Learning Recursive Filters for Low-Level Vision via a Hybrid Neural Network

Sifei Liu[1], Jinshan Pan[1,2], Ming-Hsuan Yang[1]

[1]UC Merced, [2]Dalian University of Technology
sliu32,mhyang@ucmerced.edu, sdluran@gmail.com

1 Outline

We specify more general settings and performance of the proposed network in Section 2. We visualize and analyze more weight maps generated by the deep CNN in the proposed algorithm in Section 3. We demonstrate the effectiveness of the proposed image denoising model via more quantitative and qualitative results in Section 4. In Section 5, we compare the filtered results by the proposed algorithm and the original implementations for edge-preserving smoothing and image enhancement. In Sections 6 and 7, more examples of image pixel interpolation are presented with comparisons to several state-of-the-art algorithms. We further provide an interesting application for color interpolation in Section 8.

For ease of comparisons, we show all results by different methods in one page, and the details can be clearly viewed at the original image resolution, or equivalently, by zooming in on each figure.

Meanwhile, we also create a video demo that combines the proposed RTV filtering through approximation, as well as the generated edge maps (See Fig. 2) to formulate the cartooning effect. The lightening of all frames are uniformly adjusted before LRNN processing, and the edges are directly summed to the filtered frames. The video is processed frame-by-frame, but is quit stable over the temporal domain in both filtering effect and edge producing.

2 More Settings and Performance

During the training phase, the momentum, weight decay and batch size are set to be 0.9, 10^{-3}, and 20, where the initial learning rate is set as 10^{-4}. The extensive quantitative performance with respect to SSIM is shown in Table 1. Specifically, our model takes 0.55 and 0.88 ms for an input image with $1080p$ or $2k$ resolution, respectively. The corresponding run time performance of the CNN filter can be found and compared in [1].

3 Analysis of Weight Maps

In this section, we show more weight maps that are generated through a single LRNN, which simulates the relative total variation (RTV) filter [4], as introduced in Section 5.1 and Fig. 1 and 4 in the manuscript.

Table 1. Quantitative evaluations for learning various image filters.

Methods	L_0 [2]	BLF [3]	RTV [4]	RGF [5]	WLS [6]	WMF [7]	Shock filter [8]
PSNRs of [1]	32.8	38.4	32.1	35.9	36.2	31.6	30.0
Our PSNRs	30.9	**38.6**	**37.1**	**42.2**	**39.4**	**34.0**	**31.8**
SSIM of [1]	0.99	**0.99**	0.98	**0.99**	0.98	0.98	**0.97**
SSIM ours	0.97	**0.99**	**0.98**	**0.99**	**0.99**	0.97	**0.97**

(a) input (b) x-map (c) y-map (d) smoothed

Fig. 1. Visualization of weight maps for L0 edge-preserving smoothing filter [2].

We demonstrate that the maps generated by the deep CNN are accurately associated to the image edges, without any defined priors or direct supervision. Specifically, the weight maps with respect to the x and y-axes are slightly different (see Fig. 2(b) and (c)). The weight map for the x-axis controls the connection of horizontally adjacent nodes, which therefore exhibits more obvious black vertical lines. In contrast, the weight map for the y-axis shows more horizontal black lines. Through the guidance of the weight maps, very similar smoothing effects to RTV filter can be achieved by the proposed algorithm, as shown in Fig. 2(d).

Note that the weight maps are task-dependent and generated through the proposed data-driven approach. Similar weight maps can be generated through approximating other edge-preserving filters (e.g., L0 filter [2]), which is not designed based on edge prior, as shown in Fig. 1. While one can also manually design the weight maps and feed them to the RNNs to create new type of filters, it is beyond the scope of data-driven approach and not be discussed in this work.

(a) input (b) x-map (c) y-map (d) smoothed

Fig. 2. Visualization of weight maps for RTV [4] edge-preserving smoothing filter.

