Robust Guided Image Filtering Using Nonconvex Potentials

Bumsub Ham, Minsu Cho, Jean Ponce

Robust Guided Image Filtering Using Nonconvex Potentials

Bumsub Ham, *Member, IEEE,* Minsu Cho, and Jean Ponce, *Fellow, IEEE*

Abstract—Filtering images using a guidance signal, a process called guided or joint image filtering, has been used in various tasks in computer vision and computational photography, particularly for noise reduction and joint upsampling. This uses an additional guidance signal as a structure prior, and transfers the structure of the guidance signal to an input image, restoring noisy or altered image structure. The main drawbacks of such a data-dependent framework are that it does not consider structural differences between guidance and input images, and that it is not robust to outliers. We propose a novel SD (for static/dynamic) filter to address these problems in a unified framework, and jointly leverage structural information from guidance and input images. Guided image filtering is formulated as a nonconvex optimization problem, which is solved by the majorize-minimization algorithm. The proposed algorithm converges quickly while guaranteeing a local minimum. The SD filter effectively controls the underlying image structure at different scales, and can handle a variety of types of data from different sensors. It is robust to outliers and other artifacts such as gradient reversal and global intensity shift, and has good edge-preserving smoothing properties. We demonstrate the flexibility and effectiveness of the proposed SD filter in a variety of applications, including depth upsampling, scale-space filtering, texture removal, flash/non-flash denoising, and RGB/NIR denoising.

Index Terms—Guided image filtering, joint image filtering, nonconvex optimization, majorize-minimization algorithm.

1 Introduction

Many tasks in computer vision and computational photography can be formulated as ill-posed inverse problems, and thus theoretically and practically require filtering and regularization to obtain a smoothly varying solution and/or ensure stability [1]. Image filtering is used to suppress noise and/or extract structural information in many applications such as image restoration, boundary detection, and feature extraction. In this setting, an image is convolved with a spatially invariant or variant kernel. Linear translation invariant (LTI) filtering uses spatially invariant kernels such as Gaussian and Laplacian ones. Spatially invariant kernels enable an efficient filtering process, but do not consider the image content, smoothing or enhancing both noise and image structure evenly.

Recent work on joint image filtering (or guided image filtering) [2], [3], [4] uses an additional *guidance* image to construct a spatially variant kernel. The basic idea is that the structural information of the guidance image can be transferred to an input image, e.g., for preserving salient features such as corners and boundaries while suppressing noise. This provides a new perspective on the filtering process, with a great variety of applications including stereo correspondence [5], [6], optical flow [5], [7], semantic flow [8], [9], joint upsampling [3], [10], [11], [12], dehazing [2], noise reduction [4], [13], and texture removal [14], [15], [16].

Joint image filtering has been used with either static or dynamic guidance images. *Static guidance* filtering (e.g., [2]) modulates the input image with a weight function depending only on features of the guidance image, as in the blue box of Fig. 1.

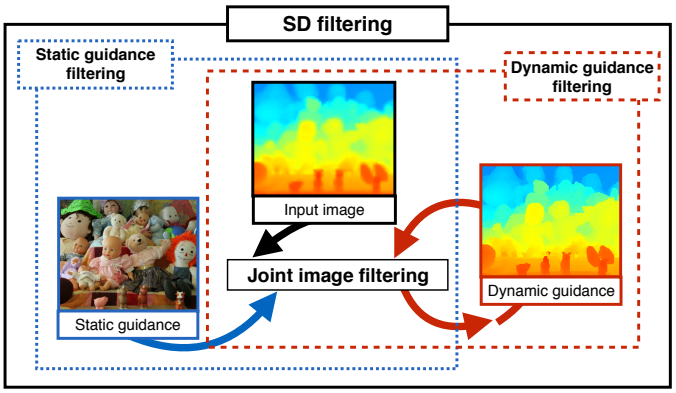


Fig. 1. SD filtering for depth upsampling: Static guidance filtering convolves an input image (a low-resolution depth image) with a weight function computed from a fixed guidance signal (a high-resolution color image), as in the blue box. Dynamic guidance filtering uses weight functions that are repeatedly obtained from filtered input images, as in the red box. We have observed that static and dynamic guidance complement each other, and exploiting only one of them is problematic, especially in the case of data from different sensors (e.g., depth and color images). The SD filter takes advantage of both, and addresses the problems of current joint image filtering. **(Best viewed in colour.)**

