Supplemental Document for Deep Photo Style Transfer

Fujun Luan
Cornell University
fujun@cs.cornell.edu

Sylvain Paris
Adobe
sparis@adobe.com

Eli Shechtman
Adobe
elishe@adobe.com

Kavita Bala
Cornell University
kb@cs.cornell.edu

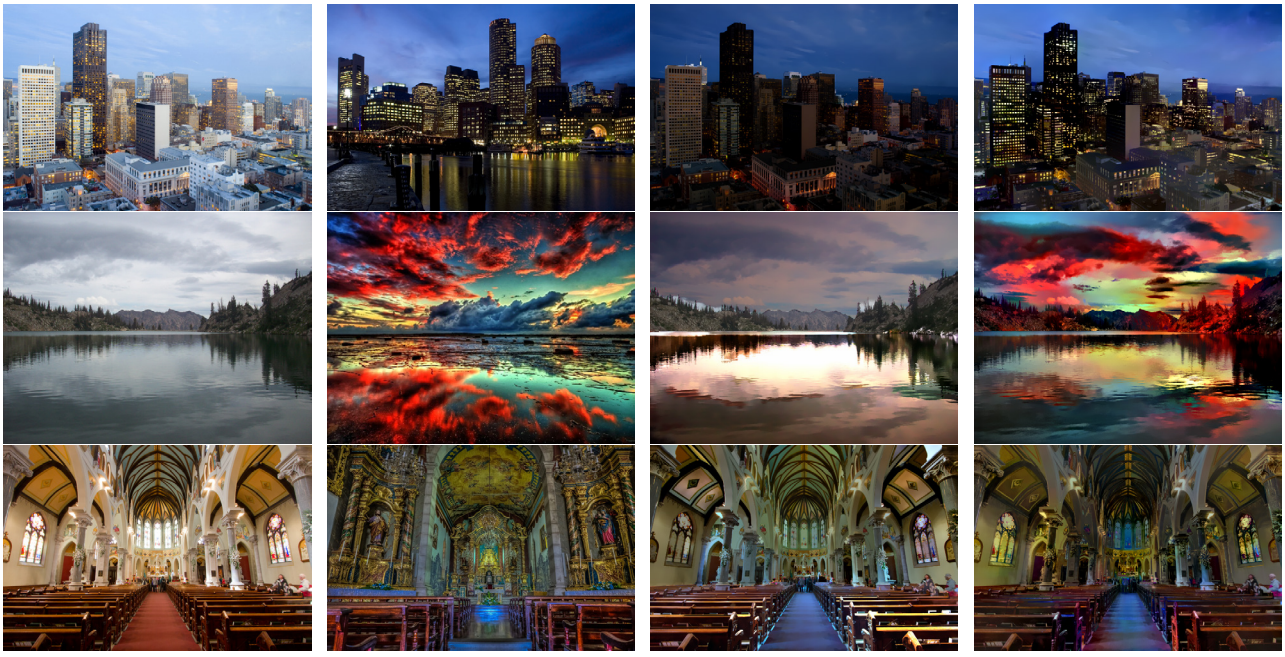
This document contains: links to two user studies (section 1), comparison against Wu et al. [6] (section 2), results with only semantic segmentation or photorealism regularization (section 3), merging classes for DilatedNet [1] Segmentation (section 4), a solution for handling noisy (section 5) or high-resolution (section 6) input, an extension for CNNMRF in photographic transfer (section 7), and some ideas we came up with, but ultimately did not work well, before reaching the matting Laplacian solution (section 8).

1. User Studies

We provide the links to our user studies that contain the interfaces as well as all the images we used:

- Photorealism evaluation: survey1, survey2;
- Style transfer comparison: survey.

2. Comparison against Wu et al. [6]



(a) Input image (b) Reference style image (c) Wu et al. [6] (d) Ours

Figure 1: Comparison of our method against Wu et al. [6]. Our method is more flexible in handling spatially-varying color changes.

3. Effects of Photorealism Regularization and Semantic Segmentation

(a) Input image (b) Reference style image (c) Without photorealism (d) Without segmentation (e) Our result

Figure 2: Transferring the style of (b) to (a). Without photorealism regularization, distortion remains in (c). Without semantic segmentation, wrong texture appears in (d). Combining the two components yields (e).

