ABSTRACT
Guided depth super-resolution is a practical task where a low-resolution and noisy input depth map is restored to a high-resolution version, with the help of a high-resolution RGB guide image. Existing methods usually view this task as a generalized guided filtering problem that relies on designing explicit filters and objective functions, or a dense regression problem that directly predicts the target image via deep neural networks. These methods suffer from either model capability or interpretability. Inspired by the recent progress in implicit neural representation, we propose to formulate the guided super-resolution as a neural implicit image interpolation problem, where we take the form of a general image interpolation but use a novel Joint Implicit Image Function (JIIF) representation to learn both the interpolation weights and values. JIIF represents the target image domain with spatially distributed local latent codes extracted from the input image and the guide image, and uses a graph attention mechanism to learn the interpolation weights at the same time in one unified deep implicit function. We demonstrate the effectiveness of our JIIF representation on guided depth super-resolution task, significantly outperforming state-of-the-art methods on three public benchmarks. Code can be found at https://git.io/JC2sU.

CCS CONCEPTS
• Computing methodologies → 3D imaging; Image representations.

KEYWORDS
Guided Super-Resolution, Implicit Neural Representation

1 INTRODUCTION
Depth maps have been widely used as a basic element in various computer vision tasks, such as semantic segmentation [5, 13, 50–52] and 3D reconstruction [4, 6, 44]. With the geometric information in the depth maps, these tasks can be facilitated and better understood. Despite the improvement of the depth sensors in recent years, high quality depth maps are still challenging to acquire. The acquired depth maps are usually low-quality due to the limitation of the sensors. In the meantime, RGB cameras have evolved rapidly and can acquire high-quality RGB images with a comparatively low cost. Hence, RGB-guided depth super-resolution, where a high-resolution (HR) RGB image is used to guide the upsampling process of a low-resolution (LR) depth input image, has become an important research topic. As illustrated in Figure 1, the detailed structures in the RGB image can be used to avoid blurry edges and suppress noises when up-sampling depth maps, but this process is non-trivial due to the complicated nature of RGB images.

![Figure 1: RGB guided noisy depth map super-resolution. Our method predicts a high-resolution target depth map from a noisy and low-resolution input depth map with the guidance from a high-resolution RGB image. The low-resolution depth map is up-sampled with bicubic interpolation for better visualization.](https://git.io/JC2sU)
aggregation doesn’t model the guide process explicitly, which lacks interpretability and may not generalize well to other datasets.

Inspired by the recent progress of implicit neural representation in 3D object/scene representation [18, 35, 39] and image super-resolution [7], we revisit the guided super-resolution problem and view it from the perspective of implicit neural representation. The idea of implicit neural representation is to use a deep implicit function (DIF) to map continuous coordinates to signals in a certain domain. To share knowledge across different input observations, an encoder is often used to extract latent codes from the input to make the DIF conditional to the current observation. Thus, a scene/image can be represented by a set of local latent codes distributed in the coordinates of the input domain, which can be used in downstream tasks such as semantic segmentation [39] and super-resolution [7]. To make the output of DIF continuous, a weighted average of the predictions from several neighboring coordinates is usually calculated, which can be viewed as a neural implicit interpolation process. However, these weights are usually empirical (e.g., distance-based) in previous work since there is no other prior knowledge between the query coordinate and the neighboring coordinates. With the extra HR guide image in the guided super-resolution task, we can learn to extract this knowledge and learn the weights in a data-driven way. We hypothesize that the guide image can benefit the learning of both interpolation weights and values, and propose to learn the interpolation weights via a graph attention mechanism. Furthermore, we integrate the learning of weights and values into one unified DIF, which we call the joint implicit image function (JIIF) representation.

To summarize, the contributions of this paper are as follows:

- We propose a novel joint implicit image function representation for guided image super-resolution, where the target image is represented by local latent codes from both the input image and the guide image.
- We learn interpolation weights at the same time via a graph attention mechanism, and integrate the learning of both interpolation weights and values into one unified representation.
- Our method outperforms existing methods by large margins on guided depth super-resolution tasks, and achieves state-of-the-art results on guided noisy depth super-resolution tasks.

