Learning Graph Regularisation for Guided Super-Resolution

Riccardo de Lutio[1,*] Alexander Becker[1,*] Stefano D'Aronco[1]
Stefania Russo[1] Jan D. Wegner[1,2] Konrad Schindler[1]
[1]EcoVision Lab, Photogrammetry and Remote Sensing, ETH Zurich
[2]Institute for Computational Science, University of Zurich
firstname.lastname@geod.baug.ethz.ch

Abstract

We introduce a novel formulation for guided super-resolution. Its core is a differentiable optimisation layer that operates on a learned affinity graph. The learned graph potentials make it possible to leverage rich contextual information from the guide image, while the explicit graph optimisation within the architecture guarantees rigorous fidelity of the high-resolution target to the low-resolution source. With the decision to employ the source as a constraint rather than only as an input to the prediction, our method differs from state-of-the-art deep architectures for guided super-resolution, which produce targets that, when downsampled, will only approximately reproduce the source. This is not only theoretically appealing, but also produces crisper, more natural-looking images. A key property of our method is that, although the graph connectivity is restricted to the pixel lattice, the associated edge potentials are learned with a deep feature extractor and can encode rich context information over large receptive fields. By taking advantage of the sparse graph connectivity, it becomes possible to propagate gradients through the optimisation layer and learn the edge potentials from data. We extensively evaluate our method on several datasets, and consistently outperform recent baselines in terms of quantitative reconstruction errors, while also delivering visually sharper outputs. Moreover, we demonstrate that our method generalises particularly well to new datasets not seen during training.

1. Introduction

Guided super-resolution takes as input two images of different resolution, a low-resolution *source* and a high-resolution *guide* from a different domain. It returns a high-resolution version of the source as output, termed the *target*. This task is relevant in many practical applications such as medical [62] and satellite imaging [26], where performing a diagnosis or analysis from low-quality images can be extremely difficult. Another very popular example in computer vision is upsampling depth maps, where the low-resolution depth is the source, a conventional grayscale or RGB image is the guide, and the target is a high-resolution depth map. Consumer-grade depth sensors provide low-resolution depth maps, but a high-resolution RGB camera is usually mounted on the same device and can acquire a high-resolution image of the same scene. Guided super-resolution methods can be divided into two main categories, conventional and deep learning-based methods. The former typically cast the task into an optimisation problem [7, 8, 10, 12]. The goal is to create a high-resolution target image that, when downsampled, matches the source, while at the same time complying with an appropriate regularisation term that favours a desired image characterstic such as (piecewise) smoothness. Deep learning methods [15, 17–19, 52] instead rely on a dataset of source/guide/target triplets to learn a feed-forward mapping from the source and guide to the target. To that end the model must learn the statistical correlations that allow it to transfer high-frequency details from the guide to the tar-

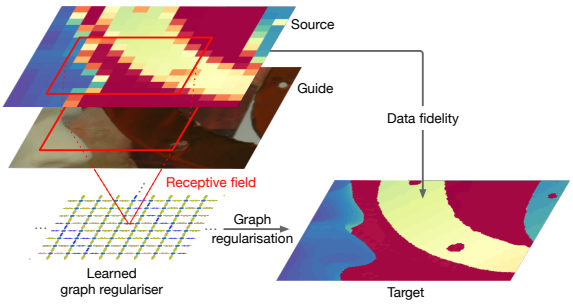


Figure 1. Our method takes as input a low-resolution *source* image and a high-resolution *guide* image of another modality to build a graph using high-level image features. The graph is then used in a differentiable optimisation layer as a regularisation to reconstruct the *target*.

*Equal contribution.

get, while, at the same time, ensuring that the predicted target stays close to the source. A considerable advantage of the conventional approach is that, by individually solving a properly formulated optimisation for each image, the prediction is usually guaranteed to match the source. On the other hand, designing an adequate regularisation term based on low-level image features is a complicated task. Deep learning methods exhibit rather complementary strengths: as long as one has access to enough training data and that data is representative of the images encountered at test time, these methods tend to perform very well, due to the unmatched ability of deep networks to mine complex, highly informative features from images. On the other hand, with limited training data, or when there is a domain shift between the training and the test set [54], feed-forward methods can no longer guarantee that downsampling the predicted target will produce the source, thus contradicting the fundamental relation behind super-resolution.

In this work, we show how to combine the two schools, and learn the graph of an optimisation-based super-resolution scheme, – see Figure 1. In particular, we learn a mapping from the two inputs (source and guide) to the edge potentials (also called edge weights) of an affinity graph between pixels of the target. The learned graph serves as the regulariser for an optimisation-based reconstruction of the high-resolution target, which is particularly suited for signals with a piecewise smooth structure. This entire mapping is trained end-to-end: the mapping function, which is parametrised as a convolutional network, is learned from training data, by back-propagating the gradients of the loss through the optimisation layer. CRF-RNN [9] also proposed to perform an online optimisation and include a graph in their network for semantic segmentation. However, they construct a dense graph and use an RNN to approximate the inference of the posterior. In contrast, we show that a sparse, local graph is sufficient while performing exact maximum a posteriori inference. We test our method on three different guided depth super-resolution datasets and show that it compares favourably against conventional and deep learning-based methods, across a range of upsampling factors from $\times 4$ to $\times 16$. We further show that our proposed method is much more robust to distribution shifts and can effectively generalise across datasets.

In summary, the contributions of this paper are the following: *(i)* We introduce a novel formulation of guided super-resolution, where a deep feature extractor is trained to derive the edge potentials for a graph-based energy minimisation from the input (source and guide) images; *(ii)* we develop a differentiable optimiser for the graph regularisation, taking advantage of the sparse graph connectivity to efficiently process large input patches up to 256^2 pixels[1]; *(iii)* in this way, our scheme therefore combines the power of learned, deep feature extractors with large receptive fields and the rigor of graph-based optimisation in an end-to-end trainable framework. As a result, it produces crisp, natural-looking images that correctly adhere to the underlying image formation model.

[1]Code is available at https://github.com/prs-eth/graph-super-resolution

2. Related Work

At a conceptual level, guided super-resolution can be seen as a form of guided filtering [13], where the source image is first naively upsampled to match the target resolution, and then enhanced with some transformation that is guided by the local structure of both the (upsampled) source and the guide.

