Guided Super-Resolution as Pixel-to-Pixel Transformation

Riccardo de Lutio, Stefano D'Aronco, Jan Dirk Wegner, and Konrad Schindler

EcoVision Lab, Photogrammetry and Remote Sensing, ETH Zürich

Abstract

Guided super-resolution is a unifying framework for several computer vision tasks where the inputs are a low-resolution source image *of some target quantity (e.g., perspective depth acquired with a time-of-flight camera) and a high-resolution* guide image *from a different domain (e.g., a grey-scale image from a conventional camera); and the target output is a high-resolution version of the source (in our example, a high-res depth map). The standard way of looking at this problem is to formulate it as a super-resolution task, i.e., the source image is upsampled to the target resolution, while transferring the missing high-frequency details from the guide. Here, we propose to turn that interpretation on its head and instead see it as a pixel-to-pixel mapping of the guide image to the domain of the source image. The pixel-wise mapping is parametrised as a multi-layer perceptron, whose weights are learned by minimising the discrepancies between the source image and the downsampled target image. Importantly, our formulation makes it possible to regularise only the mapping function, while avoiding regularisation of the outputs; thus producing crisp, natural-looking images. The proposed method is unsupervised, using only the specific source and guide images to fit the mapping. We evaluate our method on two different tasks, super-resolution of depth maps and of tree height maps. In both cases we clearly outperform recent baselines in quantitative comparisons, while delivering visually much sharper outputs.*

1. Introduction

A number of computer vision tasks can be seen as instances of *guided super-resolution*. For instance, many robots are equipped with a conventional camera as well as a time-of-flight camera (or a laser scanner). The latter acquires depth maps of low spatial resolution, respectively large pixel footprint in object space, and it is a natural question whether one can enhance its resolution by transferring details from the camera image – see Figure 1. Another example is environmental mapping, where maps of parameters like tree height or biomass are available at a mapping resolution that is significantly lower than the ground sampling distance of modern earth observation satellites.

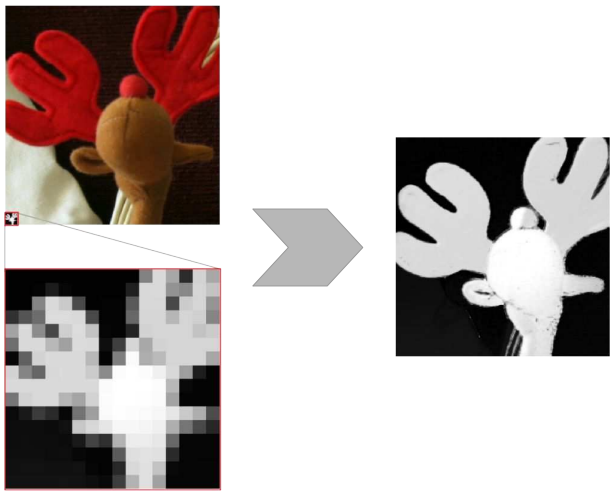

Figure 1: Guided super-resolution: given a low-resolution depth map and a high-resolution guide image, our method predicts a high-resolution depth map. The figure shows an example output of the proposed method, for an upsampling factor of $16\times$.

An alternative view of guided super-resolution is as a generalisation of guided filtering [8], widely used in image processing and analysis. A guided filter maps a source image to a target image of *the same size* by computing, at each pixel, a function that depends on the local neighbourhood in both the source and the guide image (which can be the source itself, as in the popular bilateral filter [27]). Guided super-resolution does the same, except that the source image has a lower spatial resolution and must additionally be upsampled in the process.

The standard way to model guided super-resolution is as an inverse problem: the source image is understood as the

result of downsampling the target image. The objective is to undo that operation, utilising the guide to constrain the solution, by transferring high-frequency details that were lost during downsampling, such as fine structures and sharp boundaries. Model inference can either be done by directly minimising an appropriate loss function, e.g., with variational methods [5]; or in two separate steps, e.g., upsampling with generic bilinear or bicubic interpolation followed by guided filtering [30].

Here, we propose an alternative interpretation of guided super-resolution, where the roles of the source and guide images are swapped: rather than finding a transformation from source to target and constraining the output to be consistent with the guide from a different image domain; we instead prefer to find a transformation from the guide to the target, i.e., a pixel-wise mapping from one image domain to another *without changing the resolution*, and constrain the output by demanding that its downsampled version matches the source image.

In our implementation, we parametrise the mapping as a multi-layer perceptron that takes as input all channels of the guide image *at a single pixel* (plus two additional "channels" corresponding to the pixel's x- and y-coordinates). In CNN terminology, the guide is augmented with two extra channels that encode pixel location, and then passed through a convolutional network whose layers all use only 1×1 kernels. Thus, the transformation from the guide to the target domain acts on pixels individually, without looking at their neighbours. Spatial context relations are encoded implicitly, and adaptively per image, via the structural bottleneck created by learning a single set of transformation parameters that must be valid for all pixels. We refer to this setup as "pixel-to-pixel transformation", as opposed to "image-to-image translation" with large receptive fields. Importantly, our method is *unsupervised*: while the mapping is structurally a form of CNN, we do not learn a static set of network weights from a training set and then apply those weights to every new test image. Rather, we fit an individual set of weights for each new image similarly to [28], using all its pixels as "training data" and the consistency with the low-resolution source as "supervision".