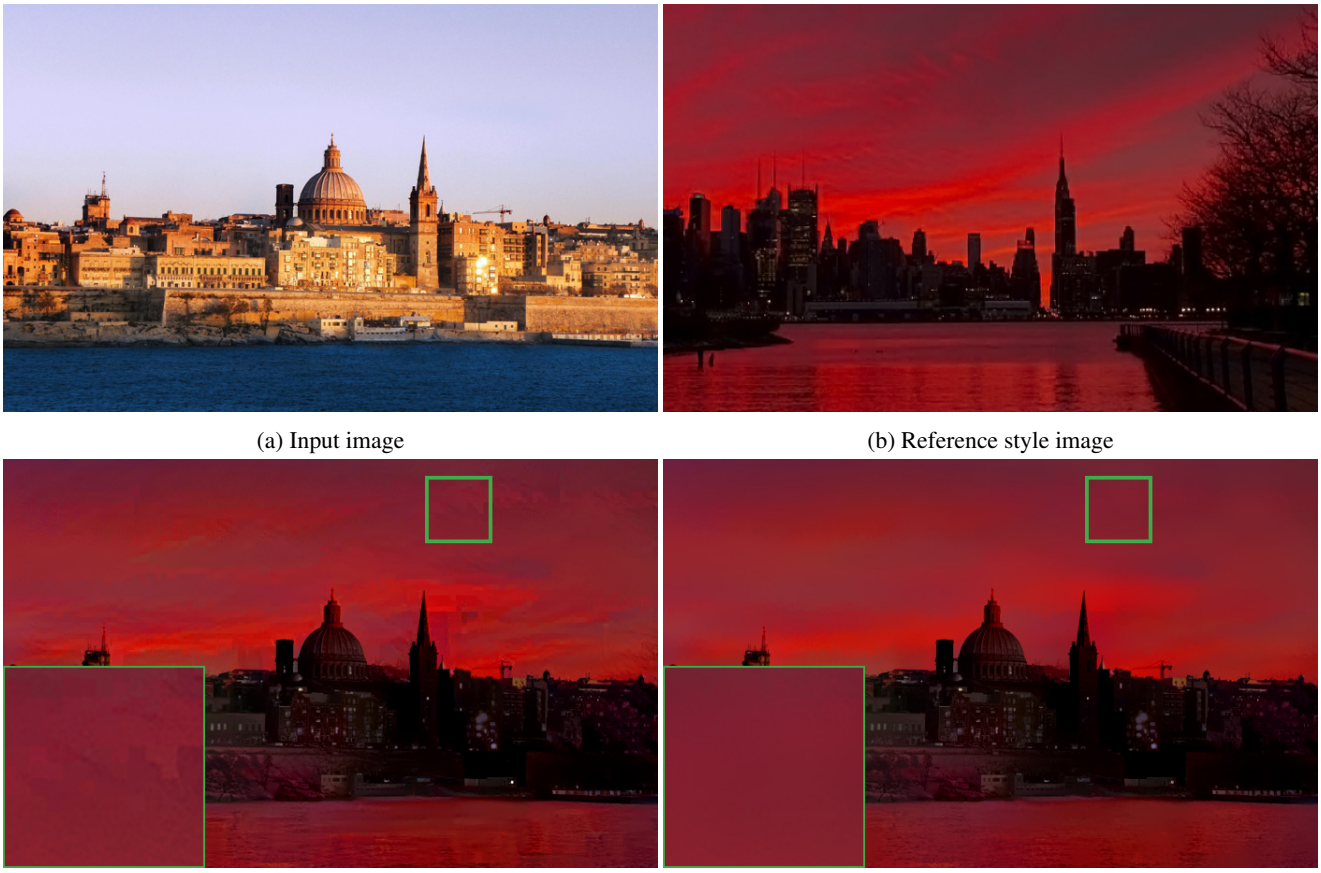
4. Merging Classes for DilatedNet [1] Segmentation

We merge similar classes to make the segmentation map simpler and cleaner. In particular, we merge the following classes:

- Water class: 'water', 'sea', 'river', 'lake';
- Tree class: 'plant', 'grass', 'tree', 'fence';
- Building class: 'house', 'building', 'skyscraper', 'wall', 'hovel', 'tower';
- Road class: 'road', 'path', 'sidewalk', 'sand', 'hill', 'earth', 'field', 'land', 'grandstand', 'stage', 'dirt track';
- Roof class: 'awning', 'ceiling';
- Mountain class: 'mountain', 'rock',
- Stair class: 'stairway', 'stairs', 'step';
- Chair class: "chair', 'seat', 'armchair', 'sofa', 'bench', 'swivel chair';
- Vehicle class: 'car', 'bus', 'truck', 'van'.