2.1. Optimisation Methods

Local optimisation methods are variants of the filtering procedure described above. Here, the source is first upsampled, then a local filter controlled by the values of the guide [24, 60] is applied to it. Extensions of these methods include the use of geodesic distances to define the filter [30], or constructing it by combining the contrast in both the guide and source images [3].

Global optimisation methods construct a global energy function over all pixels and minimise it to obtain the target. The energy generally consists of two parts: a data fidelity term that ensures that the target stays close to the source, and a prior term that regularises the otherwise ill-posed problem of super-resolution. Data fidelity is typically defined as a distance term between the source and the downsampled target. The regulariser, in the guided setting, is not an isotropic smoothing but it is modulated by the guide. Depending on the parametrisation, the global energy minimisation can be viewed as Markov Random Field (MRF) inference [7], as a form of non-local means [35], or as a variational inference with an anisotropic version of total generalised variation (TGV) [10]. Some works [58, 59] have also proposed to replace the TGV prior with an auto-regressive model. The fast bilateral solver [2] solves a sparse linear system [1] to obtain bilateral-smooth outputs with sharp discontinuities. The SD filter [12] formulates guided image filtering as a non-convex optimisation problem that exploits static and dynamic guidance. The Pixtransform method [5] estimates a mapping from guide to target individually for each pixel and spatially smoothens the mapping function, rather than the target output. In a similar spirit, [34] predicts the target as a linear function of the guide, with coefficients that vary spatially, modulated by the guide and the source. The Guided Deep Decoder (GDD) [55] adapts the deep image prior [56] to guided super-resolution of hyper-spectral images. A random noise map is decoded into a target that has maximal data fidelity to the source, guided by feature maps obtained by a joint encoder-decoder branch for the

guide. Cross-Modality Super-Resolution (CMSR) [48] also fits a neural architecture to the individual source/guide pair, allowing to optimise for individual alignment errors. [36] proposes to (over-)segment the images and attribute planar disparities to each super-pixel, while smoothness of the disparities across super-pixels is encouraged by connecting them into an MRF.

In [42], the authors also propose to build a graph to encourage smoothness of the target in regions where the guide is also smooth. Contrary to our work, their graph is based on raw colour differences (similar to [7, 36]) whereas our graph encodes affinity between deep, latent features, derived not only from the guide but also from the source, and trained in an end-to-end fashion to optimally support the super-resolution task.

2.2. Learning Methods

The other large family of guided super-resolution methods are learning-based. Following a general trend towards supervised machine learning, the hope is that one can outperform conventional models by learning from data how to best fuse the source and the guide to recover the target. Perhaps the first learning-based methods for guided super-resolution were those learning dictionaries of source, guide and target patches. At test time, the source and guide were then (soft-)matched to the dictionary to retrieve suitable target patches and assemble the target image [25, 29].

More recently, deep learning methods have become predominant for guided super-resolution. These approaches work by parametrising the non-linear mapping from the two inputs — guide and source — to the target as a convolutional neural network, and learning its weights directly. The deep joint image filter [27, 28] feeds the upsampled source and the guide directly into a standard encoder-decoder architecture. The deep primal-dual network [40] follows a similar strategy, but outputs a residual correction to the naively upsampled source. Additionally, the output is refined with non-local total variation, unrolled into a sequence of network layers. The Multi-Scale Guided network (MSG-Net) [17] implements a new strategy, to *en*code only the guide, extract rich hierarchical features at different levels of the encoder, and append them to the corresponding levels of a network that *de*codes the source into the target through a final reconstruction layer. This integrated multi-scale guidance from the guide to the upsampled source allows to resolve ambiguity in depth map upsampling. This design has inspired several other works: PMBANet [61] adds multi-branch aggregation blocks; the Fast Depth Super-Resolution network (FDSR) [15] adds a high-frequency layer to extract fine details from the guide, and strives for a computationally efficient, yet effective design. DepthSR-Net [11] integrates the idea in a residual U-Net architecture [41]. First, the source is naively upsampled to the desired resolution, then the residuals between this naive interpolation and the corresponding target are learned using the hierarchical features as input pyramid in the encoder structure. In [57], an explicit coarse-to-fine cascade of networks is used to iteratively refine the output and progressively add high-frequency details. In [52] two networks are trained collaboratively, one for monocular depth estimation from the guide and one to super-resolve the source. Furthermore, there is an auxiliary structure prediction task to mitigate differences between depth and intensity discontinuities. Also in a very recent work, [43] explores learning depth super-resolution from unpaired data, using a learnable degradation model, and surface normal estimates as additional features to obtain more accurate depth maps.

Several authors have experimented with modifications of the basic Convolutional Neural Network (CNN) layers to enable modulation based on the guide. The Pixel-Adaptative Convolutional (PAC) network [51] proposes a novel type of learned filters where the convolution is conditioned on other features. For guided super-resolution, these conditioning features are extracted from the guide. Channel attention is used in [50] to improve super-resolution of channels with abundant high-frequency content. The Deformable Kernel Network (DKN) [18] applies sparse, spatially-variant kernels to predict a set of neighbours and associated weights for each target pixel, such that their weighted mean yields the pixel's value.

3. Method

3.1. Notation and Problem Statement

Throughout, we denote matrices and higher-order tensors with uppercase bold letters $\boldsymbol{A}$, and their flattened, 1D vector version with corresponding lowercase letters $\boldsymbol{a}$. In our guided super-resolution setting we are given a guide $\boldsymbol{G}$ with spatial dimensions $H \times W$ and C channels, as well as a low-resolution source $\boldsymbol{S}$ of dimension $h \times w$. For simplicity, we will assume that the source has a single channel, as the extension to multiple channels is straight-forward. The ratio between the spatial dimensions of the guide and the source is the upsampling factor $k = H/h = W/w$. The goal is to upsample $\boldsymbol{S}$ to a target $\boldsymbol{Y}$ with the same spatial resolution of $\boldsymbol{G}$. We denote with $\boldsymbol{D}$ the **downsampling operator** that maps $\boldsymbol{y}$ to $\boldsymbol{s}$. In our case the downsampling is a weighted average over a $k \times k$ window of the image $\boldsymbol{Y}$ (a *point spread function*). Note that some authors instead assume that $\boldsymbol{S}$ is not downsampled, but rather represents a sparsely sampled version of the target $\boldsymbol{Y}$, which need not be deconvolved. Due to the finite area of pixels on the sensor, respectively the beam divergence in laser-based scanners, this sparse subsampling without low-pass filtering is not a very realistic model for most practical sensing systems.