2 RELATED WORK

2.1 Guided Super-Resolution

2.1.1 Filtering based methods. Guided filtering aims to enhance a target image by applying a filter that is dependent on a guide image. Bilateral filter [47] is the starting work where the target image also serves as the guide image. Later work includes joint bilateral filter [23], guided filter [15] and weighted median filter [32]. Guided filtering can be used for a variety of tasks such as image denoising, colourisation and stereo matching. When extended to different sized target image and guide image, it can also be used for the guided super-resolution task. We further distinguish these methods into two categories: Local methods first upsample the low-resolution target image with a traditional interpolation method, then apply a local filter which is controlled by the guide image [23, 55] or both the target image and the guide image [2]. On the other hand, Global methods formulate the filtering as an implicit energy minimization problem, and optimize values of all the pixels in the target image. This category includes Markov Random Field [8] and its non-local means variant [34]. Variational inference with anisotropic total generalized variation prior [10] and auto-regressive models [54] are other types of global methods [19, 28]. Some recent methods also combine the idea of guided filtering into the global optimization framework, such as the fast bilateral solver [1] and the SD filter [14].

2.1.2 Learning based methods. Different from the previous unsupervised filtering based methods, learning based methods provide a data-driven and supervised way to solve the guided super-resolution problem by training neural networks. Self-supervised super-resolution where the HR target image is first down-sampled to serve as the LR input has been explored by lots of methods [3, 7, 9, 24, 25, 56–58]. Guided super-resolution further introduces a HR guide image to direct the up-sampling process of the LR input. Early work like Depth Multi-Scale Guided Network (DMSG) [17], Dynamic Guidance (DG) [12] and Deep Joint Filtering (DJF) [26, 27] starts to use CNNs to extract features and directly regress the target image. Pixel-Adaptive Convolution (PAC) [45] learns a spatially variant kernel to fuse the guide image features into the LR input. Deformable Kernel Network (DKN) [20] draws ideas from both the explicit filtering based methods and learning based methods. It uses a CNN to learn a set of sparsely chosen neighbors and the interpolation weights adaptively, then apply an explicit image filter to calculate the final prediction. These methods either lack model interpretability for directly regressing the target, or rely on a simple image filter that cannot take full advantage of the guide image. Instead, our method starts from the general form of image interpolation and equips it with the effective implicit neural representation, leading to both better performance and interpretability.

2.2 Implicit Neural Representation

Implicit neural representation uses a deep implicit function (DIF) to map coordinates to signals in a specific domain. A DIF is a continuous and differentiable function, usually parameterized by an MLP. To make the DIF conditional to different input observations, a latent code extracted from the input is usually appended to the coordinate. Recent research has demonstrated the potential of implicit neural representations for 3D single objects [33, 35, 53], 3D scene surface [11, 18, 42, 43], 2D images [7, 42] and 1D audios [42]. Compared to traditional representations, DIF is shown to be more efficient, expressive, and is fully continuous. It is able to capture better structural details with fewer parameters when trained properly. For example, DeepSDF [35] takes a 3D coordinate and a categorical latent code as the input, and outputs the signed distance (SDF) at this coordinate to decide whether it is inside the target shape. Local Implicit Grid (LIG) [18] learns the common geometric features from local overlapping patches and reconstructs complicated scenes by associating them. Local Implicit Image Function (LIIF) [7] extracts a set of latent codes distributed in the LR domain to interpolate the HR target image. SIREN [42] proposed a general implicit neural representation for various domains to fit complicated signals by using periodic activation functions. Different from previous methods that focus on learning from single-modal data, we stress our work on
learning from multi-modal data, e.g., HR RGB guide and LR depth input. We focus on extracting prior knowledge from the guide image to help the representation learning of the target image. On this aspect, our method is more close to PixTransform [31]. This method explores guided super-resolution from a different perspective more similar to depth estimation, by training a DIF that maps each pixel in the guide image to the target image, and supervises only by the LR input. Different from our training pipeline, it doesn’t rely on HR target image for supervision, and can be categorized as an unsupervised depth super-resolution task. Besides, PixTransform is unconditional to observations, which means it needs to train a different set of parameters for every new image.