4 More Results for Image Denoising

In this section, we show more quantitative evaluations in Table 2 and qualitative results in Fig. 3 for image denoising. We apply the test set of berkeley segmentation dataset 500 (BSDS500) which contains 200 natural images, and compare the proposed algorithm with the state-of-the-art methods, including EPLL [9], bm3d [10] and deep CNN based model [11]. We apply a white Gaussian noise with the standard deviation of 0.01 to each input image, as introduced in Section 5.2. Considering the computational costs evaluated on Section 5.4, the proposed algorithm outperforms the state-of-the-art methods in terms of the overall performance as well as efficiency.

Table 2. Quantitative evaluations for image denoising on BSDS500-test.

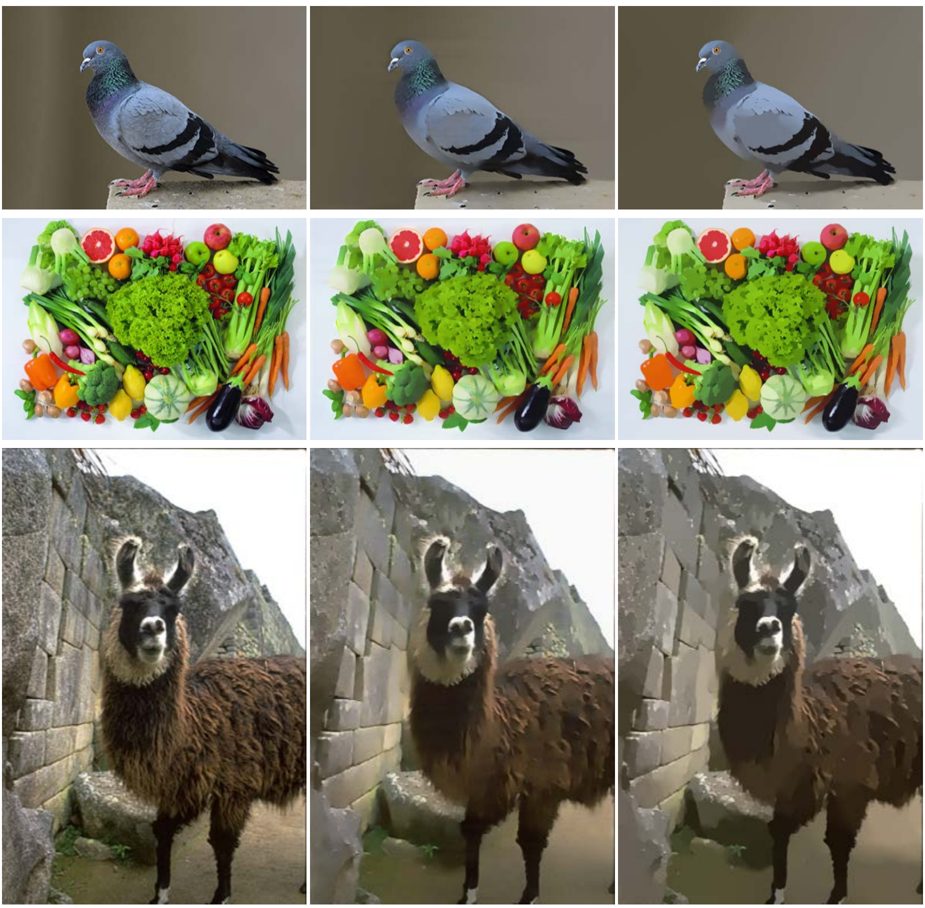
Methods	EPLL [9]	deep CNN [11]	bm3d [10]	ours
Average PSNRs	28.38	28.82	28.38	**31.05**

In Fig. 3, several patches are cropped for better visualization and comparisons. The EPLL algorithm over-smooths many regions (in all examples) especially on the background, and introduces color noise (being obvious on the first and third columns). The CNN based method preserves more details. However, it produces more texture-like noise on smooth regions. Comparatively, the results generated by the proposed algorithm (see Fig. 3, the 4-th row) are visually pleasant on both preserving details and removing noise.

Fig. 3. Image denoising. First row: image with white Gaussian noise; Second row: image denoised by EPLL [9]; Third row: image denoised by deep CNN based method [11]; Forth row: image denoised by the proposed algorithm. Best viewed with zoom-in.

