

# Edge-aware Gradient Domain Optimization Framework for Image Filtering by Local Propagation \*

Miao Hua<sup>1,2</sup>

Xiaohui Bie<sup>1,2</sup>

Minying Zhang<sup>1,2</sup>

Wencheng Wang<sup>1</sup>

<sup>1</sup>State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences

<sup>2</sup>University of Chinese Academy of Sciences

## Abstract

Gradient domain methods are popular for image processing. However, these methods even the edge-preserving ones cannot preserve edges well in some cases. In this paper, we present new constraints explicitly to better preserve edges for general gradient domain image filtering and theoretically analyse why these constraints are edge-aware. Our edge-aware constraints are easy to implement, fast to compute and can be seamlessly integrated into the general gradient domain optimization framework. The improved framework can better preserve edges while maintaining similar image filtering effects as the original image filters. We also demonstrate the strength of our edge-aware constraints on various applications such as image smoothing, image colorization and Poisson image cloning.

## 1. Introduction

In the context of image filtering, manipulation of content in an image is generally performed using gradient domain methods. As the name implies, gradient domain algorithms do not operate in the 0th order domain (i.e. color domain), but instead impose changes to the 1st order derivatives of the input image, i.e. the image gradient. This is because gradients are integral to the way in which human beings perceive images. It is widely believed that human cortical cells may be hard wired to preferentially respond to high contrast stimulus in their receptive fields [28], which directly correlate with gradients in an image. These gradient domain methods allow editing an image without introducing artifacts at the boundaries of the edited regions. Consequently, gradient domain methods are widely used in image processing for many applications including image smooth-

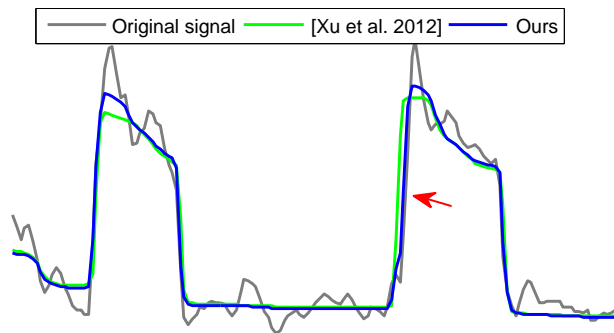


Figure 1. Signal obtained from an image scanline to illustrate the effect of our edge-aware constraints. As the red arrow indicates, with our edge-aware constraints, we can better preserve edges than [Xu et al. 2012] [35] while maintaining similar image filtering effect.

ing [11, 3, 34, 35], sparse data interpolation over images [21, 24], image inpainting [27], seamless cloning and compositing [26, 9] and matting [31].

A challenging problem in gradient domain image filtering is reducing, or if possible avoiding, artifacts at image edges. As already discussed by several researchers [23, 11, 22], both blurring and sharpening of edges may cause halos and gradient reversals. Consequently, various edge-preserving image filtering methods have been proposed, such as WLS optimization [11],  $L_0$  smoothing [34], structure extraction from texture via relative total variation [35] and so on. These gradient domain methods generally specify two constraints: zeroth-order constraints to get desired pixel values and first-order constraints to get desired pixel-gradients over space and time. There are no explicit constraints to treat edges in these methods, where the edge-preserving properties are usually implicitly achieved via the first-order constraints. As a result, in some cases, in order to get desired image filtering effect, these edge-preserving methods cannot preserve edges well, as shown in Figure 1. This is because image filtering is usually a image *coarsening* process accompanying with image smoothing. When these edge-preserving methods consider image filtering ef-

\*Corresponding author: Wencheng Wang. This work is supported in part by National Natural Science Foundation of China (61379087), the Knowledge Innovation Program of the Chinese Academy of Sciences, the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement (612627), and the National Social Science Foundation of China (12AZD118)

fect and edge-preserving effect together, edges may be smoothed inevitably.

To better preserve edges during filtering, we explicitly add first-order edge-aware constraints to the general gradient domain optimization framework. Our motivation originates from edit propagation [24, 25] which enforces the policy that similar edits are applied to spatially-close regions of similar appearance. The edge-aware constraints enforce similar image filtering effects to be propagated in local regions with similar appearance. We point out that the corresponding Laplacian matrix of the edge-aware constraints can be treated as a special case of the Laplacian matrix of KNN matting [6], thus theoretically prove its reasonability for edge preserving. The new image filter maintains the original filter properties and can better preserve edges, as shown in Figure 1. The added edge-aware constraints are fast to compute, easy to implement, and can be rewritten in the form of first-order gradient constraints, by which they can be seamlessly integrated into the original gradient domain optimization framework. We also revisit and improve upon various applications such as  $L_0$  smoothing [34] and structure extraction from texture via relative total variation [35], image colorization [24] and Poisson image cloning [26].

## 2. Related Work

**Gradient domain image filtering.** General gradient domain image processing can benefit from our added edge-aware constraints. A full review of all work in this aspect is outside the scope of the paper. Instead, we only survey the methods that are most relevant to our formulation. For a more extensive introduction to the gradient domain literature, the reader is referred to [1] and [3].

