Deep Joint Image Filtering

Yijun Li[1], Jia-Bin Huang[2], Narendra Ahuja[2], and Ming-Hsuan Yang[1](✉)

[1] University of California, Merced, Merced, USA
{yli62,mhyang}@ucmerced.edu
[2] University of Illinois, Urbana-Champaign, Champaign, USA
{jbhuang1,n-ahuja}@illinois.edu
https://sites.google.com/site/yijunlimaverick/deepjointfilter

Abstract. Joint image filters can leverage the guidance image as a prior and transfer the structural details from the guidance image to the target image for suppressing noise or enhancing spatial resolution. Existing methods rely on various kinds of explicit filter construction or hand-designed objective functions. It is thus difficult to understand, improve, and accelerate them in a coherent framework. In this paper, we propose a learning-based approach to construct a joint filter based on Convolutional Neural Networks. In contrast to existing methods that consider only the guidance image, our method can selectively transfer salient structures that are consistent in both guidance and target images. We show that the model trained on a certain type of data, e.g., RGB and depth images, generalizes well for other modalities, e.g., Flash/Non-Flash and RGB/NIR images. We validate the effectiveness of the proposed joint filter through extensive comparisons with state-of-the-art methods.

Keywords: Joint filtering · Deep convolutional neural networks

1 Introduction

Image filtering with a guidance signal, known as *joint* or *guided filtering*, has been successfully applied to a variety of computer vision and computer graphics tasks, such as depth map enhancement [1–3], joint upsampling [1,4], cross-modality noise reduction [5–7], and structure-texture separation [8,9]. The wide applicability of joint filters can be attributed to their adaptability in handling visual signals in various visual domains and modalities, as shown in Fig. 1. For a target image, the guidance image can either be the target image itself [6,10], high-resolution RGB images [2,3,6], images from different sensing modalities [5,11,12], or filtering outputs from previous iterations [9]. The basic idea behind joint image filtering is that the guidance image often contains important structural details that can be transferred to the target image. The main goal of joint filtering is to enhance the degraded target image due to noise or low spatial resolution while

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46493-0_10) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part IV, LNCS 9908, pp. 154–169, 2016.
DOI: 10.1007/978-3-319-46493-0_10

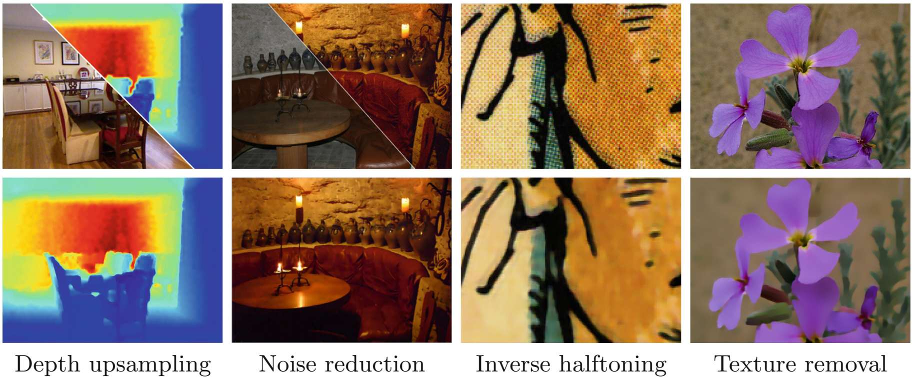


Fig. 1. Sample applications of joint image filtering: depth map upsampling, cross-modal noise reduction (flash/non-flash), inverse halftoning, and edge-preserving smoothing for texture removal. The *Target/Guidance* pair (top) can be various types of cross-modality visual data. With the help of the guidance image, important structures can be transferred to the degraded target image to help restore the blurred boundary or suppress noise (bottom).

avoiding transferring erroneous structures that are not originally presented in the target image, i.e., the texture-copying artifacts.

Several techniques have been proposed to transfer structures in the guidance image to the target image. One approach is based on explicit filter construction. For example, the bilateral filter [10] constructs spatially-varying filters that reflect local image structures (e.g., smooth regions, edges, textures) in the guidance image. These filters can then be applied to the target image to perform edge-aware smoothing [10] or joint upsampling [4]. Guided image filters [6] provide another type of filter construction by assuming a locally linear model over the guidance image. However, these filters share one common drawback. That is, the filter construction considers only the information in the guidance image and remains fixed (i.e., static guidance). When the local structures in the guidance and target images are not consistent, these techniques may transfer incorrect contents to the target image.

To address this issue, recent efforts focus on considering the contents of *both* the target and guidance images for exploiting common structures [7,9,13]. These frameworks typically build on iterative methods for minimizing a global objective function. The guidance signals are updated at each iteration (i.e., dynamic guidance) towards preserving the mutually consistent structures while suppressing structures that are not commonly shared in both images. However, these global optimization based techniques often use hand-crafted objective functions that may not reflect natural image priors well and are typically slow.

In this paper, we propose a *learning-based* joint filter based on Convolutional Neural Networks (CNNs). We propose a network architecture that consists of three sub-networks, as shown in Fig. 2. The first two sub-networks $\mathrm{CNN_T}$ and

CNN_G act as feature extractors to determine informative features from both target and guidance images. These feature responses are then concatenated as inputs for the network CNN_F to selectively transfer common structures and reconstruct the filtered output. We train the network using large quantities of real data (RGB and depth images) and learn all the network parameters simultaneously without stage-wise training. Our algorithm differs from existing methods in that our joint image filter is *completely data-driven*. This allows the network to handle complicated scenarios that may be difficult to capture through hand-crafted objective functions. While the network is trained using the RGB/D data, the network learns how to selectively transfer structures by leveraging the prior from the guidance image, rather than predicting specific depth values. As a result, the learned network generalizes well for handling images in various domains and modalities.