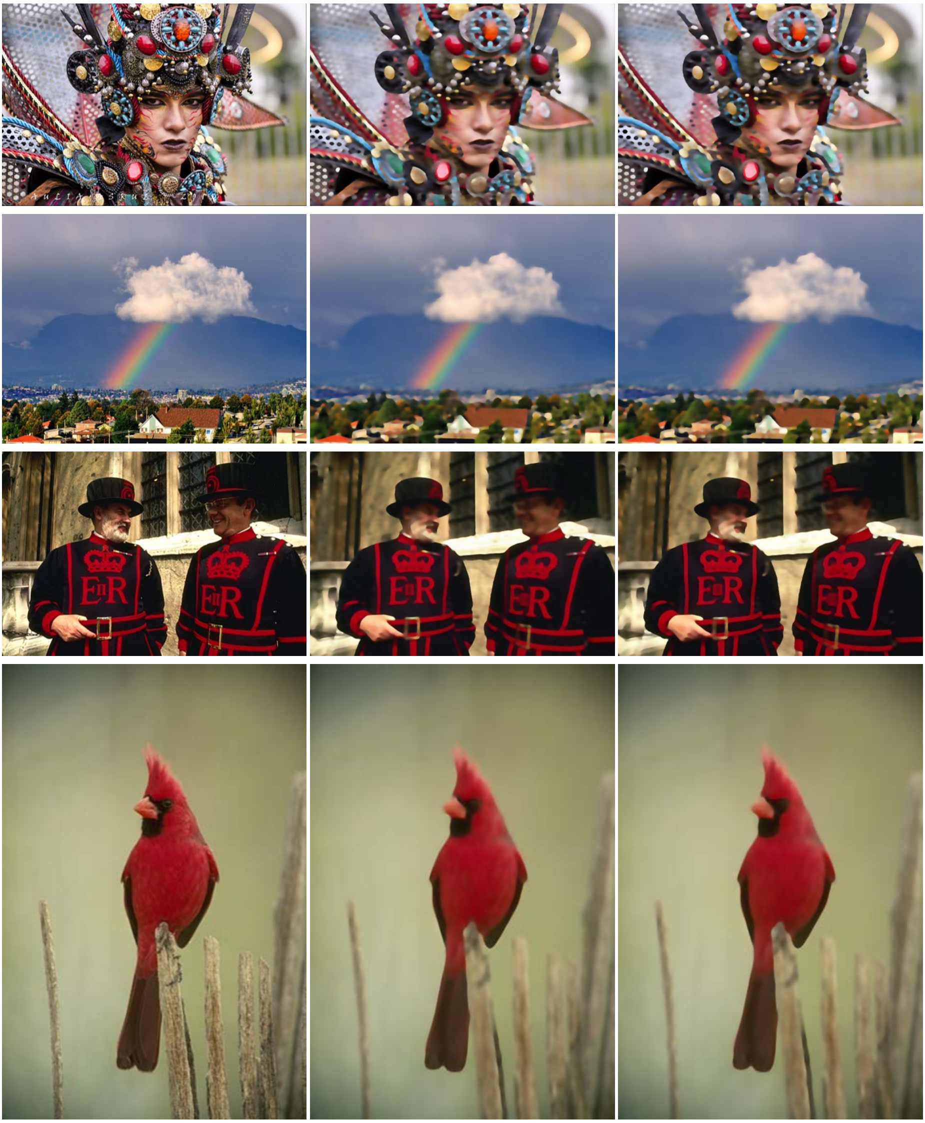
5 More Results for Image Filters

In this section, we show more qualitative results for the approximation of edge-preserving/enhancement filters to demonstrate the effectiveness of the proposed method. Specifically, we crop one patch for each image in visualizing the approximation of shock filter (see Fig. 8), for better comparisons with respect to the region details.