Gradient domain image filtering is very popular for image processing. One of the first work was tone-mapping [13]. It manipulated the gradient field of the luminance image by attenuating the magnitudes of large gradients, and then a new, low dynamic range image is obtained by solving a Poisson equation in the modified gradient domain. Using generic interpolation machinery based on solving Poisson equations, Perez et al. [26] introduced a variety of tools for seamless editing of image regions. In Poisson matting [31], the problem of natural image matting was formulated as one of solving Poisson equations with the matte gradient field. Levin et al. [21] demonstrated an gradient domain optimization approach for image colorization using a few user drawn color scribbles. This work was further generalized by Lischinski et al. [24] for interactive local tonal adjustment. They demonstrated that various user-specified local edits could be interpolated over the image in a piecewise-manner with respect to the underlying gradient field of the luminance image. Farbman et al. [11] advocated the use of weighted least squares optimization for multi-scale tone and

detail manipulation. Bhat et al. [3] presented the general gradient domain optimization framework for various image filtering tasks such as image relighting, non-photorealistic rendering, image de-blocking and image colorization. A similar optimization framework was introduced by Lang et al. [20], which introduced temporal consistency to a large class of optimization driven image-based computer graphics problems. Recently, many measures were presented to run in the gradient domain for image filtering. Xu et al. [34] presented a new image smoothing method for sharpening major edges by making use of  $L_0$  gradient minimization to globally control how many non-zero gradients were resulted in. New measure for inherent variation and relative total variation was proposed by Xu et al. [35] to extracted structure from texture.

**Edge-preserving smoothing.** To avoid artifacts in gradient domain image filtering, many edge-preserving smoothing methods have been proposed and used in a wide variety of computational photography applications [11]. Popular edge-preserving smoothing filters include the bilateral filter [32], for which a number of efficient implementations have been proposed [5, 36, 17, 15]. Subr et al. [30] proposed an edge-preserving smoothing approach based on identifying and fitting envelopes to local extrema in the image. Fattal [12] introduced edge-avoiding wavelets, and demonstrated their usefulness for very fast edge-preserving smoothing, as well as for edge-aware editing. Another class of edge-aware filtering used weights based on the geodesic-distance [8, 14], which performs pixel mixing inversely proportional to the distance over the  $\mathbf{IR}^5(\text{RGBXY})$  image manifold, as opposed to the  $L_2$  norm (used for bilateral filtering).

**Edge-aware interpolation.** Our edge-aware constraints are inspired by edit propagation, which enforce the policy that similar edits are applied to spatially-close regions of similar appearance. In this aspect, there have been many works. Levin et al. [21] and Lischinski et al. [24] edited images by smoothly interpolating sparse constraints across spatially contiguous image regions. Yatziv and Sapiro [37] propagate similar constraints by intrinsic distance based blending. AppWand approach [25] extended these methods to edit spatially- and temporally-varying measured materials. An and Pellacini [2] propagated the initial rough adjustments to the final refined ones by enforcing the editing policy over all pairs of points in the dataset. Chen et al. [7] used the locally linear embedding to represent each pixel as a linear combination of its neighbors in a feature space and sought to maintain the manifold structure formed by all pixels in the feature space. Farbman et al. [10] demonstrated the benefits of using diffusion distances in a variety of image editing contexts. Xu et al. [33] proposed to accelerate the all-pairs approach by computing the propagation between pairs of clusters in the feature space, rather than between pairs of pixels. Bie et al. [4] further used clustering and

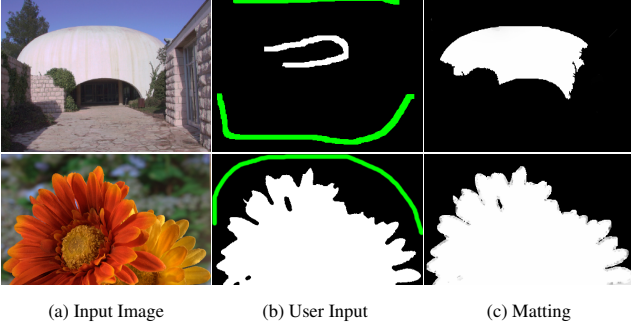


Figure 2. Matting using the Laplacian matrix of our edge-aware constraints.

efficient sampling techniques together to achieve real time edit propagation.

### 3. Edge-Aware Constraints

In this section we introduce our constraints and theoretically analyse why they are edge-aware.

**General gradient domain optimization framework.** Following the description in [3], we first briefly review the general gradient domain optimization framework for image processing. The task of an image filter is to transform the input image into the final image, and can be expressed as an energy function involving its zeroth and first order terms. Suppose  $f$  and  $u$  represent the sets of the pixels in the target and the input images respectively. That is, the final result  $f$  is generated by minimizing the following energy function:

$$E(f) = \sum_{p \in f} E_d(p) + E_g(p) \quad (1)$$

where  $p$  is a pixel of  $f$ ,  $E_d$  is the data cost function, and  $E_g$  is the gradient cost function. These two terms are quadratic functions defined as follows:

$$E_d(p) = w^d(p)[f(p) - d(p)]^2$$

and

$$E_g(p) = w^x(p)[f_x(p) - g^x(p)]^2 + w^y(p)[f_y(p) - g^y(p)]^2$$

where  $w^d$ ,  $w^x$  and  $w^y$  are per-pixel weights for constraints,  $d$  provides the data constraint for each pixel in  $f$ ,  $f_x$  and  $f_y$  denote the  $x$  and  $y$  derivative of  $f$ ,  $g^x$  and  $g^y$  specify the desired  $x$ -derivative and  $y$ -derivative of  $f$  respectively. Thus, the data and gradient energy terms are the squared errors between the desired values specified by  $d$ ,  $g$  and the actual values of the final image. As shown later, several effects can be achieved by varying these terms, including image smoothing, image colorization and Poisson image cloning.