We make the following contributions in this paper:

- We propose a learning-based framework for constructing joint image filters. Our network takes both target and guidance images into consideration and naturally handles the inconsistent structure problem.
- With the learned joint filter, we demonstrate the state-of-the-art performance on four joint depth upsampling datasets.
- We show that the model trained on the RGB/D dataset generalizes well to handle image data in a variety of domains.

2 Related Work

Joint Image Filters. Joint image filters can be categorized into two main classes: (1) explicit filter based and (2) global optimization based. First, explicit joint filters compute the filtered output as a weighted average of neighboring pixels in the target image. The bilateral filters [1,4,9,10,14,15] and guided filters [6] are representative algorithms in this class. The filter weights, however, depend solely on the local structure of the guidance image. Therefore, erroneous structures may be transferred to the target image due to the lack of consistency check. In contrast, our model considers the contents of both images through extracting feature maps and handles this consistency issue implicitly through learning from examples.

Second, numerous approaches formulate joint filtering using a global optimization framework. The objective function typically consists of two terms: data fidelity and regularization terms. The data fidelity term ensures that the filtering output is close to the input target image. These techniques differ from each other mainly in the regularization term, which encourages the output to have a similar structure with the guidance. The regularization term can be defined according to texture derivatives [16], mid-level representations [2] such as segmentation and saliency, filtering outputs [13], or mutual structures shared by the target and guidance image [7]. However, global optimization based methods rely on hand-designed objective functions that may not reflect the complexities in natural images. Furthermore, these approaches are often time-consuming.

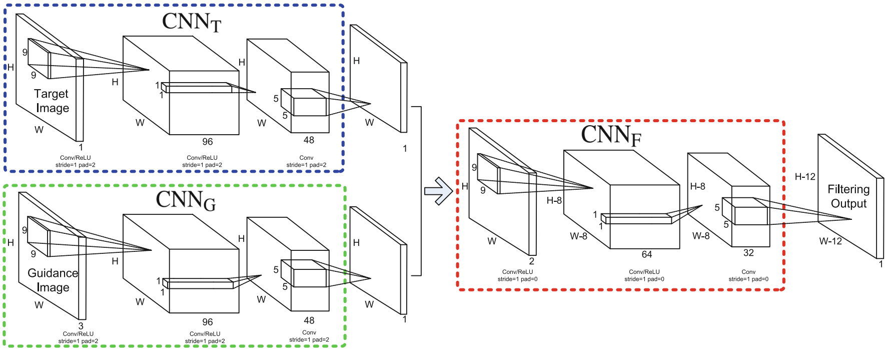


Fig. 2. The network architecture of our learning-based joint filters. The proposed model consists of three major components. Each component is a three-layer network. The sub-networks $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ aim to extract informative feature responses from the target and guidance images, respectively. These responses are then concatenated together as input for the network $\mathrm{CNN_F}$. Finally, the $\mathrm{CNN_F}$ model reconstructs the desired output by selectively transferring main structures while suppressing inconsistent structures.

In contrast, our method learns how to selectively transfer details directly from real RGB-depth datasets. Even though the training is time-consuming, the learned model is efficient during run time.

Deep Models for Low-Level Vision. While CNNs have achieved great success in high-level vision tasks [17], considerably less attention has been paid to apply these models to low-level vision problems. Recently, several methods apply CNNs for low-level vision and computational photography tasks. Examples include image denoising [18], rain drop removal [19], image super-resolution [20] and optical flow estimation [21]. Existing deep learning models for low-level vision take either one input image [18–20,22] or two images in the same domain [21]. In contrast, our network can take two streams of inputs in *heterogeneous* domains, e.g., RGB/NIR, Flash/Non-Flash, RGD/Depth, Intensity/Color. Our network architecture bears some resemblance to that in [21]. The main difference is that the merging layer in [21] uses a correlation operator while our model merges the inputs through stacking the feature responses. The closest work to ours is Xu et al. [22], which learns a CNN to approximate existing edge-aware filters from example images. Our method differs from [22] in two aspects. First, the goal of [22] is to use CNN for approximating *existing* edge-aware filters. In contrast, our goal is to learn a *new* joint image filter. Second, unlike the network in [22] that takes only one single RGB image, the proposed joint filter handles two images from different domains and modalities.

3 Learning Deep Joint Image Filters

Our CNN model consists of three sub-networks: CNN_T, CNN_G, and CNN_F, as shown in Fig. 2. First, the sub-network CNN_T takes the target image as input and extracts its feature map. Second, similar to CNN_T, the sub-network CNN_G extracts a feature map from the guidance image. Third, the sub-network CNN_F takes the concatenated feature responses from the sub-networks CNN_T and CNN_G as input and generates the final joint filtering result. Here, the major roles of the sub-network CNN_T and CNN_G are to serve as *non-linear* feature extractors that capture the local structural details in the respective target and guidance images. The sub-network CNN_F can be viewed as a non-linear regression function that maps the feature responses from both target and guidance images to the final filtered results. Note that the information from target and guidance images is simultaneously considered when predicting the final filtered result. Such a design allows us to selectively transfer structures and avoid texture-copying artifacts.