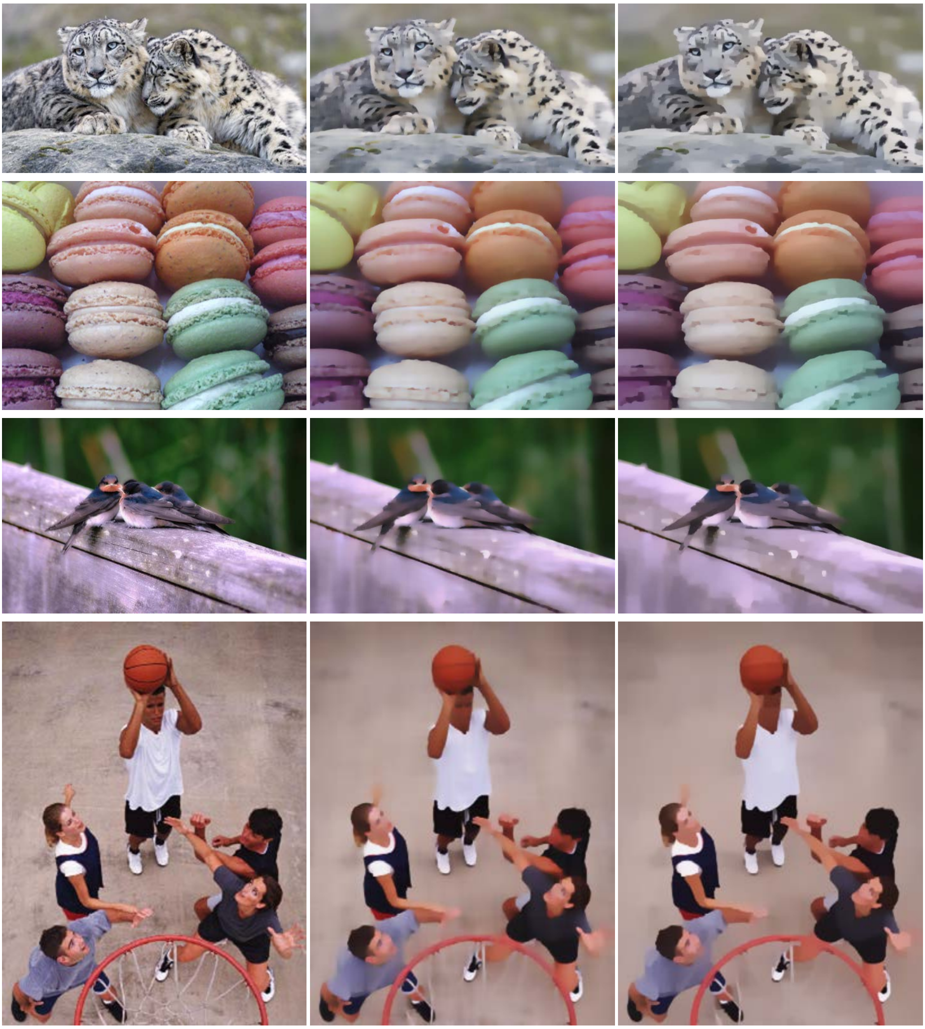
(a) input (b) proposed (c) L0

Fig. 4. Approximation of L0 edge-smoothing method [2]. Zooming in to see more details.