This guidance signal is fixed during the optimization. It can reflect internal properties of the input image itself, e.g., its gradients [3], [17], or be another signal aligned with the input image, e.g., a near infrared (NIR) image [4]. This approach assumes the structure of the input and guidance images to be consistent with each other, and does not consider structural (or statistical) dependencies and inconsistencies between them. This is problematic, especially in the case of data from different sensors (e.g., depth and color images). *Dynamic guidance* filtering (e.g., [16]) uses weight functions that are repeatedly obtained from filtered input images, as in the red box of Fig. 1. It is assumed that the weight between

- *Bumsub Ham is with the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea. E-mail: mimo@yonsei.ac.kr.*
- *Minsu Cho is with the Department of Computer Science and Engineering, POSTECH, Pohang, Korea. E-mail: mscho@postech.ac.kr.*
- *Jean Ponce is with École Normale Supérieure / PSL Research University and WILLOW project-team (CNRS/ENS/INRIA UMR 8548), Paris, France. E-mail: jean.ponce@ens.fr.*

neighboring pixels can be determined more accurately from the filtered input image than from the initial one itself [16], [18]. This method is inherently iterative, and the dynamic guidance signal (the filtered input image, i.e., a potential output image) is updated at every step until convergence. Dynamic guidance filtering takes into account the properties of the input image, but ignores the additional information available in the static guidance image, which can be used to impose image structures lost in the input data, e.g., in joint upsampling [3], [10], [11], [12].

We present in this paper a unified framework for joint image filtering taking advantage of both static and dynamic guidance, called the *SD filter* (Fig. 1). We address the aforementioned problems by adaptively fusing data from the static and dynamic guidance signals, rather than unilaterally transferring the structure of the guidance image to the input one. To this end, we combine static guidance image weighting with a nonconvex penalty on the gradients of dynamic guidance, which makes joint image filtering robust to outliers. The SD filter has several advantages over current joint image filters: First, it effectively controls image structures at different scales, and can handle a variety of types of data from different sensors. Second, it has good edge-preserving properties, and is robust to artifacts, such as gradient reversal and global intensity shift [2].

Contributions. The main contributions of this paper can be summarized as follows:

- We formulate joint image filtering as a nonconvex optimization problem, where Welsch's function [19] (or correntropy [20]) is used for a nonconvex regularizer. We solve this problem by the majorize-minimization algorithm, and propose to use an l_1 regularizer to compute an initial solution of our solver, which accelerates the convergence speed (Section 3).
- We analyze several properties of the SD filter including scale adjustment, runtime, filtering behavior, and its connection to other filters (Section 4).
- We demonstrate the flexibility and effectiveness of the SD filter in several computer vision and computational photography applications including depth upsampling, scale-space filtering, texture removal, flash/non-flash denoising, and RGB/NIR denoising (Section 5).

A preliminary version of this work appeared in [21]. This version adds (1) an in-depth presentation of the SD filter; (2) a discussion on its connection to other filtering methods; (3) an analysis of runtime; (4) an extensive experimental evaluation and comparison of the SD filter with several state-of-the-art methods; and (5) an evaluation on the localization accuracy of depth edges for depth upsampling. To encourage comparison and future work, the source code of the SD filter is available at our project webpage: http://www.di.ens.fr/willow/research/sdfilter.