**Edge-aware constraints.** Here we introduce our edge-aware constraints. The above energy function specifies the

squared errors between the desired values and the actual values of  $f$ . Per-pixel weights control the amount of influence that a constraint should have on the final image. However, there are no explicit constraints to prevent the image filter effect not to propagate to adjacent similar objects, and so edges cannot be well preserved, such as shown in the 1D example of Figure 1 and the application of image colorization in Figure 6. Inspired by edit propagation which enforces that similar edits are applied to spatially-close regions of similar appearance, we add edge-aware constraints explicitly to the energy function, as in the below:

$$E(f) = \sum_{p \in f} E_d(p) + E_g(p) + \gamma E_e(p)$$

where  $\gamma$  is for tradeoff between edge-aware constraints and original constraints, and  $E_e$  are the edge-aware constraints defined as:

$$E_e(p) = \sum_{q \in N_4(p)} w^e(p, q)[(f(p) - u(p)) - (f(q) - u(q))]^2$$

where  $N_4(p)$  is the set of 4-connected neighbors of pixel  $p$ ,  $w^e$  are weights defined as gaussians of the distance between adjacent pixels in the Lab color space:

$$w^e(p, q) = \exp(-\|u(p) - u(q)\|^2 / \sigma^2) \quad q \in N_4(p) \quad (2)$$

Note that other definitions of the weights can also be used, such as using diffusion distance [10] instead of Euclidean distance. Let  $e = f - u$ , measuring the changes by image filtering. The main purpose of the edge-aware constraints is to propagate the changes in local neighborhoods so that the image filter can cause similar effects in regions of similar appearance. Lischinski et al. [24] and Pellacini and Lawrence [25] used similar constraints to propagate edits, but they did not theoretically analyse why their constraints are edge-aware. In this paper, we prove that this is because the corresponding Laplacian matrix of the constraints is a special case of the Laplacian matrix of KNN matting [6]. The proof is denoted as follows:

$$\begin{aligned} \sum_{p \in f} E_e(p) &= \sum_{p \in f} \sum_{q \in N_4(p)} w^e(p, q)[(f(p) - u(p)) - (f(q) - u(q))]^2 \\ &= \sum_{p \in f} \sum_{q \in N_4(p)} w^e(p, q)[e(p) - e(q)]^2 \\ &= \sum_{p \in f} \sum_{q \in N_4(p)} w^e(p, q)[e(p)^2 - 2e(p)e(q) + e(q)^2] \\ &= 2 \sum_{p \in f} \sum_{q \in N_4(p)} w^e(p, q)[e(p)^2 - e(p)e(q)] \end{aligned}$$

The last equation is true since  $w^e(p, q) = w^e(q, p)$ , if we set  $w^e(p, q) = 0$  when pixel  $q \notin N_4(p)$ , then the above

equation can be rewritten as:

$$\begin{aligned}
\sum_{p \in f} E_e(p) &= 2 \sum_{p \in f} \sum_{q \in N_4(p)} w^e(p, q) [e(p)^2 - e(p)e(q)] \\
&= 2 \sum_{p \in f} \sum_{q \in f} w^e(p, q) [e(p)^2 - e(p)e(q)] \\
&= 2 \left( \sum_{p \in f} e(p)^2 \sum_{q \in f} w^e(p, q) - \sum_{p \in f} \sum_{q \in f} w^e(p, q) e(p)e(q) \right) \\
&= 2 \mathbf{e}^T (\mathcal{D} - \mathcal{A}) \mathbf{e}
\end{aligned}$$

where  $\mathbf{e}$  is the vector representation of  $e$ ,  $\mathcal{A} = [w^e(p, q)]$  is an  $N \times N$  affinity matrix and  $\mathcal{D} = \text{diag}(\mathcal{D}_p)$  is an  $N \times N$  diagonal matrix, where  $N$  is the total number of pixels and  $\mathcal{D}_p = \sum_{q \in f} w^e(p, q)$  is the sum of the  $p$ th column of matrix  $\mathcal{A}$ .

KNN matting [6] capitalizes on the nonlocal principle by using  $K$  nearest neighbors (KNN) in matching nonlocal neighborhoods, and contributes a simple and fast algorithm to give competitive results with sparse user markups. In KNN matting, if we choose  $K$  ( $K = 4$ ) nearest neighbors for each pixel and assume the  $K$  nearest neighbors are 4-connected neighbors of that pixel, then the Laplacian matrix  $L_e = \mathcal{D} - \mathcal{A}$  of the edge-aware constraints is exactly the Laplacian matrix of KNN matting. As a result, our new constraints are edge-aware.

Figure 2 shows the matting results using the Laplacian matrix of our edge-aware constraints. Although our corresponding Laplacian matrix is just a special case of the Laplacian matrix of KNN matting, it can produce satisfactory matting results.

**Computation cost and optimization.** Our new added edge-aware constraints can be categorized into the gradient domain framework. This is because:

$$\begin{aligned}
E_e(p) &= \sum_{q \in N_4(p)} w^e(p, q) [(f(p) - u(p)) - (f(q) - u(q))]^2 \\
&= \sum_{q \in N_4(p)} w^e(p, q) [(f(p) - f(q)) - (u(p) - u(q))]^2
\end{aligned}$$

Since  $q \in N_4(p)$ ,  $f(p) - f(q)$  is one of the discrete forms of  $\{f_x(p), -f_x(q), f_y(p), -f_y(q)\}$ . Thus our edge-aware constraints can be seamlessly integrated into the original gradient domain optimization framework. Our edge-aware constraints are easy and fast to implement, in a complexity of  $O(N)$ .