3.1 Network Architecture Design

To design a joint filter using CNNs, a straightforward implementation is to concatenate the target and guidance images together and directly train a generic CNN as in CNN_F. While in theory we can train a generic CNN to approximate the desired function for joint filtering, our empirical results show that such a network yields poor performance. Figure 3(c) shows one typical example of joint upsampling using only the network CNN_F. The main structures (e.g., the

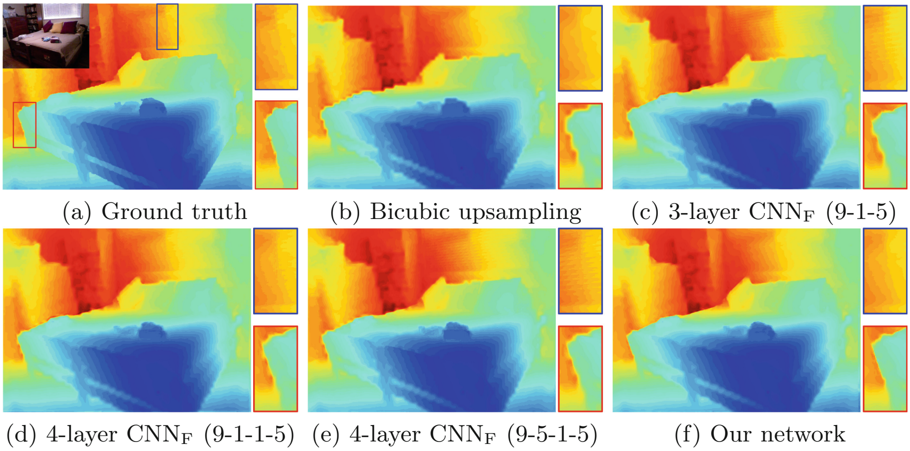

(a) Ground truth
(b) Bicubic upsampling
(c) 3-layer CNN_F (9-1-5)
(d) 4-layer CNN_F (9-1-1-5)
(e) 4-layer CNN_F (9-5-1-5)
(f) Our network

Fig. 3. Joint depth upsampling (8×) results of using different network architectures f_1-f_2-... where f_i is the filter size of the i-th layer. (a) GT depth map (inset: Guidance). (b) Bicubic upsampling. (c)–(e) Results from the straightforward implementation using CNN_F. (f) Results from the proposed model.

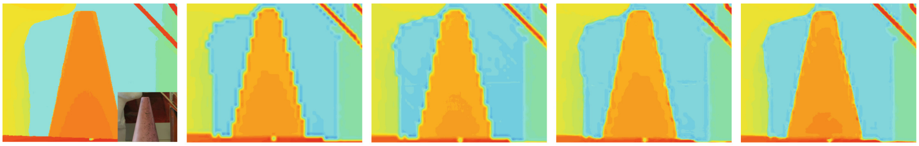

(a) Ground truth (b) Bicubic (c) RGB guided (d) Edge guided (e) Ours

Fig. 4. Joint depth upsampling (8×) results under different types of guidance images. (a) Ground truth depth map (inset: guidance). (b) Bicubic upsampling. (c) RGB guided result. (d) Edge [24] guided result. Both (c) and (d) are trained using the $\mathrm{CNN_F}$ network. (e) Result of our final network design. Note the boundary of the sculpture (left) and the cone (middle).

boundary of the bed) presented in the guidance image are *not* well transferred to the target depth image, thereby resulting in blurry boundaries. Also, inconsistent texture structures in the guidance image (e.g., the stripe pattern of the curtain on the wall) are also incorrectly copied to the target image. A possible way that may improve the result is to adjust the architecture of $\mathrm{CNN_F}$, such as increasing the network depth or using different filter sizes. However, as shown in Fig. 3(d) and (e), these variants do not show notable improvement. Blurry boundaries and the texture-copying problem still exist. Furthermore, we empirically find that there is no significant improvement using deeper models. We note that similar observations have also been reported in [23], which indicate that the effectiveness of deeper structures for low-level tasks is not as apparent as that shown in high-level tasks (e.g., image classification).

We attribute the limitation of using a generic network for joint filtering to the fact that the original RGB guidance image fails to provide direct and effective guidance as it mixes a variety of information (e.g., texture, intensity, edges). Figure 4 shows one example where we replace the original RGB guidance image with its edge map (extracted using [24]). Compared to the results guided by the RGB image (Fig. 4(c)), the result using edge map guidance (Fig. 4(d)) shows substantial improvement. Based on the above observation, we introduce two sub-networks $\mathrm{CNN_T}$ and $\mathrm{CNN_G}$ to create two separate processing streams for the two images first before concatenation. With the proposed architecture, we constrain the network to extract effective features from both images separately first and then combine them at a later stage to generate the final filtering output. This differs from conventional joint filters where the guidance information is mainly computed from the pixel-level intensity/color differences in the local neighborhood. As our models are jointly trained in an end-to-end fashion, our result (Fig. 4(e)) shows further improvements over that of using the edge guided filtering (Fig. 4(d)).

We adopt a three-layer structure for each sub-network as shown in Fig. 2. Given M training image samples $\{I_i^T, I_i^G, I_i^{gt}\}_{i=1}^M$, we learn the network parameters by minimizing the summed squared loss:

$$\|I^{gt} - \Phi(I^T, I^G)\|_2^2, \tag{1}$$

where Φ denotes the joint filtering operator, and I^T, I^G, I^{gt} denote the target image, the guidance image and the ground truth output, respectively.