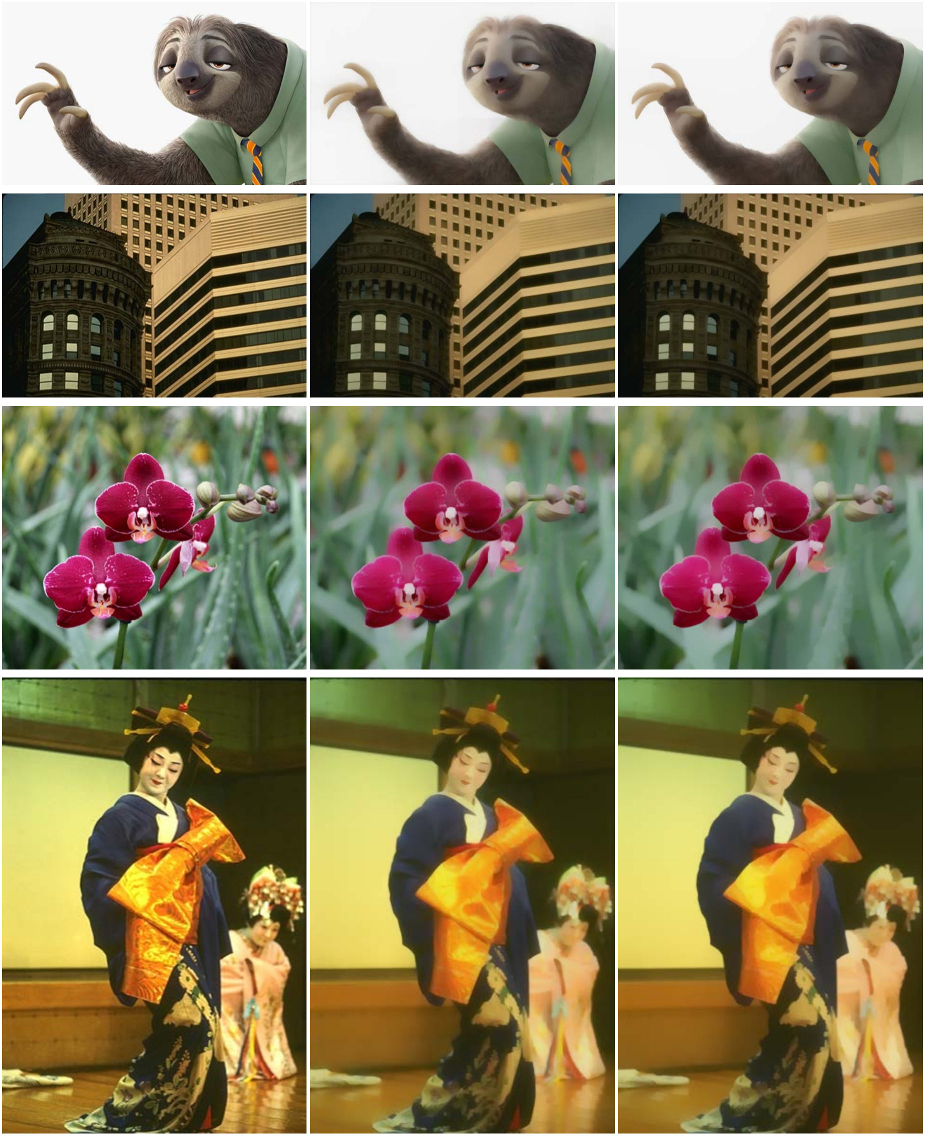
(a) input (b) proposed (c) RGF

Fig. 5. Approximation of RGF [5] edge-smoothing method. Zooming in to see details.



(a) input (b) proposed (c) RTV

Fig. 6. Approximation of RTV [4] edge-smoothing method. Zooming in to see details.



(a) input (b) proposed (c) WLS

Fig. 7. Approximation of WLS [6] edge-smoothing method. Zooming in to see details.