5. Handling Noisy Input

The photorealism regularizer might magnify the noise of the input image, such as JPEG compression noise or quantization noise, and lead to visible artifacts in the output by under constraining the locally affine matrices. We address this problem using a simple, yet efficient solution: after the optimization iterations are done, we fit the local regression of each patch to obtain optimal parameters for each locally affine model. Then we use a joint-bilateral filter guided by the input image to smooth out the noise across the coefficients of the locally affine matrices and reconstruct the output image free of noise. We illustrate its effect in Figure 3.



(a) Input image

(b) Reference style image

(c) Our result

(d) Our result w. noise removed

Figure 3: Transferring a dark-red style (b) onto a photo (a) at the golden hour. The JPEG noise in (a) leads to visible artifacts in our result (c). Our proposed method removes the noise and produces (d).

6. Handling High-Resolution Input

Due to the memory limitation of current GPU devices (e.g., 12GB), current style transfer algorithms that use deep convolutional neural networks such as Neural Style [2] or CNNMRF [3] can not support high-resolution images (e.g., 3500×2340). Nonetheless, we can still handle high-resolution inputs using a similar idea from 5. That is, we run the core style transfer algorithm in GPU using the small-resolution images (e.g, 700×468), and then fit the optimal locally affine models for each patch. After that we use joint-bilateral upsampling to upsample the optimal local affine models under the guidance of the high-resolution input to reconstruct a high-resolution output. We show five such high-resolution results in Figure 4, 5, 6, 7, and 8.

(a) Input image (3500 × 2340)

(b) Output image (3500 × 2340)

Figure 4: High-resolution example.

(a) Input image (3500 × 2340)

(b) Output image (3500 × 2340)

Figure 5: High-resolution example.

(a) Input image (3500×2185)

(b) Output image (3500×2185)

Figure 6: High-resolution example.

(a) Input image (3500×2625)

(b) Output image (3500×2625)

Figure 7: High-resolution example.

(a) Input image (3500 × 2330)

(b) Output image (3500 × 2330)

Figure 8: High-resolution example.

7. Photorealism Regularization for CNNMRF

In this section, we demonstrate that our photorealism regularization can be applied as a separate component, not limited to Neural Style but also other deep-learning style transfer algorithms. For instance, we can augment CNNMRF by adding the photorealism regularization layer and backpropagate gradients with its penalty (denoted as *CNNMRF+*), producing photorealistic results (Figure 9(c)).

(a) Input image

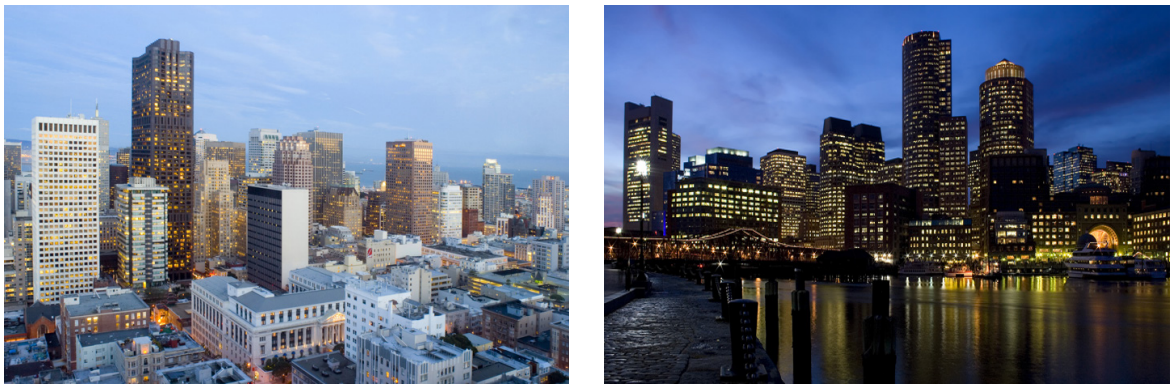
(b) Reference style image

(c) CNNMRF+

Figure 9: Our photorealism regularization can be applied as a separate component in other style transfer algorithms such as CNNMRF.