3.2 Relationship to Prior Work

The proposed framework is closely related to weighted-average, optimization-based, and CNN-based models. In each layer of the network, the convolutional filters also perform a weighted-average process. In this context, our filter is similar to weighted-average filters. The key difference is that our weights are learned from data by considering both the contents of the target and guidance images while weighted-average filters (e.g., bilateral filters) depend only on the guidance image. Compared with optimization-based filters, our network plays a similar role of the fidelity and regularization terms in optimization methods by minimizing the error in (1). Through learning, our model implicitly ensures that the output does not deviate too much from the target image while sharing salient structures with the guidance image. For CNN-based models, our network architecture can be viewed as a unified model for different tasks. For example, if we remove CNN_G and use only CNN_T and CNN_F, the resulting network architecture resembles an image restoration model, e.g., SRCNN [20]. On the other hand, in cases of removing the network CNN_T, the remaining networks CNN_G and CNN_F can be viewed as using CNNs for depth prediction [25].

4 Experimental Results

In this section, we demonstrate the effectiveness of our approach through a broad range of joint image filtering tasks, including depth upsampling, colorization, texture-structure separation, and cross-modality image restoration.

Network Training. To train our network, we randomly collect 160,000 training patch pairs of size 32 × 32 from 1,000 RGB and depth maps in the NYU v2 dataset [26]. Images in the NYU dataset are real data taken in complicated indoor scenarios. We train two kinds of models for two different tasks: (1) joint image upsampling and (2) noise reduction. For the upsampling task, we obtain each low-quality target image from a ground-truth image (4×, 8×, 16×) using nearest-neighbor downsampling. For noise reduction, we generate the low-quality target image by adding Gaussian noise to each ground-truth depth map with zero mean and variance of 1e-3. We use the MatConvNet toolbox [27] for constructing and learning our joint filters. We set the learning rate of the first two layers and the third layer as 1e-3 and 1e-4, respectively.

Testing. Using the RGB/D data for training, our model takes a 1-channel target image and a 3-channel guidance image. However, the trained model is not limited in the handling RGB/D data. We can apply our model to other modalities with a few modifications. For the multi-channel target image, we apply the model independently for each channel. For the single-channel guidance image, we replicate it three times to create the 3-channel image.

4.1 Depth Map Upsampling

Datasets. We present quantitative performance on depth upsampling in four benchmark datasets where the corresponding high-resolution RGB images are available.

- Middlebury dataset [30,31]: We collect 30 images from 2001–2006 datasets with the missing depth values provided by Lu [28].
- Lu [28]: This dataset contains six real depth maps captured with the ASUS Xtion Pro camera.
- NYU v2 dataset [26]: Since we use 1,000 images in this dataset for training, the rest of images (449) are used for testing.
- SUN RGB/D [29]: We use a random subset of 2,000 high-quality RGB/D image pairs from the 3,784 pairs obtained by the Kinect V2 sensor. These images contain a variety of complicated indoor scenes.

Evaluated Methods. We compare our model against several state-of-the-art joint image filters for depth map upsampling. Among them, JBU [4], GF [6] and Ham [13] are generic methods for joint image upsampling while MRF [16],

Table 1. Quantitative comparisons. Comparisons with the state-of-the-art methods in terms of RMSE. The depth values are scaled to the range [0, 255] for the Middlebury, Lu [28] and SUN RGB/D [29] datasets. For the NYU v2 dataset [26], the depth values are measured in centimeter. Note that the depth maps in the SUN RGB/D dataset may contain missing regions due to the limitation of depth sensors. We ignore these pixels in calculating the RMSE. Numbers in bold indicate the best performance and underscored numbers indicate the second best.

Methods	Middlebury [30,31]			Lu [28]			NYU v2 [26]			SUN RGB/D [29]		
	4×	8×	16×	4×	8×	16×	4×	8×	16×	4×	8×	16×
Bicubic	4.44	7.58	11.87	5.07	9.22	14.27	8.16	14.22	22.32	2.09	3.45	5.48
MRF [16]	4.26	7.43	11.80	4.90	9.03	14.19	7.84	13.98	22.20	1.99	3.38	5.45
GF [6]	4.01	7.22	11.70	4.87	8.85	14.09	7.32	13.62	22.03	1.91	3.31	5.41
JBU [4]	2.44	3.81	6.13	2.99	5.06	**7.51**	4.07	8.29	13.35	1.37	2.01	3.15
TGV [3]	3.39	5.41	12.03	4.48	7.58	17.46	6.98	11.23	28.13	1.94	3.01	5.87
Park [2]	2.82	4.08	7.26	4.09	6.19	10.14	5.21	9.56	18.10	1.78	2.76	4.77
Ham [13]	3.14	5.03	8.83	4.65	7.53	11.52	5.27	12.31	19.24	1.67	2.60	4.36
Ours	**2.14**	**3.77**	**6.12**	**2.54**	**4.71**	7.66	**3.54**	**6.20**	**10.21**	**1.28**	**1.81**	**2.78**

Table 2. Average run-time of depth map upsampling algorithms on images of 640 × 480 pixels from the NYU v2 dataset.