The improved gradient domain optimization framework involves a linear system with a 5-point spatially inhomogeneous sparse Laplacian matrix, which can be solved using standard, weighted least-squares techniques like the conjugate-gradient method [29]. For a fast solving, many efficient techniques can be adopted, as introduced in the following. Krishnan and Szeliski [19] unifies multigrid and multilevel (hierarchical) preconditioners, two widely-used

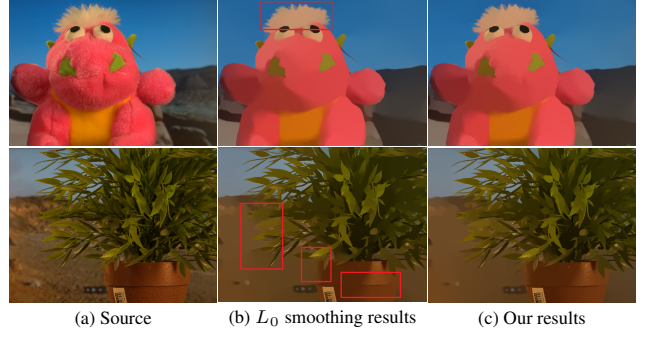


Figure 3. Visual comparison of  $L_0$  smoothing to our method. As the red boxes show, our method can better preserve edges than  $L_0$  smoothing, even the small and tiny edges such as hair, while maintaining similar image filtering effects. This figure is best viewed in electronic form and zoomed.

approaches for solving the linear system. Recently, a new multi-level preconditioning scheme with linear time and memory requirements for discrete Poisson equations was proposed by [18], which interleaved the selection of fine- and coarse-level variables with the removal of weak connections between potential fine-level variables and the compensation for these changes by strengthening nearby connections.

## 4. Applications

Our edge-aware constraints can be added to any gradient domain optimization framework. In this section, we apply our edge-aware gradient domain optimization framework in various image processing applications as examples. We also compare them with original image processing results to show our advantages.

### 4.1. Image Smoothing

As mentioned in related work, there are various gradient domain methods for image smoothing. Here we just list two ones as examples to demonstrate the effect of our edge-aware constraints.

**$L_0$  smoothing.** Xu et al. [34] proposed an optimization framework making use of  $L_0$  gradient minimization. It is by solving the following energy function:

$$\sum_{p \in f} (f(p) - u(p))^2 + \lambda C(f) \quad (3)$$

where  $C(f) = \#\{p | |f_x(p)| + |f_y(p)| \neq 0\}$ , counting the pixel  $p$  whose magnitude  $|f_x(p)| + |f_y(p)|$  is not zero. Since Eq. (3) is not the standard gradient domain optimization framework as Eq. (1), Xu et al. used a splitting scheme to solve it by splitting the optimization into two processes and iterating between these two processes: one is the standard gradient domain optimization framework and the



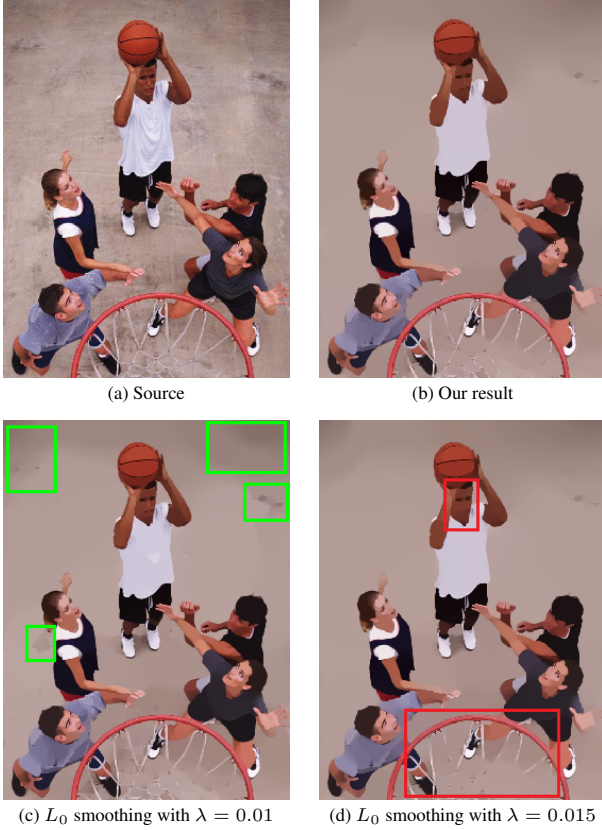


Figure 4. Visual comparison of our method to  $L_0$  smoothing with different parameters.  $L_0$  smoothing cannot smooth the background well when using smaller  $\lambda$ , as shown in the green boxes in (c), and cannot maintain structures well when using larger  $\lambda$ , as shown in the red boxes in (d). Our method can simultaneously achieve both effects well.

other one is a simple voting process. As described before, our edge-aware constraints can be seamlessly integrated into the gradient domain optimization framework and have no influence in the voting process, thus the new energy function formed by adding our edge-aware constraints to Eq. (3) can be solved using the same splitting scheme as Xu et al..