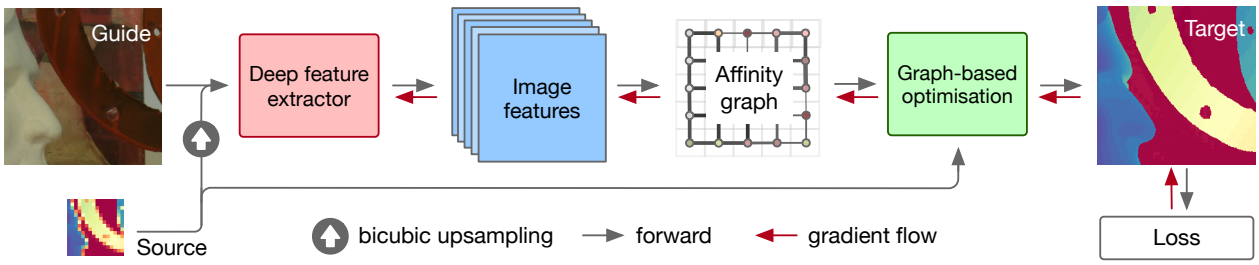


Figure 2. The architecture of our approach. A neural network backbone is employed to extract deep feature maps from the guide and source images. A graph over pixels is constructed based on pairwise affinities derived from these feature maps. Finally, a quadratic optimisation problem is solved for a target image that is in agreement with both the low-resolution source and the structure of the graph. Crucially, the graph optimisation layer is differentiable and our method is thus end-to-end trainable.

3.2. Graph Regularisation

A natural way to formalise the guided super-resolution problem mathematically is as an energy minimisation:

$$\underset{\boldsymbol{y}}{\text{argmin}} \quad f\big(\boldsymbol{D}\boldsymbol{y}, \boldsymbol{s}\big) + \lambda \cdot r(\boldsymbol{y}), \tag{1}$$

where f is the data fidelity term that measures how well the downsampled target matches the source, and r is a prior, respectively regulariser for the reconstructed target, and λ a parameter that weights the effect of the regularisation.

The **data fidelity term** serves to ensure similarity between $\boldsymbol{D}\boldsymbol{y}$ and the source $\boldsymbol{s}$, typically in the form of an l_1 or (squared) l_2 norm. In this work we use the latter, $f(\boldsymbol{D}\boldsymbol{y}, \boldsymbol{s}) = \|\boldsymbol{D}\boldsymbol{y} - \boldsymbol{s}\|_2^2$.

An effective **regulariser** that has often been used successfully for images [7, 42] is to encourage smoothness of the reconstructed signal w.r.t. some graph defined over the image pixels. The affinity matrix of that graph is denoted by $\boldsymbol{A}$ and has size $HW \times HW$. It describes which pixels are connected, i.e., have direct, first-order influence on each other. The generic element A_{ij} represents the weight of the edge connecting pixel i to pixel j, for all pairs of pixels that are not directly connected, $A_{ij} = 0$. The degree matrix $\boldsymbol{U}$ is a diagonal matrix with entries constructed by summing the weights of all edges that meet at a node, $U_{ii} = \sum_j A_{ij}$. Finally, the graph Laplacian $\boldsymbol{L}$ is defined as $\boldsymbol{L} = \boldsymbol{U} - \boldsymbol{A}$. For a signal defined on the graph nodes – in our case the image – the quantity $\boldsymbol{y}^T\boldsymbol{L}\boldsymbol{y}$ is a measure of how smooth that signal is on the graph. Encouraging smoothness is an effective regulariser, provided that the graph matches the intrinsic structure of the signal. The objective then becomes:

$$\underset{\boldsymbol{y}}{\text{argmin}} \quad \|\boldsymbol{D}\boldsymbol{y} - \boldsymbol{s}\|_2^2 + \lambda \boldsymbol{y}^T\boldsymbol{L}\boldsymbol{y}\ . \tag{2}$$

What remains is to construct the right graph, i.e., to determine the "natural" connectivity between the pixels of the target $\boldsymbol{Y}$. This is not trivial, but in guided super-resolution we can leverage the guide $\boldsymbol{g}$, which shares the same, high resolution with $\boldsymbol{Y}$. The graph thus becomes a function $\boldsymbol{L}(\boldsymbol{g})$ of the guide. Note that this does not imply constructing the graph from the guide's raw brightness (resp., contrast) values. Rather, one may as well derive it from more abstract per-pixel features. As we will show, a particularly useful procedure is to learn those features from the data, such that the graph is optimally adapted to the specific super-resolution task at hand.

In order to tailor the graph structure of the regulariser to the problem, we feed both the source and the guide through a CNN, to obtain a deep feature representation $\boldsymbol{F} = f_{\boldsymbol{\theta}}(\boldsymbol{G}, \boldsymbol{S})$ of size $H \times W \times M$, with M the channel depth of the representation and $\boldsymbol{\theta}$ the trainable parameters of the network. For efficiency, we restrict the graph to have fixed topology, where each pixel is connected (at most) to its 4-neighbours in the 2D pixel lattice. Longer-range connectivities are in principle possible, but greatly increase the computational effort, with rapidly diminishing returns. Indeed, our setup attaches deep features to the graph nodes, which encode a large receptive field and capture semantic and long-range information in the guide image beyond the 4-neighbour topology. The weights of the graph edges are defined as standard negative exponential affinities between the learned features:

$$A_{ij} = e^{-\frac{\|F_i - F_j\|_2^2}{M\mu}}, \tag{3}$$

where μ is a learnable scaling parameter.

