

Mixed-Domain Edge-Aware Image Manipulation

Xian-Ying Li, Yan Gu, Shi-Min Hu, *Member, IEEE*, and Ralph R. Martin, *Member, IEEE*

Abstract—This paper gives a novel approach to edge-aware image manipulation. Our method processes a Gaussian pyramid from coarse to fine, and at each level, we apply a nonlinear filter bank to the neighborhood of each pixel. Outputs of these spatially-varying filters are merged using global optimization, and this optimization problem is solved using an *explicit* mixed-domain (real space and DCT transform space) solution, which is efficient, accurate, and easy-to-implement. We demonstrate applications of our method to a set of problems including detail and contrast manipulation, HDR compression, non-photorealistic rendering, and haze removal.

Index Terms—mixed-domain, edge-aware image processing, optimization-based image processing, multi-scale method.

I. INTRODUCTION

EDGE-AWARE image processing is an important technique that has received much attention in the computer graphics community. The goal is to process or filter images in some way without destroying fine scale image edges. *Anisotropic diffusion* [1] and *bilateral filtering* [2] are well-known examples of such techniques, originally devised for image smoothing, but later extended to many other applications. More recently, many other edge-aware techniques have been proposed, e.g. [3], [4], [5], [6], [7], [8]. The particular topic considered in this paper is to provide a simple interface for independently adjusting the *overall* appearance and *details* of input images (in a similar way to [7]), doing so in a way which both keeps fine edges, and avoids introducing unsightly artifacts. An example of our manipulation of overall/detailed appearance is shown in Fig. 1. Different from previous work, our approach is based on a novel mixed-domain (real space and DCT transform space) processing framework, which is fast and simple, and is readily accelerated on the GPU.

A challenging problem in edge-aware image processing is reducing, or if possible avoiding, artifacts at image edges. This is particularly true for algorithms which process overall appearance and detail separately. The problem is illustrated using a simple 1D signal in Fig. 2 (left). Suppose we want to amplify the detail, without losing the edge. The grey line represents an input signal, and the blue line represents the ‘overall’ signal, produced by averaging of the input signal. The red signal represents the ‘detail’, the difference between the input and overall signals. If an enhanced version of the detail is added directly back to the overall signal to give the output (the black signal), the result will overshoot at the step, causing what is often referred to as a ‘halo’ effect in image processing. Conversely, if averaging is done in such a way as

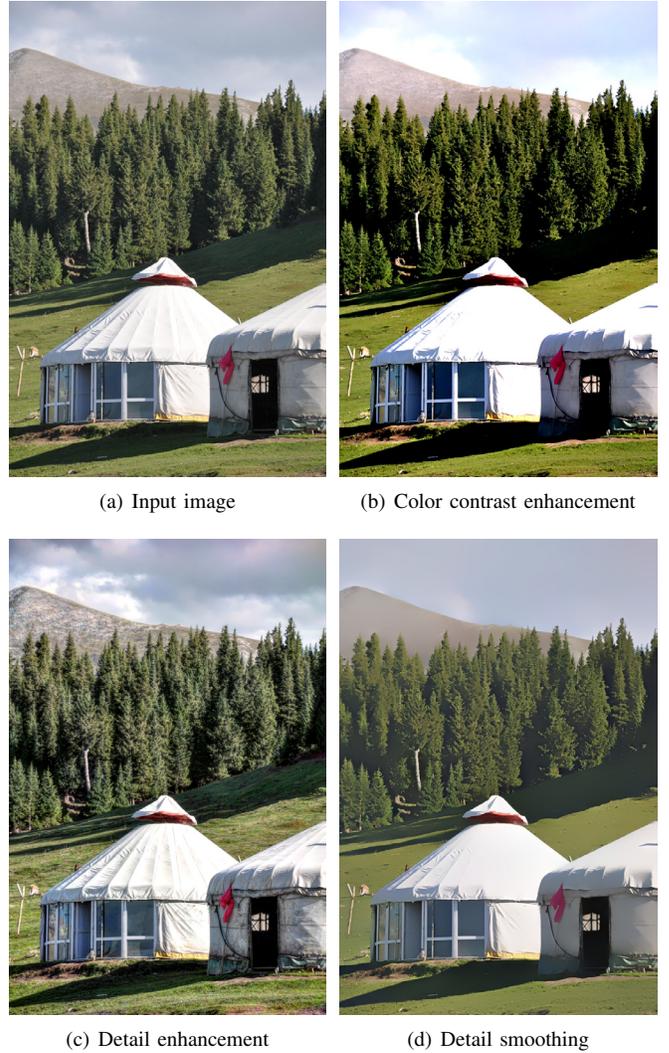


Fig. 1. An input image, and three output images produced by our method. Edges are preserved well at different scales, without unsightly artifacts such as halos, gradient reversals, or aliasing.

to sharpen the input step, the result can even be a ‘gradient reversal’, as shown in Fig. 2 (center). Although weak halos can make image edges stand out, and indeed are deliberately introduced in *unsharp masking*, stronger halos are perceived as unwanted artifacts. Many previous works have thus carefully considered how to reduce or avoid halos [3], [4], [7].

In this paper, we advocate a new method for halo-free edge-aware image manipulation. Our method processes a Gaussian pyramid level-by-level, from coarse to fine, in order to reduce halos at each scale. At each level, we consider a set of local filters defined on overlapping windows, and use optimization to merge their outputs with those of the previous level: the result at the current level has a detailed appearance in accor-

Xian-Ying Li, Yan Gu, and Shi-Min Hu are with TNList, Department of Computer Science and Technology, Tsinghua University, Beijing, P.R. China, 100084. E-mail: lee.xianying@gmail.com, shimin@tsinghua.edu.cn.

Ralph R. Martin is with the School of Computer Science & Informatics, Cardiff University, Cardiff, Wales, CF24 3AA UK.

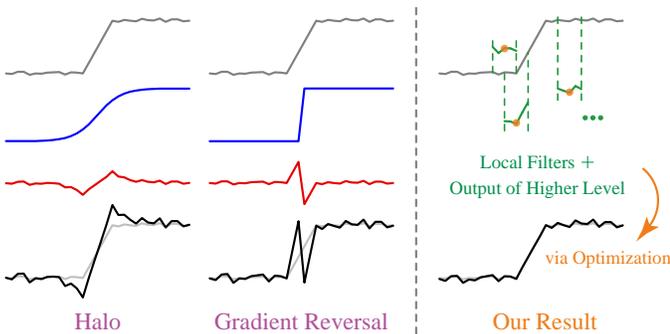


Fig. 2. A simple 1D example. Left, center: poor base-detail decomposition may cause halos or gradient reversals. Right: our artifact-free result (using $\alpha = 2.0$, $\beta = 1.0$; see later).

dance with the local filters’ output, and overall appearance in accordance with the coarser result from the next higher level. Controlling the detailed behavior of our result in this way helps us to reduce halos. An enhancement result produced by our method for the same 1D signal is shown in Fig. 2 (right).