We argue that this view of guided super-resolution has two very practical advantages. *(i)* by starting already at the desired resolution, and using only 1×1 kernels, *different input locations do not mix*, which avoids blurring. *(ii)* by using the same mapping function for all pixels and placing a shrinkage prior on its parameters, one obtains a well-posed problem *without regularisation of the output image*. In this way, blurring is also avoided at the output stage. Together, these properties lead to outputs with superior sharpness.

The **contribution** of this paper is a novel formulation of guided super-resolution, as unsupervised learning of a pixel-to-pixel transformation from the guide to the target image, constrained by the low-resolution source. We present experiments on two tasks: super-resolution of depth maps, and super-resolution of tree height maps. They demonstrate that our formulation clearly outperforms competing super-resolution methods at high upsampling factors ($\times 8$ to $\times 32$).

2. Related Work

Guided filtering. A large body of work exists about guided filtering, without the additional challenge of super-resolution. The general principle is to enhance the source image by applying a filter whose output depends not only on a local neighbourhood of the source image, but also on weights derived from the same neighbourhood in the guide image. The starting point is the bilateral filter [27], where the source image itself serves as a guide. Classical examples that employ a guide from a different domain include the joint bilateral filter [23], the guided filter (GF) [8] and the weighted median filter [20]. Guided filtering has been used to a diverse range of image processing applications, ranging from low-level tasks like denoising [8] or colourisation [13] all the way to stereo matching [10].

Guided super-resolution. Extensions of guided filtering to the super-resolution problem have been explored for super-resolving depth, as well as for low-level operations like tone mapping and image colourisation. We distinguish between *local methods* based on the above local filtering principle, and *global methods* that formulate the upsampling task as a global energy minimisation.

The local methods are variants of the two-step procedure, i.e., first upsample the low-resolution source image with naive interpolation, then enhance it by applying a filter that is controlled by the high-resolution guide [13, 30]. Variants include using the geodesic distance in the high-res image instead of the raw contrast [19], and combining the contrast in both the source *and* the guide image to determine the filter strength [3].

Global methods formulate the super-resolution as an energy minimisation problem, whose solution returns the values of all pixels in the target image. The energy function consists of a data term that measures the compatibility between the downsampled target image and the low-resolution source image, and a smoothness term to regularise the ill-posed problem. In the guided scenario the latter term is not an isotropic preference for smooth solutions, but is modulated by the guide image. The global approach has been implemented as a Markov Random Field [4], and has been extended to additionally include the idea of non-local means to enhance image structures [22]. Another possible implementation is as variational inference [5], with an anisotropic version of the total generalised variation (TGV) prior, modulated by the guide image. It has also been proposed to

replace the TGV prior by an auto-regressive model [29], whose parameters are again a function of the bilateral filter response in the guide image.

Recently some methods have appeared that embed the idea of bilateral/guided filtering in a global optimisation framework, rather than apply a local filter. In particular, the fast bilateral solver (FBS) [2] offers an optimisation mechanism based on a sparse linear system [1] to obtain bilateral-smooth outputs with sharp discontinuities. Whereas the static/dynamic (SD) filter [7] converts the guided filtering problem into a non-convex optimisation that is solved by the majorisation-minimisation algorithm. Both have been successfully used for guided super-resolution, besides other image processing tasks.

Learned guided super-resolution. The methods described so far are unsupervised. There is also a line of work that learns from examples how to upsample the source image while transferring high-frequency details from the guide image to the target output. The advantage of such data-driven methods is that learning from real image data how to optimally fuse the source and guide images can potentially give better results than a hand-crafted heuristic. The disadvantages, as for all supervised learning, are on the one hand that one must have access to a sufficient amount of training data - in our case triplets of low-resolution source, high-resolution guide *and* high-resolution target images. And on the other hand that the super-resolution algorithm is, by design, overfitted to the training data and unlikely to generalise across even mild domain shifts. Early learning-based methods were based on the idea of dictionary learning, where an image patch is seen as a (sparse) linear combination of basis functions. For super-resolution, one jointly constructs a basis (dictionary) of corresponding source, guide and target patches, so that at test time the basis coefficients can be extracted from the source and guide images and used to reconstruct the target image [18, 14]. More recently, deep convolutional networks have been used to directly learn the mapping from the two inputs to the target output, keeping the dictionary implicit in the network. The deep primal-dual network [24] employs a standard encoder-decoder architecture that takes as input the naively upsampled source image and the guide, and outputs a differential correction to the source image. The result is then further refined with non-local total variation (TV) minimisation, unrolled into a series of neural network layers. The deep joint image filter [16, 17] encodes the source image and the guide, then decodes the resulting features into the target. Most prominently, the multi-scale guided network (MSG-Net) [11] extracts features at different resolutions from the guide image with an encoder branch, and uses them to guide the upsampling of the source image by concatenating them to layers of corresponding resolutions in a decoder branch