3.3. Optimisation Layer

Let $\boldsymbol{y}^\star$ denote the minimiser of Eq. (2), and $\boldsymbol{y}_{\text{gt}}$ the ground truth target values of some training set. We can then assemble triplets $(\boldsymbol{g}, \boldsymbol{s}, \boldsymbol{y}_{\text{gt}})$ and optimise the graph construction to minimise the loss between $\boldsymbol{y}^\star$ and $\boldsymbol{y}_{\text{gt}}$:

$$\boldsymbol{\theta}* = \underset{\boldsymbol{\theta}}{\text{argmin}} \quad \mathbb{E}_{p(\boldsymbol{g},\boldsymbol{s},\boldsymbol{y}_{\text{gt}})}\big[l\big(\boldsymbol{y}^\star(\boldsymbol{\theta}), \boldsymbol{y}_{\text{gt}}\big)\big]\ , \tag{4}$$

where l is an appropriate loss function, for instance an l_1 loss or a Mean Squared Error (MSE) loss.

This means that, in order to train the feature extractor, we must compute the gradient of the loss in Eq. (4) w.r.t. the graph. Towards that end, we first notice that Eq. (2) is a quadratic problem, and equivalent to solving the linear equation system:

$$\big(\lambda \boldsymbol{L}(\boldsymbol{\theta}) + \boldsymbol{D}^T \boldsymbol{D}\big)\boldsymbol{y}^\star = \boldsymbol{D}^T \boldsymbol{s}, \tag{5}$$

here we have made it explicit that the graph Laplacian $\boldsymbol{L}$ is the only term that depends on the network parameters $\boldsymbol{\theta}$. For error backpropagation we must map the gradient $\frac{\partial l}{\partial \boldsymbol{y}^\star}$ w.r.t. the reconstructed image to the entries of $\boldsymbol{L}$. Using the implicit function theorem [2] we obtain:

$$\frac{\partial l}{\partial \boldsymbol{L}} = -\lambda \frac{\partial l}{\partial \boldsymbol{D}^T \boldsymbol{s}} \boldsymbol{y}^{\star T}, \; (\lambda \boldsymbol{L}(\boldsymbol{\theta}) + \boldsymbol{D}^T \boldsymbol{D}) \frac{\partial l}{\partial \boldsymbol{D}^T \boldsymbol{s}} = \frac{\partial l}{\partial \boldsymbol{y}^\star} \tag{6}$$

In order to backpropagate the loss, we must solve a second linear equation system, which then yields the derivatives for individual entries of the graph Laplacian. Note that the derivative w.r.t. the Laplacian is a dense matrix, which is impractical (e.g., for an image of 256^2 pixels this matrix has ≈ 4 billion elements). Fortunately, we can exploit the fact that the graph topology is fixed and compute the gradient only w.r.t. non-zero entries of $\boldsymbol{L}$ (i.e., index pairs of the 4-neighbourhood). Once the gradients for the graph weights have been computed, they are propagated through the deep feature extractor.

Finally, we summarise our proposed model, see Figure 2. The feature extractor $f_{\boldsymbol{\theta}}(\boldsymbol{G}, \boldsymbol{S})$ computes the deep features from the guide and source images, and these features inform the weights of a 4-neighbour graph. The graph, together with the source $\boldsymbol{S}$, forms the input to the optimisation problem of Eq. (2) that estimates the target. During training, a loss computed between the prediction and the ground truth steers the feature extraction such that the graph weights optimally regularise the prediction of the high-resolution target. Note that at test time we must solve a quadratic problem in order to predict the target image. To do so, very efficient algorithms are available, although it is of course not as fast as a conventional forward pass.

4. Experimental Results

In this section we describe the evaluation of our proposed method on the task of RGB-guided depth map super-resolution. We conduct our experiments on three widely used RGB-D datasets. For each dataset, we compare our approach to several guided super-resolution baselines. All algorithms are evaluated at 3 upsampling factors – $\times 4$, $\times 8$ and $\times 16$ – on the following datasets:

Middlebury [16, 44–47] We use all 50 RGB-D images available from the Middlebury 2005-2014 datasets. We split the data randomly into 40 images for training, 5 for validation and 5 for testing. A challenging aspect of this dataset is that it contains missing values in the depth ground truth. For generating the source, we therefore only take into account valid pixels during downsampling. Furthermore, we generate a pixel validity mask for both the target and source, so we can ignore the invalid pixels during training and testing.

NYUv2 [31] consists of 1449 RGB-D images captured with a Microsoft Kinect. We randomly split these into 849 images for training, 300 for validation and 300 for testing.

DIML [4, 20–22] is a large-scale dataset comprised of 2M RGB-D frames in total. For our evaluation, we use the high-resolution indoor sample subset, which was acquired using a Microsoft Kinect. From this data we construct a split of 1440 images for training, 169 images for validation and 503 images for testing.

We compare our model to the *Guided Filter* (GF) [13], the *Static/Dynamic filter* (SD) [12], the *Pixtransform* [5], the *MSG-Net* [17], the *Deformable Kernel Network* (DKN) and its fast version (FDKN) [18], the PMBANet [61], and finally to the *Fast Depth Super-Resolution* (FDSR) [15]. We were not able to compare to the recent work by [52] since no code has been released at the time of writing. For all other methods, we use the respective publicly available code. We implemented our method using PyTorch [37]. The graph-based optimisation layer is realised using the sparse matrix support of the CuPy library [33], allowing for a GPU-accelerated implementation of the forward and backward pass. In order to solve the linear systems of equations needed in the optimisation, we implement the conjugate gradient method [32]. We use a U-Net [41] network with a ResNet-50 [14] encoder pretrained on ImageNet [6] as feature extractor for the graph weights prediction. We also use a simple gradient clipping as it improves the stability of the training procedure. As baseline we further compare to a version of the proposed method where the RGB features of the guide and the bicubic upsampling of the source are the only pixel features used to construct the graph, i.e., no deep feature extractor is used.

We train all learned methods using the Adam [23] optimiser. Varying with the specific dataset, we fix the same batch size, initial learning rate and scheduling strategy for all methods. For fairness of comparison, we trained all learned methods with both our configuration for these hyperparameters and their original ones (when indicated) and report the best results. The detailed hyperparameter settings for each dataset and method as well as additional experimental results can be found in the supplementary material. For all learned methods we further make use of data augmentations during training, consisting of random cropping, random horizontal flipping and random rotation, where the rotation angle is sampled from $\mathcal{U}(-15°, 15°)$. All methods are evaluated on patches of 256^2 pixels, however, for some methods training on such large patches was infeasible due to memory constraints, in which case we used patches of size 128^2 or even 64^2 (for PMBANet with a factor $\times 4$).