Contributions of our work in this paper include:

- A new, rotationally invariant optimization-based formulation for edge-aware image manipulation, without causing unsightly artifacts such as halos, gradient reversals, and aliasing.
- A direct mixed-domain solution to the optimization formulation above, which is exact, efficient, and easy-to-implement.

II. RELATED WORKS

A. Edge-aware image processing

Edge-aware image processing for computer graphics is a challenging problem that has received much recent attention. Here we briefly review the techniques most closely related to our work.

Anisotropic diffusion [1] uses a non-linear PDE that depends on local image variation to iteratively smooth an image without blurring important features. However, being based on iteration, anisotropic diffusion and related PDE-based methods are slow, and furthermore, parameters are difficult to set [9]. Tumblin and Turk [10] show how anisotropic diffusion can be used as a basis for high dynamic range (HDR) compression, but as pointed out by [3], [4], it tends to over-sharpen image edges.

Bilateral filtering (BLF) [2] provides an alternative approach to edge-aware image smoothing. It uses a local, non-iterative, explicit, data dependent filter, whose simplicity, efficiency, and effectiveness [11], [12], [13], [14], [15] have led to its widespread use [16], [17], [18]. The survey in [19] gives an in-depth treatment. However, as discussed in [4], BLF involves a trade off between edge preservation and data smoothing. Methods relying on BLF to separate a mean surface from detail may lead to halos at image edges if too much smoothing is applied.

Weighted least squares (WLS) and related methods attempt to avoid such halos by use of more careful *edge-aware decomposition*. WLS [4] computes the smooth (overall) component of the input image by optimizing a quadratic energy based on

squared gradients with spatially-varying weights; the weights are set to be small at sharp image edges, to preserve them. As an alternative, Subr et al. [6] follow the ideas of empirical mode decomposition (EMD) [20]: they smoothly connect the local extrema of the input image to form a maximal envelope and a minimal envelope, and then average these two envelopes to obtain the smooth component. Compared to classical EMD, Subr et al.’s approach behaves better at sharp image edges. However, both of these methods have to solve large sparse linear systems. Although some fast numerical solvers like multi-grid methods can potentially be used, these complex methods are not easy to implement. Our algorithm avoids solving large linear equations.

Pyramid based methods provide a further approach to edge-aware image processing, using a multi-scale representation such as a Gaussian/Laplacian pyramid, or a wavelet decomposition. Li et al. [21] perform HDR compression by directly manipulating the wavelet coefficients with a careful correction scheme, but as noted by [7], this method can also produce halos with poor choice of parameter values. Later, Fattal [5] developed two novel data-dependent wavelets more suitable for edge-aware image applications than traditional wavelets. Most recently, Paris et al. [7] introduced the *local Laplacian filter (LLF)*, which uses a set of simple local filters to directly and ingeniously manipulate the Laplacian pyramid. However, the LLF is computationally costly because sub-pyramids must be constructed for each element of the Laplacian pyramid. Aubry et al.’s unpublished work [22] reveals that LLF is closely related to anisotropic diffusion and BLF, and provides an approximate sampling-based algorithm to accelerate LLF. However, for a desired accuracy, its ability to accelerate the method depends on the parameter settings used. Our method in this paper uses Paris et al.’s local filters, but merges them in a different, multi-scale optimization-based scheme. Unlike the above techniques, we do not directly manipulate either the pyramid or the wavelet coefficients.

Domain transform is a recent technique that proposed by Gastal and Oliveira [8] for real-time edge-aware image processing. This method is based on a data-dependent transform from 5D image manifold to 2D real space that keeps geodesic distance. However, like many other fast filtering techniques, domain transform is not rotationally invariant (i.e. filtering a rotated image and rotating a filtered image produce different results). Our approach is rotationally invariant.

B. Optimization-based image processing

Optimization is widely used in image processing, with a variety of objective functions.

Gradient domain image processing [3], [23], [24], [25], [26], [27], [28] all solve problems by manipulating image gradients via optimization instead of directly changing pixel values, taking advantage of the fact that the human visual system (HVS) is more sensitive to local contrast than absolute intensities. Fattal et al. [3] utilized a Gaussian pyramid to compute a gradient attenuation function for gradient domain HDR compression without halos. Similarly, we use a multi-scale scheme to produce halo-free results. Recently, Bhat et

al. [29] provided a uniform quadratic optimization framework, *Gradient Shop*, supporting a number of applications, such as saliency sharpening, pseudo re-lighting, non-photorealistic rendering, de-blocking, and colorization. However, its ability to obtain various effects relies on carefully designed spatially-varying weights for their constraints on gradients, and as a result, the algorithm can not be accelerated using frequency-domain approaches like those in [30]. We further explore the relation of our work to gradient domain techniques later in the paper.

Quadratic energies defined on image sub-windows have also been utilized for image matting [31] and haze removal [32]. However, these methods again require to solve large linear systems.

Other techniques such as total variation optimization [33] and L_0 smoothing [34] have considered non-quadratic energies in order to obtain outputs that have ‘sparse gradients’. However, these require elaborate iterative algorithms, which are generally slow and may lack mathematical guarantees of convergence and stability [35].

We give a novel multi-scale optimization-based approach to edge-aware image manipulation. At each scale, we optimize a quadratic objective function over sub-windows to merge a set of local edge-aware filters’ outputs. Unlike WLS, coefficients of our objective function have a spatially-invariant structure, which permits a fast frequency-domain solution to this optimization problem. Use of a multi-scale scheme helps to reduce halos.

III. ALGORITHM

In this section, we explain our algorithm for single-channel images only. As for color images, we decompose the input image into channels (in e.g. RGB space or CIELAB space), and process each channel independently. Note that this is a common strategy for processing color images [29], [36].

A. Overview

We express the input image as a real-valued 2D signal I .

Our algorithm allows the user to freely manipulate the *detailed* appearance and the *overall* appearance of I separately, without having to precisely specify how to decompose the image into these components. Following LLF, we use three parameters to control the output: α controls detail ($\alpha > 1$ implies detail enhancement and $\alpha < 1$ implies detail smoothing), β controls the overall appearance ($\beta > 1$ implies intensity range expansion and $\beta < 1$ implies range compression), and σ provides a threshold to determine what comprises detail.

To achieve the desired effects, we first create a Gaussian pyramid for I with l levels from bottom to top: I_0, I_1, \dots, I_{l-1} , where $I = I_0$. We then compute the output O gradually from top to bottom: $O_{l-1}, O_{l-2}, \dots, O_0$, where $O = O_0$.

In the following, we assume that I_k and O_k share a rectangular definition domain D_k , and the pixel values for I_k and O_k are denoted by $I_{k,p}$ and $O_{k,p}$ for each pixel $p \in D_k$.