Methods	MRF [16]	GF [6]	JBU [4]	TGV [3]	Park [2]	Ham [13]	**Ours**
Time(s)	0.76	0.08	5.6	68	45	8.6	1.3

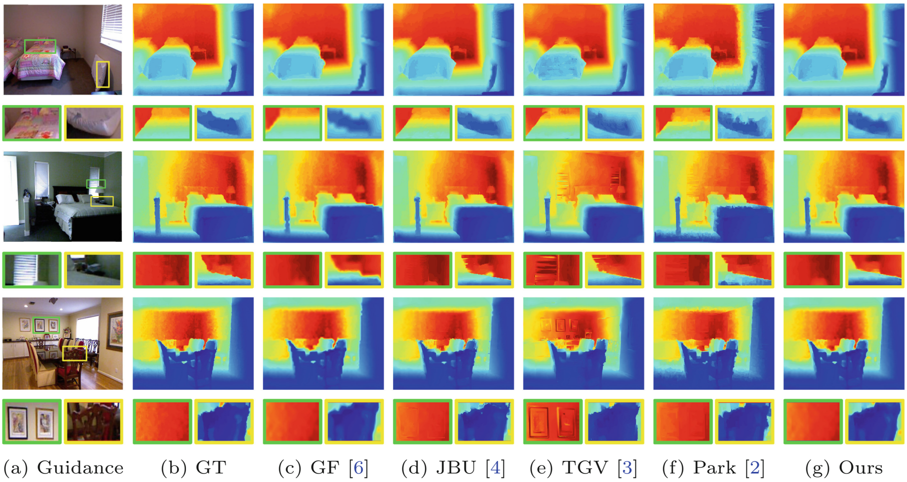


Fig. 5. Qualitative comparisons of joint depth upsampling algorithms for a scaling factor of 8×.

Table 3. Quantitative comparisons of different upsampling methods for colorization.

Methods	Bicubic	GF [6]	Ham [13]	Ours
RMSE	6.01	5.74	6.31	5.48

TGV [3] and Park [2] are algorithms specifically designed for image guided depth upsampling. The low-quality target image is obtained from the ground-truth via nearest-neighbor downsampling [2,3,13].

Table 1 shows the quantitative results in terms of the root mean squared errors (RMSE). For other methods, we use default parameters suggested in their papers. The proposed model performs well against state-of-the-art methods on both synthetic and real datasets. The extensive evaluations on real depth maps demonstrate the effectiveness of our algorithm in handling complicated indoor scenes in the real world. We also compare the average run-time of different methods on the NYU v2 dataset in Table 2. We carry out the experiments on the same machine with an Intel i7 3.6 GHz CPU and 16GB RAM. Compared with other methods, the proposed algorithm performs efficiently with high-quality results.

We show in Fig. 5 three indoor scene examples (real data) for qualitative comparisons. The main advantage of the proposed joint filter is to selectively transfer salient structures in the guidance image while avoiding artifacts (see the green boxes). The GF [6] method does not recover the degraded boundary well under a large upsampling factor (e.g., 8×). The JBU [4], TGV [3] and Park [2] approaches are agnostic to structural consistency between the target and the guidance images, and thus transfer erroneous details. In contrast, our results are smoother, sharper and more accurate with respect to the ground truth.

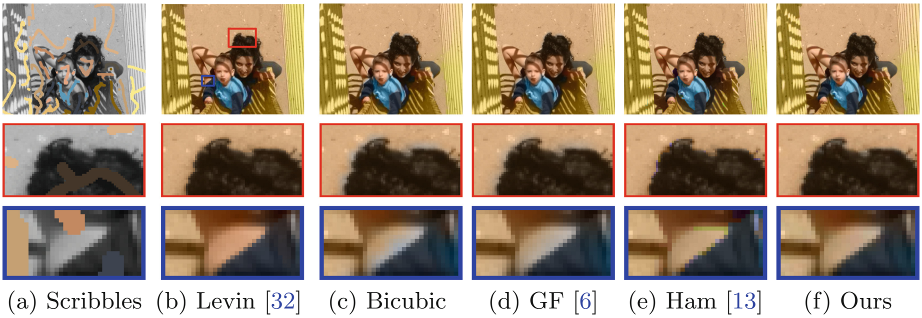


Fig. 6. Joint image upsampling applied to colorization. The computational cost: (b) 8.2 s (c) 1.3 s (d) 1.5 s (e) 28.8 s (f) 2.8 s. The close-up areas clearly show that our joint upsampling results have fewer color bleeding artifacts and are comparable with the results computed using the full resolution image.

4.2 Joint Image Upsampling

Numerous computational photography applications require computing a solution (e.g., chromaticity, disparity, labels) over the pixel grid. However, it is often time-consuming to directly obtain the high-resolution solution maps. We demonstrate the use of joint image upsampling with colorization [32] as an example. We first compute the solution map (chromaticity) on the downsampled image using the user-specified color scribbles [32], and then use the original high-resolution intensity image as guidance to upsample the low-resolution chromaticity map. Figure 6 shows that our model is able to achieve visually pleasing results with much less color bleeding artifacts while being more efficient. Our results are visually similar to the direct solutions on the high-resolution intensity images (Fig. 6(b)). We also show quantitative comparisons in Table 3. We use the direct solution of [32] on the high-resolution image as GT and compute the RMSE over seven test images in [32]. Table 3 shows that our results approximate the direction solution best.

4.3 Structure-Texture Separation

We apply our model for texture removal and structure extraction. We use the target image itself as the guidance and adopt a similar strategy as in the rolling guidance filter (RGF) [9] to remove small-scale textures. We use inverse half-toning task as an example. A halftoned image is generated by the reprographic technique that simulates continuous tone imagery using various dot patterns [33], as shown in Fig. 7(a). The goal of inverse half-toning is to remove these dots and preserve main structures. We compare our results with those from RGF [9], L0 [34], Xu [8] and Kopf [33] for halftoned images reconstruction. Figure 7 shows that our filter can well preserve edges and achieve comparable performance against Kopf [33].