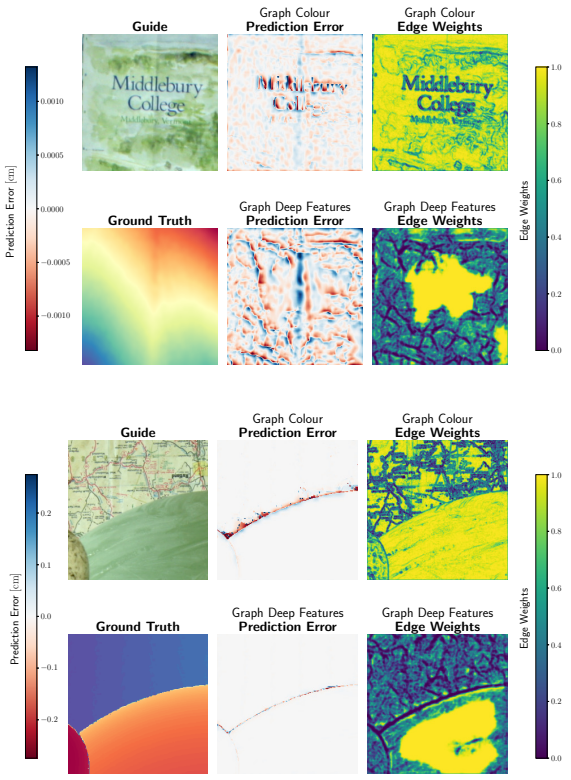


Figure 3. Importance of learned edge potentials. We visualise the total affinity of each pixel to its four neighbours when derived from raw colour (top) or from deep features (bottom). Examples are from the Middlebury test set.

4.1. Learning Graph Weights

As previously mentioned, we believe that smoothing based on a graph that is defined on four *local* neighbours only is adequate for the problem at hand, as long as the features used to create the graph encode a sufficiently large context. This is of course the case when using deep features that are learned from an entire dataset using a CNN with a large receptive field. In Figure 3, we compare the graph obtained from plain colour information to the graph obtained from learned deep features for two selected examples from the Middlebury test set. The graph is visualised by displaying, for each pixel, the sum of the four edges that connect it to its neighbours. Areas where the connections of the nodes are strong appear in yellow, and in these areas the graph regularisation will enforce smoothness in the predicted target. On the other hand, in areas where the edge weights are small, the graph smoothness term is weak (it disappears as weights approach zero) and the target is allowed to reveal depth discontinuities. Figure 3 demonstrates that the proposed method is able to extract semantical information from the guide image and to transfer it to the edge potentials. In the top example, the model has learned that a high-contrast text is part of the surrounding object, thus it predicts high edge weights for the respective image area and effectively enforces smooth depth. This correctly results in the text not being carried over to the target, an effect that is observable in the prediction based on colour features. On the other hand, the bottom image shows that our model has learned to detect object boundaries and highlights them even if both background and foreground have very similar colours. In fact, the learned edge weights are much lower around the depth discontinuities compared to the colour-based weights, which implies that the learned graph is able to recreate a sharp edge in the target prediction. In contrast, the graph built from colour information cannot perform a proper cut, leading to bleeding artifacts.

4.2. Depth Super-Resolution Evaluation

In Table 1 and Figure 4 we quantitatively and qualitatively compare our method to all selected baselines for the Middlebury, NYUv2 and DIML datasets. We outperform all other methods w.r.t. both MSE and MAE metrics for upsampling factors of $\times 4$, $\times 8$ and $\times 16$. It is observable from the table that the tested methods perform rather differently among the three datasets. Conventional methods tend to perform generally worse than the learned ones. In particular, Pixtransform [5] shows a rather flat performance curve with mediocre performance on low upsampling factors but also no abrupt performance drop for higher upsampling factors. In terms of visual results, the method reveals many artifacts being carried over from the guide. The SD filter [12] instead has good performance on MAE but MSE performance degrades fast for larger upsampling factors. Visually it captures some edges very well, whereas it completely misses and smoothes out others, as seen in Figure 4. FDKN and DKN [18] attain worse performance than expected across the datasets, especially quantitatively. It appears that these methods are tuned to sparsely downsampled source images and not well suited for realistic (not impulse-shaped) point spread functions. Our method by contrast achieves good quantitative performance across all three datasets, while producing visually crisp images. It is particularly effective at larger upsampling factors, showing the advantages of a hybrid model that leverages a deep learning backbone alongside a conventional online optimisation layer. Finally, it is more robust to domain shifts, as we explain in the next paragraph.

Cross-dataset generalisation. A major advantage of our work is that the prediction, after downsampling, is constrained to match the source. This additional constraint affords the model better robustness against domain shifts between training and testing (at the cost of added computation at inference time). To quantify this behaviour, we perform

			GF [13]	SD filter [12]	Pixtransform [5]	MSG-Net [17]	DKN [18]	FDKN [18]	PMBANet [61]	FDSR [15]	Ours - Colour	Ours
Middlebury	×4	MSE	33.3	24.9	39.8	4.13	4.29	3.60	4.72	7.72	14.8	**3.04**
		MAE	1.27	0.46	0.79	0.22	0.18	0.16	0.25	0.35	0.42	**0.13**
	×8	MSE	40.5	82.5	32.7	10.5	11.2	10.4	9.48	23.2	68.3	**7.26**
		MAE	1.49	0.86	0.82	0.43	0.38	0.37	0.38	0.69	0.83	**0.24**
	×16	MSE	67.4	511	41.5	34.2	47.6	38.5	30.6	55.4	297	**24.7**
		MAE	2.21	1.73	1.24	1.06	1.42	1.18	0.89	1.51	1.69	**0.67**
NYUv2	×4	MSE	114	36.0	112	6.85	11.4	8.07	10.8	10.5	19.0	**6.45**
		MAE	3.91	1.31	3.61	0.81	1.03	0.85	0.93	0.94	1.11	**0.73**
	×8	MSE	142	105	122	24.1	29.8	29.9	31.5	35.4	68.4	**19.6**
		MAE	4.47	2.57	3.86	1.66	1.82	1.80	1.79	1.96	2.30	**1.42**
	×16	MSE	249	533	219	84.5	115	113	84.9	179	264	**67.5**
		MAE	6.34	5.07	5.40	3.35	4.01	3.95	3.26	4.68	4.56	**2.90**
DIML	×4	MSE	25.6	10.5	20.7	1.73	3.47	2.2	3.05	2.75	7.02	**1.68**
		MAE	1.45	0.40	1.15	0.22	0.33	0.23	0.31	0.29	0.35	**0.20**
	×8	MSE	34.1	44.9	23.0	4.13	5.47	5.95	5.87	8.40	15.2	**3.51**
		MAE	1.77	0.83	1.26	0.40	0.45	0.47	0.47	0.66	0.67	**0.31**
	×16	MSE	66.3	411	39.3	13.0	19.3	20.8	13.8	32.9	133	**9.45**
		MAE	2.74	1.91	1.78	0.93	1.20	1.24	0.87	1.66	1.72	**0.68**