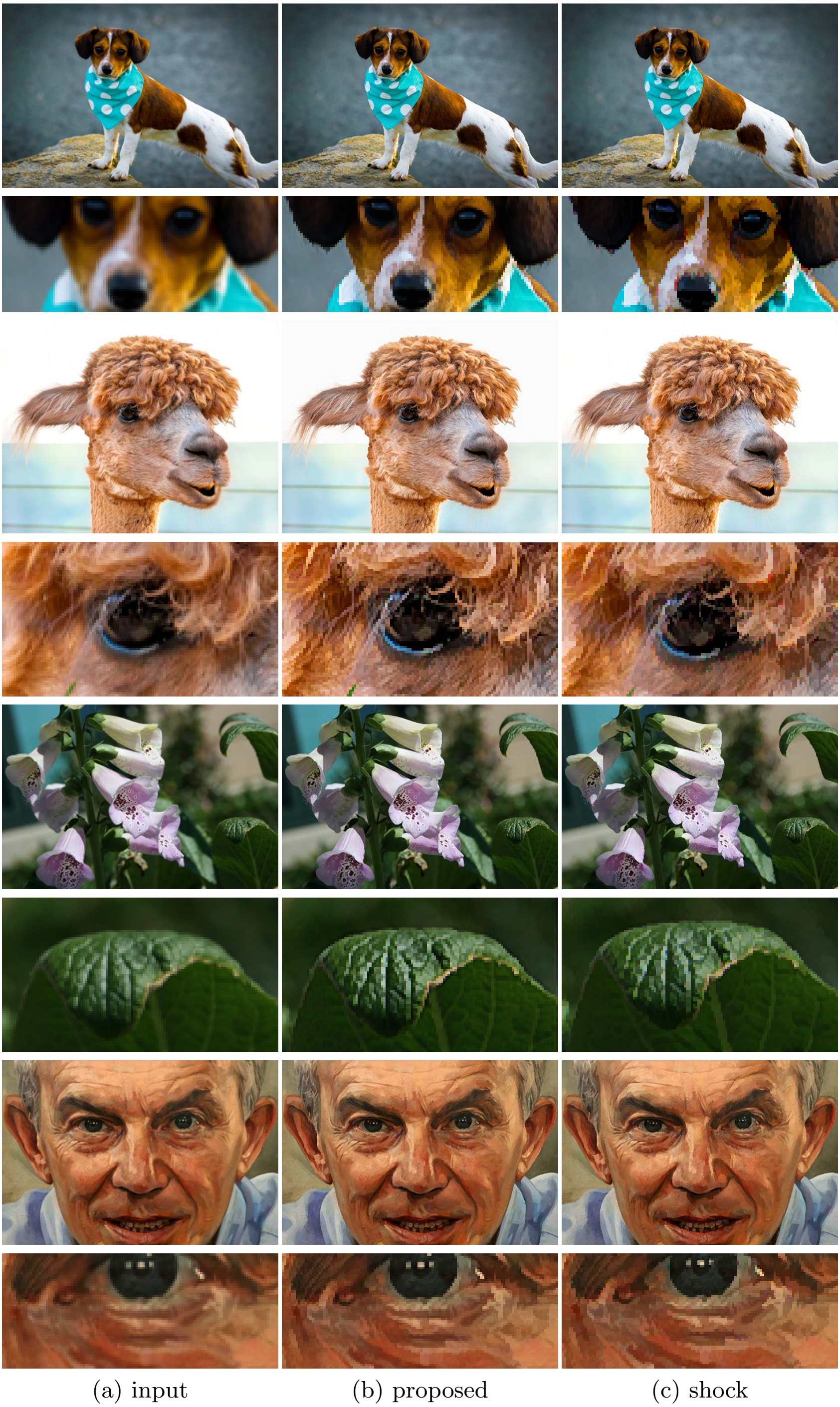
