Fast Local Laplacian Filters: Theory and Applications

Mathieu Aubry, Sylvain Paris, Samuel Hasinoff, Jan Kautz, Frédo Durand

Fast Local Laplacian Filters: Theory and Applications

MATHIEU AUBRY
INRIA / ENS
and
SYLVAIN PARIS
Adobe
and
SAMUEL W. HASINOFF
Google Inc.
and
JAN KAUTZ
University College London
and
FRÉDO DURAND
Massachusetts Institute of Technology

Multi-scale manipulations are central to image editing but they are also prone to halos. Achieving artifact-free results requires sophisticated edge-aware techniques and careful parameter tuning. These shortcomings were recently addressed by the local Laplacian filters, which can achieve a broad range of effects using standard Laplacian pyramids. However, these filters are slow to evaluate and their relationship to other approaches is unclear. In this paper, we show that they are closely related to anisotropic diffusion and to bilateral filtering. Our study also leads to a variant of the bilateral filter that produces cleaner edges while retaining its speed. Building upon this result, we describe an acceleration scheme for local Laplacian filters on gray-scale images that yields speed-ups on the order of 50×. Finally, we demonstrate how to use local Laplacian filters to alter the distribution of gradients in an image. We illustrate this property with a robust algorithm for photographic style transfer.

Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation—*Photo Editing*

General Terms: Photo editing, Image processing

Additional Key Words and Phrases: Computational photography, photo editing, image processing, Laplacian pyramid, bilateral filter, photographic style transfer

ACM Reference Format:

Aubry, M., Paris, S., Hasinoff, S. W., Kautz, J., Durand, F. Fast and Robust Pyramid-based Image Processing.

1. INTRODUCTION

Manipulating images at multiple scales is a challenging task. Direct linear manipulation of the frequency bands yields unsightly halos. This issue has been addressed by many nonlinear approaches. While these techniques produce halo-free images, they also come with their own shortcomings such as limited scalability due to the need of solving a global optimization problem, e.g. [Fattal et al. 2002; Farbman et al. 2008; Subr et al. 2009; Xu et al. 2011], or edge defects that require corrections in post-process [Durand and Dorsey 2002; Bae et al. 2006; Kass and Solomon 2010]. Recently, Paris et al. [2011] described the *local Laplacian filters* that address these shortcomings and produce high-quality results over a wide range of parameters. However, while these filters achieve similar effects to existing edge-aware filters, their relationship to other approaches is unclear. Further, these filters are prohibitively slow in their original form. Paris and colleagues [2011] mitigate this issue with a heuristic approximation but its properties and accuracy are unknown, and even so, it remains slow.

In this paper, we study these filters to gain a better understanding of their behavior. First, we rewrite them as the averaging at each scale of the signal variations in the local neighborhood around each pixel. From this formulation, we show that local Laplacian filters can be interpreted as a multi-scale version of anisotropic diffusion, and that they are closely related to bilateral filtering, the main difference being their multi-scale nature and how they are normalized. While the difference is minor in uniform regions, it becomes large in configurations such as edges, corners, and isolated pixels where the bilateral filter is known not to perform well [Durand and Dorsey 2002; Buades et al. 2006]. We use this insight to design a variant of the bilateral filter, which we name *unnormalized bilateral filtering*, and we show that it yields significantly cleaner edges. We also propose a signal-processing interpretation of local Laplacian filtering applied to gray-scale images and derive a new acceleration scheme grounded on sampling theory. Our analysis shows that we can quantize the intensity scale while introducing only negligible differences with the original scheme. Our tests show that, on gray-scale images, our algorithm is about 50 times faster than the heuristics of Paris et al. and that it runs at interactive rates on CPUs without resorting to parallelism. Further, our GPU implementation processes one-megapixel images at 20 Hz. Finally, we show how to use these filters to alter the gradient distribution of an image without introducing halos. We illustrate this approach in the context of photographic style transfer (Fig. 1). Our experiments show that our method achieves satisfying transfers on a larger set of photos than previous work.

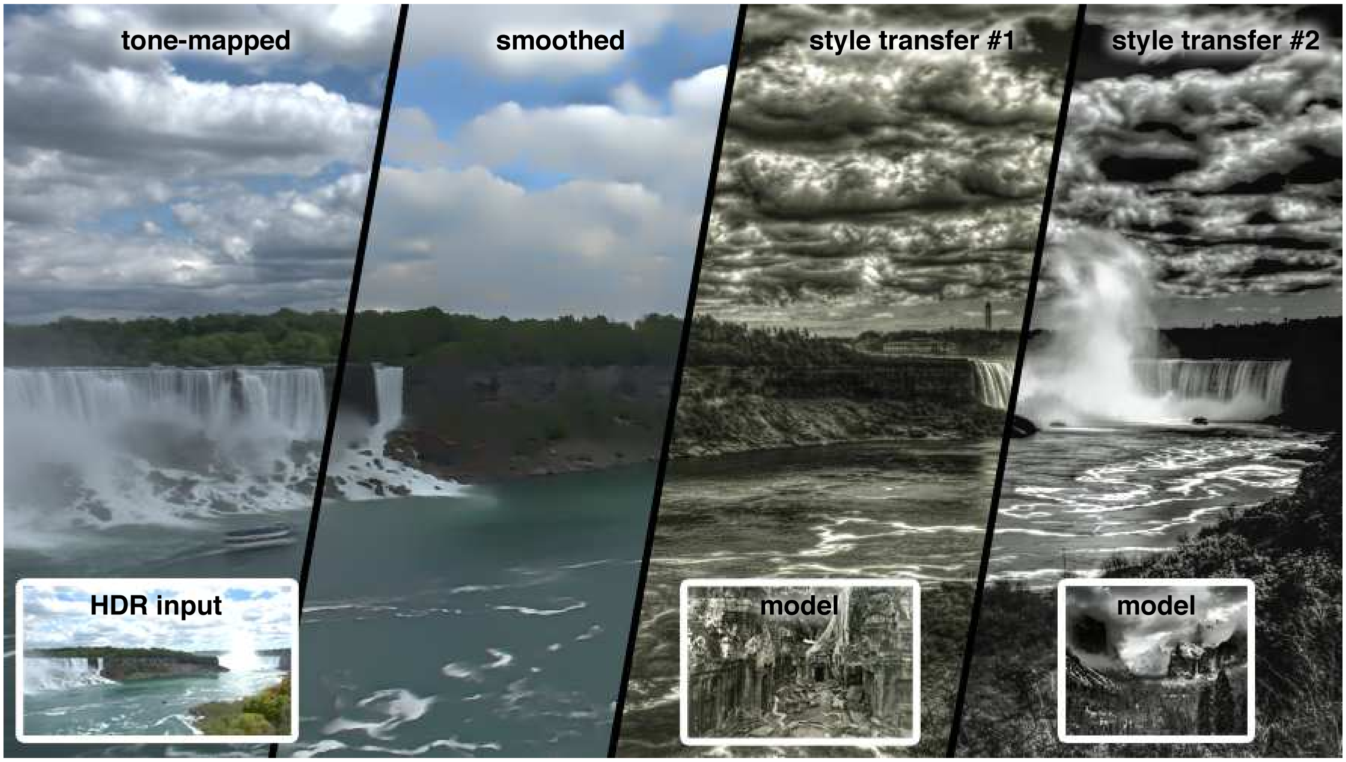


Fig. 1. In this paper, we show that the local Laplacian filter [Paris et al. 2011] is related to anisotropic diffusion and that it can be also understood as a multiscale variant of bilateral filtering. This enables us to derive an efficient algorithm that is 50× faster, processing a one-megapixel grayscale image in 350 ms on a single core and achieving, for instance, interactive tone-mapping (left) and detail reduction (second left). Our analysis also leads to a better understanding of local Laplacian filtering, which allows us to modify gradient distributions without artifacts, yielding robust and efficient style transfer (two examples on right, models inset at the bottom).

We make the following contributions.

—We formally characterize the similarities between local Laplacian filters, anisotropic diffusion, and bilateral filtering. In contrast, Paris et al. presented those filters independently of other existing methods.

—Based on this understanding, we build a new single-scale filter, the unnormalized bilateral filter, that behaves similarly to the bilateral filter in smooth areas while producing cleaner edges.

—We describe a fast algorithm for Laplacian filtering on gray-scale images that is about 50 times faster than the heuristic of Paris et al. and that is guaranteed to approximate the original scheme.

—We explain how to transfer the gradient histogram from one image to another using Laplacian filtering and demonstrate its application to photographic style transfer.

1.1 Related work

Local Laplacian filters are part of the edge-preserving filter category. While some of these filters were initially proposed for denoising [Aubert and Kornprobst 2002], they have now been superseded by texture-preserving filters such as non-local means [Buades et al. 2005b] and BM3D [Dabov et al. 2006]. In this paper, we focus on photo editing applications for which local Laplacian filters and other edge-preserving filters are most useful.

Relationship between nonlinear filters. A dense net of studies relate methods as diverse as bilateral filtering, anisotropic diffusion, mean shift, neighborhood filtering, mode filtering, and robust statistics, e.g. [Black et al. 1998; Durand and Dorsey 2002; Elad 2002; van de Weijer and van den Boomgaard 2002; Barash and Comaniciu 2004; Buades et al. 2005a; Mrzek et al. 2006; Paris et al. 2009]. However, the recently proposed local Laplacian filters are not yet part of this mesh and little is known about their relationship to the existing body of work on nonlinear image filtering. A contribution of our paper is to show that they are closely related to anisotropic diffusion and to the bilateral filter.