For the top level, we simply scale the signal, relative to its average, to obtain the output:

$$O_{l-1,p} = \text{Avg}(I_{l-1}) + \beta(I_{l-1,p} - \text{Avg}(I_{l-1})) \quad (1)$$

where $\text{Avg}(I_{l-1})$ is the average pixel value over I_{l-1} . Note that the top level output has the desired properties for overall appearance, but lacks detail. However, we gradually correct the detail in the output as we proceed through the levels, as explained below.

For each $k < l - 1$, we compute O_k while taking two concerns into account.

On one hand, we would like O_k to be like the up-sampled signal of the output from the level above: $\text{Upsample}(O_{k+1})$. However, because of the blurring effects of *both* the down-sampling *and* up-sampling, they should differ by a Gaussian convolution. Thus, we would like the following to be satisfied:

$$G * O_k = \text{Upsample}(O_{k+1}) \quad (2)$$

where G is a Gaussian kernel and $*$ is the convolution operator. In the discrete setting, binomial kernels replace Gaussian kernels: we use a standard 5×5 binomial kernel for both the down-sampling and up-sampling processes [37], and so G here is a 9×9 binomial kernel.

On the other hand, we would like our result O_k to have the desired detail. We consider a neighboring sub-window w_p (not required to be a rectangle) for each pixel $p \in D_k$: w_p is composed of the pixels inside the disk that centered at p and radius to N . Within this sub-window, we use a carefully designed local filter that takes the parameters α, β, σ to provide an ideal output, and we desire that:

$$O_{k,q} = \text{LocalFilter}_{\alpha,\beta,\sigma,w_p}(I_{k,q}) + C_{k,p} \quad (3)$$

for every $p \in D_k$, and for each pixel $q \in w_p$. At first, it might seem that we need $O_{k,q} = \text{LocalFilter}_{\alpha,\beta,\sigma,w_p}(I_{k,q})$ for each $q \in w_p$. However, this term is meant to control the detail, which is essentially determined by local differences in values, or gradients, to which human perception is particularly sensitive [29]. Thus, we may add some constants over each sub-window w_p , as doing so does not affect the gradient; the $C_{k,p}$ above are these constants.

Our local filter design follows the ideas in [7] but with certain changes to better fit our algorithm, as explained in Subsection III-B. Our main algorithmic contribution is an optimization framework to trade-off the two requirements in Eqs. (2) and (3) (Subsection III-C), via a frequency-domain solution that is efficient and exact (Subsection III-D).

B. Local filtering

For each pixel p , $\text{LocalFilter}_{\alpha,\beta,\sigma,w_p}$ is actually a *filter bank*: it leaves the center pixel value $I_{k,p}$ unchanged, but generates a new value for each of the other pixels in the window centered on p . It treats each other pixel in a two-scale manner using threshold σ . Specifically, $\text{LocalFilter}_{\alpha,\beta,\sigma,w_p}(I_{k,q})$ generates the following output for each pixel $q \in w_p$:

$$I'_{k,p,q} = I_{k,p} + \text{sign}(I_{k,q} - I_{k,p})M_{\alpha,\beta,\sigma}(|I_{k,q} - I_{k,p}|) \quad (4)$$

where $M_{\alpha,\beta,\sigma}(d)$ is a non-linear map:

$$M_{\alpha,\beta,\sigma}(d) = \begin{cases} \sigma(d/\sigma)^{1/\alpha} & \text{if } d \leq \sigma, \alpha \leq 1 \\ \tau_d \sigma(d/\sigma)^{1/\alpha} + (1 - \tau_d)d & \text{if } d \leq \sigma, \alpha > 1 \\ (\sigma^2 + \beta^2(d^2 - \sigma^2))^{1/2} & \text{if } d > \sigma \end{cases} \quad (5)$$

Here, $\tau_d = \max(0, \min(1, 100 \times (d - 0.01)))$ to avoid noise amplification while enhancing detail. Examples of $M_{\alpha,\beta,\sigma}(d)$ are shown graphically in Fig. 3. The multiple values produced for each output pixel are resolved by our optimization step.

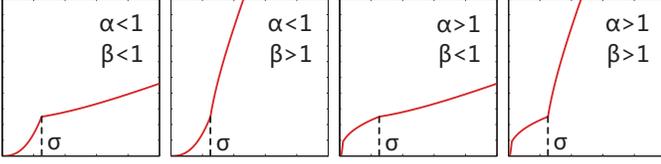


Fig. 3. Mapping function $M_{\alpha,\beta,\sigma}(d)$ for different α and β .

The only difference between our local filter in Eq. (4) and the one in [7] is that we take a hyperbolic curve instead of a straight line when $d > \sigma$. This ensures that, if the difference d is large, the mapping function $M_{\alpha,\beta,\sigma}(d)$ produces a result very close to βd (and does not differ from βd by a constant like the approach in [7]). As a result, our local filter better follows the overall appearance inherited from the top level, via Eq. (1).

C. Global optimization

We next formulate a global optimization scheme to trade-off Eqs. (2) and (3); it also reconciles the multiple outputs of Eq. (4). Our objective function has two terms.

The first measures directly the extent to which Eq. (2) is not satisfied. Its role is to control the overall appearance of the target O_k :

$$E_b(O_k) = \|G * O_k - \text{Upsample}(O_{k+1})\|^2 \quad (6)$$

For the second, we use ‘‘averaged squared differences’’ to express the requirement in Eq. (3):

$$E_d(O_k, C_k) = \sum_{p \in D_k} \sum_{q \in w_p} (O_{k,q} - I'_{k,p,q} - C_{k,p})^2 / W \quad (7)$$

where $I'_{k,p,q}$ comes from Eq. (4), and $C_{k,p}$ are the constants explained earlier. W is the number of pixels in w_p .

These two terms are combined to give an overall energy to be optimized:

$$E(O_k, C_k) = E_d(O_k, C_k) + \lambda E_b(O_k) \quad (8)$$

We simultaneously determine O_k and C_k by minimizing this energy. λ is a positive parameter; simply setting $\lambda = 1$ works well in practice (see later).

Note that Eq. (8) is a quadratic function, with a unique minimum, so optimizing it is equivalent to solving a linear equation. However, the coefficients of Eq. (8) have a spatially-invariant structure, unlike an arbitrary quadratic function. This observation allows us to instead use an efficient and exact frequency-domain method to optimize Eq. (8).

D. Frequency-domain solution

Suppose that O_k and C_k minimize the energy in Eq. (8). Therefore $\partial E / \partial C_{k,p} = 0$, which implies, for each $p \in D_k$:

$$C_{k,p} = \sum_{q \in w_p} (O_{k,q} - I'_{k,p,q}) / W \quad (9)$$

Algorithm 1 Mixed-domain edge-aware image manipulation

Input : $I, \alpha, \beta, \sigma, \lambda$.

Output: O .

Construct the Gaussian pyramid of I (with l levels).

Compute O_{l-1} according to Eq. (1).

for k from $l-1$ to 0 **do**

 Compute $I'_{k,p,q} = \text{LocalFilter}_{\alpha,\beta,\sigma,w_p}(I_{k,q})$ for each pixel $q \in w_p$.