Table 1. Performance comparison with the state-of-the-art algorithms on the Middlebury [16, 44–47], NYUv2 [49] and DIML [4, 20–22] datasets for different values of upsampling factors. The table shows the MSE (in cm^2) and MAE (in cm).

Testing Dataset		GF [13]	SD filter [12]	Pixtransform [5]	MSG-Net [17]	FDKN [18]	PMBANet [61]	FDSR [15]	Ours - Colour	Ours
DIML	MSE	34.1	44.9	23.0	5.76	6.74	7.35	7.73	20.5	**4.95**
	MAE	1.77	0.83	1.26	0.51	0.53	0.59	0.74	0.77	**0.40**
	MSE (low-resolution)	17.7	1.45	6.19	6.16	0.20	0.04	0.45	0.03	$\mathbf{2.4 \cdot 10^{-3}}$
Middlebury	MSE	40.5	82.5	32.7	11.0	10.0	9.62	18.4	23.9	**8.25**
	MAE	1.49	0.86	0.82	0.54	0.43	0.46	0.73	0.91	**0.35**
	MSE (low-resolution)	17.9	1.86	22.5	5.01	0.20	0.06	7.20	0.08	$\mathbf{1.1 \cdot 10^{-3}}$

Table 2. Performance comparison with the state-of-the-art algorithms on cross-dataset generalisation. All learned methods have been trained on the NYUv2 dataset [49]. The table shows the performance of the methods on the DIML [4, 20–22] and Middlebury [16, 44–47] datasets for a ×8 upsampling factor. The table shows the MSE (in cm^2), MAE (in cm) and low-resolution MSE (in cm^2).

a cross-dataset generalisation experiment. For all methods, we train on NYUv2 and test the resulting model on Middlebury and DIML. As can be seen in Table 2, we outperform all other methods by a significant margin. Furthermore, our prediction matches the source almost perfectly when downsampled, as measured by the low-resolution MSE.

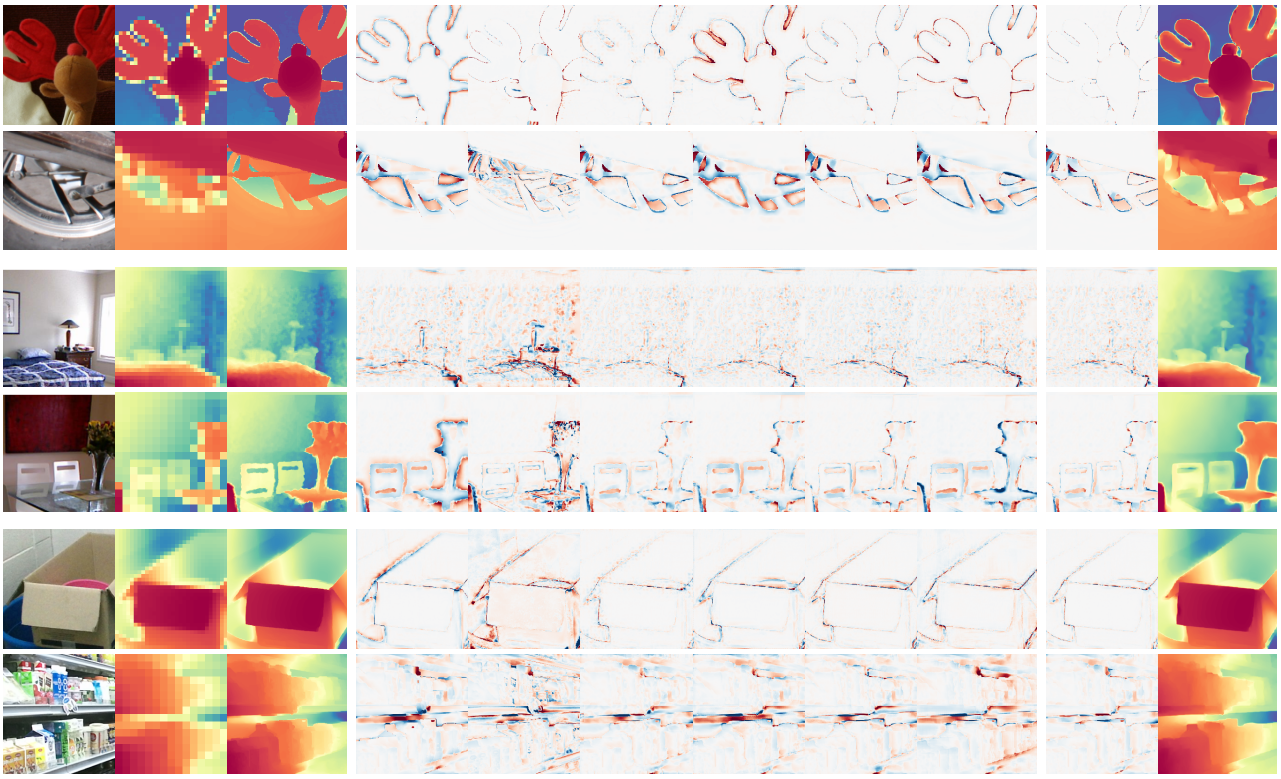
4.3. Feature Extractor Comparison

We go on to investigate the performance of our method with different feature extractors. Table 3 compares the errors on the NYUv2 ×8 upsampling task with different backbones used to extract the features for the graph regularisation layer, as described in Section 3.2. We have tested several well-known backbones, always initialising them with weights pretrained on ImageNet [6]. In addition to these generic backbones, we also evaluate feature maps extracted from the FDSR network that was specifically designed for guided depth super-resolution. Finally, to explore the boundaries of feature extraction, we employ a variant of the *Dense Prediction Transformer* [38, 39], which extracts multi-scale features for dense prediction tasks (like monodepth or semantic segmentation) from a Vision Transformer (ViT). We adapted the model to account for the (appropriately re-sampled) source image at each feature level and call it the *Guided Dense Prediction Transformer* (GDPT).