Figure 3 shows the visual comparison of  $L_0$  smoothing to our method. Our method can better preserve edges even the small, tiny edges such as hair shown in the first row of Figure 3, while maintaining the similar image filtering results as  $L_0$  smoothing.

As shown in Figure 4, our edge aware effects cannot be achieved with  $L_0$  smoothing via adjustment of parameters. To maintain the face and basket structures, a smaller parameter  $\lambda = 0.01$  should be used in Eq. (3) with  $L_0$  smoothing, but this cannot smooth the background well, as shown in the green boxes in Figure 4(c). On the contrary, using a large  $\lambda = 0.015$  to smooth the background well, the face and basket structures cannot be maintained using  $L_0$  smoothing, as

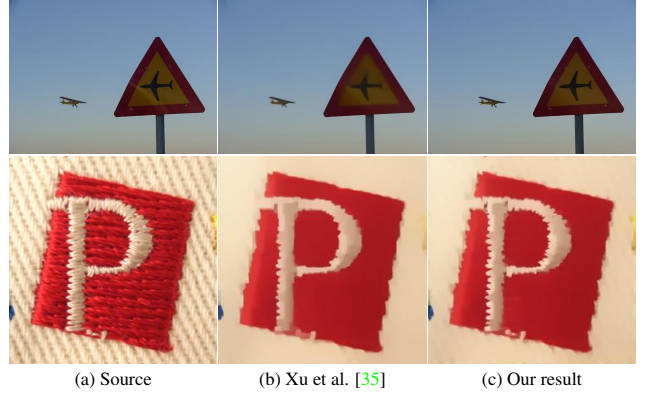


Figure 5. Visual comparison of Xu et al. [35] to our method. Our method can better preserve structure edges than Xu et al. [35], even the tiny edges shown in the second row, while maintaining similar image filtering effects.

shown in the red boxes in Figure 4(d). As for our method, it can simultaneously smooth the background and maintain structures well.

**Structure extraction from texture.** Xu et al. [35] proposed new measures for inherent variation and relative total variation, which capture the essential difference of these two types of visual forms, and developed an efficient optimization system to extract main structures. As  $L_0$  smoothing, [35] is not a standard gradient domain optimization framework, Xu et al. approximated the relative total variation measures and transformed the optimization process into the framework as Eq. (1). Clearly, our edge-aware constraints can be seamlessly integrated into the optimization framework of [35].

Figure 5 shows the visual comparison of [35] to our method. Our method can better preserve structure edges even the tiny edges shown in the second row of the figure, while maintaining similar image filtering effects as [35].

## 4.2. Image Colorization

Levin et al. [21] proposed a gradient-domain method for interpolating colors from a sparse set of user-drawn color scribbles over a grayscale image to transform it into a photorealistic color image. Lischinski et al. [24] observed that Levin’s method was in fact a general and powerful technique for interpolating sparse data over images, and showed that a variety of data types including tonal values, blurring amounts, and white balance corrections can also be locally adjusted using Levin’s method. As shown in [3], Lischinski’s method can be easily mapped to the general gradient domain formulation:

$$w^d(p) = \begin{cases} \infty & \text{if } d(p) \text{ is defined} \\ 0 & \text{otherwise} \end{cases}$$

$$g^x(p) = 0; g^y(p) = 0$$

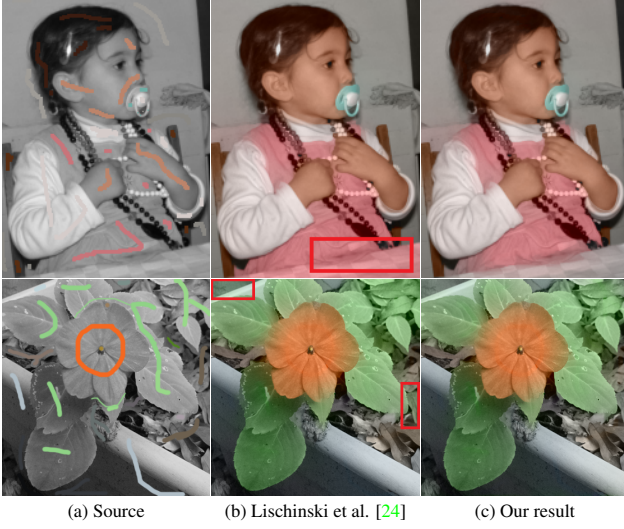


Figure 6. Visual comparison of image colorization results of Lischinski et al. [24] to our method. Our method can better prevent color strokes from bleeding into adjacent regions with similar appearance, as shown in the red boxes in the middle column. This figure is best viewed in electronic form and zoomed.

$$E_g(p) = \frac{f_x(p)^2}{|L_x(p)|^\alpha + \varepsilon} + \frac{f_y(p)^2}{|L_y(p)|^\alpha + \varepsilon}$$

where the data constraint  $d$  contains the user data (e.g., scribbles, paint strokes), the weights  $w^d$  of the data constraints encourage the result to maintain fidelity to the user input,  $L_x$  and  $L_y$  are the  $x$ -derivative and  $y$ -derivative of the log-luminance channel  $L$  respectively,  $\alpha$  and  $\varepsilon$  are controlling parameters. Clearly, our edge-aware constraints can be seamlessly integrated into the optimization framework.