 Compute S_k and T_k .

$DG \leftarrow \text{DCT}(G)$, $DL \leftarrow \text{DCT}(L)$.

$U_k \leftarrow \lambda G * \text{Upsample}(O_{k+1}) + T_k - L * S_k$.

$DU_k \leftarrow \text{DCT}(U_k)$.

for all frequencies f **do**

$\mathcal{D}O_{k,f} \leftarrow \mathcal{D}U_{k,f} / (\lambda(\mathcal{D}G_f)^2 + 1 - (\mathcal{D}L_f)^2)$.

end for

$O_k \leftarrow \text{InverseDCT}(\mathcal{D}O_k)$.

end for

$O \leftarrow O_0$.

By expanding the signals to cover the whole plane, we can rewrite Eq. (9) in the following form involving a convolution:

$$C_k = L * O_k - S_k \quad (10)$$

where L is the kernel for unweighted averaging within a centered sub-window, and S_k satisfies $S_{k,p} = \sum_{q \in w_p} I'_{k,p,q} / W$.

Also, $\partial E / \partial O_k = 0$, which yields:

$$O_k - T_k - L * C_k + \lambda G * (G * O_k - O'_k) = 0 \quad (11)$$

where $O'_k = \text{Upsample}(O_{k+1})$, and $T_{k,p} = \sum_{q \in w_p} I'_{k,q,p} / W$.

Taking a Discrete Cosine Transform (DCT), the convolutions in Eqs. (10) and (11) become multiplications in the frequency-domain. Substituting Eq. (10) into Eq. (11) to eliminate C_k , and denoting the DCT transform by $\mathcal{D}(\cdot)$, we obtain:

$$O_k = \mathcal{D}^{-1} \left(\frac{\lambda \mathcal{D}(G) \mathcal{D}(O'_k) + \mathcal{D}(T_k) - \mathcal{D}(L) \mathcal{D}(S_k)}{\lambda \mathcal{D}(G)^2 + 1 - \mathcal{D}(L)^2} \right) \quad (12)$$

which provides a direct and exact solution.

To extend the signals to the whole plane for use with DCT, we reflect signals about the image boundaries. This prevents artifacts at the boundaries (if we used with cyclic filling with DFT, signal values would leak from one side to the other). Note that using Fourier transform to speed up optimization is a standard technique (e.g. see [30] and [38]). However, unlike previous work, we have extra variables (the C_k) in our problem: we carefully eliminate these variables and obtain an explicit solution. Another feature of our solution is that we have intermediate variables (the S_k and T_k), which must be calculated in real space, so that our computation is done in a mixed-domain (real space and DCT transform space).

In detail, using Eq. (12), we can calculate O_k with just one 2-dimensional DCT and one 2-dimensional inverse DCT, using the observations that:

- The DCTs of G and L could be efficiently calculated. For G , we write $G = G_x * G_y$ where G_x and G_y are the 1-dimensional 9-tap binomial kernels in the x and y directions, giving $\mathcal{D}(G)_{x,y} = \mathcal{D}(G_x)_x \mathcal{D}(G_y)_y$. For

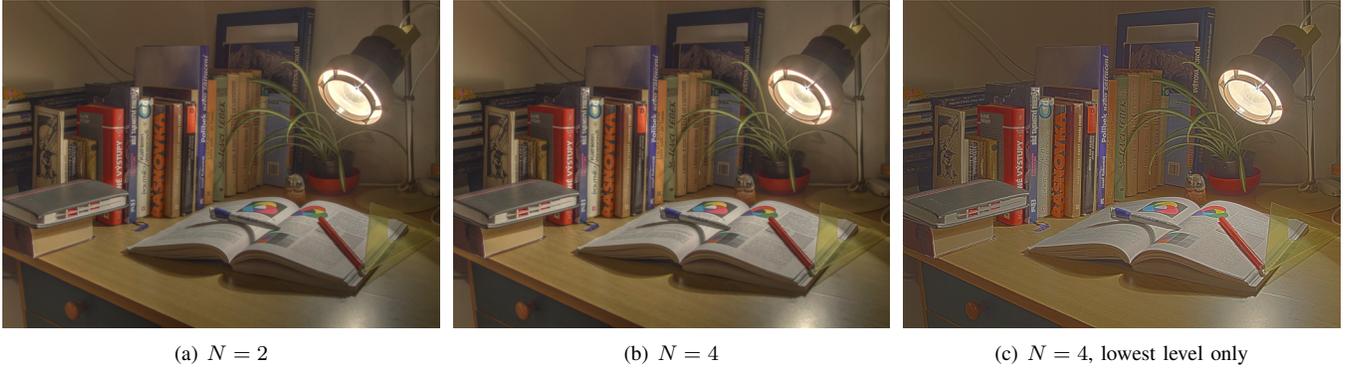


Fig. 4. HDR compression results. (a,b): $N = 2$ and $N = 4$, processing the whole pyramid. (c): $N = 4$, processing the lowest level only. All the results are with parameters $\alpha = 1.0$, $\beta = 0.1$, $\sigma = \ln(2.5)$.

L , we calculate each element of $D(L)$ according to the formal definition of DCT; this is also efficient because L has few identical non-zero elements. (These DCTs must actually be computed—although G and L are local, and the same for all images, their DCTs depend on the size of D_k .)

- We can first calculate an intermediate signal: $U_k = \lambda G * \text{Upsample}(O_{k+1}) + T_k - L * S_k$. This is again efficient because a convolution with G or L can be calculated via 1-dimensional convolutions. The numerator in Eq. (12) equals $\mathcal{D}(U_k)$ and thus can be calculated via a single 2-dimensional DCT.

Pseudocode for our entire mixed-domain edge-aware image manipulation is summarized in Algorithm 1.

IV. DISCUSSION

Since we use a quadratic objective function, our approach is closely related to gradient-domain image processing techniques. To see this, consider a 1×2 (or 2×1) window w with pixels q_1, q_2 . The energy term used to preserve details in this w would be:

$$E_{d,w}(O_{q_1}, O_{q_2}, C) = (O_{q_1} - I'_{q_1} - C)^2 + (O_{q_2} - I'_{q_2} - C)^2 \quad (13)$$

whose minimum over arbitrary constant C is achieved when $C = (O_{q_1} - I'_{q_1} + O_{q_2} - I'_{q_2})/2$, and the minimum is:

$$\begin{aligned} \min_C E_{d,w}(O_{q_1}, O_{q_2}, C) &= ((O_{q_1} - I'_{q_1}) - (O_{q_2} - I'_{q_2}))^2 / 2 \\ &= ((O_{q_1} - O_{q_2}) - (I'_{q_1} - I'_{q_2}))^2 / 2 \end{aligned} \quad (14)$$

This corresponds to requiring the gradient of the output, $O_{q_1} - O_{q_2}$ to be like the desired gradient, $I'_{q_1} - I'_{q_2}$, which is the basic principle used in gradient-domain image processing [29]. However, in our approach, we choose a set of larger sub-windows providing more extensive overlap, so that our output is ‘more consistent’. Bigger windows also provide more information allowing us to produce better local detail. Fig. 4 (a,b) demonstrate results of using our method for HDR compression using different sub-window sizes of $N = 2$ and $N = 4$. Note that the smaller N produces a dull result, while the larger N better provides the desired detail.