Bilateral filtering. The bilateral filter [Tomasi and Manduchi 1998] is a popular edge-aware smoothing filter for computational photography applications, e.g. [Durand and Dorsey 2002; Bae et al. 2006; Paris et al. 2009], because it achieves satisfying results while being fast [Chen et al. 2007; Paris and Durand 2009; Adams et al. 2009; Adams et al. 2010; Gastal and Oliveira 2012]. However, it is also known to suffer from over-sharpening, which introduces unsightly edge defects [Buades et al. 2006] and requires applying a fix in post-processing, e.g. [Durand and Dorsey 2002; Bae et al. 2006; Kass and Solomon 2010]. This additional step requires more computation and introduces new parameters to set. In comparison, our variant modifies the original bilateral filter in minor ways that preserve its speed and ease of use, while significantly reducing over-sharpening.

Local Laplacian filtering. Paris et al. [2011] introduced local Laplacian filtering as an alternative to existing edge-aware filters. They demonstrated that these filters generate high-quality results for detail manipulation and tone mapping for a wide range of parameters. In particular, they showed that these filters produce strong detail enhancement whereas existing techniques suffer from halos [Li et al. 2005; Farbman et al. 2008] or aliasing artifacts [Fattal 2009]. Figure 3 shows that, for such large detail enhancement, the local Laplacian filters also compare favorably to recent filters such as the Guided Filter [He et al. 2010] and Domain Transform [Gastal and Oliveira 2011] in terms of visual quality. However, the running times of the local Laplacian filters are slow, on the order of a minute per megapixel with a single thread, which requires a parallel implementation and an approximation scheme to reach interactive rates. Whereas the effects of this approximation are unclear, we propose an acceleration technique firmly grounded on signal processing analysis [Chen et al. 2007; Paris and Durand 2009], which allows us to control the trade-off between speed and accuracy. And as previously discussed, we also describe a theoretical relationship between local Laplacian filters, anisotropic diffusion, and bilateral filtering. Since local Laplacian filters are at the core of our work, we describe them in more detail in Section 1.2.

Photographic style transfer. Bae et al. [2006] transfer the "look" of one photographer's masterpiece onto another photo by matching statistics such as the intensity and texture histograms of the two pictures. While they demonstrate convincing results, the method consists of many steps, including solving the Poisson equation several times, which limits the ability of the approach to process high-resolution images and makes the technique difficult to implement and tune. Sunkavalli et al. [2010] propose a simpler alternative based on image pyramids but their results do not match the look of the model photograph as well. In this paper, we demonstrate that Laplacian filtering can be used for manipulating the gradient histogram of an image. Our approach generates visual matches in the same spirit as Bae's technique. For strongly stylized examples, it often performs better, because the robustness of local Laplacian filters allows for larger image modifications.

1.2 Background on local Laplacian filters

We now summarize how local Laplacian filters are defined and computed. We use the notation shown in Table 1.2.

Local Laplacian filters are edge-aware operators that define the output image O by constructing its Laplacian pyramid $\{L[O]\}$ coefficient by coefficient. The computation of each coefficient is independent of the others. To estimate $L_\ell[O](x,y)$, the Laplacian coefficient at level ℓ and position (x,y), one first applies a simple pixel-wise filter to the input image, then computes a Laplacian pyramid of this transformed image, and finally uses the (ℓ,x,y) coefficient in that pyramid as the value of the output coefficient $L_\ell[O](x,y)$. We now detail each of these three steps. First, the input image I is processed with a point-wise nonlinearity $r(\cdot)$ that depends on $g = G_\ell[I](x,y)$, the coefficient of the Gaussian pyramid at level ℓ and position (x,y). Intuitively, $r(I)$ is an image that looks like the desired result where the intensity I is close to g. For instance, to increase the amount of detail, we apply a local S-shaped tone curve centered on g which makes I values close to g farther away from it, and leaves more distant values unchanged (Fig. 2). By combining the results from various g values, we affect the entire intensity range and obtain the final output. We will discuss the formal definition of r later. Then, given $r(I)$ for a particular g value corresponding to the position (x,y) and scale ℓ, one builds the Laplacian pyramid of that transformed image, that is $\{L[r(I)]\}$. Finally, we use the coefficient $L_\ell[r(I)](x,y)$ in that pyramid as the value of the output coefficient $L_\ell[O](x,y)$. We repeat this process for all the coefficients of the output pyramid.

Variable	Description	Variable	Description
$\mathbf{p} = (x,y)$	spatial location	$L_\ell[I]$	level ℓ of pyramid
ℓ	level in pyramid	$\{L[I]\}$	Laplacian pyramid
I	input image	g	Gaussian coefficient
O	output image	G_σ	Gaussian kernel
$r(i)$	remapping func.		

Table 1.2: Common notation used in this paper.

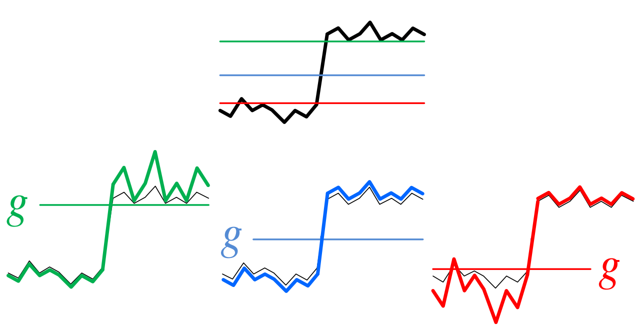


Fig. 2. Effect of a detail enhancement ($0 \le \alpha < 1$) remapping function r with several reference values g near an edge. Details are enhanced for values similar to the reference value, but not for values far from it.

A direct application of this approach yields an $\mathcal{O}(N^2)$ algorithm where N is the number of image pixels, but since only a portion of the image needs to be processed to obtain $L_\ell[r(I)](x,y)$ at a given (x,y), this complexity can be reduced to $\mathcal{O}(N \log N)$. Paris et al. [2011] further accelerate the process and obtain an $\mathcal{O}(N)$ method by using heuristic that amounts to applying r to a downsampled version of I when processing coarse pyramid levels. While satisfying results are obtained in practice, there is no clear understanding of the speed-versus-accuracy trade-off inherent in this heuristic and a multi-core implementation is still required to achieve interactive running times. A contribution of our work is to propose a novel acceleration scheme that is faster and grounded on signal processing theory. For the nonlinearity r, the original article focuses only on a small set of options defined by three parameters. While these are sufficient for detail manipulation and tone mapping, in this paper, we reinterpret r in terms of first-order image statistics and explore more general designs in the context of style transfer.

Design of the remapping function. Paris and colleagues demonstrated the local Laplacian filters for tone mapping and detail manipulation. In this context, they proposed the following remapping functions to compute the coefficient (ℓ,x,y):

$$\tilde{r}(i) = \begin{cases} g + \operatorname{sign}(i-g)\,\sigma_\mathrm{r}\,(|i-g|/\sigma_\mathrm{r})^\alpha & \text{if } i \le \sigma_\mathrm{r} \\ g + \operatorname{sign}(i-g)\big(\beta(|i-g|-\sigma_\mathrm{r})+\sigma_\mathrm{r}\big) & \text{if } i > \sigma_\mathrm{r} \end{cases} \quad (1)$$

where g is the coefficient of the Gaussian pyramid at (ℓ,x,y), which acts as a reference value, α controls the amount of detail increase ($0 \le \alpha < 1$) or decrease ($\alpha > 1$), β controls the dynamic range compression ($0 \le \beta < 1$) or expansion ($\beta > 1$), and σ_r defines the intensity threshold the separates details from edges. For the sake of clarity, we omit these parameters in the notation and use the concise form $\tilde{r}(i)$. Sample functions are plotted in Figure 4.

2. BILATERAL FILTERING, ANISOTROPIC DIFFUSION, AND LOCAL LAPLACIAN FILTERS

We now study the local Laplacian filter and relate it to anisotropic diffusion and bilateral filtering. We start by formalizing this relationship and then define a new variant of the bilateral filter inspired by this result.

Background. With our notation, anisotropic diffusion as introduced by Perona and Malik [1990] is defined by a partial differential equation:

$$\frac{\partial I}{\partial t} = \operatorname{div}\big(w(\nabla I)\ \nabla I\big) \tag{2}$$

where t represents the diffusion time, i.e., how long the process has been run, and w is a weighting function that is equal to 1 for $\nabla I = \mathbf{0}$ and decreases for larger gradients. This equation is discretized as an iterative process:

$$I_{t+1}(\mathbf{p}) \ = \ I_t(\mathbf{p}) \ + \sum_{\mathbf{q}\in\mathcal{N}_4(\mathbf{p})} w\big(I_t(\mathbf{q}) - I_t(\mathbf{p})\big)\ \big[I_t(\mathbf{q}) - I_t(\mathbf{p})\big] \tag{3}$$

where t now counts how many iterations have been performed, and $\mathcal{N}_4$ is the 4-neighborhood of $\mathbf{p}$. The process is initialized at $t = 0$ with I being the input image.