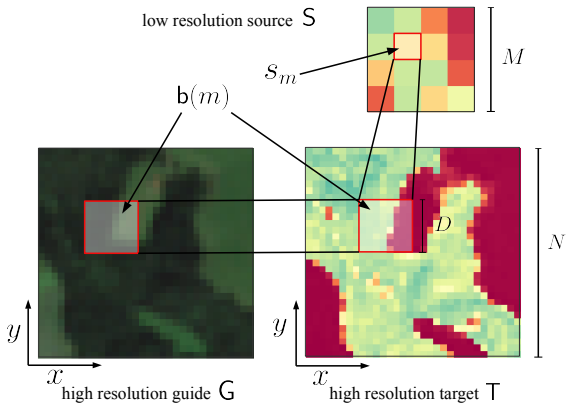

Figure 2: Illustration of the problem setting and notation.

that upsamples the source image. As before, the network is trained to output a differential correction of the naive upsampling. [21] targets the specific case of super-resolving semantic segmentations. The high-resolution "guide" image is passed through a standard semantic segmentation network to generate a "target" segmentation map, using a loss function that encourages the target to have the same label distribution as the low-resolution source map.

3. Method

Notation and Preliminaries

We denote the low-resolution source map as S, the high-resolution target map that we aim to recover as T, and the high-resolution guide image as G. For simplicity, and w.l.o.g., we assume square images, with source S of size $M \times M$, target T of size $N \times N$, and guide G of size $N \times N \times C$, where C is the number of channels. To simplify the notation, we use 1-dimensional pixel indices $m \in [1 \dots M^2]$ for the low resolution and $n \in [1 \dots N^2]$ for the high resolution, which can be expanded to 2-dimensional pixel coordinates $[x_m, y_m] = \mathbf{x}_m$ when needed. The relation between N and M is given by the upsampling factor $D \in \mathbb{N}^+$: $N = D \cdot M$. In other words, each source pixel S_m covers a block $\mathsf{b}(m)$ of $D \times D$ target pixels; see Fig. 2. The value of a low-resolution pixel is the average of the underlying high-resolution pixels (weighted averaging with a known point spread function is also possible, but omitted to simplify the notation):

$$s_m = \frac{1}{D^2} \sum_{n \in \mathsf{b}(m)} t_n = \langle t_n \rangle_{\mathsf{b}(m)}. \tag{1}$$

Our goal is to obtain an estimate $\hat{\mathsf{T}}$ of the high-resolution map, given S and G.

Proposed Solution

Instead of directly estimating the unknown target pixels t_n, we reformulate the problem as trying to find a function $f_{\boldsymbol{\theta}} : \mathbb{R}^C \to \mathbb{R}$ with parameters $\boldsymbol{\theta}$ that maps every guide pixel to a target pixel, $\hat{t}_n = f_{\boldsymbol{\theta}}(\mathbf{g}_n)$, such that the result is consistent with the source image according to Eq. (1). As a loss function to measure the consistency, we empirically use the ℓ^1-distance, leading to:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_m \left| s_m - \left\langle f_{\boldsymbol{\theta}}(\mathbf{g}_n) \right\rangle_{\mathsf{b}(m)} \right| . \tag{2}$$

That problem is obviously ill-posed, since many different target images T can be constructed that have loss 0. Moreover, even for given S, T and G a perfect solution can always be found by choosing a sufficiently complex function[1] $f_{\boldsymbol{\theta}}$.

Here, we parametrise the function $f_{\boldsymbol{\theta}}$ as a multi-layer perceptron (MLP), which takes as input the $(C \times 1)$-vector of intensities at a guide pixel $\mathbf{g}_n$ and outputs the corresponding target value $\hat{t}_n$. Restricting $f_{\boldsymbol{\theta}}$ to a function with reasonably low complexity ensures the problem is solvable. But since some input images are easier to upsample than others, always utilising the full capacity of $f_{\boldsymbol{\theta}}$ is prone to overfit in most cases. A core insight of our method is that, instead of regularising the output $\hat{\mathsf{T}}$, one can also combat overfitting by encouraging the choice of a simpler $f_{\boldsymbol{\theta}}$ through a suitable regulariser, in our case an ℓ^2-penalty on the network weights:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_m \left| s_m - \left\langle f_{\boldsymbol{\theta}}(\mathbf{g}_n) \right\rangle_{\mathsf{b}(m)} \right| + \lambda \left\| \boldsymbol{\theta} \right\|^2 , \tag{3}$$

with a hyper-parameter λ that controls the strength of the regularisation. There is still one issue with Eq. (3), namely that the model in this form is too restrictive: it imposes a one-to-one relationship between guide pixels $\mathbf{g}_n$ and output pixels $\hat{t}_n$. If, for instance, two pixels have the same colour in the guide, then they will be mapped to the same target depth, which is clearly not reasonable. Our trick to inject the necessary flexibility is to additionally allow the mapping to vary across the image plane:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_m \left| s_m - \left\langle f_{\boldsymbol{\theta}}(\mathbf{g}_n, \mathbf{x}_n) \right\rangle_{\mathsf{b}(m)} \right| + \lambda \left\| \boldsymbol{\theta} \right\|^2 . \tag{4}$$

Note that the regulariser $\|\boldsymbol{\theta}\|^2$ enforces low complexity of the network $f_{\boldsymbol{\theta}}$ not only w.r.t. the guide pixel values, but also w.r.t. the spatial location. In practice, we found it beneficial to train separate branches for the intensities $\mathbf{g}_n$ and the coordinates $\mathbf{x}_n$, which are then merged by adding their activations, as depicted in Fig. 3. With this architecture it is also possible to regularise each branch differently, by setting individual hyper-parameters $\lambda_{\mathbf{g}}, \lambda_{\mathbf{x}}, \lambda_{\text{head}}$. This is convenient

[1] Except for the pathological case of two or more identical blocks in G with different source values.