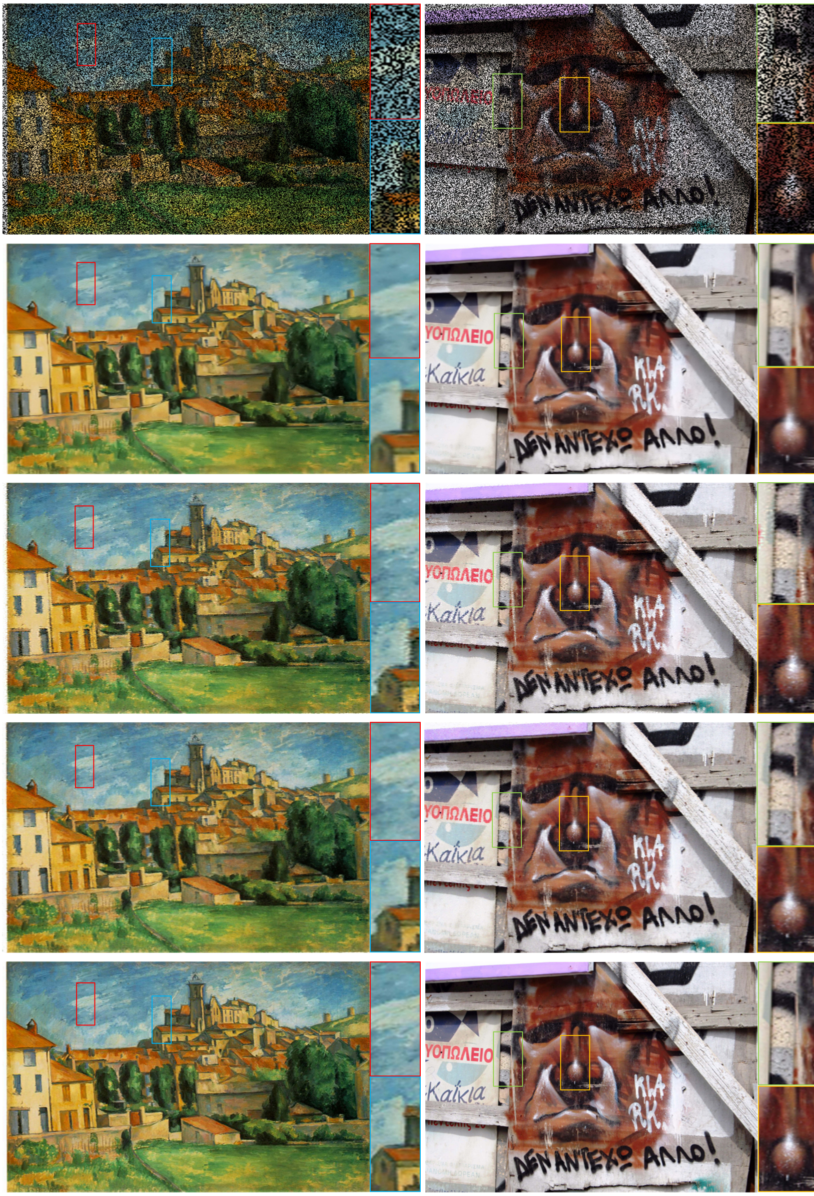