Using $I_\mathbf{p}$ as a shorthand for $I(\mathbf{p})$, the bilateral filter is defined as [Tomasi and Manduchi 1998]:

$$BF_\mathbf{p} = \frac{1}{W_\mathbf{p}} \sum_\mathbf{q} G_{\sigma_\mathrm{s}}(\mathbf{q}-\mathbf{p})\ G_{\sigma_\mathrm{r}}(I_\mathbf{q} - I_\mathbf{p})\ I_\mathbf{q} \tag{4a}$$

$$W_\mathbf{p} = \sum_\mathbf{q} G_{\sigma_\mathrm{s}}(\mathbf{q}-\mathbf{p})\ G_{\sigma_\mathrm{r}}(I_\mathbf{q} - I_\mathbf{p})\ , \tag{4b}$$

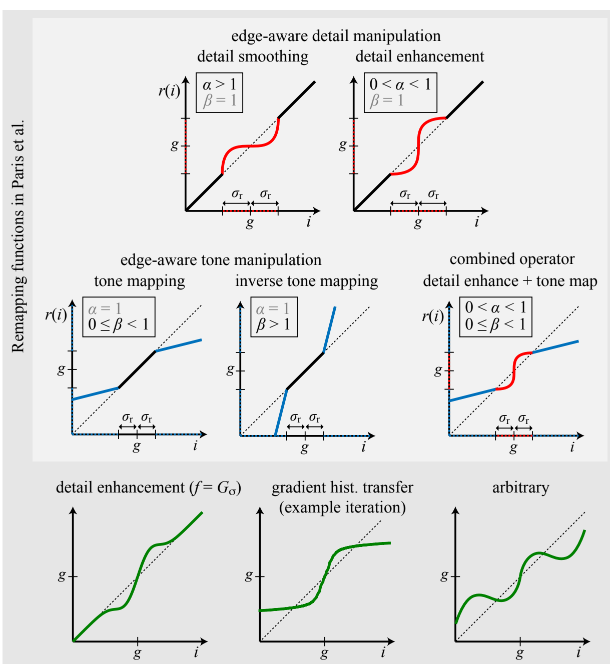


Fig. 4. Remapping functions proposed by Paris et al. [2011] (reproduced from [Paris et al. 2011]). Our paper considers a wider class of functions that includes all these as well as others, a few of which are shown at the bottom of the figure.

where G_{σ_r} and G_{σ_s} are Gaussian kernels of variance σ_r^2 and σ_s^2 defined by $G_\sigma(x) = \exp(-x^2/2\sigma^2)$ that are called the *range weight* and *space weight* respectively. Formally, the sums $\sum_\mathbf{q}$ cover the entire image but in practice are limited to local windows of radius $3\sigma_\mathrm{s}$ since G_{σ_s} becomes almost zero for distant pixels. Using the symmetry of the Gaussian kernel and the fact that the weights sum up to 1, Equation 4a can be rewritten to make bilateral filtering appear as a multi-scale version of anisotropic diffusion as described in Equation 2 with $w = G_{\sigma_\mathrm{r}}$ [Elad 2002; Barash and Comaniciu 2004]:

$$BF_\mathbf{p} = I_\mathbf{p} + \frac{1}{W_\mathbf{p}} \sum_{d>0} G_{\sigma_\mathrm{s}}(d) \sum_{\substack{\mathbf{q}\ \text{s.t.} \\ \|\mathbf{q}-\mathbf{p}\|=d}} G_{\sigma_\mathrm{r}}(I_\mathbf{q} - I_\mathbf{p})\ (I_\mathbf{q} - I_\mathbf{p}) \tag{5}$$

As we shall see later, it is also convenient to rewrite Equation 4a as:

$$BF_\mathbf{p} = I_\mathbf{p} + \frac{1}{W_\mathbf{p}} \sum_\mathbf{q} G_{\sigma_\mathrm{s}}(\mathbf{q}-\mathbf{p})\ G_{\sigma_\mathrm{r}}(I_\mathbf{q} - I_\mathbf{p})\ (I_\mathbf{q} - I_\mathbf{p})\ . \tag{6}$$

2.1 Analysis of local Laplacian filters

In this section, we relate local Laplacian filtering to anisotropic diffusion and bilateral filtering. We show that the Gaussian kernel used to build the image pyramids acts as a spatial weight and that the remapping function r as a range weight, thereby resembling bilateral filtering. The main differences with standard bilateral filtering are the absence of normalization factor and the multiscale aspect. We then further decompose the spatial kernel into rings of fixed radius to make appear a link with anisotropic diffusion akin to the studies by Elad [2002] and Barash and Comaniciu [2004]. We show that local Laplacian filters can be interpreted as a multi-scale diffusion process controlled by a specific set of parameters.

For this study, we consider the space of remapping functions with the form

$$r(i) = i - (i-g)\ f(i-g)\ , \tag{7}$$

where f is a continuous function. This space includes the functions of Paris et al. as a special case when $f(i) = (i - \tilde{r}(i))/(i-g)$ where $\tilde{r}$ is the remapping function defined in Equation 1.

2.1.1 *Single-scale filter.* We first consider a two-level pyramid, that is, we seek to compute the levels $L_0[O]$ and $L_1[O]$ of the Laplacian pyramid of the output image O. We assume for now that the residual remains unprocessed, that is $L_1[O] = L_1[I]$. For a pixel $\mathbf{p} = (x,y)$ on the 0th level we have:

$$L_0[O](\mathbf{p}) = r(I_\mathbf{p}) - \big[\bar{G}_{\sigma_\mathrm{p}} * r(I)\big](\mathbf{p})\ , \tag{8}$$

where $\bar{G}_{\sigma_\mathrm{p}} = \frac{1}{\sqrt{2\pi\sigma_\mathrm{p}^2}} G_{\sigma_\mathrm{p}}$ is a normalized Gaussian kernel of variance σ_p^2 used to build the pyramids, and $*$ is the convolution operator. Expanding r and using $L_0[I] = I - \bar{G}_{\sigma_\mathrm{p}} * I$ and $g = I_\mathbf{p}$ since we are at the finest level of the pyramid, we obtain:

$$L_0[O](\mathbf{p}) = L_0[I](\mathbf{p}) + \big[\bar{G}_{\sigma_\mathrm{p}} * (I - I_\mathbf{p})\ f(I - I_\mathbf{p})\big](\mathbf{p})\ . \tag{9}$$

Then, by upsampling the unmodified residual, adding it to both sides, and expanding the convolution, we get the formula that we seek:

$$O_\mathbf{p} = I_\mathbf{p} + \sum_\mathbf{q} \bar{G}_{\sigma_\mathrm{p}}(\mathbf{q}-\mathbf{p})\ f(I_\mathbf{q} - I_\mathbf{p})\ (I_\mathbf{q} - I_\mathbf{p}) \tag{10}$$

This equation shows that the two-level local Laplacian filter computes a local average in the same spirit as the bilateral filter (Eq. 6) using $\bar{G}_{\sigma_\mathrm{p}}$ as the spatial weight and f as the range weight. If we

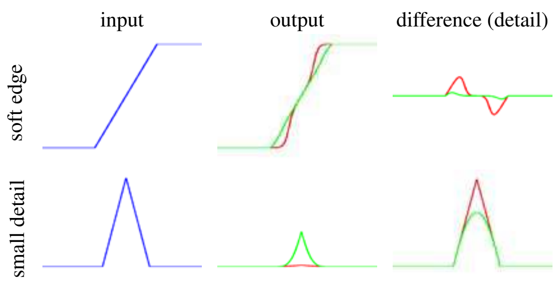


Fig. 5. We compare how the standard bilateral filter (red) and our unnormalized variant (green) process simple features. The top row shows a soft edge (left). The standard filter over-sharpens the edge (center) and introduces non-negligible content in the detail layer (right). These spurious variations are a source of halos and gradient reversals in applications such as tone mapping. In comparison, the unnormalized filter produces a cleaner edge with almost no spurious content in the detail layer. The bottom row shows an isolated small detail (left), with a different vertical scale, whose height is 10% of the edge amplitude. The standard bilateral filter smooths such details more aggressively than the unnormalized version. This explains why the tone-mapping results of the unnormalized version tend to be softer, but this is a minor side effect compared to the creation of edge artifacts. Both filters were set with σ_r equal to 10% of the edge amplitude.

choose $f = G_{\sigma_r}$, then the two-level local Laplacian filter becomes almost the same as the bilateral filter—the only difference is that the weights are *not* normalized by $\frac{1}{W_\mathbf{p}}$. This simple modification defines a new filter we call the *unnormalized bilateral filter* which we examine in Section 2.2.

This parallel with the bilateral filter also suggests that one may achieve cross filtering akin to Eiseman et al. [2004] and Petschnigg et al. [2004] by replacing the input image i by a guiding image $\hat{\imath}$ in the f function in Equation 7, that is, $r(i,\hat{\imath}) = i - (i-g)f(\hat{\imath}-g)$. We leave the exploration of this option as future work.