8. Ideas That Did Not Work

While our approach may seem relatively simple, it was discovered after various efforts that fared poorly. We describe several of these failed attempts to better explore our understanding of the difficulty of the problem, and how that helped us develop our final solution. The original algorithm by Gatys et al. [2] is complex and the origin of the painterly distortion is not obvious, which makes it difficult to remove. For the task in Figure 10, we will show a few reasonable ideas we came up with, but ultimately that did not work well, before reaching our final matting Laplacian solution.



(a) Input image (b) Reference style image

Figure 10: We seek to transfer the style of (b) Reference style image onto (a) Input image.

L2 norm of the color gradients. We modify the loss function as $\mathcal{L}_{\text{total}} = \sum_{l=1}^{L} \alpha_\ell \mathcal{L}_c^\ell + \Gamma \sum_{\ell=1}^{L} \beta_\ell \mathcal{L}_{s+}^\ell + \tau \mathcal{L}_g$ where

$$\mathcal{L}_g[O] = \sum_{i,j} ((O_{i,j+1} - O_{i,j})^2 + (O_{i+1,j} - O_{i,j})^2) \tag{1}$$

The results are shown in Figure 11.

Neural Style + Shih et al. [4]. Shih et al. [4] proposed a multiscale portrait style transfer algorithm. However, this algorithm is designed specifically for human portraits that utilizes a face landmark template to align faces. In order to apply this to our problem, we need to align our style image onto the input image. This is done by using the Neural Style output as the style image. The output of this algorithm is shown in Figure 12a.

Neural Style + Shih et al. [5] Shih et al. [5] proposed a locally affine transform for time-of-day hallucination. We adapt this algorithm and fit the local regression using the input image and Neural Style output (as the target image). The output of this algorithm is shown in Figure 12b.

Multiscale gain maps. We propose a solution of Neural Style + Shih et al. [4] and use a multiscale decomposition to our task. For a Laplacian pyramid with n levels we have

$$L_l[I] = \begin{cases} I - I \otimes G(2), & \text{if } l = 0 \\ I \otimes G(2^l) - I \otimes G(2^{l+1}), & \text{if } l > 0 \end{cases} \tag{2}$$

and residual

$$R[I] = I \otimes G(2^n). \tag{3}$$

We define *gain map* as $Gain_l = L_l[O]/L_l[I]$ and represent O using I and $Gain$:

$$\begin{aligned} O &= L_0[O] + L_1[O] + ... + L_{n-1}[O] + R[O] \\ &= L_0[I] \times Gain_0 + L_1[I] \times Gain_1 + ... + L_{n-1}[I] \times Gain_{n-1} + R[I] \times Gain_n \end{aligned} \tag{4}$$

(a) $\tau = 10^{-6}$ (b) $\tau = 10^{-3}$

(c) $\tau = 10^{-1}$ (d) $\tau = 10^{3}$

Figure 11: Regularizing the L2 norm of color gradients. There is no sweet spot that works. It either can't prevent distortions in (a)(b) or generates a half-transferred style in (c)(d).

(a) Neural Style + Shih et al. [4] (b) Neural Style + Shih et al. [5]

Figure 12: Using Neural Style output as the target image and applying Shih et al. [4] or Shih et al. [5] generate photorealistic outputs but the style is not faithfully transferred.

Now the unknown are a set of Gain maps: $\{Gain_l\}$. We can perform gradient descent on the Gain maps using the chain rule:

$$\frac{d\mathcal{L}_{total}}{dGain_l} = \frac{d\mathcal{L}_{total}}{dO} \times \frac{dO}{dGain_l} \tag{5}$$

where $\mathcal{L}_{total}$ is the Neural Style loss. Assuming that Gain maps are independent of each other, we have

$$\frac{dO}{dGain_l} = \begin{cases} L_l[I], & \text{if } l < n \\ R[I], & \text{if } l = n \end{cases} \tag{6}$$

and $\frac{d\mathcal{L}_{total}}{dO}$ the same as in Neural Style optimization.