2 Related work

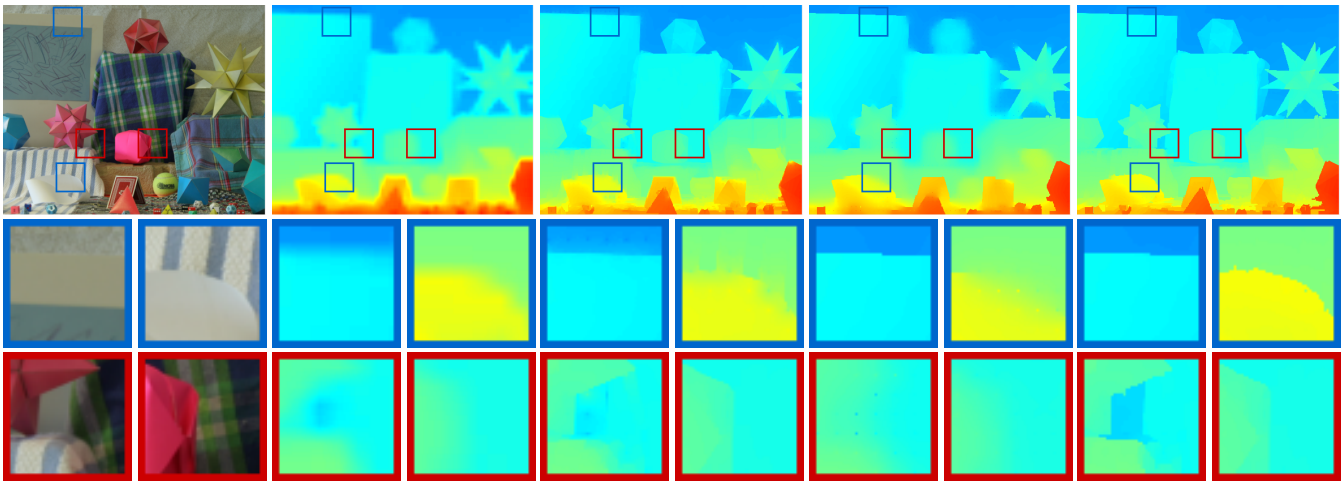
2.1 Image filtering and joint image filtering

Joint image filters can be derived from minimizing an objective function that usually involves a fidelity term (e.g., [2], [3], [22]), a prior smoothness (regularization) term (e.g., [17]), or both (e,g., [15], [23], [24]). Regularization is *implicitly* imposed on the objective function with the fidelity term only. The smoothness term encodes an *explicit* regularization process into the objective function. We further categorize joint image filtering (static or dynamic guidance filtering) into implicit and explicit regularization methods. The SD filter belongs to the explicit regularization that takes advantage of both static and dynamic guidance.

Implicit regularization stems from a local filtering framework. The input image is filtered using a weight function that depends on the similarity of features within a local window in the guidance image [3], allowing the structure of the guidance image to be transferred to the input image. The bilateral filter (BF) [22], guided filter (GF) [2], weighted median filter (WMF) [11], and weighted mode filter (WModF) [25] are well-known implicit regularization methods, and they have been successfully adapted to static guidance filtering. They regularize images through a weighted averaging process [2], [22], a weighted median process [11], or a weighted mode process [25]. Two representative filtering methods using dynamic guidance are iterative nonlocal means (INM) [18] and the rolling-guidance filter (RGF) [16]. These methods share the same filtering framework, but differ in that INM aims at preserving textures during noise reduction, while RGF aims at removing them through scale-space filtering. The implicit regularization involved is simple and easy to implement, but it has some drawbacks. For example, it has difficulties handling sparse input data (e.g., in image colorization [26]), and often introduces artifacts (e.g., halos and gradient reversal [2]) due to its local nature. Accordingly, implicit regularization has mainly been applied in highly controlled conditions, and is typically used as pre- and/or post-processing for further applications [11], [27].

An alternative approach is to explicitly encode the regularization process into the objective function, while taking advantage of the guidance image. The weighted least-squares (WLS) framework [23] is the most popular explicit regularization method in static guidance filtering [12]. The objective function typically consists of two parts: A fidelity term captures the consistency between input and output images, and a regularization term, modeled using a weighted l_2 norm [23], encourages the output image to have a similar structure to the guidance one. Many other regularization terms, e.g., l_1 norm in total generalized variation (TGV) [10], l_0 norm in l_0 norm minimization [24], or relative total variation (RTV) [15], have also been employed, so that the filtered image is forced to have statistical properties similar to those of the desired solution. Anisotropic diffusion (AD) [17] is an explicit regularization framework using dynamic guidance. In contrast to INM [18] and RGF [16], AD updates both input and guidance images at every step. Filtering is performed iteratively with the filtered input and updated guidance images. This explicit regularization enables formulating a task-specific model, with more flexibility than implicit regularization. Furthermore, it overcomes several limitations of implicit regularization, such as halos and gradient reversal for example, at the cost of global intensity shift [2], [23].