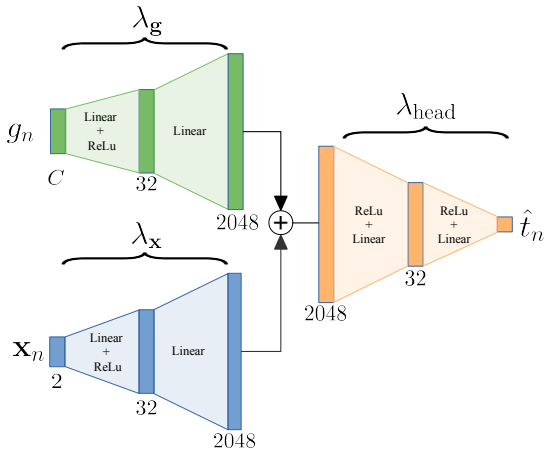

Figure 3: Architecture of the neural network used to model the mapping between the guide image and the high resolution map.

when one has corresponding a priori knowledge, e.g., when super-resolving semantic segmentations one may not want the mapping to strongly vary across the image plane.

Super-resolving a given input image S with the help of a guide image G now amounts to solving the optimisation problem (4). This can be done with simple stochastic gradient descent, but it may be beneficial to use more advanced optimisation schemes for this specific problem structure – finding the best numerical scheme is left for future work. Note that the optimisation over all pixels can be performed efficiently in any deep learning framework, by implementing $f_{\boldsymbol{\theta}}$ as a convolutional network $F^{1\times1}_{\boldsymbol{\theta}}$ with (1×1) kernels on all layers. The network takes as input the complete guide image, augmented with two additional channels for the pixel indices x_n, y_n, and outputs the complete target image. Once the network parameters have been fitted, the target is recovered by applying the function $f_{\boldsymbol{\theta}}$ to each pixel of the guide, which corresponds to a forward pass in the convolutional version:

$$\hat{t}_n = f_{\hat{\boldsymbol{\theta}}}(\mathbf{g}_n) \quad , \quad \hat{\mathsf{T}} = F^{1\times1}_{\hat{\boldsymbol{\theta}}}(\mathsf{G}) \, . \tag{5}$$

4. Experimental Results

In the following, we analyse the performance of the proposed pixel-to-pixel transformation method on two different datasets, and compare it to three state-of-the-art guided super-resolution methods, as well as two baselines.

Evaluation Settings

In all experiments, we set the target resolution to 256^2 pixels. We evaluate the algorithms at different upsampling factors, namely $\times 4$, $\times 8$, $\times 16$, and $\times 32$, correspond-

ing to source resolutions of 64^2, 32^2, 16^2, and 8^2 respectively. We test the proposed method on two different applications, super-resolving depth and super-resolving vegetation height. For depth we use the data from the 2005 version of the Middlebury benchmark [25, 9], from which we extract 120 high-resolution RGB images and depth maps. For vegetation height, the test set is composed of 40 maps extracted from the Swiss national forest inventory [6] (we use an updated version issued after the publication). As guide images we use multi-spectral images from ESA's Sentinel-2 satellite[2]. The satellite sensor records 13 channels at three different resolutions, we limit ourselves to the four channels with the highest resolution of 10 m per pixel, which are recorded in blue, green, red and near infra-red. In both cases the source images are generated by downsampling the ground truth targets with the appropriate scaling factor.

As baselines we adopt on the one hand naive *bicubic interpolation*, without guide image; and on the other hand the classical *guided filter* [8]. We further compare to two state-of-the-art methods for guided super-resolution, namely the *Fast Bilateral Solver* (FBS) [2] and the *static-dynamic filter* (SD) [7]. For the former we used the authors' original implementation[3], for the latter we ported the authors' implementation[4] to Python, and modified the data fidelity term of the optimisation to match the per-block consistency of Eq. (1). We select the parameters of FBS and SD according to the authors' guidelines and keep them constant for all experiments. We have verified that the quantitative results are consistent with the original publications.

The last method we compare to is a recent supervised learning algorithm, *MSG-Net* [11], also in the authors' original implementation[5]. We argue that guided super-resolution is most useful if it is not easily possible to record large amounts of data at the target resolution (e.g., large-scale vegetation height maps at 10 m resolution cannot be produced at a reasonable cost). Therefore, we follow a common procedure from the literature [26, 15]: under the assumption that the upsampling model is to some degree scale-invariant, one can *downsample* the available $M \times M$ data by the factor D to obtain synthetic training data for $\times D$ upsampling. The model thus trained for upsampling $(M/D)^2 \rightarrow M^2$ is then, at test time, applied to the actual super-resolution task $M^2 \rightarrow N^2$. We found that, due to the repeated downsampling, the data provided by [11] was not enough, so we additionally used the training data of [24]. Overall, we train MSG-Net on 5'000 images for depth and on 8'000 images for vegetation height. Still, the data was not sufficient to train for factors larger than $\times 8$. For practical applications of guided super-resolution, the need for large amounts of labelled training data is a real issue, and a serious limitation.