The results show that our method is fairly insensitive to the choice of architecture, across a range of capacities (respectively, parameter counts). This seems to indicate that the graph-based regularisation, although clearly benefiting from high-level features, bounds the expressive power of the feature extractor. We speculate that this is due to the fact that the graph cannot represent long-range patterns in the output image, but only enforces local smoothness where needed, thus limiting the amount of information that can be usefully transferred to the predicted target. However, we do not recommend very low capacity backbones: when using FDSR, the performance is better than the original FDSR model, but clearly lower than with higher capacity models.

Feature extractor	# Params	MSE (cm^2)	MAE (cm)
Colour	2	68.4	2.30
UEfficientNet-B2 [53]	10M	24.9	1.63
UResNet-18 [14]	14M	21.7	1.52
UResNet-50 [14]	32M	**19.6**	**1.42**
FDSR [15]	0.6M	30.4	1.75
GDPT [38, 39]	127M	22.3	1.54

Table 3. Performance comparison of various feature extractors for the ×8 upsampling task on NYUv2.



(a) Guide. (b) Source. (c) GT. (d) SD. (e) Pixtr. (f) MSG. (g) FDKN. (h) PMBA. (i) FDSR. (j) Ours. (k) Ours Pred.

Figure 4. Qualitative comparison of upsampled depth maps. From top to bottom each group of two rows shows the error of upsampled images, defined as the difference between the prediction and the ground truth, on the Middlebury [16, 44–47], NYUv2 [31] and DIML [4, 20–22] datasets, respectively; alternating between upsampling factors $\times 8$ and $\times 16$ for each dataset. From left to right, the first group of columns are (a) Guide, (b) Source and (c) Ground Truth; the second group includes selected methods from our quantitative evaluation, (d) SD filter [12], (e) Pixtransform [5], (f) MSG-Net [17], (g) FDKN [18], (h) PMBANet [61] and (i) FDSR [15]; the last two columns represent (j) the error for the prediction of our model and (k) the prediction itself.

Nevertheless, independent of the backbone used, our approach achieves the lowest MAE among all evaluated methods; except when using the raw *Colour* as features, i.e., low-level image contrast is not sufficient as a regulariser and using a learned feature extractor is essential.

5. Discussion

The graph regularisation layer effectively acts as a bottleneck on the amount of information that can be carried from the guide to the target – the regularisation is not able to create arbitrary patterns in the target image. This can be seen as a limitation, but also a desirable property, as it increases model robustness. One drawback of our method w.r.t. to most of the conventional deep forward architectures is the inference time, which is the price to pay for an online optimisation that guarantees rigorous fidelity to the source. A forward pass on a single 256^2 pixel patch, for upsampling factor $\times 8$, takes on average 111 ms on an NVIDIA GeForce RTX 2080 Ti. This number varies depending on the complexity of the image, and the upsampling factor.

6. Conclusion

We have presented a novel formulation for guided super-resolution based on a learnable graph regulariser. The method employs a deep feature extractor that takes a *guide* and a *source* as input, and infers an affinity graph over adjacent pixels in the *target* image. The learned graph serves as a regulariser in the upsampling of the source, implemented as a differentiable optimisation layer. This explicit optimisation within the architecture guarantees rigorous fidelity of the high-resolution target to the low-resolution source. Our proposed method combines desirable properties from both, conventional and deep learning based methods: the optimisation layer guarantees that the fidelity w.r.t. the source image is satisfied, even in case of domain shifts in the test set, while the deep feature extractor enables the learned affinity graph to encapsulate valuable information extracted from a large context. The experimental evaluation confirms that our graph regulariser is effective for signals that exhibit a piecewise smooth structure, such as depth maps.

References

[1] Jonathan T. Barron, Andrew Adams, YiChang Shih, and Carlos Hernández. Fast bilateral-space stereo for synthetic defocus. In *CVPR*, 2015. 2

[2] Jonathan T. Barron and Ben Poole. The fast bilateral solver. In *ECCV*, 2016. 2, 5

[3] Derek Chan, Hylke Buisman, Christian Theobalt, and Sebastian Thrun. A noise-aware filter for real-time depth upsampling. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2*, 2008. 2

[4] Cho, Jaehoon and Min, Dongbo and Kim, Youngjung and Sohn, Kwanghoon. Deep monocular depth estimation leveraging a large-scale outdoor stereo dataset. *Expert Systems with Applications*, 2021. 5, 7, 8

[5] Riccardo de Lutio, Stefano D'Aronco, Jan D. Wegner, and Konrad Schindler. Guided super-resolution as pixel-to-pixel transformation. In *ICCV*, 2019. 2, 5, 6, 7, 8

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. Ieee, 2009. 5, 7

[7] James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. In *NIPS*, 2006. 1, 2, 3, 4

[8] Weisheng Dong, Guangming Shi, Xin Li, Kefan Peng, Jinjian Wu, and Zhenhua Guo. Color-guided depth recovery via joint local structural and nonlocal low-rank regularization. *IEEE Transactions on Multimedia*, 2016. 1

[9] Zheng et al. Conditional random fields as recurrent neural networks. In *ICLR*, 2015. 2

[10] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *ICCV*, 2013. 1, 2

[11] Chunle Guo, Chongyi Li, Jichang Guo, Runmin Cong, Huazhu Fu, and Ping Han. Hierarchical features driven residual learning for depth map super-resolution. *TIP*, 2019. 3

[12] Bumsub Ham, Minsu Cho, and Jean Ponce. Robust guided image filtering using nonconvex potentials. *TPAMI*, 2018. 1, 2, 5, 6, 7, 8