We applied this idea onto Figure 10. We used $n = 5$ and obtained the final output as well as 18 Gain maps ($(n + 1) \times 3$ for 6 layers with RGB channels):

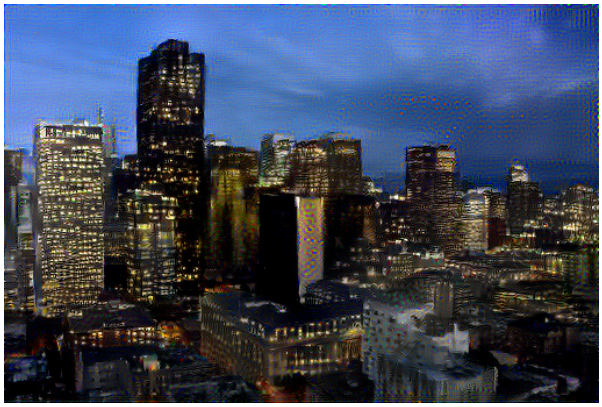


Figure 13: Result of using multiscale gain maps. Note the high frequency artifacts.

Figure 14: Gain maps. Note the maps are not smooth due to high frequency components.

Band-limited multiscale gain maps. A reasonable next step from the multiscale gain maps idea is to apply band-limited constraints onto the gain maps in the optimization. We tried several options: multiscale Gaussian post reconstruction, gradient constraint of gain maps in the optimization, Gaussian filter on gain maps in the optimization, and Sinc filtering of gain maps in the optimization.

- Multiscale Gaussian post reconstruction: Apply Gaussian filter on the gain maps and reconstruct the final output.