Figure 6 shows the visual comparison of image colorization results of [24] to our method. As shown in the red boxes in the middle column of the figure, our improvement significantly reduces the amount of data bleeding in the result (or conversely the number of user strokes required to produce the desired result).

### 4.3. Poisson Image Cloning

Image composition is the process of creating a new image by pasting an object or a region from a source image onto a target image. Poisson image cloning [26] has been demonstrated an effective approach for seamless image composition. By solving Poisson equations using the user-specified boundary condition, Poisson image cloning seamlessly blends the colors from both images without visible discontinuities around the boundary of the source image. We first show that Poisson image cloning can be mapped to the general gradient domain optimization framework and then demonstrate that our edge-aware constraints can improve Poisson image cloning.

To paste a region of interest from the source image  $u$  to the target image  $f^*$  and to get the final image  $f$ , Poisson image cloning solves the following minimization problem:

$$\min_f \int_{p \in \Omega} |\nabla f - \mathbf{v}|^2 dp \quad \text{with} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (4)$$

where  $\mathbf{v} = \nabla u$  is the guidance field for seamlessly image cloning,  $\Omega$  is the user-drawn region of interest and  $\partial\Omega$  is the exterior boundary of  $\Omega$ . For discrete images, the problem can be discretized using the underlying discrete pixel grid. Without loss of generality, suppose  $S$  and  $\Omega$  represent the pixels of the target and the source images respectively. For each pixel  $p$  in  $S$ , let  $N_4(p)$  be the set of its 4-connected neighbors, and let  $\langle p, q \rangle$  denote a pixel pair with  $q \in N_4(p)$ . The boundary of  $\Omega$  is now  $\partial\Omega = \{p \in S \setminus \Omega : N_4(p) \cap \Omega \neq \emptyset\}$ . Let  $f_p$  be the value of  $f$  at  $p$ , then the finite difference discretization of Eq. (4) yields the following discrete, quadratic optimization problem:

$$\min_{f|_{\Omega}} \sum_{p \in S} w^d(p) (f_p - f_p^*)^2 + \sum_{\langle p, q \rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2 \quad (5)$$

where  $\langle p, q \rangle \cap \Omega \neq \emptyset$  is the set of the pixel pair  $\langle p, q \rangle$  that at least one of  $p$  and  $q$  belong to  $\Omega$ ,  $v_{pq}$  is the projection of  $\mathbf{v}(\frac{p+q}{2})$  on the oriented edge  $[p, q]$ . Here  $v_{pq} = u_p - u_q$ , with  $u_p, u_q$  being the values of  $u$  at  $p, q$ , and

$$w^d(p) = \begin{cases} \infty & \text{if } p \in \partial\Omega \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The weights  $w^d$  encourage the result to maintain fidelity to the boundary condition in Eq. 4. As energy function (5) could be treated as the discrete form of the general gradient domain optimization framework (1), it can be solved fast for interactive editing of medium-sized color image regions.

**Improvement:** Poisson image cloning is very effective for image composition. However, in some cases it may cause the color bleeding problem: the colors of boundary pixels in the target image will bleed into the source image to change the appearance of the objects in the source image, as shown in Figure 7(c). Our edge-aware constraints can effectively prevent such color bleeding. To further alleviate color bleeding artifacts, we add another zeroth order data constraints to preserve original colors in the source image, that is, we minimize the following energy function:

$$\begin{aligned} \min_{f|_{\Omega}} & \sum_{p \in S} w^d(p) (f_p - f_p^*)^2 + \sum_{\langle p, q \rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2 + \\ & \alpha \left( \sum_{p \in \Omega} s^d(p) (f_p - u_p)^2 + \right. \\ & \left. \beta \sum_{p \in \Omega} \sum_{q \in N_4(p)} w^e(p, q) [(f_p - u_p) - (f_q - u_q)]^2 \right) \end{aligned} \quad (7)$$