[13] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *TPAMI*, 2013. 2, 5, 7

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 7

[15] Lingzhi He, Hongguang Zhu, Feng Li, Huihui Bai, Runmin Cong, Chunjie Zhang, Chunyu Lin, Meiqin Liu, and Yao Zhao. Towards fast and accurate real-world depth super-resolution: Benchmark dataset and baseline. In *CVPR*, 2021. 1, 3, 5, 7, 8

[16] Heiko Hirschmüller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, 2007. 5, 7, 8

[17] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *ECCV*, 2016. 1, 3, 5, 7, 8

[18] Beomjun Kim, Jean Ponce, and Bumsub Ham. Deformable kernel networks for joint image filtering. *IJCV*, 2021. 3, 5, 6, 7, 8

[19] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1

[20] Sunok Kim, Dongbo Min, Bumsub Ham, Seungryong Kim, and Kwanghoon Sohn. Deep stereo confidence prediction for depth estimation. In *ICIP*, 2017. 5, 7, 8

[21] Youngjung Kim, Bumsub Ham, Changjae Oh, and Kwanghoon Sohn. Structure selective depth superresolution for rgb-d cameras. *TIP*, 2016. 5, 7, 8

[22] Youngjung Kim, Hyungjoo Jung, Dongbo Min, and Kwanghoon Sohn. Deep monocular depth estimation via integration of global and local predictions. *TIP*, 2018. 5, 7, 8

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[24] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ToG*, 2007. 2

[25] HyeokHyen Kwon, Yu-Wing Tai, and Stephen Lin. Data-driven depth map refinement via multi-scale sparse representation. In *CVPR*, 2015. 3

[26] Charis Lanaras, José Bioucas-Dias, Silvano Galliani, Emmanuel Baltsavias, and Konrad Schindler. Super-resolution of Sentinel-2 images: Learning a globally applicable deep neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018. 1

[27] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *ECCV*, 2016. 3

[28] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Joint image filtering with deep convolutional networks. *TPAMI*, 2019. 3

[29] Yanjie Li, Tianfan Xue, Lifeng Sun, and Jianzhuang Liu. Joint example-based depth map super-resolution. In *ICME*, 2012. 3

[30] Ming-Yu Liu, Oncel Tuzel, and Yuichi Taguchi. Joint geodesic upsampling of depth images. In *CVPR*, 2013. 2

[31] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 5, 8

[32] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006. 5

[33] Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. 5

[34] Jinshan Pan, Jiangxin Dong, Jimmy S. Ren, Liang Lin, Jinhui Tang, and Ming-Hsuan Yang. Spatially variant linear representation models for joint filtering. In *CVPR*, 2019. 2

[35] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S. Brown, and Inseo Kweon. High quality depth map upsampling for 3D-TOF cameras. In *ICCV*, 2011. 2

[36] Min-Gyu Park and Kuk-Jin Yoon. As-planar-as-possible depth map estimation. *Computer Vision and Image Understanding*, 2019. 3

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming

Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, 2019. 5

[38] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 7

[39] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020. 7

[40] Gernot Riegler, David Ferstl, Matthias Rüther, and Horst Bischof. A deep primal-dual network for guided depth super-resolution. In *BMVC*, 2016. 3

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 3, 5

[42] Mattia Rossi, Mireille El Gheche, Andreas Kuhn, and Pascal Frossard. Joint graph-based depth refinement and normal estimation. In *CVPR*, 2020. 3, 4

[43] Aleksandr Safin, Maxim Kan, Nikita Drobyshev, Oleg Voynov, Alexey Artemov, Alexander Filippov, Denis Zorin, and Evgeny Burnaev. Unpaired depth super-resolution in the wild, 2021. 3

[44] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *GCPR*, 2014. 5, 7, 8

[45] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *CVPR*, 2007. 5, 7, 8

[46] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, 2001. 5, 7, 8

[47] Daniel Scharstein, Richard Szeliski, and Ramin Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002. 5, 7, 8

[48] Guy Shacht, Dov Danon, Sharon Fogel, and Daniel Cohen-Or. Single pair cross-modality super resolution. In *CVPR*, 2021. 3

[49] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *TPAMI*, 2017. 7

[50] Xibin Song, Yuchao Dai, Dingfu Zhou, Liu Liu, Wei Li, Hongdng Li, and Ruigang Yang. Channel attention based iterative residual learning for depth map super-resolution. In *CVPR*, 2020. 3

[51] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *CVPR*, 2019. 3

[52] Baoli Sun, Xinchen Ye, Baopu Li, Haojie Li, Zhihui Wang, and Rui Xu. Learning scene structure guidance via cross-task knowledge transfer for single depth super-resolution. In *CVPR*, 2021. 1, 3, 5

[53] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 2019. 7

[54] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 2

[55] Tatsumi Uezato, Danfeng Hong, Naoto Yokoya, and Wei He. Guided deep decoder: Unsupervised image pair fusion. In *ECCV*, 2020. 2

[56] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *CVPR*, 2018. 2

[57] Yang Wen, Bin Sheng, Ping Li, Weiyao Lin, and David Dagan Feng. Deep color guided coarse-to-fine convolutional network cascade for depth image super-resolution. *TIP*, 2019. 3

[58] Jingyu Yang, Xinchen Ye, Kun Li, and Chunping Hou. Depth recovery using an adaptive color-guided auto-regressive model. In *ECCV*, 2012. 2

[59] Jingyu Yang, Xinchen Ye, Kun Li, Chunping Hou, and Yao Wang. Color-guided depth recovery from RGB-D data using an adaptive autoregressive model. *TIP*, 2014. 2

[60] Qingxiong Yang, Ruigang Yang, James Davis, and David Nister. Spatial-depth super resolution for range images. In *CVPR*, 2007. 2

[61] Xinchen Ye, Baoli Sun, Zhihui Wang, Jingyu Yang, Rui Xu, Haojie Li, and Baopu Li. PMBANet: Progressive multi-branch aggregation network for scene depth super-resolution. *TIP*, 2020. 3, 5, 7, 8

[62] Yongqin Zhang, Feng Shi, Jian Cheng, Li Wang, Pew-Thian Yap, and Dinggang Shen. Longitudinally guided super-resolution of neonatal brain magnetic resonance images. *IEEE Transactions on Cybernetics*, 2018. 1