2.3 Graph Attention Mechanism
Graph Convolution Networks (GCNs) focus on problems residing in graph-structured data, by defining graph convolutions on the vertices and edges of a graph [22, 46, 48, 49]. An undirected graph $G = \langle V, E \rangle$ is composed of $N$ vertices $v_i \in V$, edges $\{v_i, v_j\} \in E$, and an adjacency matrix $A \in \mathbb{R}^{N \times N}$ which is binary or weighted. For tasks where $A$ is binary, some work explores to learn the edge weights from vertex features to facilitate the feature propagation. Graph Attention Networks [48] leverages masked self-attention layers to regress a continuous weight between each two connected vertices. EdgeConv [49] learns different adjacency matrices in different layers to extract vertex features in a dynamic way. For the task of guided super-resolution, we propose to first divide the image into pixels with implicit neural representation, then treat each pixel query as a graph problem. Thus, the interpolation weights can be interpreted as graph edge weights and learned through the graph attention mechanism.

3 METHOD
In this section, we introduce our JIIF representation for guided super-resolution task. We first review the recent neural implicit interpolation methods in Section 3.1, then detail our JIIF representation for guided super-resolution in Section 3.2. Finally, we describe our design of the JIIF-Net to learn the representation from data in Section 3.3.

3.1 Neural Implicit Image Interpolation
We start from a general formulation of the image interpolation problem for image upsampling, then view it from the perspective of implicit neural representation to introduce the neural implicit image interpolation method. For each LR input image $M$, we want to calculate the corresponding HR target image $I$:

$$I(x_q) = \sum_{i \in N_q} w_{q,i} v_{q,i},$$

where $x_q$ is the coordinate of the query pixel $q$ in the HR domain, $N_q$ is the set of neighbor pixels for $q$ in the LR domain, $w_{q,i}$ is the interpolation weight between $i$ and $q$, and $v_{q,i}$ is the interpolation value for $i$. The interpolation weights are usually normalized so that $\sum_{i \in N_q} w_{q,i} = 1$. We use a continuous image representation by scaling the image coordinates into $(-1, 1)$ to make it possible to share the coordinate in both the HR and LR domain. Due to the nature of 2D images, $N_q$ is usually chosen as the four nearest corner pixels of $q$ in the LR domain (as illustrated in Figure 2). Different interpolation methods have different ways to calculate the interpolation weights and values. The most commonly used bilinear interpolation is implemented with:

$$\begin{align*}
w_{q,i} &= \frac{S_i}{S}, \\
v_{q,i} &= M(x_i),
\end{align*}$$

where $S_i$ is the partial area diagonally opposite to the corner pixel $i$, $S = \sum_{i \in N_q} S_i$ is the total area serving as a normalization factor, and $M(x_i)$ is the pixel value of the LR input image at $x_i$.

In implicit neural representation, instead of directly using the pixel value $M(x_i)$, a DIF is applied to calculate the interpolation value $v_{q,i}$. For example, LIIF [7], LIG [18] and SCSSNet [38, 39] all take the following form:

$$v_{q,i} = f_\theta(z_i, x_q - x_i),$$

where $f_\theta(\cdot)$ is a MLP with parameters $\theta$ that takes a local latent code $z_i$ and a relative coordinate $x_q - x_i$ as input. In this setting, the target image is represented by a set of local latent codes distributed at the pixel coordinates of the LR domain, each storing information about its local area [7]. The latent codes map is the output feature map from an encoder network, and it is of the same resolution as the LR input image:

$$z_i = E_\phi(M(x_i)),
$$

where $E_\phi(\cdot)$ is the encoder network with parameters $\phi$. The DIF thus models a local area centered at the coordinate of the given latent code. By querying the conditioned DIF $f_\theta(z_i, \cdot)$ with a relative query coordinate $x_q - x_i$, it returns the estimated target value at the query coordinate $x_q$, e.g., the depth value in depth super-resolution. The weighted average of these estimated values from the four corners is further calculated to avoid discontinuous prediction (which is called local ensemble in [7]).