For our method, we train the mapping $f_{\boldsymbol{\theta}}$ on batches of 32 low-resolution pixels/blocks, using the ADAM optimiser [12] with learning rate 0.001. We centre the image values and normalise them to unit standard deviation, for both the source and the guide image. If the guide has more than one channel, we normalise them separately. Pixel coordinates $\mathbf{x}_n$ are rescaled to the interval $[-0.5, 0.5]$. We train for 32'000 iterations (independent of the upsampling factor), which takes about 120 seconds for a $\times 8$ upsampling on a standard GPU (Nvidia GTX 1080 Ti). The implementation of our method is available online [6].

As quantitative error metrics we use the Mean Squared Error (MSE) and the Mean Absolute Error (MAE), both in the original units of the respective datasets (pixel disparity for depth, metres for tree height). Moreover, we also measure the Percentage of Bad Pixels (PBP) as defined in [7]:

$$\mathrm{PBP}_\delta = \frac{1}{\mathrm{N}^2} \sum_n \left[|\hat{t}_n - t_n| > \delta \right] \tag{6}$$

with $\delta = 1$ pixel for disparity, and $\delta = 3$ metres for vegetation height.

Analysis

In this subsection we analyse the mapping learned by our method, and illustrate the influence of the regularisation.

First, we visualise the mapping function $f_{\boldsymbol{\theta}}$. In Fig. 4 we plot the learned dependence between intensity g_n in the guide and depth t_n in the target image, at different image locations $\mathbf{x}_n$. Close to the discontinuity the function has a steep slope, as the network learns to translate the large intensity change into a large depth change, so as to be consistent with the depth change seen, at coarser resolution, in the source image. As one moves away from the discontinuity and into the homogeneous depth region to its right, the network response flattens out, indicating that all colours of the guide shall be translated to similar depth values. The picture nicely illustrates the mechanism behind our algorithm's ability to reproduce sharp edges: imposing smoothness on the mapping function $f_{\boldsymbol{\theta}}$ is very different from imposing smoothness on the target output. The function $f_{\boldsymbol{\theta}}$ indeed changes slowly and has similar shape at the two leftmost locations. But since that shape corresponds to a steep gradient, the depths at the two locations are very different. Regularising the mapping function instead of the output image is a lot more robust to variations in image content.

Figure 5 depicts the effect of changing the regularisation parameters $\lambda_{\mathbf{g}}$, $\lambda_{\mathbf{x}}$ and λ_{head}. The figure shows four cases:

[2] Copernicus Sentinel data 2016, processed by ESA. https://scihub.copernicus.eu/

[3] https://github.com/poolio/bilateral_solver

[4] https://github.com/bsham/SDFilter

[5] https://github.com/twhui/MSG-Net

[6] https://github.com/riccardodelutio/PixTransform

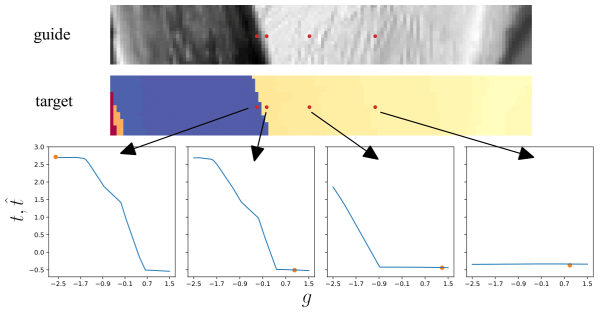

Figure 4: Illustration of the location-dependent mapping function $f_{\boldsymbol{\theta}}$.

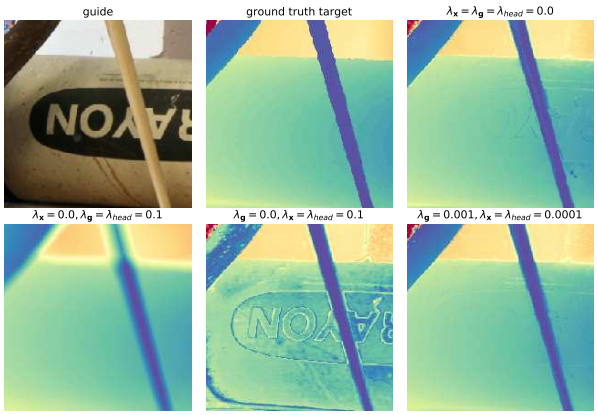

Figure 5: Illustration of different regularisation settings (upsampling factor $\times 8$).

with no regularisation, the network $f_{\boldsymbol{\theta}}$ has more capacity than needed and overreacts to intensity contrasts in the guide. That behaviour is amplified if one excessively regularises only w.r.t. the location $\mathbf{x}_m$, thus forcing $f_{\boldsymbol{\theta}}$ to base its outputs mostly on the colour values $\mathbf{g}_m$. Conversely, regularising heavily only w.r.t. $\mathbf{g}_m$ causes the network to ignore the colours of the guide, leading to blurry outputs. In the bottom right, the regularisation weights are set to a sensible compromise: $\lambda_{\mathbf{g}} = 10^{-3}$ and $\lambda_{\mathbf{x}} = \lambda_{\text{head}} = 10^{-4}$. These are the settings used in all our experiments.