Existing image filtering methods typically apply to a limited range of applications and suffer from various artifacts: For example, RGF is applicable to scale-space filtering only, and suffers from poor edge localization [16]. In contrast, our approach provides a unified model for many applications, gracefully handles most artifacts, and outperforms the state of the art in all the applications considered in the paper. Although the proposed model is superficially similar to WLS [23] and RGF [16], our *nonconvex* objective function requires a different solver, which gives theoretical reasoning for RGF. Our model is also related to a diffusion-reaction equation [28], the steady-state solution of which has a similar form to the SD filter. However, the SD filter has an additional weight function using static guidance, as described



(a) Color image. (b) Depth image. (c) Static guidance [23]. (d) Dynamic guidance [16]. (e) Our model.

Fig. 2. Comparison of static and dynamic guidance filtering methods. Given (a) a high-resolution color image, (b) a low-resolution depth image is upsampled ($\times 8$) by our model in (1) using (c) static guidance only [23], (d) dynamic guidance only [16], and (e) joint static and dynamic guidance. Static guidance filtering restores smoothed depth edges, as in the red boxes of (c). However, this method transfers all the structural information of the color image to the depth image, such as the weak color edges between the board and background and the color stripes on the tablecloth in the blue boxes of (c). Dynamic guidance filtering avoids this problem, as in the blue boxes of (d), but this method does not use the structural information of the high-resolution color image, smoothing or even eliminating depth edges as in the red boxes of (d). The SD filter jointly uses the structure of color and depth images, and does not suffer from these problems as in the blue and red boxes of (e). See Section 5.1 for details. **(Best viewed in colour.)**

in Section 4.4 in detail. While Badri *et al*. [29] used a similar nonconvex objective function for image smoothing, our model provides a more generalized objective for joint image filtering. Shen *et al*. [30] used the common structure in input and guidance images, but the local filtering formulation of the filter introduces halo artifacts and limits the applicability. The closest work to ours is a recently introduced a learning-based joint filter [31] using convolutional neural networks (CNNs). This deep joint filter considers the structures of both input and guidance images, but requires a large number of annotated images for training the deep model for each task.

2.2 Joint image filtering for high-level vision

Joint image filtering have shown great success in low-level vision tasks, but little attention has been paid to applying it to high-level vision problems. Motivated by the fact that contrast-sensitive potentials [32] typically used in conditional random fields (CRFs) have a similar role to static guidance image weighting (e.g., see (5) in [32].), Krähenbühl and Koltun [33] used joint image filtering for an efficient message passing scheme, enabling an approximate inference algorithm for fully connected CRF models. In a similar manner that joint image filtering is used for many applications (e.g., joint upsampling), the CRF model has been widely used to refine low-resolution CNN outputs, e.g., in semantic segmentation [34], [35]. Recent work [36], [37] learns parameters of both a joint image filter and a CNN model in an end-to-end manner to upsample low-resolution network outputs. As in existing joint image filters, however, the refinement methods use the structure of high-resolution color images (static guidance) only, while ignoring additional information available in the CNN outputs. We believe that using the SD filter will improve the refinement by considering the structures of both inputs and outputs of the CNN in the end-to-end learning.

3 Proposed Approach

3.1 Motivation and problem statement

There are pros and cons in filtering images under static or dynamic guidance. Let us take the example of depth upsampling in Fig. 2, where a high-resolution color image (the guidance image) in (a) is used to upsample ($\times 8$) an input low-resolution depth image in (b). Static guidance filtering reconstructs destroyed depth edges using the color image with high signal-to-noise ratio (SNR) [3], [10], as in the red boxes of (c). However, this approach does not handle differences in structure between depth and color images, transferring all the structural information of the color image to the depth image, as in the blue boxes of (c). For regions of high-contrast in the color image, the depth image is altered according to the texture pattern, resulting in texture-copying artifacts [38]. Similarly, depth edges are smoothed in low-contrast regions on the color image (e.g., weak color edges), causing jagged artifacts in scene reconstruction [39]. Dynamic guidance filtering utilizes the content of the depth image[1], avoiding the drawbacks of static guidance filtering, as in the blue boxes of (d). The depth edges are preserved, and unwanted structures are not transferred, but dynamic guidance does not take full advantage of the abundant structural information in the color image. Thus, depth edges are smoothed, and even eliminated for regions of low-contrast in the depth image, as in the red boxes of (d).