Fig. 8. Approximation of Shock filter [8] image enhancement method. Zooming in to see details.

6 More Results for Pixel Interpolation

In this section, we show more qualitative results for pixel interpolation with 50% pixels randomly masked, as introduced in Section 5.3 in the paper. Specifically, we compare the results with two state-of-the-art inpainting algorithms [9,12] for images with various artistic photography and painting work in Fig. 9. For ease of comparisons, we show all results by different methods on one page, and the details can be clearly viewed at the original image resolution, or equivalently by zooming in on Fig. 9.

The EPLL algorithm can recover the edges but over-smooths many details (in all examples). The CNN based method, on the other hand, produces jagged boundaries (e.g., edges along houses on the hill in (a), long edges in (b)). Comparatively, the results generated by the proposed algorithm (fourth row of Fig. 9) are visually pleasant on both detail and edge preserving, and are visually similar to the ground truth images (fifth row of Fig. 9).

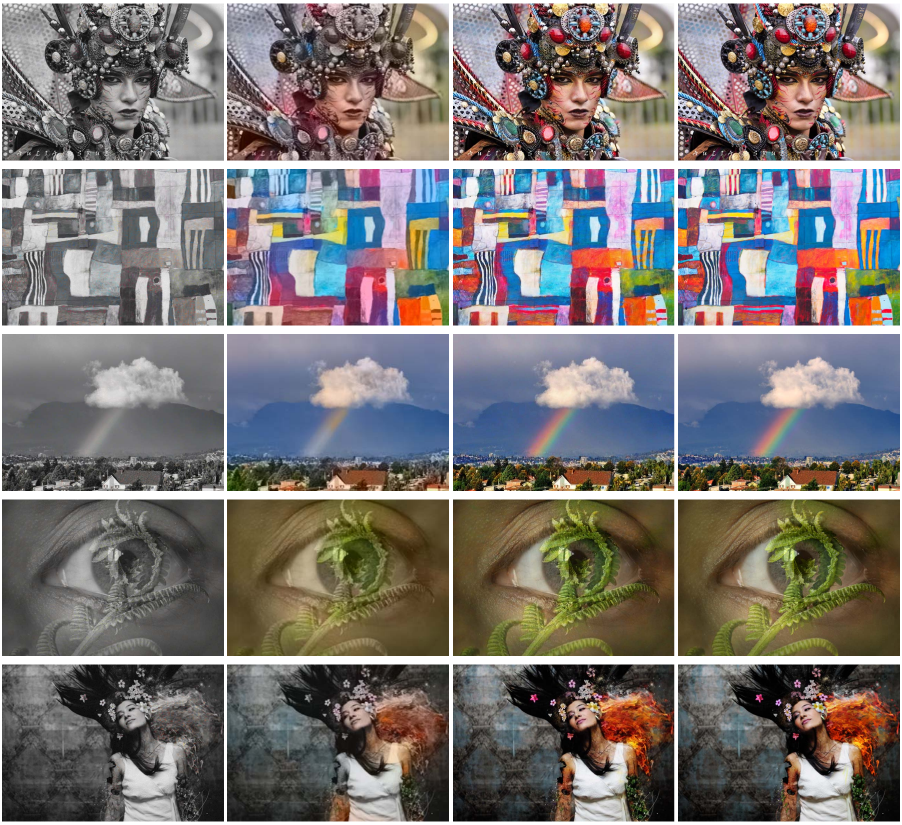


(a) (b)

Fig. 9. Pixel interpolation. First row: occluded image; Second row: EPLL based inpainting [9]; Third row: CNN based inpainting [12]; Fourth row: restored by proposed algorithm; Fifth row: the original image. Best viewed with zoom-in.

7 More Results for Color Interpolation

In this section, we show more qualitative results for color interpolation with a random 3% color pixels retained, as introduced in Section 5.3 in the paper. We compare the results of the proposed algorithm with those generated by the state-of-the-art method [13] in Fig. 10, and show more results in Fig. 11.



(a) degraded (b) Levin *et al.* (c) proposed (d) original

Fig. 10. Color interpolation with comparison to Levin *et al.* [13].

(a) degraded (b) proposed (c) original

Fig. 11. Color interpolation via proposed algorithm.

8 Re-colorization Examples

The proposed method can be extended to image re-colorization, as shown in Fig. 12. Given an input image and a reference image, the goal of re-colorization is to apply the color style of the reference image (see Fig. 12(b)) to the input image, such that the input image can be rendered with different colors without changing any content (see Fig. 12(c)). Specifically, the reference image is matched and warped to the input image by obtaining their dense pixel-correspondences through SIFT-flow [14]. 3% of the color pixels are randomly selected to be transfered from the warped reference image. The results of re-colorization, shown in Fig. 12(c), reveal the great potential for applications related to image colorization.

(a) original (b) reference (c) re-colored

Fig. 12. Re-colorization by applying the brightness channel of (a) and taking 3% color pixels from the monochrome channels in the reference images of (b). (c) shows the re-colored images with the contents of (a) and the color style of (b). Best viewed with zoom-in.

References

1. Xu, L., Ren, J.S., Yan, Q., Liao, R., Jia, J.: Deep edge-aware filters. In: ICML. (2015) 1669–1678 1, 2
2. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via l-0 gradient minimization. In: ACM TOG. Volume 30. (2011) 174 2, 6
3. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV. (1998) 839–846 2
4. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. ACM TOG **31**(6) (2012) 139 1, 2, 3, 8
5. Zhang, Q., Shen, X., Xu, L., Jia, J.: Rolling guidance filter. In: ECCV. (2014) 815–830 2, 7
6. Farbman, Z., Fattal, R., Lischinski, D., Szeliski, R.: Edge-preserving decompositions for multi-scale tone and detail manipulation. In: ACM TOG. Volume 27. (2008) 67 2, 9
7. Zhang, Q., Xu, L., Jia, J.: 100+ times faster weighted median filter (wmf). In: CVPR. (2014) 2830–2837 2
8. Osher, S., Rudin, L.I.: Feature-oriented image enhancement using shock filters. SIAM Journal on Numerical Analysis **27**(4) (1990) 919–940 2, 10
9. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: ICCV. (2011) 479–486 4, 5, 11, 12
10. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing **16**(8) (2007) 2080–2095 4
11. Ren, J.S.J., Xu, L.: On vectorization of deep convolutional neural networks for vision tasks. In: AAAI. (2015) 1840–1846 4, 5
12. SJ, R.J., Li, X., Qiong, Y., Wenxiu, S.: Shepard convolutional neural networks. In: NIPS. (2015) 901–909 11, 12
13. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. ACM TOG **23**(3) (2004) 689–694 13
14. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. IEEE PAMI **33**(5) (2011) 978–994 15