Equation 10 can also be written as follows using the symmetry of the Gaussian kernel [Elad 2002; Barash and Comaniciu 2004]:

$$O_\mathbf{p} = I_\mathbf{p} + \sum_{d>0} \bar{G}_{\sigma_\mathrm{p}}(d) \sum_{\substack{\mathbf{q} \text{ s.t.} \\ \|\mathbf{q}-\mathbf{p}\|=d}} f(I_\mathbf{q} - I_\mathbf{p})\,(I_\mathbf{q} - I_\mathbf{p}) \tag{11}$$

This formulation shows that, similar to the bilateral filter (Eq. 6), a two-scale local Laplacian filter can be seen as a multiscale version of anisotropic diffusion as described in Equation 2 [Perona and Malik 1990]. The main difference between bilateral filtering and Laplacian filtering is how each scale is weighted. The bilateral filter uses weights that sum to 1 because of the $1/W_\mathbf{p}$ normalization factor whereas the local Laplacian filters apply unnormalized weights that do not sum to 1.

2.1.2 *Multi-scale filter.* In the case of more than two levels, g is not equal to $I_\mathbf{p}$, and we cannot collapse the pyramid as above. Nonetheless, we can write:

$$L_\ell[O] = D_\ell * (I-g)\, f(I-g)\ , \tag{12}$$

where $D_\ell = \bar{G}_{2^{\ell-1}\sigma_\mathrm{p}} - \bar{G}_{2^\ell\sigma_\mathrm{p}}$ is the difference-of-Gaussians filter defining the pyramid coefficients at level ℓ. This expression can be rewritten as

$$L_\ell[O](\mathbf{p}) = \sum_\mathbf{q} D_\ell(\mathbf{q}-\mathbf{p})\, f(I_\mathbf{q} - g)\,(I_\mathbf{q} - g)\ . \tag{13}$$

This shows that each level of the output pyramid is a local average of differences over a neighborhood of $\mathbf{p}$. Similarly to Equation 11, the above formula can be factored into rings to make appear a link with anisotropic diffusion. The above formula will also be useful in our design of an acceleration scheme in Section 3.

Discussion. Previous work has described multi-scale filters based on the bilateral filter, e.g. [Fattal et al. 2007; Fattal 2009], and anisotropic diffusion, e.g. [Zhong and Sun 2008]. Our study shows that local Laplacian filters belong to the same class of filters. From a practical perspective, the multi-scale filter of Zhong et al. [2008] aims for image denoising and it is unclear how it would behave on detail manipulation and tone mapping. In comparison, bilateral pyramids have been developed for these applications. However, these methods tend to suffer from halos (Fig. 3 and [Farbman et al. 2008, Fig. 10]). In comparison, local Laplacian filters are less prone to this issue although this comes at the cost of slower running times [Paris et al. 2011]. In practice, the choice of filter depends on the application and one's priorities. Further, the local Laplacian filters are not restricted to $f = G_{\sigma_r}$ and offer additional flexibility through the design of the f function. We exploit this property in Section 4.1 to achieve gradient histogram transfer.

As we have seen earlier, the bilateral filter can also be seen as a multi-scale variant of anisotropic diffusion (Eq. 5). The difference between the two filters is how they exploit scale information in the weighting function. Bilateral filtering weights all the pixels relatively to the center pixel $I_\mathbf{p}$, i.e., relatively to the finest image scale only. In comparison, the local Laplacian filters adapt the reference pixel depending on the scale (Eq. 13), effectively defining a weighting scheme for each scale. Collapsing the output pyramid sums all the filtered coefficients and creates a sophisticated combination that mixes image information at all scales.

2.1.3 *Recap.* In this section, we have shown that, when applied to a two-level pyramid with $f = G_{\sigma_r}$, the local Laplacian filters can be expressed in forms closely related to bilateral filtering (Eq. 10) and anisotropic diffusion (Eq. 11). By doing so, we also highlighted a few key differences. Compared to bilateral filtering that is a normalized sum, local Laplacian filters are not normalized. We further explore this point in the next section and show that it helps produce clean results at edges. Compared to anisotropic diffusion, even when restricted to a two-level pyramid, local Laplacian filters consider a neighborhood defined by $\bar{G}_{\sigma_\mathrm{p}}$ that is larger than the 4 adjacent pixels in $\mathcal{N}_4$.

2.2 Unnormalized bilateral filtering

In the previous analysis, we showed that local Laplacian filters share similarities with bilateral filtering. But they are not identical; Laplacian filters are multiscale and do not normalize the contribution of the pixels, whereas the bilateral filter is two-scale and normalized. This suggests a variant of the bilateral filter where we remove the overall normalization but keep the two-scale design. We call it the *unnormalized bilateral filter*. As we shall see, this filter produces better results than the standard bilateral filter without being as good as the local Laplacian filters. But since the running times are shorter, this filter can be useful when speed is important.

Formally, we define the unnormalized bilateral filter as:

$$UBF_\mathbf{p} = I_\mathbf{p} + \sum_\mathbf{q} \bar{G}_{\sigma_s}(\mathbf{q}-\mathbf{p})\, G_{\sigma_r}(I_\mathbf{q} - I_\mathbf{p})\,(I_\mathbf{q} - I_\mathbf{p}) \tag{14}$$

Compared to the bilateral filter, this unnormalized version has a weaker effect when the sum of the weights W is small. This occurs when the center pixel is different from many of its neighbors, which typically happens at edges. Durand and Dorsey [2002] interpret W from a robust statistics standpoint and explain that when it is small,

the bilateral filter returns an estimate based on limited data, which causes the artifacts that appear at some strong edges. The unnormalized version can be interpreted as a filter that is weaker at those ambiguous locations (Fig. 5). This tends to generate slightly softer images but greatly reduces the artifacts as shown in Figure 7 and in supplemental material. We also compared to the post-process fix described by Durand and Dorsey that blends the bilateral filter output with a blurred version of the input using $\log(W)$ as blending control. Similarly, the unnormalized filter achieves cleaner and softer outputs without requiring post-processing. Although the results are not as detailed as with the multiscale local Laplacian filter, the unnormalized filter is about 5 times faster, which can make it a useful alternative.

Alternative interpretation. The unnormalized bilateral filter can also be seen as a spatially varying blend between the input and the result of the bilateral filter:

$$UBF_{\mathbf{p}} = (1-\alpha_{\mathbf{p}})\, I_{\mathbf{p}} + \alpha_{\mathbf{p}}\, BF_{\mathbf{p}} \tag{15}$$

where the mixing coefficient $\alpha_{\mathbf{p}}$ is $\sum_{\mathbf{q}} \bar{G}_{\sigma_s}(\mathbf{q}-\mathbf{p})\, G_{\sigma_r}(I_{\mathbf{q}} - I_{\mathbf{p}})$. This formula comes directly from Equations 6 and 14. The $\alpha_{\mathbf{p}}$ coefficient is close to 1 in uniform regions and takes lower values near edges because of the range term. This also explains why the unnormalized bilateral filter has a weaker effects near discontinuities.

Effect on brightness. The removal the normalization does not alter the overall brightness of the output because it is done on the form of the bilateral filter which applies weights to the pixel differences (Eq. 6 and 14). That is, changing the weights affects how the output fluctuates around $I_{\mathbf{p}}$, it does not affect the absolute brightness since $I_{\mathbf{p}}$ remains the "base value". Equation 15 also shows that the brightness of the unnormalized filter is between the value of the input and that of the normalized filter.

Acceleration method. We build upon Equation 15 and use the bilateral grid [Chen et al. 2007; Paris and Durand 2009] to efficiently compute the unnormalized bilateral filter.

Discussion. The unnormalized bilateral filter is a middle ground between bilateral filtering and local Laplacian filtering. As shown in the close-ups in Figure 6 it produces cleaner edges than the former without reaching the quality of the latter. The bilateral filter tends to produce overly sharp edges (e.g., at the border of the lamp shade) whereas its unnormalized counterpart and the local Laplacian filters produce a properly anti-aliased edge. However, both versions of the bilateral filter can generate reverse gradients on thin features (e.g., on the window frame) while the local Laplacian filters are not prone to this problem. Further, we found that the local Laplacian filters can generate stronger effects because it affects several frequency bands unlike the unnormalized bilateral filter that alters details at a single scale (see the scene through the window for instance). Figure 7 shows the effects of varying the pyramid depth to transition from the unnormalized bilateral filter with one level to local Laplacian filtering with a full pyramid. Halos may be observed at intermediate depths whereas they are typically not an issue at small and large scales. This observation is consistent with the findings of Trentacoste et al. [2012].

3. EFFICIENT LOCAL LAPLACIAN FILTERING

We propose an acceleration technique to evaluate local Laplacian filters on single-channel images. This encompasses many practical cases such as detail manipulation and tone mapping [Paris et al. 2011] as well as photographic style transfer that we discuss later in Section 4. Our strategy is based on the fact that the nonlinearity comes from the dependency on g. We characterize this dependency in terms of signal processing, which allows us to design a theoretically grounded subsampling scheme that is more than an order of magnitude faster than the algorithm proposed by Paris et al. [2011]. In practice, we precompute a small set of pyramids $\{L[r_j(I)]\}$ over different values γ_j of g, where r_j is the remapping function for $g = \gamma_j$. Whenever we need a pyramid coefficient for a particular g value, instead of remapping the image and computing a new pyramid, which is expensive, we find j such that $\gamma_j \leq g < \gamma_{j+1}$ and interpolate the coefficients of precomputed pyramids j and $j+1$. Formally, we seek to sample r as sparsely as possible without losing accuracy. If r is band-limited, using the sampling theorem, the optimal sampling is the Nyquist limit, i.e., half the smallest wavelength present in the signal. To estimate this value, we observe r as a function of g. From that perspective, only the term $(i-g)\, f(i-g)$ is not constant and what actually matters is the frequency content of $x\,f(x)$. Denoting the Fourier transform by $\mathcal{F}[\cdot]$ and using $'$ for derivatives, the property $\mathcal{F}[x\,f(x)] \propto \mathcal{F}[f]'$ ensures that, if f is band-limited, r is as well. This means that, if f is band-limited, we can sparsely sample the intensity domain with only minimal loss. We further discuss the accuracy of this approach at the end of this section.