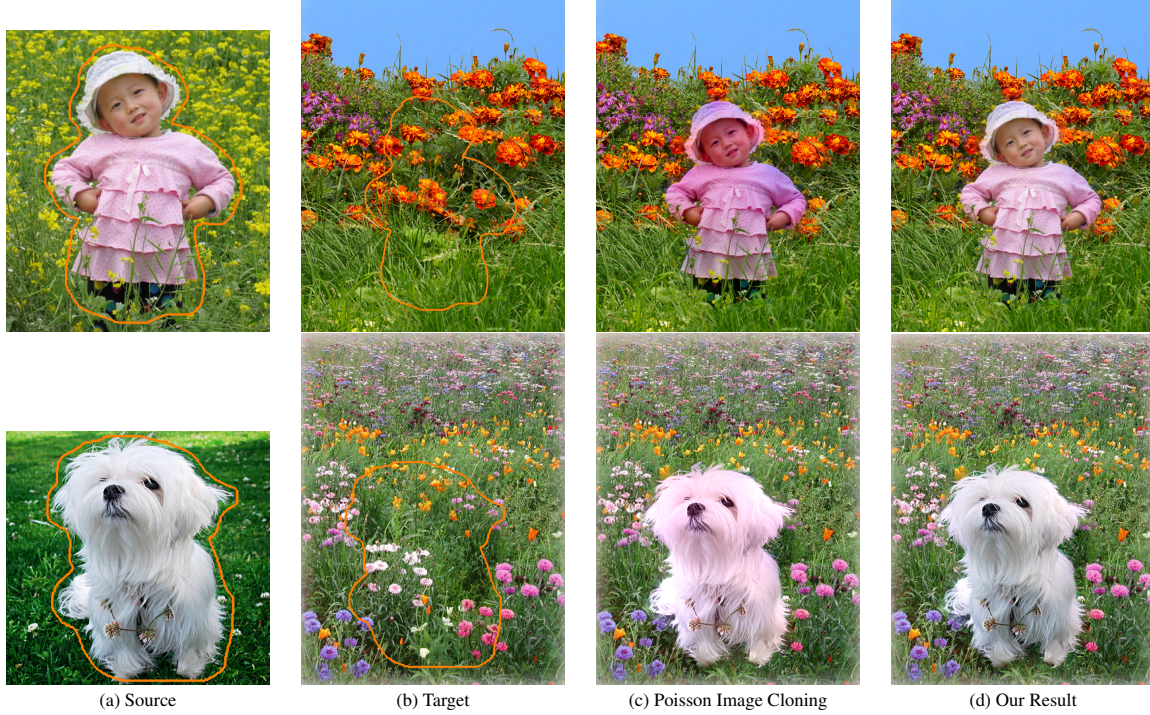


Figure 7. Visual comparison of Poisson image cloning to our method. Poisson image cloning may cause color bleeding problem, as shown in (c), while our method produces more photo-realistic results.

where  $w^e(p, q)$  and  $w^d(p)$  are defined as in Eq. (2) and (6) respectively, and  $s^d(p)$  is the context-aware saliency value of  $p$  for detecting the image regions that represent the scene [16].  $\alpha$  and  $\beta$  are two weights. When  $\alpha = 0$ , the above function returns to the original Poisson image cloning.

Note that besides the edge-aware constraints, we additionally add another zeroth order data constraints. If we treat the image cloning process as some kind of image filter, the new added data constraints perform like the original data constraints in Eq. (1), which enforce the resulting image being similar with the original image. As a result, from the point of image filtering, Eq. (7) becomes the general gradient domain optimization framework with edge-aware constraints and user input sparse data constraints (i.e., the first item of Eq. (7)). Thus our new approach can effectively alleviate the artifacts from color bleeding.

Figure 7 shows the visual comparison of Poisson image cloning to our method. Original Poisson image cloning may generate severe color bleeding artifacts, as shown in the third column of the figure. Colors on the boundary bleed into the child and dog, which makes image composition unreal. Our method can significantly alleviate such artifacts and produces more natural compositions.

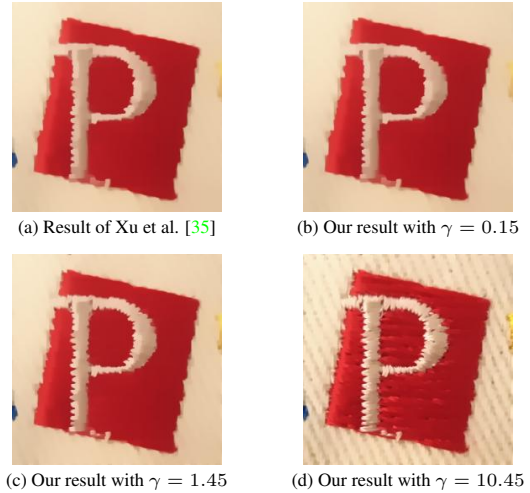


Figure 8. Results of structure extraction from texture with different  $\gamma$ . Please refer to the lower left of Figure 5 for the input.  $\gamma$  should be selected suitably, otherwise it cannot preserve edges well, as shown in (b), or it may contain undesired edges for image filtering, as shown in (d).

## 5. Evaluation

Our edge-aware constraints are fast to compute, and the overall optimization framework can perform nearly as fast as the original image filter. For example, on a PC with an



Intel i7 3.40GHz CPU and 4GB memory, Xu et al. [35] takes 3.7 seconds to process an  $800 \times 600$  color image, while our method takes 3.9 seconds to process the same image on the same PC.

We evaluate our method with different weights  $\gamma$  in Figure 8, which control the tradeoff between edge-aware constraints and original gradient domain constraints. Typically,  $\gamma$  should be selected properly (Figure 8(c)), since small  $\gamma$  value cannot preserve edges well, as shown in Figure 8(b), and large  $\gamma$  value may introduce undesired image filtering effect (Figure 8(d)). By experiences, we choose  $\gamma$  in [1, 10] in our tests, by which good results can be obtained in general.

**Limitation.** The Laplacian matrix of our edge-aware constraints is a special case of the Laplacian matrix of KNN matting. In extreme cases, they cannot preserve edges well. Fortunately, as shown by Farbman et al. [10], such a problem can be well treated by using diffusion distance instead of Euclidean distance when computing weights  $w^e(p, q)$  in Eq. (2). Of course, this will cost much more time.

## 6. Conclusion

We present edge-aware constraints for improving the general gradient domain optimization framework and attest the superiority over existing methods in edge preserving. We demonstrate the effectiveness of our method in various applications including image smoothing, image colorization and Poisson image cloning.