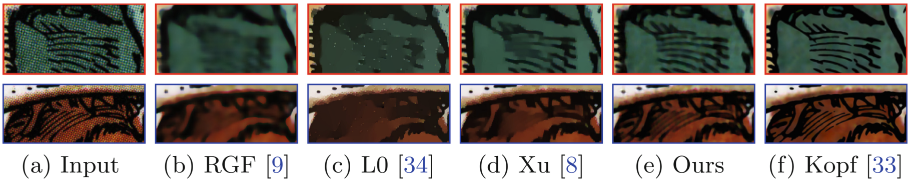
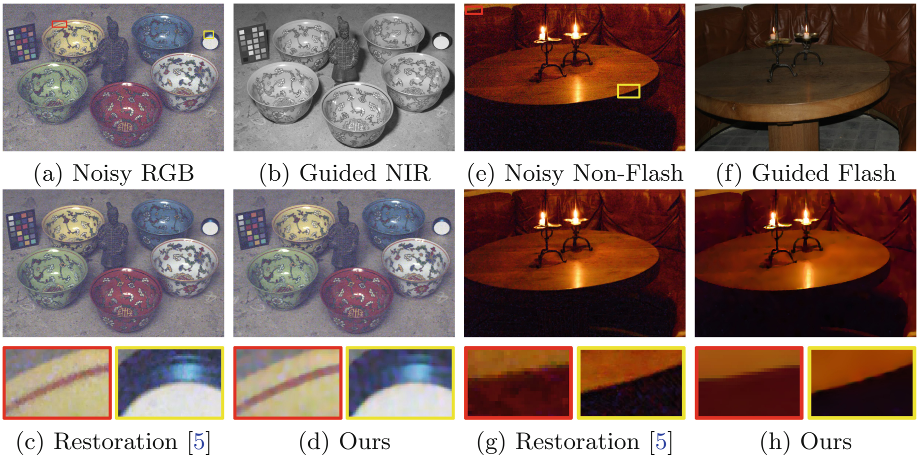

(a) Input (b) RGF [9] (c) L0 [34] (d) Xu [8] (e) Ours (f) Kopf [33]

Fig. 7. Comparisons of inverse halftoning results. For each method, we carefully select the parameter for the optimal results. (b) $\sigma_s = 3, \sigma_r = 0.1$. (c) $\lambda = 0.06$. (d) $\lambda = 0.005, \sigma = 3$. (e) Our result. (f) Result of [33]. Note that [33] is an algorithm specifically designed for reconstructing halftoned images.


(a) Noisy RGB (b) Guided NIR (e) Noisy Non-Flash (f) Guided Flash

(c) Restoration [5] (d) Ours (g) Restoration [5] (h) Ours

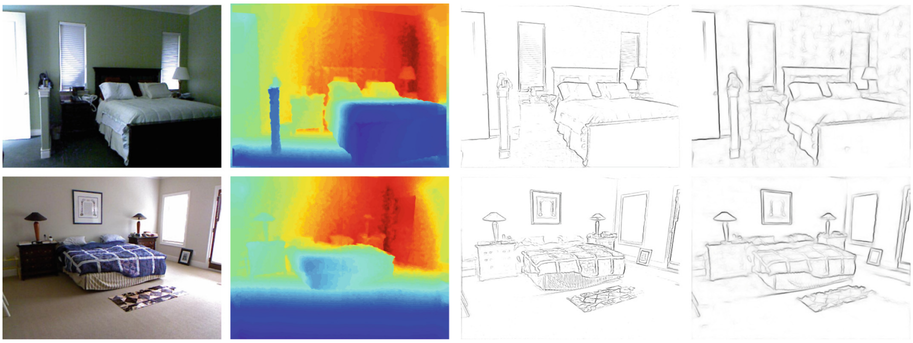
Fig. 8. Sample results of noise reduction using RGB/NIR image pairs (a)–(d) and Flash/Non-Flash image pairs (e)–(h).

4.4 Cross-Modality Filtering for Noise Reduction

Finally, we demonstrate that our model can handle various visual domains through two noise reduction applications using RGB/NIR and Flash/Non-Flash image pairs. Figure 8(a)–(d) show sample results on joint image denoising with the NIR guidance image. The filtering results by our method are comparable to those of the state-of-the-art technique [5]. For Flash/Non-Flash image pairs, we aim to merge the ambient qualities of the no-flash image with the high-frequency details of the flash image. Guided by a flash image, the filtering result of our method is comparable to that of [5], as shown in Fig. 8(e)–(h).

5 Discussions

What Has the Network Learned? In Fig. 9(c), we visualize the learned guidance from CNN_G using two examples from the NYU v2 dataset [26].



(a) Guidance (b) Depth map (c) Learned guidance (d) Edge map [24]

Fig. 9. Comparison between the learned guidance feature maps from CNN$_\mathrm{G}$ and edge maps from [24]. It suggests that the network extracts informative, salient structures from the guidance image for content transfer.