The use of a pyramid is important in our method to reduce halos. Fig. 4 (c) shows the result of our method for $N = 4$

if we only process the bottom level (i.e. we simply scale I_1 like Eq. (1) to obtain O_1 , and then solve a single optimization on the bottom level to obtain O_0). Unsightly halos appear near the edges of the desk, as well as the open book’s pages. These halos result from the combination of the constraints of both detailed and overall appearances. The output near an edge may be enhanced/reduced due to the detail constraint, but the output far away edges tends to be quite close to the “target base”: $\text{Upsample}(O_1)$; incompatibility of detail constraint and target base causes halos. Similar problems arise in WLS [4], where WLS simply uses input signal as the target base, and reduces halos by introducing data-dependent weights to the terms of detail constraint. Our approach does not rely on those complicated weights; halos are reduced due to the using of a series of carefully estimated target bases that better suit the detail constraint. These target bases are computed via a multi-scale process. However, the computational cost of our algorithm decreases rapidly at higher levels; processing the whole pyramid costs about $4/3$ the time of processing just the bottom level. For the remaining examples in this paper, we set $N = 4$, constructing the Gaussian pyramid such that the width and height of the top level image I_{l-1} are at least 8.

In our objective function in (8), λ is a positive parameter to trade-off the detail constraint and the overall constraint. Using a small λ produces better detailed appearance and worse overall appearance; an extremely small λ causes color tone deviation of the output (Fig. 5 (b)). Using a large λ produces better overall appearance and worse detailed appearance, and an extremely large λ causes ringing artifacts (Fig. 5 (c)). We set $\lambda = 1$ for all examples in this paper and it works well.

Our method and Aubry et al.’s work [22] are two different approaches to speed up local Laplacian filters. In some ways, they are complimentary—Aubry et al. use sample to efficiently estimate coefficients of the Laplacian pyramid, while we use an optimization-based framework to merge the local filters’ outputs level by level with a Gaussian pyramid. However, the ability of Aubry et al.’s approximation algorithm to achieve a speed up for a desired accuracy depends on the upper cutoff frequency of the mapping function $M_{\alpha,\beta,\sigma}$ (see [22]). It is faster than our method for e.g. HDR compression (where the parameter α is typically small and therefore the upper cutoff frequency of $M_{\alpha,\beta,\sigma}$ is low), however, for detail enhancement,

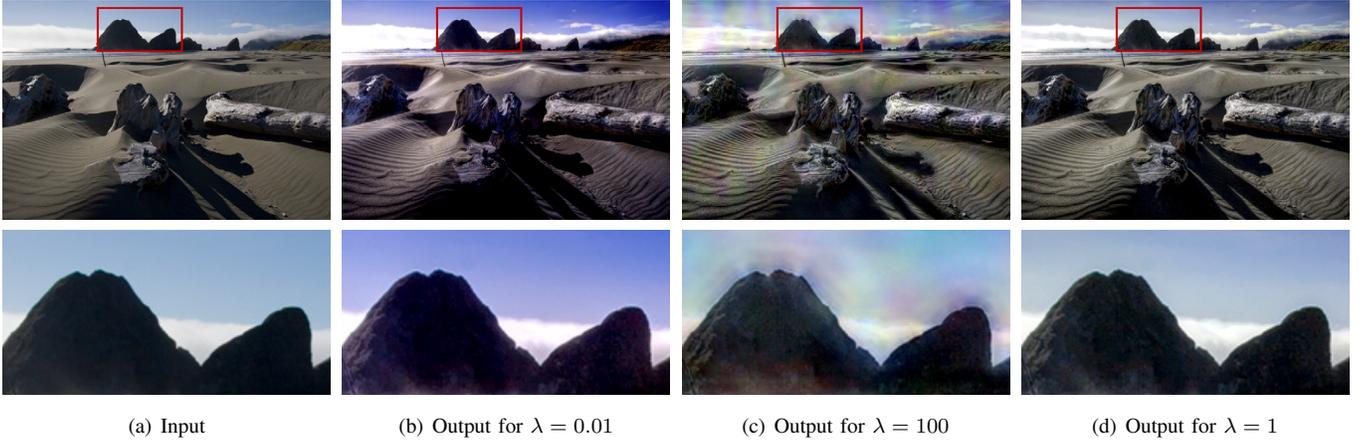


Fig. 5. Our detail enhancement results with different settings of λ . Extremely small λ leads to color tone deviation (b), while extremely large λ causes ringing artifacts (c). Setting $\lambda = 1$ works well in practice (d). All the results are with parameters $\alpha = 4.0, \beta = 1.0, \sigma = 0.2$.



Fig. 6. Rotationally invariance. (a): input images. (b): our detail smoothing results; $\alpha = 0.0, \beta = 1.0, \sigma = 0.15$.

where the parameter α is large, it would become much slower. The performance and accuracy of our method do not depend on the choosing of mapping function $M_{\alpha, \beta, \sigma}$.

Another feature of our algorithm is the rotationally invariance (i.e. filtering a rotated image and rotating a filtered image produce identical results), because we used disk-shaped sub-windows. This makes our method different from many other fast filtering techniques that are not rotationally invariant (e.g. domain transform [8]). An example is shown in Fig. 6.

V. APPLICATIONS

We now showcase several applications to demonstrate the effectiveness of our method. Further results can be found in the supplementary material.

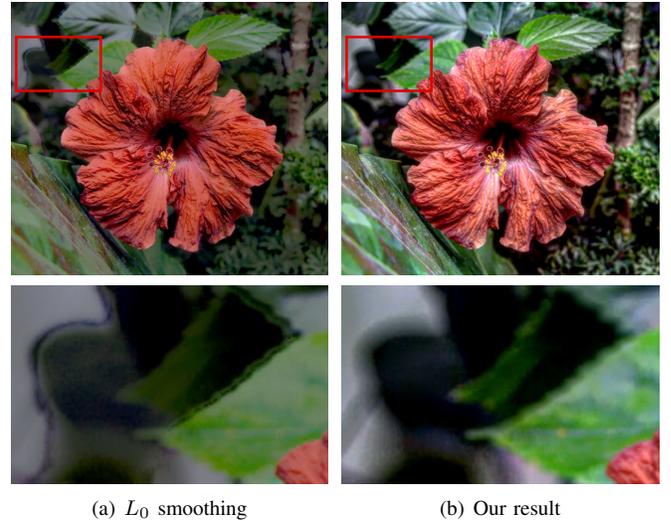


Fig. 7. Detail enhancement. (a): using L_0 smoothing; $\lambda = 0.015$; $3\times$ detail boosting. (b): our result; $\alpha = 4.0, \beta = 1.0, \sigma = 0.3$.