## References

- [1] A. Amit and R. Ramesh. Gradient domain manipulation techniques in vision and graphics. 2007. 2
- [2] X. An and F. Pellacini. Approp: all-pairs appearance-space edit propagation. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 40:1–40:9, New York, NY, USA, 2008. ACM. 2
- [3] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Trans. Graph.*, 29(2):10:1–10:14, Apr. 2010. 1, 2, 3, 5
- [4] X. Bie, H. Huang, and W. Wang. Real time edit propagation by efficient sampling. *Comput. Graph. Forum*, 30(7):2041–2048, 2011. 2
- [5] J. Chen, S. Paris, and F. Durand. Real-time edge-aware image processing with the bilateral grid. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. 2
- [6] Q. Chen, D. Li, and C.-K. Tang. Knn matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pages 869–876, June 2012. 2, 3, 4
- [7] X. Chen, D. Zou, Q. Zhao, and P. Tan. Manifold preserving edit propagation. *ACM Trans. Graph.*, 31(6):132:1–132:7, Nov. 2012. 2
- [8] A. Criminisi, T. Sharp, C. Rother, and P. Pérez. Geodesic image and video editing. *ACM Trans. Graph.*, 29(5):134:1–134:15, Nov. 2010. 2
- [9] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image Melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2012)*, 31(4):82:1–82:10, 2012. 1
- [10] Z. Farbman, R. Fattal, and D. Lischinski. Diffusion maps for edge-aware image editing. In *ACM SIGGRAPH Asia 2010 papers*, SIGGRAPH ASIA '10, pages 145:1–145:10, New York, NY, USA, 2010. ACM. 2, 3, 8
- [11] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 67:1–67:10, New York, NY, USA, 2008. ACM. 1, 2
- [12] R. Fattal. Edge-avoiding wavelets and their applications. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 22:1–22:10, New York, NY, USA, 2009. ACM. 2
- [13] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 249–256, New York, NY, USA, 2002. ACM. 2
- [14] E. S. L. Gastal and M. M. Oliveira. Domain transform for edge-aware image and video processing. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 69:1–69:12, New York, NY, USA, 2011. ACM. 2
- [15] E. S. L. Gastal and M. M. Oliveira. Adaptive manifolds for real-time high-dimensional filtering. *ACM Trans. Graph.*, 31(4):33:1–33:13, July 2012. 2
- [16] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, pages 2376–2383. IEEE, 2010. 7
- [17] K. He, J. Sun, and X. Tang. Guided image filtering. In *Proceedings of the 11th European conference on Computer vision: Part 1, ECCV'10*, pages 1–14, Berlin, Heidelberg, 2010. Springer-Verlag. 2
- [18] D. Krishnan, R. Fattal, and R. Szeliski. Efficient preconditioning of laplacian matrices for computer graphics. volume 28, pages 1–10, New York, NY, USA, 2013. ACM. 4
- [19] D. Krishnan and R. Szeliski. Multigrid and multilevel preconditioners for computational photography. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 177:1–177:10, New York, NY, USA, 2011. ACM. 4
- [20] M. Lang, O. Wang, T. Aydin, A. Smolic, and M. Gross. Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph.*, 31(4):34:1–34:8, July 2012. 2
- [21] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 689–694, New York, NY, USA, 2004. ACM. 1, 2, 5
- [22] X.-Y. Li, Y. Gu, S.-M. Hu, and R. R. Martin. Mixed-domain edge-aware image manipulation. *IEEE Transactions on Image Processing*, 22(5):1915–1925, 2013. 1
- [23] Y. Li, L. Sharan, and E. H. Adelson. Compressing and companding high dynamic range images with subband architectures. *ACM Trans. Graph.*, 24(3):836–844, July 2005. 1
- [24] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski. Interactive local adjustment of tonal values. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 646–653, New York, NY, USA, 2006. ACM. 1, 2, 3, 5, 6
- [25] F. Pellacini and J. Lawrence. Appwand: editing measured materials using appearance-driven optimization. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. 2, 3
- [26] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 313–318, New York, NY, USA, 2003. ACM. 1, 2, 6
- [27] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *ICCV'09*, pages 151–158, Kyoto, Sept 2009. 1
- [28] J. H. Reynolds and R. Desimone. Interacting roles of attention and visual salience in V4. *Neuron*, 37(5):853–63, Mar. 2003. 1
- [29] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Pittsburgh, PA, USA, 1994. 4
- [30] K. Subr, C. Soler, and F. Durand. Edge-preserving multiscale image decomposition based on local extrema. In *ACM SIGGRAPH Asia 2009 papers*, SIGGRAPH Asia '09, pages 147:1–147:9, New York, NY, USA, 2009. ACM. 2
- [31] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 315–321, New York, NY, USA, 2004. ACM. 1, 2
- [32] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision, ICCV '98*, pages 839–, Washington, DC, USA, 1998. IEEE Computer Society. 2
- [33] K. Xu, Y. Li, T. Ju, S.-M. Hu, and T.-Q. Liu. Efficient affinity-based edit propagation using k-d tree. In *ACM SIGGRAPH Asia 2009 papers*, SIGGRAPH Asia '09, pages 118:1–118:6, New York, NY, USA, 2009. ACM. 2
- [34] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l0 gradient minimization. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, SA '11, pages 174:1–174:12, New York, NY, USA, 2011. ACM. 1, 2, 4
- [35] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *ACM Trans. Graph.*, 31(6):139:1–139:10, Nov. 2012. 1, 2, 5, 7, 8
- [36] Q. Yang, K.-H. Tan, and N. Ahuja. Real-time o(1) bilateral filtering. In *CVPR*, pages 557–564, 2009. 2
- [37] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *Trans. Img. Proc.*, 15(5):1120–1129, May 2006. 2