Our algorithm is as follows:

(1) Compute the Gaussian pyramid of I.

(2) Regularly sample the intensity range with the $\{\gamma_j\}$ values.

(3) Compute the remapped images $\{r_j(I)\}$ and their Laplacian pyramids $\{L[r_j(I)]\}$.

(4) For each pyramid coefficient (ℓ, x, y):
   i. Get the corresponding coefficient g in the Gaussian pyramid.
   ii. Compute a and j such that $g = (1-a)\gamma_j + a\,\gamma_{j+1}$.
   iii. Linearly interpolate the output coefficient from the precomputed pyramids: $L_\ell[O](x,y) = (1-a)\, L_\ell[r_j(I)](x,y) + a\, L_\ell[r_{j+1}(I)](x,y)$.

(5) Collapse the output pyramid $\{L[O]\}$.

In practice, a Gaussian function G_σ is used for f, e.g., for the detail enhancement, we recommend sampling the intensity range every standard deviation σ. For other applications such as tone mapping, we proceed similarly and use a sampling that matches the bandwidth of f.

Because the number of precomputed pyramids is fixed, this algorithm has linear complexity in the number of pixels. In most applications, r is not strictly band-limited. Further, we use linear interpolation instead of a sinc kernel for reconstructing the signal. The effect of these simplifications is that our algorithm does not perfectly reproduce the result of the original algorithm. Nevertheless, it produces accurate approximations above 30 dB and the differences are invisible in practice. Most importantly, when sampling every standard deviation, it runs at interactive rates, about 350 ms per megapixel on a 2.66 GHz Intel Core i7, which is about 50× faster than the heuristic of Paris and colleagues for the same accuracy (Fig. 8). We also ported our algorithm to graphics hardware (an NVIDIA GeForce 480 GTX), where it runs at about 49 ms for a one megapixel image and about 116 ms for a four megapixel image, which is about 10 times faster than Paris's heuristic implemented on the same card. Beside this, using the Halide programming language dedicated to optimizing image-processing algorithms, Ragan-Kelley et al. [2012] recently reported 49 ms on an NVIDIA Tesla C2070 GPU (equivalent to a mid-range consumer

GPU) using our algorithm also on a single channel of a 4-megapixel image, thereby approximately gaining another 2× factor.

Discussion. While these running times are not as fast as other filters, e.g. [Chen et al. 2007; Fattal 2009; Gastal and Oliveira 2011; 2012], they are sufficient to enable user interaction while still producing clean results on a wider range of parameters than what these faster options support. In practice, the choice between our filter and these other techniques depend on the application and the priority between visual quality and speed. In terms of memory, a straightforward implementation of the algorithm requires to store all the precomputed pyramids $\{L[r_j(I)]\}$ in memory at the same time. If memory consumption is an issue, one can instead compute one such pyramid at a time and add directly its contribution to the output pyramid. While this approach requires more updates of the output pyramids, only one precomputed pyramid needs to be in memory at a given time, which can be beneficial depending on the available memory.

4. PHOTOGRAPHIC STYLE TRANSFER

As we showed in Section 2, the re-centering of the remapping makes local Laplacian filters rely on differences between the value of a pixel and its neighbors. This makes them closely related to image gradients. In this section, we propose a new method using local Laplacian filters to alter the *distribution* of these differences and, by extension, the distribution of image gradients. Although our approach is mostly empirical, it builds upon a strong intuition and performed well in practice. We illustrate this capability with an algorithm to transfer photographic style akin to the method of Bae et al. [2006].

4.1 Manipulating gradient distributions

To gain intuition, we first consider a two-level pyramid and further assume that the pixel $\mathbf{p}$ has a single neighbor $\mathbf{q}$ and ignore the spatial weight $\bar{G}_{\sigma_p}$. The output of the filter then becomes: $O_\mathbf{p} = I_\mathbf{p} + f(I_\mathbf{q} - I_\mathbf{p})\ (I_\mathbf{q} - I_\mathbf{p})$. We highlight the role of pixel differences by subtracting $I_\mathbf{q}$ on both sides. Assuming that f is symmetric, which is always the case in practice, we get: $O_\mathbf{p} - I_\mathbf{q} = (I_\mathbf{p} - I_\mathbf{q}) - f(I_\mathbf{p} - I_\mathbf{q})\ (I_\mathbf{p} - I_\mathbf{q})$. Defining $h(x) = [1 - f(x)]x$, this can be rewritten in the more concise form $O_\mathbf{p} - I_\mathbf{q} = h(I_\mathbf{p} - I_\mathbf{q})$ which shows that the filter remaps $I_\mathbf{p}$ so that its difference with its neighbor $I_\mathbf{q}$ has a desired value specified by the h function. Since $\mathbf{p}$ and $\mathbf{q}$ are neighbors, this can be seen as remapping the image gradient at $\mathbf{p}$. If we now consider again a larger neighborhood as in Equation 10, the filter can be interpreted as making a trade-off between the desired gradient values coming from different neighbors. $\bar{G}_{\sigma_p}$ weights the contribution of each pixel $\mathbf{q}$ and h defines the desired output gradients. Further, h is sufficient to define a local Laplacian filter since $\tilde{r}(i) = g + h(i - g)$ where $\tilde{r}$ is the remapping function originally defined by Paris et al. (§ 1.2). This comes directly from the definitions of f and h.

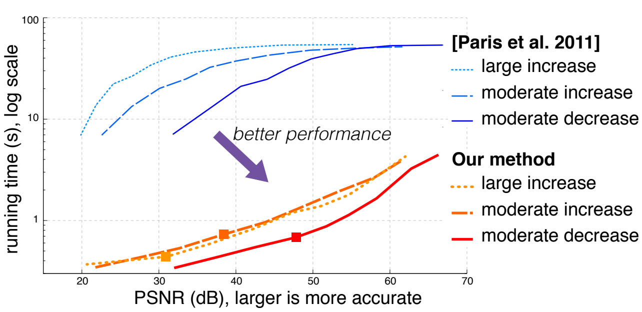
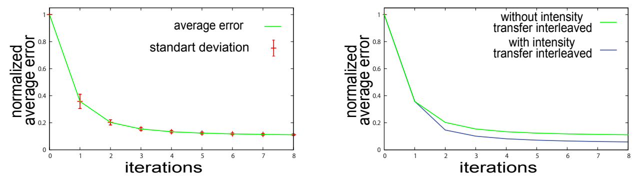


Fig. 8. We measured the running time and accuracy of Paris's acceleration scheme and ours for several settings. For Paris's algorithm, we varied the depth of the intermediate sub-pyramids; for ours, we varied the sampling rate of the intensity range. For each setting of both methods, we applied three different sets of parameters to achieve a large detail increase, a moderate increase, and a moderate decrease. For the same accuracy, our scheme is about 50× faster than Paris's. The square marks on the curve of our method indicate the one-sample-per-standard-deviation sampling rate. We used a 1600 × 1200 image.



(a) (b)

Fig. 10. To validate our histogram transfer method, we measured the Earth Mover's distance between the gradient histogram of the output image and the gradient histogram of the model image as a function of the number of iterations. We report the average and standard deviation of this distance over 15 image pairs. We normalized the distances so that the difference between the input and model histograms is 1. The output histogram quickly becomes closer to the model and then the convergence slows down (a). We tested more iterations, up to 1000, the distance keeps reducing but slowly. Visually the results becomes stable after a few iterations (Fig. 9). Interleaving standard intensity histogram transfer with our local Laplacian method speeds up the convergence at a minor computational cost (b).

Building upon this intuition, we describe a method to transfer the histogram of gradient amplitudes from a model image M to the input image I. We apply local Laplacian filtering with a remapping function r defined such that gradient statistics of M are transferred to I. For both images, we compute the histogram of the gradient amplitudes $\|\nabla I\|$ and $\|\nabla M\|$ and the corresponding histogram transfer function t, i.e. $t(x) = CDF\big[\|\nabla M\|\big]^{-1}\big(CDF\big[\|\nabla I\|\big](x)\big)$ where $CDF\big[\cdot\big]$ denotes the cumulative distribution function, i.e., $CDF\big[F\big](x) = \sum_{\mathbf{p}|F(\mathbf{p})<x} 1$ for F a scalar function defined over the image domain. The transfer function t means that when two pixels have a difference with amplitude $|x|$ in I, we want a difference with amplitude $t(|x|)$ in O. Finally, we seek to preserve the sign of the difference, and we define $h(x) = \operatorname{sign}(x)\ t(|x|)$ that leads to the remapping function:

$$r(i) = g + \operatorname{sign}(i - g)\ t(|i - g|) \tag{16}$$

We use this function to run a local Laplacian filter on I. While we built our intuition on a simplified case based only on two levels and two pixels, in practice, the situation is more complex and we need to iterate to obtain the desired result. Figures 9 and 10a show that our approach quickly converges after a few iterations. The only fixed point of the iteration is when r is the identity function, which implies that t is also the identity function, which finally implies $CDF\big[\|\nabla M\|\big] = CDF\big[\|\nabla I\|\big]$. This guarantees that the process can only converge to the desired result where the output has the gradient histogram of the model. In supplemental material, we tested our method on a variety of image pairs and it always successfully transferred the gradient histogram.