A. Detail and contrast manipulation

Our method can be directly applied for purposes of image detail and contrast manipulation. Fig. 8 shows the results of processing a single input image (in RGB color space) with different parameter settings. In each case, edges are preserved well without halos resulting (see, for example, the edges of the flower, and the stigma of the flower). The parameter α controls the detail in the output; $\alpha < 1$ causes detail smoothing and $\alpha > 1$ causes detail enhancement. The parameter β controls the color contrast of the output; $\beta < 1$ causes contrast reduction, while $\beta > 1$ leads to contrast enhancement. The parameter σ allows a tradeoff between detail and contrast manipulation: a larger σ increases the effects of α and decreases the effects of β . Thus, Fig. 8 (c,d) show stronger detail smoothing and enhancement than Fig. 8 (a,b); however, the contrast manipulation effects in Fig. 8 (c,d) are weaker than in Fig. 8 (a,b). σ is an important parameter which must be carefully chosen since it distinguishes details from edges. If we use $\alpha < 1$ to smooth an image with a large σ , e.g. as in



Fig. 8. Detail and contrast manipulation (in RGB color space). Top to Bottom: $\alpha = 0.3, 0.6, 1.5, 4.0$. (a,b): contrast reduction and enhancement with a normal threshold $\sigma = 0.1$. (c,d): contrast reduction and enhancement with an extreme threshold $\sigma = 0.3$.

the top-right of Fig. 8, some red color ‘leaks’ from the flower onto the green leaf. We suggest σ should lie within $[0.1, 0.2]$ for typical detail and contrast manipulation tasks.

In Fig. 7 we compare detail enhancement results using L_0 smoothing [34] with those from our method. Note that L_0 smoothing tends to produce smooth base components with sharp edges. Boosting from those components causes gradient reversals as discussed in Section I. Our method does not produce such artifacts.

B. HDR compression

HDR compression, or tone mapping, is concerned with compressing the intensity range of an HDR image while keeping details. Our implementation follows previous works,

i.e. we first compute the luminance channel using a linear combination of RGB values: $L_i = (20r_i + 40g_i + b_i)/61$ [16], and then process the logarithm of the luminance $\ln(L_i)$ [10], [3], using our method to compress the range without reducing detail, setting $\beta < 1$ and $\alpha \geq 1$. Given the output $\ln(L_o)$, we offset and scale $\ln(L_o)$ to make its maximum 0 and its minimum $-\ln(100/(\max(L'_i) - \min(L'_i)))$, where L'_i is the sub-signal of L_i obtained by throwing away its greatest and smallest 0.5% values [7]. Finally, we convert $\ln(L_o)$ back to L_o , and compute the output RGB values. We set $(r_o, g_o, b_o) = (r_i L_o / L_i, g_i L_o / L_i, b_i L_o / L_i)$, then gamma correct each channel with an exponent of $1/2.2$ [7], and finally clamp values to $[0, 1]$ for display.

In our experiments, we set $\beta = 0.1$, $\sigma = \ln(2.5)$; $\alpha = 1$ to give ‘photorealistic’ output and $\alpha = 4$ to give ‘exaggerated’

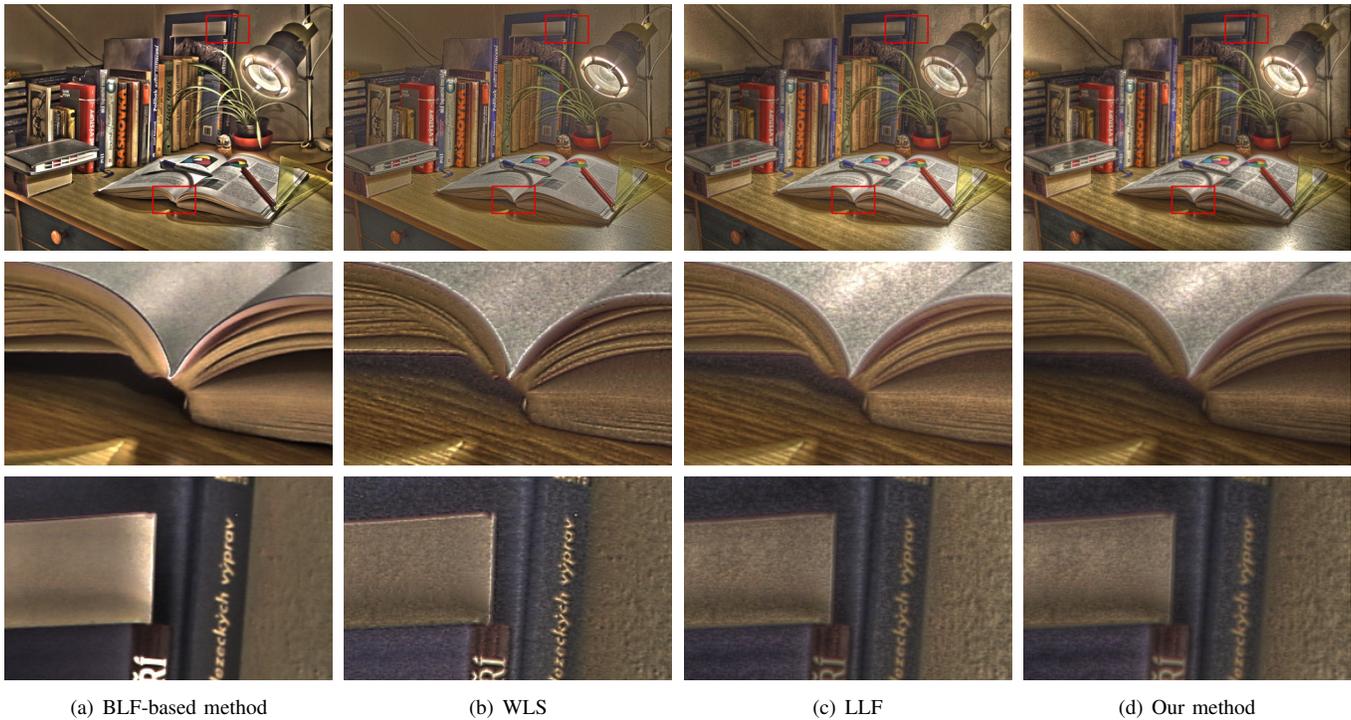


Fig. 9. Comparison of our method with previous HDR compression methods. (a): BLF [16]. (b): Weighted Least Squares [4]. (c): Local Laplacian Filter [7]. (d): our method; $\alpha = 4.0$, $\beta = 0.1$, $\sigma = \ln(2.5)$.