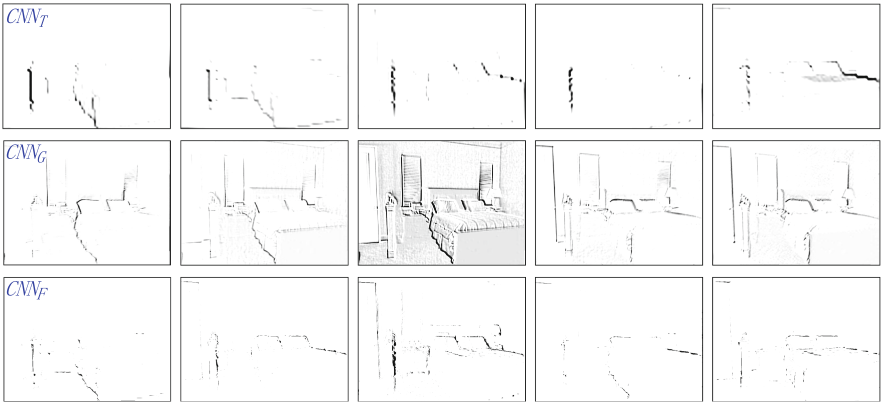


Fig. 10. Sample feature responses of the input in Fig. 9(a) at the first layer of CNN$_\mathrm{T}$ (top), and CNN$_\mathrm{G}$ (middle), and the second layer of CNN$_\mathrm{F}$ (bottom). Pixels with darker intensities indicate stronger responses. Note that with the help of CNN$_\mathrm{F}$, inconsistent structures (e.g., the window on the wall) are successfully suppressed.

In general, the learned guidance appears like an edge map highlighting the salient structures in the guidance image. We show edge detection results from [24] in Fig. 9(d). Both results show strong responses to the main structures, but the guidance map generated by CNN$_\mathrm{G}$ appears to detect sharper boundaries while suppressing responses to small-scale textures, e.g., the wall in the first example. This is why using only CNN$_\mathrm{F}$ (Fig. 3(c)) does not perform well as it lacks the salient feature extraction step from the sub-network CNN$_\mathrm{G}$. Similar observations are also found in [35] where a reference edge map is learned first from intermediate CNN features for the semantic segmentation.

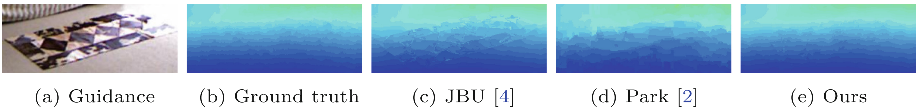


Fig. 11. Comparisons of different joint upsampling methods on the texture-copying issue (the area carpet on the floor contains unwanted texture structures).

Selective Structure Transfer. Using the learned guidance alone to transfer details may sometimes be erroneous. In particular, the structures extracted from the guidance image may not exist in the target image. In Fig. 10, the top and middle rows show typical responses at the first layer of CNN_T and CNN_G. These two sub-networks show strong responses to edges from the target and guidance image respectively. Note that there are inconsistent structures (e.g., the window on the wall). The bottom row of Fig. 10 shows sample responses at the second layer of CNN_F. We observe that the sub-network CNN_F re-organizes the extracted structural features and suppresses inconsistent details.

We present another example in Fig. 11. We note that the ground truth depth map of the selected region is smooth. However, due to the high-contrast patterns on the mat in the guidance image, several methods, e.g., [2,4], incorrectly transfer the mat structure to the upsampled depth map. The reason is that these methods [2,4] rely only on structures in the guidance image. The problem, commonly known as texture-copying artifacts, often occurs when the texture in the guidance image has strong color contrast. With the help of the CNN_F, our filter successfully blocks the texture structure in the guidance image. Figure 11(e) shows our joint upsampling result.

Network Architecture. Based on our network configurations in Fig. 2, we analyze the effects of the performance under different hyper-parameter settings. As suggested in [23] that the number of layers does not play a significant role for low-level tasks (e.g., super-resolution), we vary the filter number n_i and size f_i $(i = 1, 2)$ of the first two layers in each sub-network. The training process is the same as described in Sect. 3.1 and the evaluation is conducted on the NYU v2 dataset [26] (449 test images). Table 4 shows that larger number and larger size of the filter may not always yield performance improvements. Therefore, the parameter selection of our method (shown in Fig. 2) strikes a good balance between performance and efficiency.

We set the output feature maps extracted from the target and guidance images as one single channel. That is, the input of CNN_F is of size $H \times W \times 2$. Intuitively, using multi-dimensional features may further improve the model capacity and performance. However, our experimental results (see the supplementary material) indicate that using multi-dimensional feature maps only slows down the training process without clear performance improvements.

Table 4. Depth upsampling results (RMSE in centimeters) of using different filter numbers and sizes in each sub-network. We apply the same parameters to all three sub-networks.

Size fixed	$n_1 = 128, n_2 = 64$	$n_1 = 96, n_2 = 48$	$n_1 = 64, n_2 = 32$
$Upscale = 8$	6.44	6.32	6.35
Number fixed	$f_1 = 11, f_2 = 1, f_3 = 7$	$f_1 = 9, f_2 = 1, f_3 = 7$	$f_1 = 9, f_2 = 1, f_3 = 5$
$Upscale = 8$	6.28	6.40	6.20

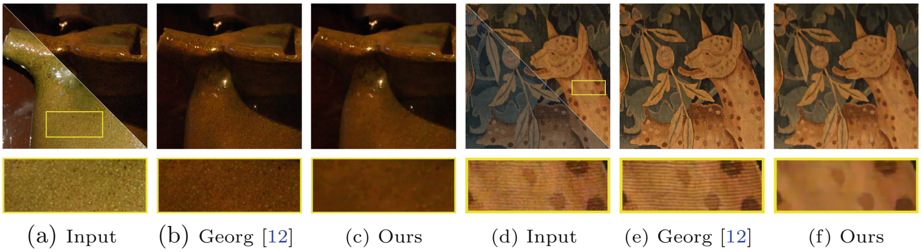

(a) Input (b) Georg [12] (c) Ours (d) Input (e) Georg [12] (f) Ours

Fig. 12. Failure cases. Detailed small-scale textures (yellow rectangle) in the guidance image are over-smoothed by our filter. (Color figure online)

Failure Cases. We note that in some images, our model fails to transfer small-scale details from the guidance map. That is, our model incorrectly treats some small-scale details as noise. This can be explained by the fact that our training data is based on depth images. The depth map usually tends to be smooth and does not contain many details.