3.2 Joint Implicit Image Function
We focus on the problem of guided super-resolution, where an extra HR guide image $G$ is provided with the LR input image $M$. Previous methods either directly regress the target image values by fusing CNN features which lacks interpretability [26, 27], or treat it as an explicit filtering problem which cannot fully take advantage of the information in the guide image [20]. We hypothesize that the information in the guide image can benefit the learning of both interpolation weights and values, and these two terms can be learned jointly to boost the performance. Inspired by the recent neural implicit image interpolation methods, we propose to use DIFs to model both the interpolation weights and values, which we call the Joint Implicit Image Function representation.

Similar to the LIIF representation, the target image is represented by a set of local latent codes, but our latent codes are extracted from both the LR input image and the HR guide image, allowing the detailed information from the guide image to help the upsampling process. In particular, we apply two encoder networks to extract two sets of latent codes from the guide image and the input image respectively:

$$\begin{align*}
z_i &= E_\Theta(M(x_i)), \\
g_j &= E_\Phi(G(x_j)),
\end{align*}$$

with a relative distance in the LR domain.
where $E_\phi$ is another encoder network with parameters $\psi$. Then, the interpolation values can be naturally calculated by querying the DIF with these two latent codes and a relative coordinate:

$$v_{q,i} = f_\theta(z_i, g_i, x_q - x_i), \quad (6)$$

where $i$ is one of the neighbors of $q$ in the LR domain ($i \in N_q$). Please note that due to the different resolutions of HR and LR images, we could not obtain the HR latent code at position $x_i$ directly. In such cases, we conduct the bicubic interpolation operation to approximate the HR latent code at position $x_i$.

Furthermore, we propose to learn the interpolation weights at the same time. As illustrated in the neural implicit interpolation part in Figure 2, we view the interpolation at each query pixel as a graph problem. The four corner pixels and the query pixel are the vertices, and each corner pixel is connected to the query pixel with an edge. Previous methods usually use an empirical value for the edge weights [7], or directly regress the weights from the CNN features [20]. Inspired by recent research in Graph Convolutional Networks [48, 49], we propose to use a graph attention mechanism to calculate the edge weights. Specifically, we extract the guide latent code of each corner pixel $g_i$ and the query pixel $g_q$ in the HR domain, and apply a MLP to learn the weight in an asymmetric way:

$$a_{q,i} = f_\eta(g_i, g_q - g_i), \quad (7)$$

where $a_{q,i}$ is the learned edge weight, and $f_\eta$ is a MLP with parameters $\eta$.

We notice the representation of the interpolation weights (Equation 7) and values (Equation 6) are of a similar form. Hence, we propose to integrate these two separate functions into a unified one:

$$a_{q,i}, v_{q,i} = f_\theta(z_i, g_i, g_q - g_i, x_q - x_i), \quad (8)$$

By integrating the learning of interpolation weights and values, we reduce the parameters needed to model the representation, and allow interaction between these two processes, which is demonstrated to be more effective in our experiments. Finally, the edge weights are normalized by applying the softmax function to calculate the final interpolation weights:

$$w_{q,i} = \frac{\exp(a_{q,i})}{\sum_{i \in N_q} \exp(a_{q,i})}, \quad (9)$$

### 3.3 Network Architecture and Training

After defining the JIIF representation, we design a neural network to learn the representation from large datasets. As shown in Figure 2, the network contains two image encoders and one JIIF decoder. The input image and the guide image are fed into two encoders respectively, generating two feature maps as the latent codes for the JIIF representation. During training, we sample a set of pixels from the HR image with their coordinates, and query the JIIF decoder with these coordinates to predict the pixel values. A standard L1 loss is applied to optimize the network for predicting accurate results:

$$L = \frac{1}{N} \sum_{i} |f(x_i) - I(x_i)|, \quad (10)$$

where $N$ is the total number of sampled pixels, $x_i$ is the coordinate of any sampled pixel, $I(x_i)$ is the ground truth pixel value, and $f(x_i)$ is the predicted pixel value. In testing, we query all pixels’ coordinates in the target domain to recover the full up-sampled image.
Table 1: Quantitative comparison with the state of the art on depth map upsampling in terms of average RMSE.