4.2 Style transfer algorithm

We demonstrate how to achieve photographic style transfer in the spirit of Bae et al. [2006] using the gradient transfer method described in the previous section. We seek to transfer the "look" of the model image M to an input image I. Typically, M is a picture by a master, such as Ansel Adams, and I is a photo by a casual photographer who wishes to mimic the master's style. Bae's technique involves solving the Poisson equation twice to mitigate the over-sharpening artifacts inherent in bilateral filtering. This has two drawbacks: first, the global optimization limits scalability, and second, these corrections can limit the large image transforms necessary to achieve more extreme looks. In comparison, our method is optimization-free and its increased robustness enables more strongly stylized renditions.

Our algorithm follows the same overall approach as Bae's: we seek to match both the global contrast, i.e. the large-scale intensity variations, and the local contrast, i.e. the amount of texture of the model image M. We use an iterative process. For each iteration, we first compute the histograms of the input and model gradients, $\|\nabla I\|$ and $\|\nabla M\|$. We build the transfer remapping function r as described in Section 4.1 and apply the corresponding local Laplacian filter. Then, we apply a standard intensity histogram transfer to match the intensity distribution of the model M. We typically apply a few such iterations, 4 in all the results presented in this paper and in supplemental material. Intuitively, the local Laplacian filtering step transfers the local contrast, i.e. the gradients, and the intensity matching step transfer the global variations, i.e. the intensity distribution. Further, we found empirically that interleaving the histogram matching step speeds up the convergence of the gradient transfer as shown in Figure 10b. To produce the final result, we observe that after gradient matching, the local contrast is accurate but the dynamic range may be too large, i.e., some values may be outside $[0,1]$. Conversely, after intensity matching, the global contrast and the dynamic range match that of the model but the local contrast is sometimes slightly too weak. We found that after the last iteration, averaging the result after gradient transfer with the result after intensity transfer yields a satisfying trade-off on all our examples (Fig. 9).

4.3 Results

We demonstrate the robustness of our method using a variety of input and model images for which we successfully transfer the look of the artist's photo to the input (Fig. 13 and 14). Our technique handles standard images and HDR images seamlessly since it produces output with the same dynamic range as the model photograph.

A limitation of our approach is the lack of semantic understanding of the scene. For instance, the method can sometimes introduce unnatural variations in a uniform sky or darken regions that one would expect to be lit (Fig. 11), or generate overly smooth results when used with low-detail models (see supplemental material). Fixing these problems in general requires either the user in the loop or a higher-level analysis of the scene, both of which are beyond the scope of this paper but would be interesting future work. Also, some look may be more appropriate for some photos than others, and we rely on users to make this subjective choice. That said, these cases are rare and our approach performs well most of the time. Compared to the method by Bae et al. [2006], our approach often performs better, especially in its ability to reproduce the style of texture. Because Bae and colleagues use the bilateral filter, they need to correct their results using strong gradient-domain constraints that limit their ability to modify the input image. This is visible when comparing Bae's results with and without imposing these constraints (Fig. 12). While Bae's results without constraints better match the models, they suffer from halos at strong edges. The gradient constraints mitigate these artifacts but come at the cost of significantly duller renditions. Since our approach does not require such strong constraints, we are able to obtain high-quality results that better reproduce the texture in the models without introducing halos. We also compared our method to histogram transfer applied to the gradient amplitudes followed by a Poisson reconstruction. As shown in Figure 13, this naïve approach matches the amount of details in the models poorly and does not yield satisfying results. Finally, we also experimented with the multiscale approach of Sunkavalli et al. [2010] and found that the produced image does not match the model look as well as our approach (Fig. 13 and 14).

5. CONCLUSION

We have studied local Laplacian filters and have shown that they are closely related to bilateral filtering and anisotropic diffusion. This insight has led to several practical contributions: we have described the unnormalized bilateral filter which improves the results of the bilateral filter at edges, sped up the local Laplacian filters, and described a technique to manipulate image gradients that leads to a robust algorithm for transferring photographic style. We believe that these improvements make local Laplacian filtering usable and suitable for interactive image editing.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their constructive comments and to Mark Fairchild's HDR Photographic Survey for providing high quality HDR images. Frdo Durand, Sam Hasinoff, and Jan Kautz acknowledge gifts from Adobe.

REFERENCES

ADAMS, A., BAEK, J., AND DAVIS, A. 2010. Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum (Proc. Eurographics).*

ADAMS, A., GELFAND, N., DOLSON, J., AND LEVOY, M. 2009. Gaussian KD-trees for fast high-dimensional filtering. *ACM Transactions on Graphics 28,* 3. Proceedings of the ACM SIGGRAPH conference.

AUBERT, G. AND KORNPROBST, P. 2002. *Mathematical problems in image processing: Partial Differential Equations and the Calculus of Variations*. Applied Mathematical Sciences, vol. 147. Springer.

Fig. 11. Failure cases. Our algorithm only has a low-level view of an image and does not know its semantics. In rare cases, it can produce unexpected results such as low-frequency variations in the sky (left) or darkening of a tower that we expect to be well lit (right).

(a) HDR input naive gamma compression | (b) Bae et al. without gradient correction | (c) Bae et al. with gradient correction | (d) our method

Fig. 12. Since the method by Bae et al. [2006] relies on the bilateral filter, it suffers from rim halos at strong edges (b). This can be corrected by manipulating the gradient field of the result. However, this slows down the computation and yields duller results (c). In comparison, our approach directly produces satisfying results and does not require solving a costly Poisson equation (d).

Bae, S., Paris, S., and Durand, F. 2006. Two-scale tone management for photographic look. *ACM Transactions on Graphics (Proc. SIGGRAPH) 25,* 3, 637 – 645.

Barash, D. and Comaniciu, D. 2004. A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Journal of Image and Video Computing 22*, 73–81.

Black, M. J., Sapiro, G., Marimont, D. H., and Heeger, D. 1998. Robust anisotropic diffusion. *IEEE Transactions on Image Processing 7,* 3 (March), 421–432.

Buades, A., Coll, B., and Morel, J.-M. 2005a. Neighborhood filters and PDE's. Tech. Rep. 2005-04, CMLA.

Buades, A., Coll, B., and Morel, J.-M. 2005b. A non local algorithm for image denoising. In *Proceedings of the conference on Computer Vision and Pattern Recognition*.

Buades, A., Coll, B., and Morel, J.-M. 2006. The staircasing effect in neighborhood filters and its solution. *IEEE Transactions on Image Processing 15,* 6, 1499–1505.

Chen, J., Paris, S., and Durand, F. 2007. Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26,* 3.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. 2006. Image denoising with block-matching and 3D filtering. In *Proceedings of SPIE Electronic Imaging*.

Durand, F. and Dorsey, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics (Proc. SIGGRAPH) 21,* 3.

Eisemann, E. and Durand, F. 2004. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics 23,* 3 (July). Proceedings of the ACM SIGGRAPH conference.

Elad, M. 2002. On the bilateral filter and ways to improve it. *IEEE Transactions On Image Processing 11,* 10.

Farbman, Z., Fattal, R., Lischinski, D., and Szeliski, R. 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 27,* 3.

Fattal, R. 2009. Edge-avoiding wavelets and their applications. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28,* 3.

Fattal, R., Agrawala, M., and Rusinkiewicz, S. 2007. Multiscale shape and detail enhancement from multi-light image collections. *ACM Transactions on Graphics (Proc. SIGGRAPH) 26,* 3.

Fattal, R., Lischinski, D., and Werman, M. 2002. Gradient domain high dynamic range compression. *ACM Transactions on Graphics (Proc. SIGGRAPH) 21,* 3.

Gastal, E. S. L. and Oliveira, M. M. 2011. Domain transform for edge-aware image and video processing. *ACM Transactions on Graphics 30,* 3. Proceedings of the ACM SIGGRAPH conference.

Gastal, E. S. L. and Oliveira, M. M. 2012. Adaptive manifolds for real-time high-dimensional filtering. *ACM Transactions on Graphics 31,* 4. Proceedings of the ACM SIGGRAPH conference.

He, K., Sun, J., and Tang, X. 2010. Guided image filtering. In *Proceedings of European Conference on Computer Vision*.

Kass, M. and Solomon, J. 2010. Smoothed local histogram filters. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29,* 3.

Li, Y., Sharan, L., and Adelson, E. H. 2005. Compressing and companding high dynamic range images with subband architectures. *ACM Transactions on Graphics 24,* 3. Proceedings of the ACM SIGGRAPH conference.

Mrzek, P., Weickert, J., and Bruhn, A. 2006. *Geometric Properties from Incomplete Data*. Springer, Chapter On Robust Estimation and Smoothing with Spatial and Tonal Kernels.

Paris, S. and Durand, F. 2009. A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision*.

Paris, S., Hasinoff, S. W., and Kautz, J. 2011. Local Laplacian filters: Edge-aware image processing with a Laplacian pyramid. *ACM Transactions on Graphics (Proc. SIGGRAPH) 30,* 4.

Paris, S., Kornprobst, P., Tumblin, J., and Durand, F. 2009. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*.

