spatio-temporal analysis. volume 89, pages 47–59, 2006. 2

[16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, 2016. 6

[17] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5835–5843, 2017. 1, 6

[18] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 1

[19] Yeonkun Lee, Jaeseok Jeong, Jongseob Yun, Wonjune Cho, and Kuk-Jin Yoon. Spherephd: Applying cnns on 360° images with non-euclidean spherical polyhedron representation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3, 4, 7

[20] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. Multi-scale residual network for image super-resolution. In *ECCV*, 2018. 6

[21] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 2, 6, 8

[22] Hongying Liu, Zhubo Ruan, Chaowei Fang, Peng Zhao, Fanhua Shang, Yuan yuan Liu, and Lijun Wang. A single frame and multi-frame joint network for 360-degree panorama video super-resolution. In *ArXiv*. 2

[23] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465, 2019. 2

[24] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. 2, 5

[25] Hajime Nagahara, Yasushi Yagi, and Masahiko Yachida. Super-resolution from an omnidirectional image sequence. In *2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies*, volume 4, pages 2559–2564. IEEE, 2000. 1

[26] Hajime Nagahara, Yasusi Yagi, and Masahiko Yachida. Super-resolution from an omnidirectional image sequence. volume 4, pages 2559–2564 vol.4, 2000. 2

[27] Akito Nishiyama, Satoshi Ikehata, and Kiyoharu Aizawa. 360° single image super resolution via distortion-aware network and distorted perspective images. In *ICIP 2021*, 2021.

2

[28] Cagri Ozcinar, Aakanksha Rana, and Aljosa Smolic. Super-resolution of omnidirectional images using adversarial learning. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2019. 2, 6, 8

[29] Yajun Qiu, Ruxin Wang, Dapeng Tao, and Jun Cheng. Embedded block residual network: A recursive restoration model for single-image super-resolution. pages 4179–4188, 2019. 6

[30] Vida Fakour Sevom, Esin Guldogan, and J. Kämäräinen. 360 panorama super-resolution using deep convolutional networks. In *VISIGRAPP*, 2018. 2

[31] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 7

[32] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019. 2

[33] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[34] Sanghyun Son and Kyoung Mu Lee. SRWarp: Generalized image super-resolution under arbitrary transformation. In *CVPR*, 2021. 2

[35] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. pages 9434–9443, 2019. 3

[36] Yule Sun, Ang Lu, and Lu Yu. Weighted-to-spherically-uniform quality evaluation for omnidirectional video. In *IEEE Signal Processing Letters*, volume 24, pages 1408–1412, 2017. 2

[37] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4549–4557, 2017. 1, 6

[38] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. volume abs/2006.10739, 2020. 5

[39] Longguang Wang, Yingqian Wang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning a single network for scale-arbitrary super-resolution. In *IEEE/CVF International Conference on Computer Vision*, 2021. 2

[40] Lin Wang and Kuk-Jin Yoon. Semi-supervised student-teacher learning for single image super-resolution. *Pattern Recognition*, 121:108206, 2022. 1

[41] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 1

[42] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2695–2702. IEEE, 2012. 6

[43] Chao Zhang, Stephan Liwicki, William Smith, and Roberto Cipolla. Orientation-aware semantic segmentation on icosahedron spheres. *IEEE/CVF International Conference on Computer Vision*, pages 3532–3540, 2019. 3

[44] Yulun Zhang, Kunpeng Li, Kai Li, WangLichen, Bineng Zhong, and Yun Raymond Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 1, 6, 7, 8

[45] Yupeng Zhang, Hengzhi Zhang, Daojing Li, Li-Yan Liu, Hong Yi, Wei Wang, Hiroshi Suitoh, and Makoto Odamaki. Toward real-world panoramic image enhancement. pages 2675–2684, 2020. 2

[46] Yufeng Zhou, Mei Yu, Hualin Ma, Hua Shao, and Gangyi Jiang. Weighted-to-spherically-uniform ssim objective quality evaluation for panoramic video. In *2018 14th IEEE International Conference on Signal Processing (ICSP)*, pages 54–57, 2018. 2, 6