Figure 15: Result of multiscale Gaussian post reconstruction

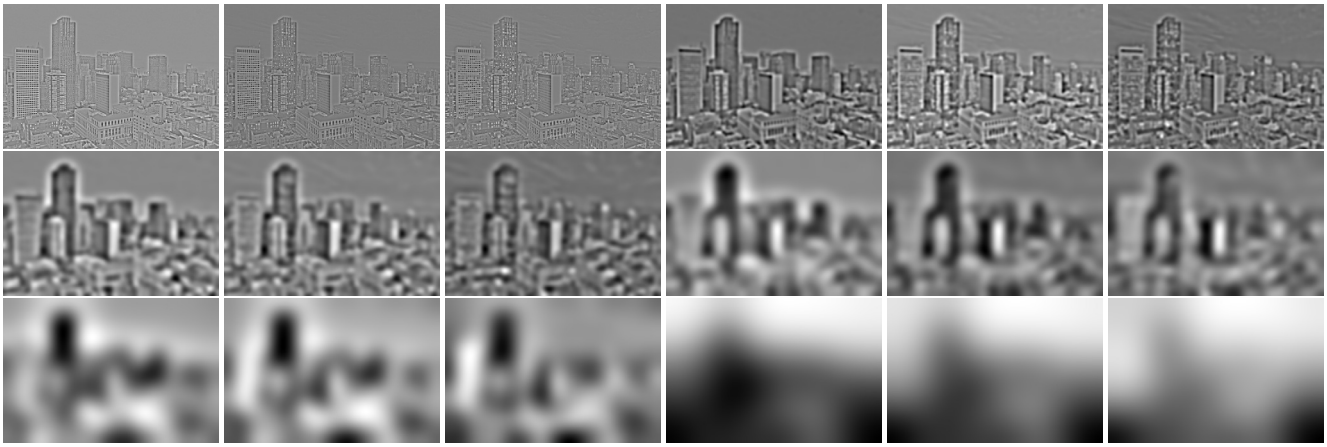
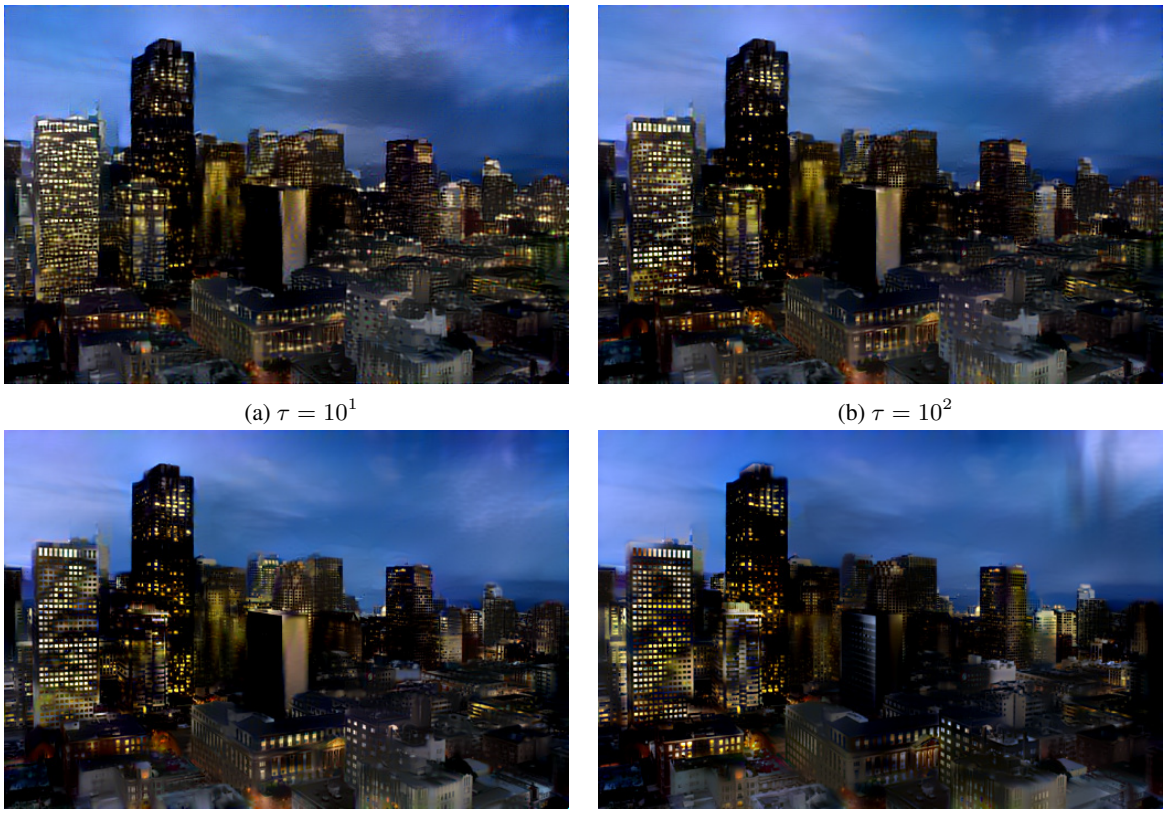


Figure 16: Gain maps w. multiscale Gaussian post filtering.

- Gradient constraint on gain maps in the optimization: Apply L2-norm gradient constraint on the gain maps and multiply a parameter τ to control the weight of it during the optimization, similar to Equation 1.



(a) $\tau = 10^1$ (b) $\tau = 10^2$

(c) $\tau = 10^3$ (d) $\tau = 10^4$

Figure 17: L2-norm gradient constraint on gain maps. Again, there is no sweet spot.