Depth super-resolution

As commonly done, we run the super-resolution in disparity (inverse depth) space. In Table 1 we show the means and standard deviations of the three error metrics MSE, MAE and PBP over the images in the depth dataset, for upsampling factors of $\times 4$, $\times 8$, $\times 16$ and $\times 32$. For a $\times 4$ upsampling factor all methods achieve similar performance. MSG-Net stands out for having very low MSE, probably since it was optimised on a huge training set to minimise that error. The SD filter has a slight edge in terms of robustness and reaches the lowest MAE and PBP. It is worth pointing out that even naive bicubic upsampling is competitive, i.e., upsampling by a moderate $\times 4$ is quite an easy problem, for which the guide image has only limited effect.

For larger upsampling factors our method outperforms all others w.r.t. all three metrics. We could not run MSG-Net for factors above $\times 8$, because not enough training data was left after downsampling the low-resolution source images.

Fig. 6a shows a depth upsampling result for upsampling factor $\times 8$. Although our method on average achieves the best results for this task – see Tab. 1 – we deliberately show an image where MSG-Net has lower MSE. Nevertheless, our output is visibly sharper and better preserves discontinuities. The top right corner of the image shows a particularly difficult situation where the contrast is high, and nearby pixels have similar colours, but different depths. In this situation several methods, including ours, exhibit a tendency to rely too much on the guide image and hallucinate spurious depth patterns. In such cases, an additional regularisation of the output, e.g., with a total variation prior, could potentially be helpful.

Fig. 6b shows the results for depth upsampling by a factor $\times 32$. As can be seen, our method greatly outperforms the competitors. Not only it achieves a much lower MSE, but also the resulting image is a lot sharper and exhibits fewer artefacts. Notice in particular the two thin sticks at the bottom, where only our method reaches a reasonable reconstruction quality. Another impressive feature is the reconstruction of the hole in the middle of the image. While it is not that surprising that the boundary can be transferred from the guide; it is remarkable that from seeing the red colour of the foreground, the white colour in the background outside the object, and the area-weighted depth average of the two in the source, the network is able to extract enough information to choose the correct depth in the hole.

Super-resolution by a factor as high as $\times 32$ is evidently pushing things to the limit of what is possible, and satisfactory results are not reached for all images. Figure 7 shows a failure case. The guide image has a lot of texture details, and nearby pixels with the same colour but different depths. The target is still consistent with the source and contains the true depth boundaries, but our method also transfers a lot of spurious texture details where there should not be depth discontinuities. It may be possible to mitigate the problem – but probably not completely solve – by stronger regularisation, perhaps making the regulariser $\lambda_{\mathbf{g}}$ dependent on the upsampling factor.

Super-resolution of vegetation height

Table 2 again shows the means and standard deviations of the three error metrics over the images of the vegetation height dataset, for upsampling factors $\times 8$, $\times 16$, and