| Datasets | Middlebury | Lu | NYU v2 |
|----------|------------|----|--------|
|          | ×4         | ×8 | ×16    | ×4 | ×8 | ×16 | ×4 | ×8 | ×16 |
| Bicubic  | 2.28       | 3.98 | 6.37  | 2.42 | 4.54 | 7.38  | 4.28 | 7.14 | 11.58 |
| DMSG [17]| 1.88       | 3.45 | 6.28  | 2.30 | 4.17 | 7.22  | 3.02 | 5.38 | 9.17  |
| DG [12]  | 1.97       | 4.16 | 5.27  | 2.06 | 4.19 | 6.90  | 3.68 | 5.78 | 10.08 |
| DJF [26] | 1.68       | 3.24 | 5.62  | 1.65 | 3.96 | 6.75  | 2.38 | 4.94 | 9.18  |
| DJFR [27]| 1.32       | 2.62 | 4.58  | 1.20 | 2.33 | 5.19  | 1.89 | 3.33 | 6.78  |
| PAC [45] | 1.32       | 2.12 | 4.24  | 0.96 | 2.16 | 5.11  | 1.62 | 3.26 | 6.51  |
| DKN [20] | 1.23       | 1.82 | 3.31  | 0.85 | 1.73 | 4.16  | 1.37 | 2.76 | 5.27  |
| Ours     | 1.09       | 1.82 | 3.31  | 0.85 | 1.73 | 4.16  | 1.37 | 2.76 | 5.27  |

Table 2: Quantitative comparison with the state of the art on noisy depth map upsampling in terms of average RMSE.

| Datasets | Art | Books | Moebius |
|----------|-----|-------|---------|
|          | ×4  | ×8    | ×16     | ×4  | ×8 | ×16 |
| Bicubic  | 6.07 | 7.27 | 9.59 | 5.15 | 5.45 | 5.97 | 5.51 | 5.68 | 6.11 |
| DMSG [17]| 6.19 | 7.26 | 9.53 | 5.38 | 5.18 | 5.20 | 5.48 | 5.06 | 5.36 |
| PDN [37] | 3.11 | 4.48 | 7.35 | 1.56 | 2.24 | 3.46 | 1.68 | 2.48 | 3.62 |
| DG [12]  | 2.96 | 4.41 | 7.06 | 1.64 | 2.35 | 3.50 | 1.74 | 2.57 | 3.79 |
| DJF [27] | 4.25 | 6.43 | 9.05 | 2.20 | 3.35 | 4.94 | 2.39 | 3.51 | 4.56 |
| PAC [45] | 5.34 | 7.69 | 10.66 | 2.11 | 3.12 | 4.60 | 2.21 | 3.38 | 4.72 |
| DKN [20] | 3.01 | 4.14 | 7.01 | 1.44 | 2.10 | 3.09 | 1.63 | 2.39 | 3.55 |
| Ours     | 2.79 | 3.87 | 7.14 | 1.30 | 1.75 | 2.47 | 1.40 | 2.03 | 3.18 |

4 EXPERIMENTS

In this section, we apply our method to guided depth map super-resolution and guided noisy depth map super-resolution tasks to demonstrate the effectiveness of our method.

4.1 Guided depth map super-resolution

4.1.1 Datasets and Evaluation Metrics. We adopt three widely-used benchmarks for the guided depth super-resolution task:

- **NYU v2 dataset [41]**: This dataset provides 1449 RGBD pairs of indoor scenes captured by Microsoft Kinect [59] using structural light. We use the first 1000 pairs as the training set and the rest 449 pairs as the evaluation set following previous work [20, 27].
- **Middlebury dataset [16, 40]**: we use a subset of 30 RGBD pairs from the 2001-2006 datasets provided by Lu et al. [30] for testing.
- **Lu dataset [30]**: This dataset consists of 6 RGBD pairs acquired by ASUS Xtion Pro camera. We use it for testing.

Following Kim et al. [20], we train our model on the NYU v2 dataset, and test it on all the three datasets. We do not fine-tune the model on Middlebury dataset or Lu dataset in order to test the generalization ability of the model. The LR input images are generated at different ratios (×4, ×8, ×16) through bicubic down-sampling from the HR target images. We use average RMSE as the evaluation metric for the depth map super-resolution task.

4.1.2 Implementation Details. We choose EDSR-baseline [29] as the backbone for the two encoders, and discard the up-sampling modules to generate feature maps of the same size as the input image. The output dimension of the encoder is set to 128, and thus the input dimension of the DIF is 386. A 5-layer MLP is used to model the DIF with decreasing hidden dimensions (1024, 512, 256, 128).