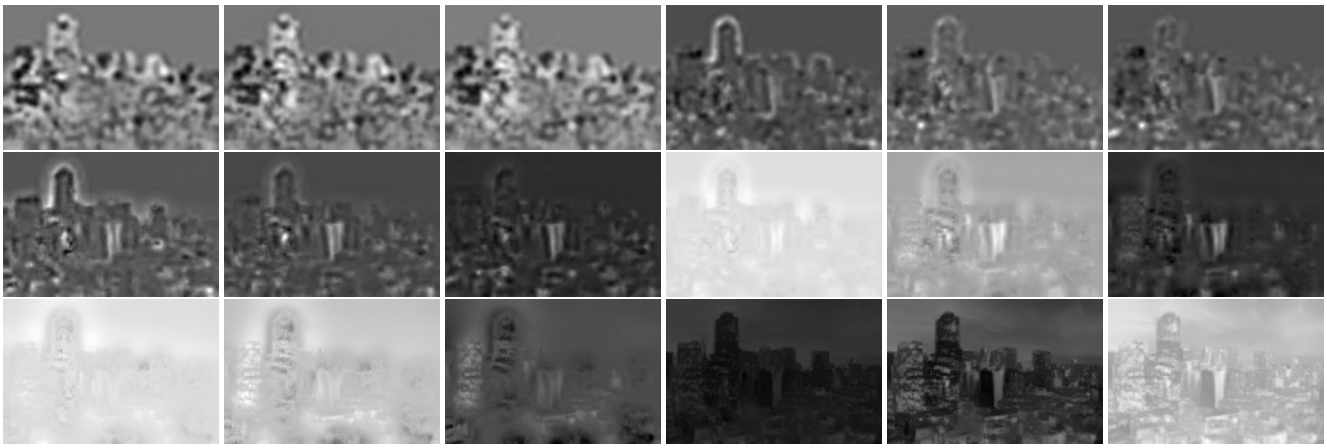


Figure 18: Gain maps using L2-norm gradient constraint on gain maps with $\tau = 10^3$.

- Gaussian filter of gain maps in the optimization: Denote the histogram-matched output image as O^h. Modify 4 as

$$O = L_0[I] \times Gaussian(Gain_0) + L_1[I] \times Gaussian(Gain_1) + ... + L_{n-1}[I] \times Gaussian(Gain_{n-1}) + R[O^h] \quad (7)$$

Figure 19: Result of Gaussian filter of gain maps in the optimization.

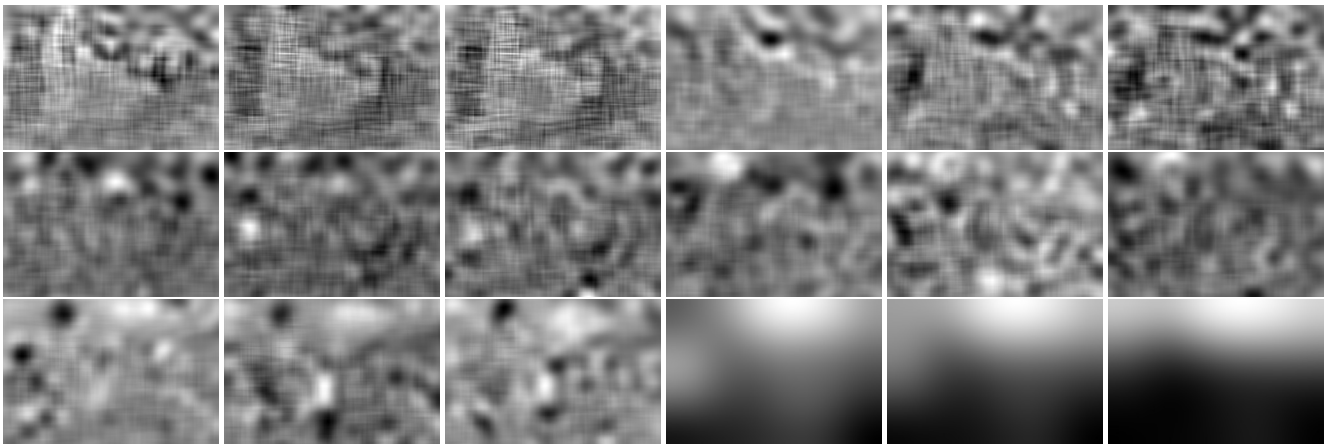


Figure 20: Gain maps w. Gaussian filter of gain maps in the optimization and residual maps of O^h (bottom-right 3 maps).

- Sinc filter of gain maps in the optimization: Denote the histogram-matched output image as O^h. Modify 4 as

$$O = L_0[I] \times Sinc(Gain_0) + L_1[I] \times Sinc(Gain_1) + ... + L_{n-1}[I] \times Sinc(Gain_{n-1}) + R[O^h] \tag{8}$$

where

$$Sinc(x) = \begin{cases} 1, & \text{if } x = 0 \\ \frac{sin(x)}{x}, & \text{otherwise} \end{cases} \tag{9}$$

Figure 21: Result of Sinc filter of gain maps in the optimization.