Fig. 10. HDR compression. (a): ‘photorealistic’ rendition with parameters $\alpha = 1.0$, $\beta = 0.1$, $\sigma = \ln(2.5)$. (b): ‘exaggerated’ rendition with parameters $\alpha = 4.0$, $\beta = 0.1$, $\sigma = \ln(2.5)$.

output. It would also be possible to post-process our results using interactive techniques like those in [39] but doing so is beyond the scope of this paper.

We compare our method with several previous methods (BLF-based method [16], WLS [4], and LLF [7]) in Fig. 9,

by setting parameters for ‘exaggerated’ effects. Note that the BLF-based method causes edge halos (e.g. see close-ups of the edges of the book’s pages, the corner of the shadow on the desk, and the white square on the blue book). WLS largely reduces halos, but some remain visible. In contrast, our method

produces halo-free output similar to that of the state-of-the-art LLF method. This is unsurprising, since both methods use almost the same local filters. However, our approach merges those local filters' multiple outputs via a novel optimization framework, with an exact and efficient mixed-domain solution, unlike LLF.

More 'photorealistic' and 'exaggerated' results using our method are shown in Fig. 10.

C. Non-photorealistic abstraction

The edge-aware smoothing ability of our method is well suited to the needs of image abstraction [36], [40]. In Fig. 11, we process an input image (in RGB color space) with parameters $\alpha = 0$, $\beta = 1$, $\sigma = 0.15$ to obtain a highly smoothed image, and then overlay this smoothed image with its DoG (Difference of Gaussian) edges to make a non-photorealistic abstraction effect.

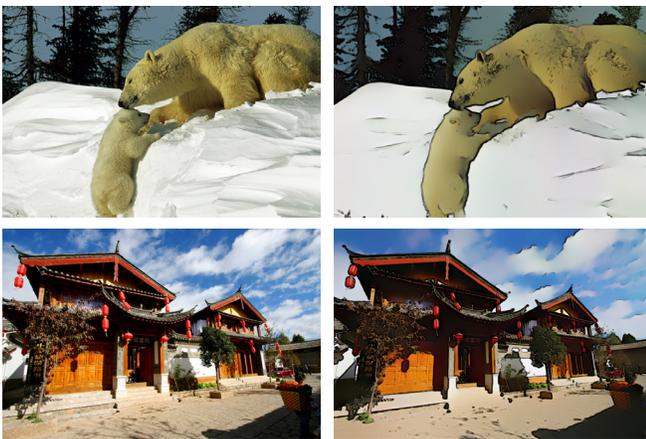


Fig. 11. Image Abstraction (use RGB color space). (a): input images. (b): our abstraction results; $\alpha = 0.0$, $\beta = 1.0$, $\sigma = 0.15$.

D. Haze removal

Our method can also be applied to *joint image filtering*, i.e. to process an image with the edge information provided by another reference image, denoted R . We simply replace the local filters in Eq. (4) by:

$$I'_{k,p,q} = I_{k,p} + (I_{k,q} - I_{k,p}) \frac{M_{\alpha,\beta,\sigma}(|R_{k,q} - R_{k,p}|)}{|R_{k,q} - R_{k,p}|} \quad (15)$$

We demonstrate haze removal as an application of this idea, in Fig. 12. Following [32], we first use the dark channel prior to roughly estimate a haze transmission map. We then use our method with the joint local filters in Eq. (15) to smooth the transmission map, taking the input hazy image as the reference image. As the reference is a color image, we use the Euclidean distance in CIELAB color space for the term $|R_{k,q} - R_{k,p}|$ in Eq. (15).

VI. IMPLEMENTATION

We have implemented our method on both the CPU and GPU. On the CPU, we used the FFTW 3.3 library [41] to compute DCTs. Our implementation takes less than 3.2 seconds to

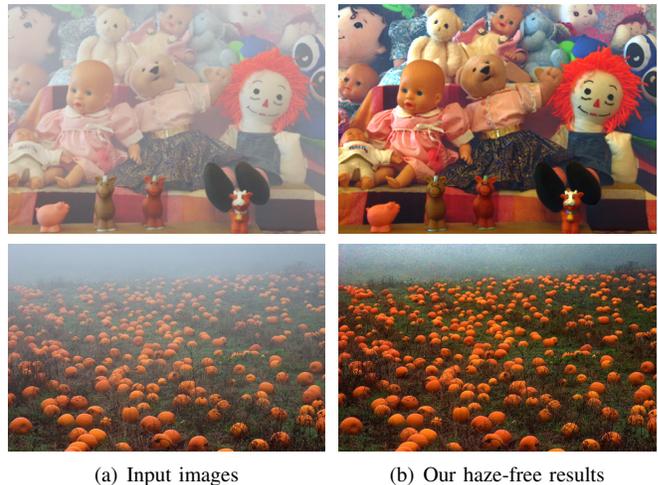


Fig. 12. Haze removal (use CIELAB color space for reference images). (a): input hazy images. (b): our haze-free results; $\alpha = 0.0$, $\beta = 1.0$, $\sigma = 20$.

process each channel of the 800×1200 image in Fig. 1, with an Intel Core 750 @ 2.66GHz using a single thread, with the sub-window size $N = 4$. Our GPU implementation was based on an NVIDIA GeForce 560 GTX card, using the NVIDIA CUFFT [42] for DCTs. The processing time was reduced to under 280 milliseconds.

Our algorithm is easy to program, since it involves only DCTs (widely supported by libraries) and a few pixel-wise operations, rather than complex solvers for large sparse linear systems (e.g. multi-grid methods) or complex signal decomposition.

VII. SUMMARY AND FUTURE WORK

This paper has presented a novel approach to edge-aware image manipulation, controlled by several intuitive parameters directly affecting the output's detailed and overall appearances. Our method processes a Gaussian image pyramid level-by-level from coarse to fine. At each level, a set of local filters is defined on overlapping sub-windows, and a spatial-domain optimization problem is solved in the frequency-domain to merge the local filters' outputs. This frequency-domain solution is exact, fast, and easy-to-implement. The results are halo-free, and can match the state-of-art results in several applications.

Our mixed-domain method is best suited to rectangular images; others can be padded to a rectangle at a slightly increased computational cost. A minor drawback of our method is that we can currently only support two-scale manipulation (detail versus overall appearance), as we have to process the whole pyramid to reduce halos. However, we believe that the idea of mixing spatial-domain and frequency-domain processing will have other useful applications in image processing for computer graphics. In particular, it would be interesting to adapt our frequency-domain solver to other more general optimization problems (e.g., we could use a linear transformation instead of the offsetting model in Eq. (7), like in [31]). We also hope to generalize our optimization approach to other spatially-varying local filters for further applications such as up-sampling and deblurring. Finally, extending our approach to

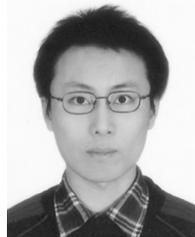
video by taking temporal consistency into account is a further problem of interest.