Perona, P. and Malik, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions Pattern Analysis Machine Intelligence 12,* 7 (July), 629–639.

Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., and Toyama, K. 2004. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics 23,* 3 (July). Proceedings of the ACM SIGGRAPH conference.

Ragan-Kelley, J., Adams, A., Paris, S., Levoy, M., Amarasinghe, S., and Durand, F. 2012. Decoupling algorithms from schedules for easy optimization of image processing pipelines. *ACM Transactions on Graphics 31,* 3. Proceedings of the ACM SIGGRAPH conference.

Subr, K., Soler, C., and Durand, F. 2009. Edge-preserving multiscale image decomposition based on local extrema. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 28,* 5.

Sunkavalli, K., Johnson, M. K., Matusik, W., and Pfister, H. 2010. Multi-scale image harmonization. *ACM Transactions on Graphics (Proc. SIGGRAPH) 29,* 3.

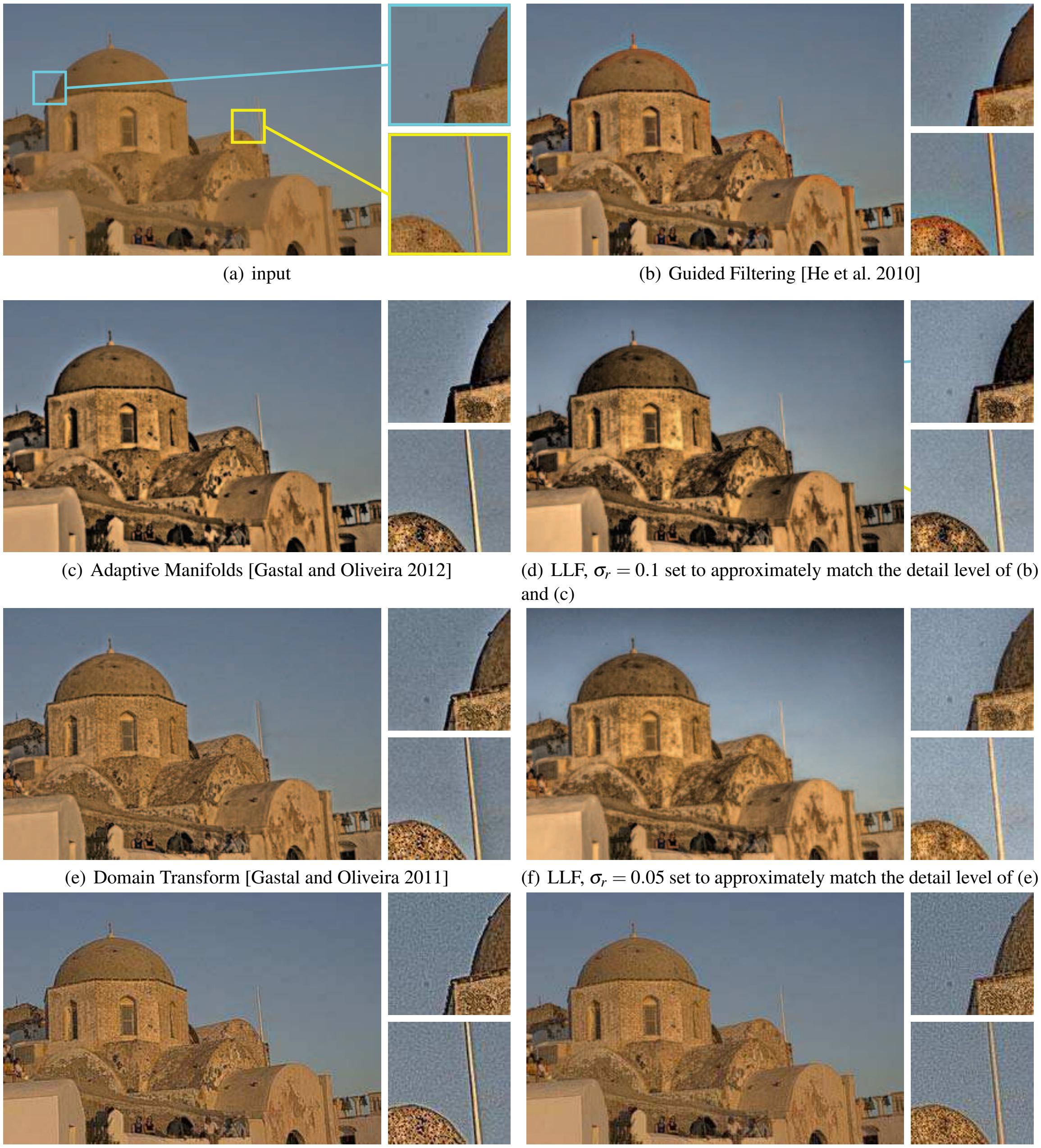
Tomasi, C. and Manduchi, R. 1998. Bilateral filtering for gray and color images. In *Proceedings of the International Conference on Computer Vision*. IEEE, 839–846.

Trentacoste, M., Mantiuk, R., Heidrich, W., and Dufrot, F. 2012. Unsharp Masking, Countershading and Halos: Enhancements or Artifacts? In *Proceedings of Eurographics*.

van de Weijer, J. and van den Boomgaard, R. 2002. On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks. In *Proceedings of the International Conference on Pattern Recognition*.

Xu, L., Cewu, Xu, Y., and Jia, J. 2011. Image smoothing via L0 gradient minimization. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia) 30,* 6.

ZHONG, J. AND SUN, H. 2008. Wavelet-based multiscale anisotropic diffusion with adaptive statistical analysis for image restoration. *IEEE Transactions on Circuits and Systems I 55,* 9.



(a) input

(b) Guided Filtering [He et al. 2010]

(c) Adaptive Manifolds [Gastal and Oliveira 2012]

(d) LLF, $\sigma_r = 0.1$ set to approximately match the detail level of (b) and (c)

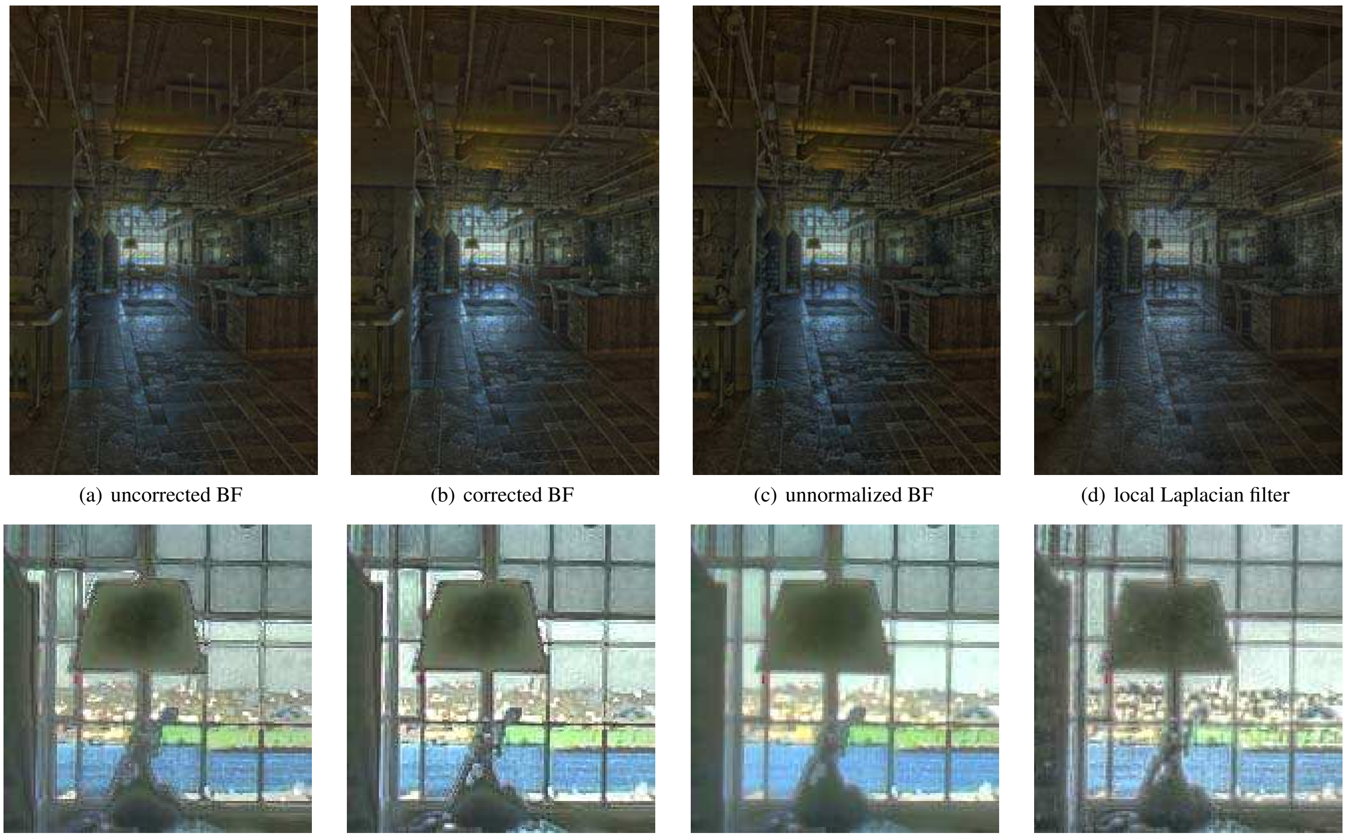
(e) Domain Transform [Gastal and Oliveira 2011]