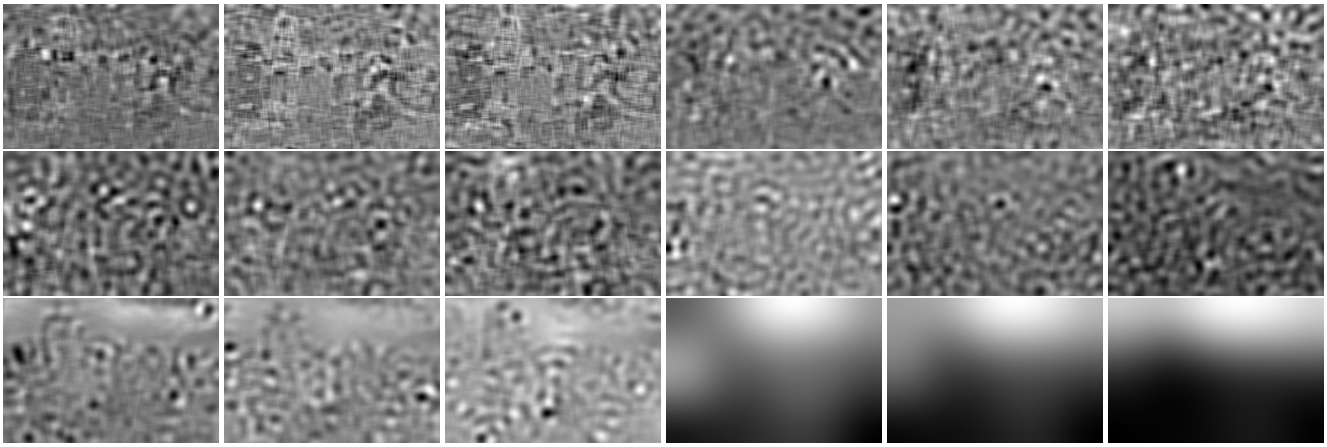


Figure 22: Gain maps w. Sinc filter of gain maps in the optimization and residual maps of O^h (bottom-right 3 maps).

Non-overlapping local affine patches in the optimization. We divide the image space into non-overlapping patches (e.g., 5×5), and represent every patch in the output image $\mathcal{P}_k[O]$ as the local affine transform of the corresponding patch in the input image $\mathcal{P}_k[I]$. For a patch containing N pixels, $\mathcal{P}_k[\cdot]$ is a $4 \times N$ matrix and each column represents the color of a pixel augmented by one as $(r, g, b, 1)^\intercal$. The local affine model for the k-th patch is a 3×4 matrix A_k.

For every patch, we represent output image patch $\mathcal{P}_k[O]$ using input image patch $\mathcal{P}_k[I]$ and a local affine model A_k:

$$\mathcal{P}_k[O] = A_k \mathcal{P}_k[I] \tag{10}$$

Now the unknown are a set of local affine matrices $\{A_k\}$. We can perform gradient descent on the affine matrices using the chain rule for every patch. However, due to the lack of constraint on neighboring patches, the result shows artifacts at the boundary of patches as shown in Figure 23.

Figure 23: Result of non-overlapping local affine optimization. Note the "square" artifacts at the boundary of patches.

We further address this problem by introducing jointly-bilateral smoothing on the local affine matrices as a post-processing step and reconstruct the output, as shown in Figure 24.

Figure 24: Result of non-overlapping local affine optimization w. post jointly-bilateral smoothing on local affine matrices. Note the lights on the left building are off.

Final solution. Instead of converting the unknowns to be a set of local affine matrices, we still use gradient descent to update the RGB pixels of the output image by introducing the overlapping locally-affine photorealism energy term into the Neural Style optimization. We find it both preserves the photorealism and matches the style of the reference image well.

Figure 25: Our final solution.

References

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 1, 2

[2] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 3, 10

[3] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. *arXiv preprint arXiv:1601.04589*, 2016. 3

[4] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand. Style transfer for headshot portraits. 2014. 10, 11

[5] Y. Shih, S. Paris, F. Durand, and W. T. Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG)*, 32(6):200, 2013. 10, 11

[6] F. Wu, W. Dong, Y. Kong, X. Mei, J.-C. Paul, and X. Zhang. Content-based colour transfer. In *Computer Graphics Forum*, volume 32, pages 190–203. Wiley Online Library, 2013. 1