We train the model for 200 epochs with the batch size of 1. The HR image is randomly cropped into (256, 256) patches and we sample 30720 pixels per patch for each training step. The depth maps are scaled to [0, 1] before fed into the neural networks. For the Middlebury dataset and the Lu dataset, we interpret the provided disparity map as the depth map according to [20]. We use the Adam optimizer [21] to train our models. The initial learning rate is set to 0.0001 and is divided by 0.2 for every 60 epochs. We apply data augmentation by flipping the image pairs vertically or horizontally in training. When testing, all of the pixels in the HR domain are queried to recover the target image. The model is implemented and trained using the PyTorch framework [36].

4.1.3 Quantitative Comparisons. We compare the proposed method with state-of-the-art methods, including recent learning based methods such as DJFR [27] and DKN [20]. Table 1 shows the detailed results on the three datasets. We report the average RMSE on the test set. For the NYU v2 dataset, the average RMSE is measured in centimeters. For the Middlebury dataset and the Lu dataset, the average RMSE is measured in the original scale of the provided disparity. Our method outperforms the existing methods by large margins.
in all datasets and settings. With the proposed JIFF representation, our method predicts more accurate target image in all up-sampling ratios, and generalizes well into data from other sources (e.g., disparity maps acquired by different devices). This improvement is from the strong capability of implicit neural representation and the joint leaning of interpolation values and weights.

4.1.4 Qualitative Comparisons. We provide visual comparison of the \( \times 8 \) super-resolution results on the NYU v2 dataset in Figure 3. Also, generalization results on the Middlebury dataset and the Lu dataset are shown in Figure 5. Our method produces more accurate and sharper edges in areas of complicated structures, where other methods fail to model the geometry and generate blurred results. Besides, our method can restore reasonable structure even when the RGB guidance is ambiguous, e.g., too dark to provide any useful information. This confirms the advantages of the proposed JIFF representation.
4.2 Guided noisy depth map super-resolution

To show the robustness of our method on noisy data, we further perform experiments to restore noisy low-resolution input depth maps to noise-free high-resolution target depth maps.

4.2.1 Datasets. The Noisy Middlebury dataset [34] is used as the evaluation dataset for this task. It contains three standard RGBD pairs from the Middlebury 2005 dataset, i.e., Art, Books and Moebius. We simulate noisy LR input following previous work [20, 37] by adding a conditional Gaussian noise to the LR input:

\[ n(x) \sim N(0, \sigma x) \]  

(11)

where \( x \) is proportional to the depth value (e.g., if the input is disparity \( d \), we use \( x = \frac{d}{2} \)), and \( \sigma \) is the magnitude of the noise. For training, we use the NYU v2 dataset with the same type of noise added to the input images, and do not fine-tune the model on the Noisy Middlebury dataset. In particular, the \( \sigma \) is set to 651 for the Noisy Middlebury dataset following [37], and 0.04 for the NYU v2 dataset to simulate similar magnitude of noise. The other experimental settings are the same as in Section 4.1.2.

4.2.2 Quantitative Comparisons. From Table 2, we can see our method outperforms other methods on most settings of the guided noisy depth map super-resolution task. Although trained on depth maps from the NYU v2 dataset, our method generalizes well to the disparity maps from the Noisy Middlebury dataset. This agrees with the previous experiments and demonstrates the noise suppression ability of our method.

4.2.3 Qualitative Comparisons. The visual comparison of noisy depth map super-resolution is shown in Figure 4. Even the input image is corrupted severely by the noise and at \( \times 16 \) up-sampling ratio, our method successfully restores reasonable structural details in the predicted HR depth map. Also, our method can better suppress noises and restore sharper edges compared to other methods.
We show two examples of the query pixel crossing an edge in the HR target image. The query pixel is in red, and the four corner pixels' color indicates the learned interpolation weights. Higher weights are in bluer color, while lower weights are in greener color.

Figure 6: Visualization of the learned interpolation weights.

4.3 Ablation Study

We conduct ablation studies on different proposed modules in our method, and verify the effect of these modules for the ×8 guided depth super-resolution task on the NYU v2 dataset.