(f) LLF, $\sigma_r = 0.05$ set to approximately match the detail level of (e)

(g) Multiscale [Fattal et al. 2007]

(h) Unormalized billateral filter, one level of LLF, $\sigma_r = 0.05$ set to approximately match the detail level of (g)

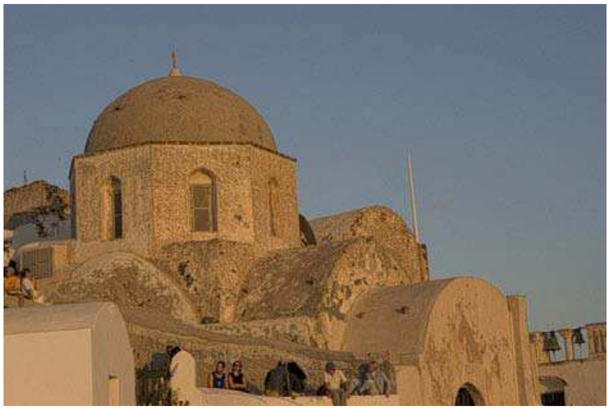
Fig. 3. Comparison with the multiscale method of Fattal et al. [2007], Guided Filtering [He et al. 2010], the Domain Transform [Gastal and Oliveira 2011], and Adaptive Manifolds [Gastal and Oliveira 2012] on detail enhancement. Although these filters produce mostly acceptable results, halos remain visible around the dome (b,c,e,f). In comparison, local Laplacian filters (LLF) generate clean images without halos (d,g). We used the code provided by the authors of each method and set the parameters to achieve a large detail increase. For the domain transform and guided filter, we used the parameters suggested in the papers and a 5x enhancement. For [2007] multiscale method, we used the parameters of the paper and multiplied the output by 2 to have an exposure more similar to the other methods. For adaptative manifolds, we used the parameters suggested in the paper on the grayscale image to avoid strange color changes.We adjusted the settings of the local Laplacian filters to produce an approximately similar level of detail enhancement. Despite our efforts, visible differences remain because of the specificities of each filter. The absence of halos in the results from the local Laplacian filters is not specific to settings that we selected (see [Paris et al. 2011] for details). We used the fast algorithm described in this paper to compute the local Laplacian filters. More comparison are available in the supplementary material.



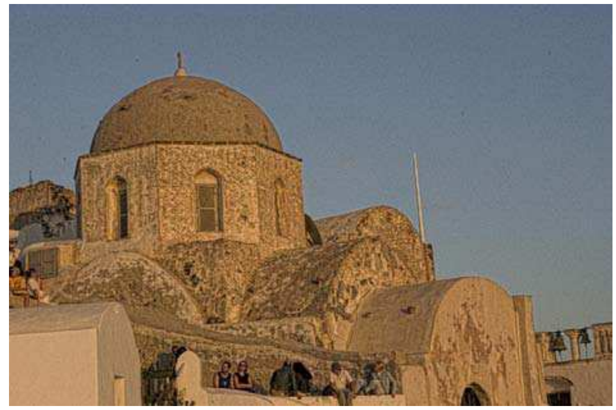
(a) uncorrected BF (b) corrected BF (c) unnormalized BF (d) local Laplacian filter

(e) uncorrected BF (f) corrected BF (g) unnormalized BF (h) local Laplacian filter

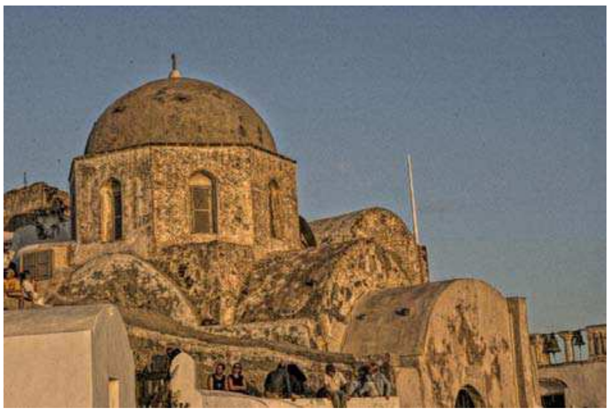
Fig. 6. Compressing an HDR image (a) with strong detail enhancement with the bilateral filter (b) leads to artifacts on the border of strong edges, e.g. on the window structure and on the lamp base. Some of them can be fixed by a postprocessing step (c), for instance the window structure is significantly improved– other parts such as the lamp base remains problematic. With our unnormalized bilateral filer (d), most these artifacts are avoided although the rendition is overall softer, and none appear with our fast version of the local Laplacian filter (e). However, the local Laplacian filter remains slower than the unnormalized filter. (b,c,d,e) are close-ups of the lamp at the end of corridor (a).



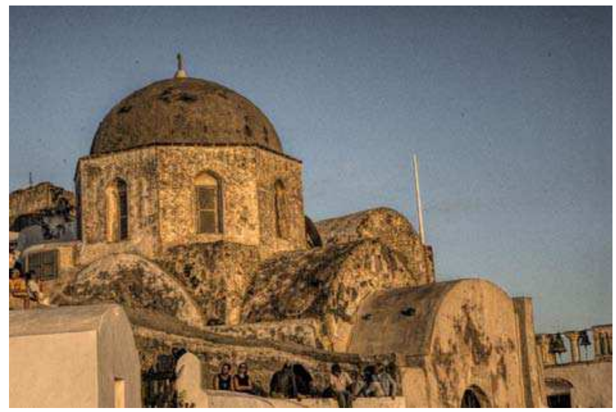
(a) 1 level - unnormalized bilateral filter



(b) 2 levels



(c) 4 levels



(d) 8 levels

Fig. 7. Changing the depth of the pyramid used to evaluate the local Laplacian filters progressively transitions from the unnormalized bilateral filter (a) to the local Laplacian filters. We used $\sigma_r = 0.05$ for all these results. Figure 3 (f) shows the result with a complete 9-level pyramid which visually similar to the 8-level result (d). Unsightly halos are visible at intermediate depths (c); this observation is consistent with the results of Trentacoste et al. [2012].

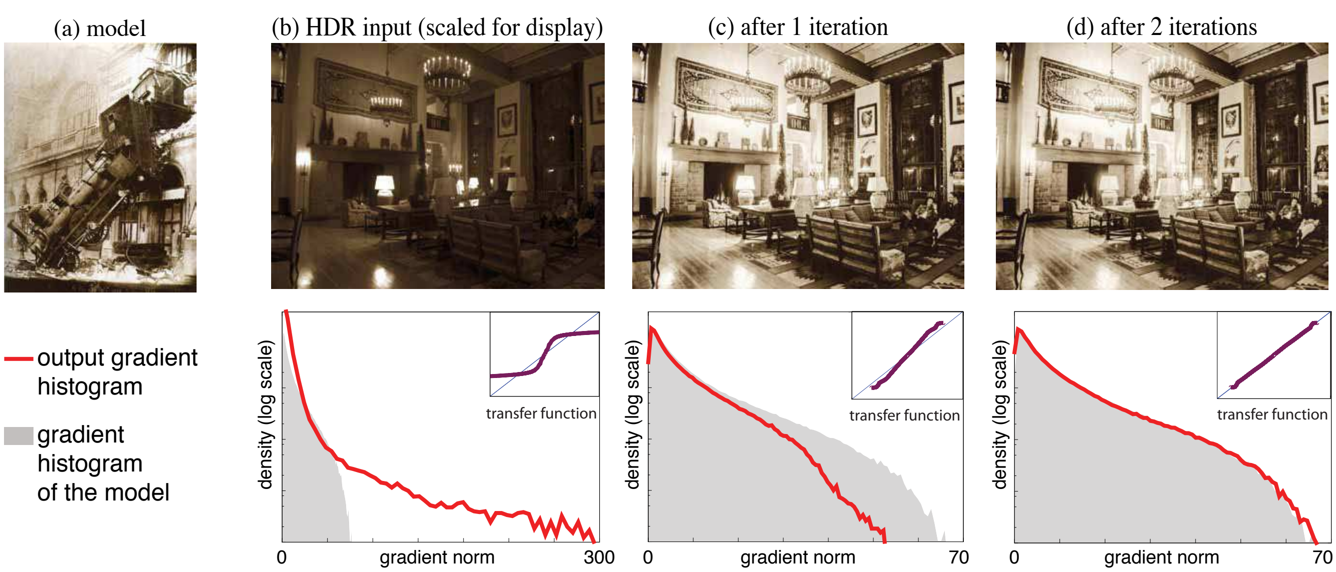


Fig. 9. Our iterative method to transfer gradient histograms stabilizes quickly. Visually, the results do not change after 2 iterations.

input image (HDR, gamma compressed)

model

our method

Sunkavalli et al. [2010]

Naive gradient transfer using a Poisson equation

Bae et al. [2006]

Fig. 13. In contrast to other methods, our style transfer method is able to reveal a lot of small variations in the input image.

Fig. 14. In the method of Bae and colleagues, the Poisson reconstruction is computationally expensive, and often breaks the style transfer while trying to fix the strong halos. In contrast, our direct method does not create halos and achieves acceptable style transfer even on these difficult examples.