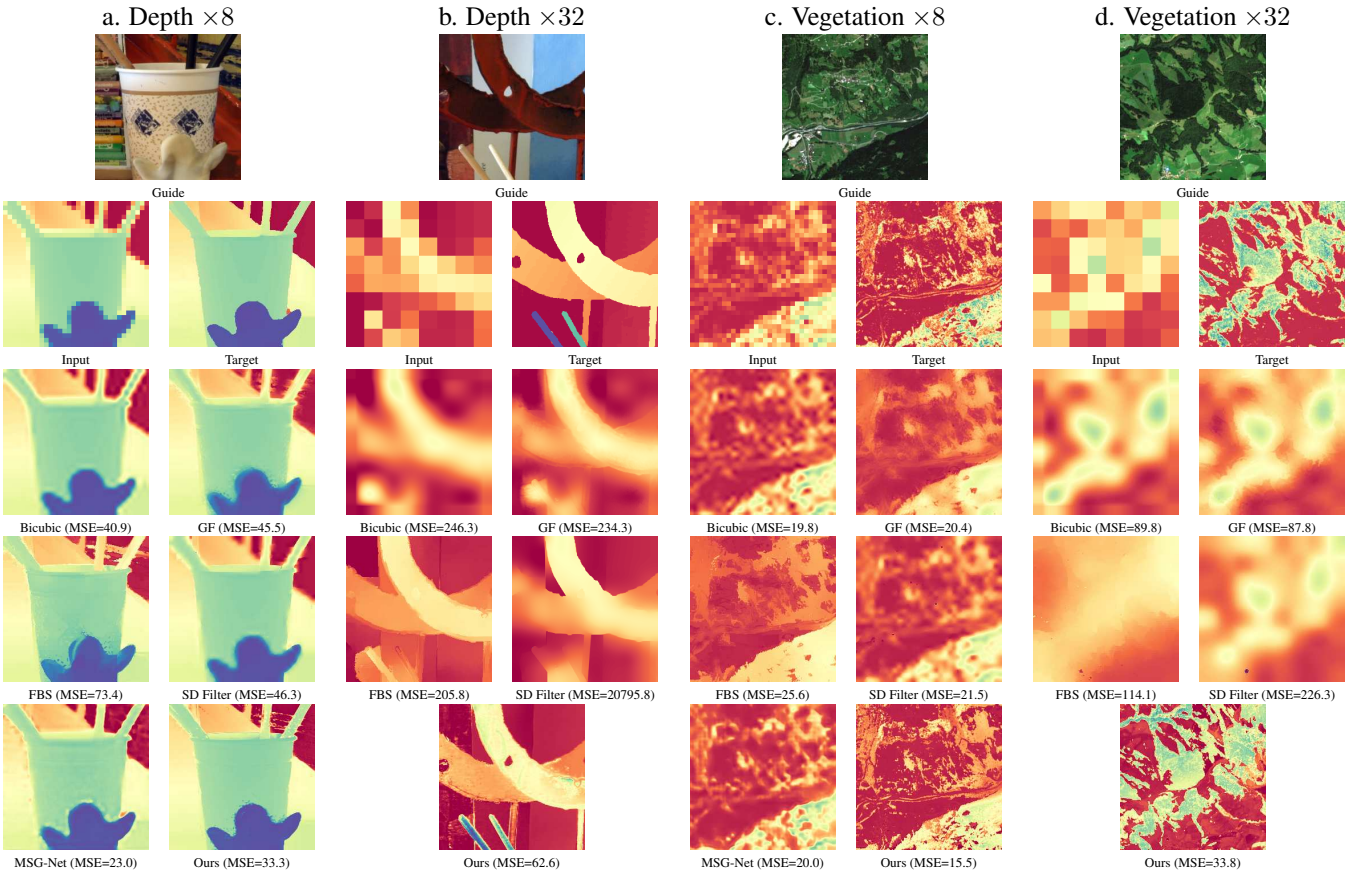

Figure 6: Qualitative results of different guided super-resolution methods.

×32. On this dataset most methods, including the bicubic upsampling, still have comparable performance at upsampling factor ×8, likely because vegetation height maps are in general smoother than depth maps. Visually, our method is again clearly sharper and recovers more high-frequency details than its competitors, see Fig. 6c-d. As for the depth case, our method outperforms others by a considerable margin at higher upsampling factors, in all three metrics.

Fig. 6c shows the results for vegetation upsampling by a factor ×8. While the MSE values are not that different, there is nevertheless a pronounced qualitative difference between our method and all others. The one that comes closest is MSG-Net, but even after having seen thousands of low-res / high-res pairs during training, the network is not able to fully recover the high-frequency details and misses a lot of the fine structures. FBS produces fairly sharp discontinuities, but has a bias towards piece-wise constant outputs, such that many of the fine details are also lost. In a sense, all methods except for ours fail, in that they perform similar to bicubic interpolation without a guide image, or even worse.

Fig. 6d shows an example for the extreme case of ×32 upsampling. The example illustrates that methods which start by blowing up the low-resolution source image cannot bridge such large resolution differences and essentially produce a smoothed version of the input. On the contrary, our method, which relies more strongly on the guide image, shines in this difficult scenario. In the pixel-to-pixel transformation from the image domain to vegetation height, no spatial detail is lost. While it appears that even the average values over large blocks of 32 × 32 pixels provide enough information to constrain the super-resolution in the target domain. Obviously, it depends also on the nature of the images whether such extreme super-resolution is feasible. In the case of the remote sensing images, the function $f_{\boldsymbol{\theta}}$ is mostly driven by the colours $\mathbf{g}_n$ of the guide, with only little spatial variation. Still, while it is less surprising that the height 0 m is correctly recovered outside the forest, which largely corresponds to a semantic segmentation of the guide; it is pleasing that within the forest regions a large portion of the height variability is correctly reconstructed too (yellow to green tones in Fig. 6d).

5. Conclusions

We have proposed a novel, unsupervised method for guided super-resolution. The key idea is to view the problem as a pixel-wise transformation of the high-res guide image to the domain of the low-res source image. By choosing

		Bicubic	GF [8]	FBS [2]	SD filter [7]	MSG-Net [11]	Ours
×4	MSE	6.5 (11.5)	7.3 (13.0)	6.6 (10.9)	5.5 (9.9)	**1.9 (3.0)**	5.0 (8.6)
	MAE	0.6 (0.5)	0.8 (0.6)	0.8 (0.5)	**0.4 (0.4)**	**0.4 (0.2)**	0.5 (0.3)
	$PBP_{\delta=1}$	7.5 (5.8)	12.3 (8.4)	14.3 (9.4)	**4.5 (3.8)**	6.0 (4.9)	6.9 (5.1)
×8	MSE	12.2 (21.9)	10.2 (18.5)	11.9 (18.5)	15.1 (27.4)	8.3 (11.2)	**5.6 (9.7)**
	MAE	1.0 (0.9)	1.0 (0.9)	1.3 (0.9)	0.7 (0.7)	1.4 (0.5)	**0.6 (0.4)**
	$PBP_{\delta=1}$	14.6 (10.0)	16.3 (10.8)	29.9 (16.6)	9.1 (7.1)	43.7 (8.5)	**8.8 (6.8)**
×16	MSE	26.5 (48.7)	21.6 (40.9)	19.3 (34.9)	115.5 (369.7)	-	**8.4 (14.9)**
	MAE	1.9 (1.8)	1.7 (1.6)	1.8 (1.5)	1.3 (1.5)	-	**0.9 (0.7)**
	$PBP_{\delta=1}$	27.3 (15.8)	26.8 (15.4)	38.8 (19.3)	18.7 (12.5)	-	**15.5 (10.9)**
×32	MSE	54.1 (95.2)	49.7 (88.3)	40.2 (72.3)	1343.3 (3374.5)	-	**26.0 (42.9)**
	MAE	3.3 (2.9)	3.2 (2.8)	3.0 (2.5)	2.7 (2.6)	-	**2.0 (1.7)**
	$PBP_{\delta=1}$	44.9 (21.6)	45.0 (21.7)	50.6 (22.5)	37.2 (19.4)	-	**36.3 (20.6)**