4.3.1 Learning interpolation weights. Firstly, we do ablation studies on different strategies to learn the interpolation weights. From Table 4, our graph attention based weights learning achieves the best performance. ‘Bilinear’ means the bilinear interpolation weights are used. ‘Direct Regression’ means we use directly a convolution layer to regress the weights from the guide image features, and ‘Graph Attention’ means we apply a graph attention layer to regress the weights from the guide image features. Compared to the baseline bilinear interpolation weights, our method reduces the average RMSE by 0.92. Direct regression of the weights used in DKN [20] also fails to learn meaningful interpolation weights. This verifies the effectiveness of the graph attention mechanism for leaning edge weights. We also provide the visualization of the learned interpolation weights in Figure 6. The graph attention module can adapt to different locations dynamically, predicting higher weights if the two vertices share common guidance features. For example, when the query pixel crosses an edge, the interpolation weights will switch to the correct side too. This avoids assigning large weights to wrong values from the opposite side, which is one of the main causes of blur in traditional image interpolation.

4.3.2 Joint learning of interpolation weights and values. We argue that the interpolation weights and values are correlated and can be learned together to boost the performance. Our JIFF representation is designed to exploit this correlation by predicting them in one DIF. We conduct experiments to prove this hypothesis in Table 3. ‘Baseline’ means that we break the JIFF into two DIFs that learn interpolation weights and values separately as described in Equation 6 and 7. ‘Joint Repr.’ means we use one unified MLP to learn interpolation weights and values as described in Equation 8. Note that we use the same architecture for the DIFs, which means the ‘Joint Repr.’ setting also reduces the last four MLP layers’ parameters by half. The experimental results validate our hypothesis that the joint representation can further enhance the final performance.

4.3.3 Residual Learning. Previous work [20, 27] has shown that residual learning, i.e. first up-sample the input with bicubic interpolation and then correct it by predicting the a residual image, can speed up convergence and improve the final performance. We also adopt this idea and perform experiments to validate the effect of residual learning in Table 3. ‘Residual’ means we adopt a residual learning framework. With both residual learning and joint representation applied, our method achieves the best performance.

5 CONCLUSION

In this paper, we propose a Joint Implicit Image Function (JIFF) representation for the guided super-resolution task. In JIFF representation, we take the form of a general image interpolation and multi-modal sensor fusion algorithms and applications-M2SFA2 2008 Workshop on Multicamera Noise-Aware Filter for Real-Time Depth Upsampling. In Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008. Springer, 617–632.

REFERENCES

[1] Jonathan T Barron and Ben Poole. 2016. The fast bilateral solver. In European Conference on Computer Vision. Springer, 617–632.
[2] Derek Chan, Hylke Buismans, Christian Theobalt, and Sebastian Thrun. 2008. A noise-aware filter for real-time depth upsampling. In Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008.
[3] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. 2004. Super-resolution through neighbor embedding. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, Vol. 1. IEEE, 1–1.
[4] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 2020. 3D sketch-aware semantic scene completion via semi-supervised structure prior.
[53] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 2019. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. arXiv preprint arXiv:1905.10711 (2019).

[54] Jingyu Yang, Xichen Ye, Kun Li, Chunping Hou, and Yao Wang. 2014. Color-guided depth recovery from RGB-D data using an adaptive autoregressive model. IEEE transactions on image processing 23, 8 (2014), 3443–3458.

[55] Qingxiong Yang, Ruigang Yang, James Davis, and David Nistér. 2007. Spatial-depth super resolution for range images. In 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 1–8.

[56] Xichen Ye, Baoli Sun, Zhuhui Wang, Jingyu Yang, Rui Xu, Haojie Li, and Baopu Li. 2020. Depth Super-Resolution via Deep Controllable Slicing Network. In Proceedings of the 28th ACM International Conference on Multimedia. 1809–1818.

[57] Kai Zhang, Luc Van Gool, and Radu Timofte. 2020. Deep unfolding network for image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 3217–3226.

[58] Yuhun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2018. Residual dense network for image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2472–2481.

[59] Zhengyou Zhang. 2012. Microsoft kinect sensor and its effect. IEEE multimedia 19, 2 (2012), 4–10.