Figure 12 shows two examples of a Flash/Non-Flash pair for noise reduction. There are several spotty textures on the porcelain in the guided flash image that should be preserved when filtering the noisy non-flash image, and likewise the small-scale strip textures on the carpet. Compared with Georg [12] (Fig. 12(b) and (d)) that deals with Flash/Non-flash images, our filter treats these small-scale details as noise and tends to over-smooth the contents. We will use non-depth data to address the over-smoothing problem in our future work.

6 Conclusions

We present a learning-based approach for joint filtering based on convolutional neural networks. Instead of relying only on the guidance image, we design two sub-networks CNN_T and CNN_G to consider the contents of both the target and guidance images by extracting informative features respectively. These feature maps are then concatenated as inputs for the network CNN_F to selectively transfer salient structures from the guidance image to the target image while suppressing structures that are not consistent in both images. While we train our network on one type of data (RGB/D images), our model generalizes well on handling images in various modalities, e.g., RGB/NIR and Flash/Non-Flash image pairs. We show that the proposed model is efficient and achieves competitive performance against state-of-the-art techniques on various computer vision and computational photography applications.

Acknowledgment. This work is supported in part by the NSF CAREER Grant #1149783, gifts from Adobe and Nvidia, and Office of Naval Research under grant N00014-16-1-2314.

References

1. Yang, Q., Yang, R., Davis, J., Nistér, D.: Spatial-depth super resolution for range images. In: CVPR (2007)
2. Park, J., Kim, H., Tai, Y.W., Brown, M.S., Kweon, I.: High quality depth map upsampling for 3D-TOF cameras. In: ICCV (2011)
3. Ferstl, D., Reinbacher, C., Ranftl, R., Rüther, M., Bischof, H.: Image guided depth upsampling using anisotropic total generalized variation. In: ICCV (2013)
4. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. In: SIGGRAPH (2007)
5. Yan, Q., Shen, X., Xu, L., Zhuo, S., Zhang, X., Shen, L., Jia, J.: Cross-field joint image restoration via scale map. In: ICCV (2013)
6. He, K., Sun, J., Tang, X.: Guided image filtering. PAMI **35**(6), 1397–1409 (2013)
7. Shen, X., Zhou, C., Xu, L., Jia, J.: Mutual-structure for joint filtering. In: ICCV (2015)
8. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM Trans. Graph. **31**(6), 139 (2012)
9. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 815–830. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10578-9_53
10. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV (1998)
11. Eisemann, E., Durand, F.: Flash photography enhancement via intrinsic relighting. In: SIGGRAPH (2004)
12. Georg, P., Maneesh, A., Hugues, H., Richard, S., Michael, C., Kentaro, T.: Digital photography with flash and no-flash image pairs. In: SIGGRAPH (2004)
13. Ham, B., Cho, M., Ponce, J.: Robust image filtering using joint static and dynamic guidance. In: CVPR (2015)
14. Liu, M.Y., Tuzel, O., Taguchi, Y.: Joint geodesic upsampling of depth images. In: CVPR (2013)
15. Jampani, V., Kiefel, M., Gehler, P.V.: Learning sparse high dimensional filters: image filtering, dense crfs and bilateral neural networks. In: CVPR (2016)
16. Diebel, J., Thrun, S.: An application of Markov random fields to range sensing. In: NIPS (2005)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS (2012)
18. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: can plain neural networks compete with BM3D? In: CVPR (2012)
19. Eigen, D., Krishnan, D., Fergus, R.: Restoring an image taken through a window covered with dirt or rain. In: ICCV (2013)

20. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10593-2_13
21. Philipp, F., Alexey, D., Eddy, I., Philip, H., Caner, H., Vladimir, G., van der Patrick, S., Daniel, C., Thomas, B.: FlowNet: learning optical flow with convolutional networks. In: ICCV (2015)
22. Xu, L., Ren, J., Yan, Q., Liao, R., Jia, J.: Deep edge-aware filters. In: ICML (2015)
23. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. PAMI **38**(2), 295–307 (2015)
24. Dollár, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV (2013)
25. David, E., Christian, P., Rob, F.: Depth map prediction from a single image using a multi-scale deep network. In: NIPS (2014)
26. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33715-4_54
27. Andrea, V., Karel, L.: MatConvNet – convolutional neural networks for matlab. In: ACM MM (2015)
28. Lu, S., Ren, X., Liu, F.: Depth enhancement via low-rank matrix completion. In: CVPR (2014)
29. Song, S., Lichtenberg, S.P., Xiao, J.: Sun RGB-D: a RGB-D scene understanding benchmark suite. In: CVPR (2015)
30. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: CVPR (2007)
31. Hirschmüller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: CVPR (2007)
32. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: SIGGRAPH (2004)
33. Kopf, J., Lischinski, D.: Digital reconstruction of halftoned color comics. In: SIGGRAPH (2012)
34. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via ℓ_0 gradient minimization. In: ACM SIGGRAPH ASIA (2011)
35. Chen, L.C., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L.: Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In: CVPR (2016)