Table 1: Performance comparison with the state-of-the-art algorithms on the depth map dataset for different values of upsampling factors. The tables shows the means and (standard deviations) over all images of the MSE (in pixel^2), MAE (in pixels), and PBP (in %).

		Bicubic	GF [8]	FBS [2]	SD filter [7]	MSG-Net [11]	Ours
×8	MSE	18.1 (13.3)	19.0 (14.1)	28.2 (24.8)	20.7 (15.8)	17.9 (13.3)	**17.6 (15.1)**
	MAE	2.4 (1.5)	2.5 (1.7)	3.1 (2.2)	2.4 (1.6)	2.3 (1.5)	**2.1 (1.5)**
	$PBP_{\delta=3}$	26.2 (19.2)	28.2 (21.4)	32.3 (25.0)	26.8 (19.8)	26.1 (19.3)	**23.5 (18.2)**
×16	MSE	29.1 (22.5)	27.7 (21.1)	33.7 (27.8)	45.1 (45.4)	-	**19.7 (17.2)**
	MAE	3.1 (2.1)	3.1 (2.1)	3.5 (2.5)	3.8 (2.2)	-	**2.3 (1.7)**
	$PBP_{\delta=3}$	33.0 (24.4)	33.7 (25.5)	36.9 (28.0)	34.2 (25.6)	-	**24.2 (18.9)**
×32	MSE	41.5 (33.6)	40.2 (32.6)	42.3 (34.4)	160.0 (228.3)	-	**21.2 (17.5)**
	MAE	4.0 (2.8)	3.9 (2.8)	4.1 (2.9)	4.2 (3.0)	-	**2.6 (1.8)**
	$PBP_{\delta=3}$	39.1 (29.4)	39.3 (29.8)	42.0 (31.8)	40.9 (30.7)	-	**29.2 (22.4)**

Table 2: Performance comparison with the state-of-the-art algorithms on the vegetation height map dataset for different values of upsampling factors. The tables shows the means and (standard deviations) over all images of the MSE (in m^2), MAE (in m), and PBP (in %).

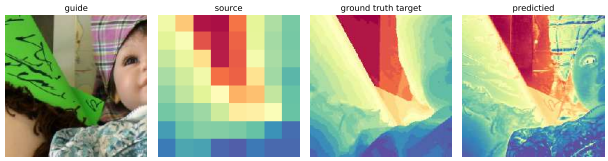

Figure 7: An example of ×32 super-resolution where our method fails. The predicted target is corrupted with lots of high-frequency details from the highly textured guide.

a multi-layer perceptron as mapping function, inference in our model is the same as fitting a CNN with only (1×1) kernels to the guide, where the loss function is the compatibility between the downsampled output and the source image. The advantage of our model is that, by construction, it avoids all unnecessary blurring. On the one hand, it does not involve any upsampling of the source image by interpolation. On the other hand, the reconstruction of the super-resolved target image is regularised at the level of the mapping function, in the spirit of CNNs, by fitting the same kernels to tens of thousands of pixels, and by penalising large weights (weight decay). Consequently, our method is able to recover very fine structures and extremely sharp edges even at high upsampling factors, setting a new state of the art.

In future work, we hope to extend the approach to handle not only super-resolution of coarse source images, but also inpainting of sparse source images, so as to recover for instance vegetation height from sparse field samples.

References

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

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

[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.

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

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

[6] Christian Ginzler and Martina L. Hobi. Countrywide stereo-image matching for updating digital surface models in the framework of the Swiss National Forest Inventory. *Remote Sensing*, 2015.

[7] Bumsub Ham, Minsu Cho, and Jean Ponce. Robust guided image filtering using nonconvex potentials. *TPAMI*, 2018.

[8] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *TPAMI*, 2013.

[9] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, 2007.

[10] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *TPAMI*, 2013.

[11] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *ECCV*, 2016.

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

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

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

[15] Charis Lanaras, Jos 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.

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

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

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

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

[20] Ziyang Ma, Kaiming He, Yichen Wei, Jian Sun, and Enhua Wu. Constant time weighted median filtering for stereo matching and beyond. In *ICCV*, 2013.

[21] Kolya Malkin, Caleb Robinson, Le Hou, Rachel Soobitsky, Jacob Czawlytko, Dimitris Samaras, Joel Saltz, Lucas Joppa, and Nebojsa Jojic. Label super-resolution networks. In *ICLR*, 2019.

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

[23] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ToG*, 2004.

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

[25] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *CVPR*, 2007.

[26] Assaf Schocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *CVPR*, 2018.

[27] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *ICCV*, 1998.

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

[29] 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.

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