ACKNOWLEDGEMENTS

We thank all the reviewers for their carefully review and many helpful comments. Most of our source images in this paper are from other researchers' websites, we thank all of them for their great work; we thank Shanshan Lu for sharing the input image in Fig. 1(a); we thank Dr. Sylvain Paris for sharing their results in [7] for a comparison. This work was supported by Grant **BLABLABLA**. Xian-Ying Li was partially supported by the New PhD Researcher Award from Ministry of Education of China.

REFERENCES

- [1] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–846.
- [3] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 249–256, 2002.
- [4] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graphics*, vol. 27, no. 3, pp. 67:1–10, 2008.
- [5] R. Fattal, "Edge-avoiding wavelets and their applications," *ACM Trans. Graphics*, vol. 28, no. 3, pp. 22:1–10, 2009.
- [6] K. Subr, C. Soler, and F. Durand, "Edge-preserving multiscale image decomposition based on local extrema," *ACM Trans. Graphics*, vol. 28, no. 5, pp. 147:1–9, 2009.
- [7] S. Paris, S. W. Hasinoff, and J. Kautz, "Local laplacian filters: edge-aware image processing with a laplacian pyramid," *ACM Trans. Graphics*, vol. 30, no. 4, pp. 68:1–12, 2011.
- [8] E. S. L. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. Graphics*, vol. 30, no. 4, pp. 69:1–11, 2011.
- [9] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer-Verlag, 2006.
- [10] J. Tumblin and G. Turk, "LCIS: A boundary hierarchy for detail-preserving contrast reduction," in *SIGGRAPH '99: Proc. 24th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1999, pp. 83–90.
- [11] B. Weiss, "Fast median and bilateral filtering," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 519–526, 2006.
- [12] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Trans. Graphics*, vol. 26, no. 3, pp. 103:1–9, 2007.
- [13] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *Int. J. Computer Vision*, vol. 81, no. 1, pp. 24–52, 2009.
- [14] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time $O(1)$ bilateral filtering," in *IEEE CVPR*, 2009, pp. 557–564.
- [15] J. Baek and D. E. Jacobs, "Accelerating spatially varying gaussian filters," *ACM Trans. Graphics*, vol. 29, no. 6, pp. 169:1–10, 2010.
- [16] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 257–266, 2002.
- [17] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes," in *ACM SIGGRAPH 2005 Courses*. New York, NY, USA: ACM, 2005.
- [18] R. Fattal, M. Agrawala, and S. Rusinkiewicz, "Multiscale shape and detail enhancement from multi-light image collections," *ACM Trans. Graphics*, vol. 26, no. 3, pp. 51:1–9, 2007.
- [19] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "Bilateral filtering: Theory and applications," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 1, pp. 1–73, 2009.
- [20] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.
- [21] Y. Li, L. Sharan, and E. H. Adelson, "Compressing and companding high dynamic range images with subband architectures," *ACM Trans. Graphics*, vol. 24, no. 3, pp. 836–844, 2005.
- [22] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand, "Fast and robust pyramid-based image processing," *Technical Report*, pp. 1–11, 2011.
- [23] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 313–318, 2003.
- [24] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless image stitching in the gradient domain," in *ECCV '04: Proceedings of the Eighth European Conference on Computer Vision*. Springer-Verlag, 2004, pp. 377–389.
- [25] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 294–302, 2004.
- [26] A. Agarwala, "Efficient gradient-domain compositing using quadtrees," *ACM Trans. Graphics*, vol. 26, no. 3, pp. 94:1–10, 2007.
- [27] A. Agrawal and R. Raskar, "Gradient domain manipulation techniques in vision and graphics," in *ICCV 2007 Courses*. Washington, DC, USA: IEEE Computer Society, 2007.
- [28] J. McCann and N. S. Pollard, "Real-time gradient-domain painting," *ACM Trans. Graphics*, vol. 27, no. 3, pp. 93:1–7, 2008.
- [29] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, "Gradientshop: A gradient-domain optimization framework for image and video filtering," *ACM Trans. Graphics*, vol. 29, no. 2, pp. 10:1–14, 2010.
- [30] P. Bhat, B. Curless, M. Cohen, and C. L. Zitnick, "Fourier analysis of the 2d screened poisson equation for gradient domain problems," in *ECCV '08: Proceedings of the Eighth European Conference on Computer Vision*. Springer-Verlag, 2008, pp. 114–128.
- [31] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, 2007.
- [32] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," in *IEEE CVPR*, 2009, pp. 1956–1963.
- [33] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, 1992.
- [34] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L_0 gradient minimization," *ACM Trans. Graphics*, vol. 30, no. 6, pp. 174:1–12, 2011.
- [35] S. J. Wright, "Sparse optimization methods," in *Conference on Advanced Methods and Perspectives in Nonlinear Optimization and Control*, 2010.
- [36] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 1221–1226, 2006.
- [37] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans. Communication*, vol. 31, no. 4, pp. 532–540, 1983.
- [38] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graphics*, vol. 28, no. 5, pp. 145:1–7, 2009.
- [39] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, "Interactive local adjustment of tonal values," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 646–653, 2006.
- [40] D. DeCarlo and A. Santella, "Stylization and abstraction of photographs," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 769–776, 2002.
- [41] FFTW, "<http://www.fftw.org/>"
- [42] CUFFT, "<http://developer.nvidia.com/cufft>."



Xian-Ying Li received his bachelor's degree in computer science in 2008 from Tsinghua University, Beijing. He is now a PhD candidate at Tsinghua University. He received a gold medal of the International Mathematical Olympiad in 2004 when he was 16 years old. His research interests include computer graphics, geometric modeling, and computational origami.



Yan Gu is now an undergraduate student in the department of computer science and technology of Tsinghua University, Beijing. His research interests include computer graphics, geometric modeling, and rendering.



Shi-Min Hu received the PhD degree from Zhejiang University in 1996. He is currently a professor in the Department of Computer Science and Technology at Tsinghua University, Beijing. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He is associate Editor-in-Chief of *The Visual Computer* (Springer), and on the editorial boards of *Computer-Aided Design* and *Computer & Graphics* (Elsevier). He is a member of the IEEE and ACM.



Ralph R. Martin obtained his PhD in 1983 from Cambridge University. Since then he has been at Cardiff University, as Professor since 2000, where he leads the Visual Computing research group. He is also a Guest Professor at Tsinghua and Shandong Universities in China, and Director of Scientific Programmes of the One Wales Research Institute of Visual Computing. His publications include over 200 papers and 12 books covering such topics as solid modelling, surface modelling, reverse engineering, intelligent sketch input, mesh processing,

video processing, computer graphics, vision based geometric inspection, and geometric reasoning. He is a Fellow of the Learned Society of Wales, the Institute of Mathematics and its Applications, and the British Computer Society. He is on the editorial boards of “*Computer Aided Design*”, “*Computer Aided Geometric Design*”, “*Geometric Models*”, the “*International Journal of Shape Modeling*”, “*CAD and Applications*”, and the “*International Journal of CAD/CAM*”.