This example illustrates the fact that static and dynamic guidances complement each other, and exploiting only one of them is not sufficient to infer high-quality structural information from the input image. This problem becomes even worse when input and guidance images come from different data with different statistical characteristics. The SD filter proposed in this paper jointly leverages the structure of static (color) and dynamic (depth) guidance, taking advantage of both, as shown in (e).

1. Dynamic guidance is initialized with the interpolated depth image using static guidance as shown in Fig. 2(c), not with the low-resolution depth image itself.

3.2 Model

Given the input image f, the static guidance g, and the output image u (dynamic guidance), we denote by f_i, g_i, and u_i the corresponding image values at pixel i, with i ranging over the image domain $\mathcal{I} \subset \mathbb{N}^2$. Our objective is to infer the structure of the input image by jointly using static and dynamic guidance, and to make joint image filtering robust to outliers in a unified framework. The influence of the guidance images on the input image varies spatially, and is controlled by weight functions that measure the similarity between adjacent pixels in the guidance images. Various features (e.g., spatial location, intensity, and texture [12], [40]) and metrics (e.g., Euclidian and geodesic distances [7], [23]) can be used to represent distinctive characteristics of pixels on images and measure their similarities.

We minimize an objective function of the form:

$$\mathcal{E}(u) = \sum_i c_i (u_i - f_i)^2 + \lambda \Omega(u, g), \tag{1}$$

which consists of fidelity and regularization terms, balanced by the regularization parameter λ. The first (fidelity) term helps the solution u to harmonize well with the input image f with confidence $c_i \geq 0$. The regularization term makes the solution u smooth while preserving salient features (e.g., boundaries). We propose the following regularizer:

$$\Omega(u, g) = \sum_{i,j \in \mathcal{N}} \phi_\mu(g_i - g_j) \psi_\nu(u_i - u_j), \tag{2}$$

where

$$\psi_\nu(x) = (1 - \phi_\nu(x))/\nu, \tag{3}$$

and

$$\phi_\lambda(x) = \exp(-\lambda x^2). \tag{4}$$

Here, μ and ν control the smoothness bandwidth, and $\mathcal{N}$ is the set of image adjacencies, defined in our implementation on a local 8-neighborhood system. The first term, ϕ_μ in the regularizer is a weight function using intensity difference between adjacent pixels in the guidance g. This function approaches to zero as the gradient of the guidance g becomes larger (e.g., at the discontinuities of g), preventing smoothing the output image u in those regions. That is, the weight function ϕ_μ transfers image structures from the static guidance g to the output image u. The second one ψ_ν, called Welsch's function [19], regularizes the output image u, and makes joint filtering *robust* to outliers, due to its nonconvex shape and the fact that it saturates at 1 (Fig. 4). Note that convex potentials in existing joint image filters regularize the structure of the output image u and bring the structure through static guidance image weighting ϕ_μ. In contrast, the nonconvex potential ψ_ν penalizes large gradients of the output image u (dynamic guidance) less than a convex one during filtering [17], [41], [42], and preserves features having high-frequency structures (e.g., edges and corners) better. This means the use of nonconvex potentials relax the strict dependence of the existing joint image filters on the static guidance g, and allows to leverage the structure of the dynamic guidance u as well. Accordingly, by combining static guidance image weighting ϕ_μ with the nonconvex penalty ψ_ν on the gradients of the dynamic guidance u, the SD filter adaptively fuses data from the static and dynamic guidances. This effectively reflects structural differences between guidance and input images, and makes joint image filtering robust to outliers. Although nonconvex potentials preserve some noise, the static guidance g with high SNR in our model alleviates this problem. Note that in order to adaptively fuse data from the static and dynamic guidances, one might attempt to add a weight function ϕ_ν using the dynamic guidance u in the regularizer as follows:

$$\Omega(u, g) = \sum_{i,j} \phi_\mu(g_i - g_j) \phi_\nu(u_i - u_j) \psi_\nu(u_i - u_j). \tag{5}$$

This regularizer, however, is hard to optimize and may be unstable.

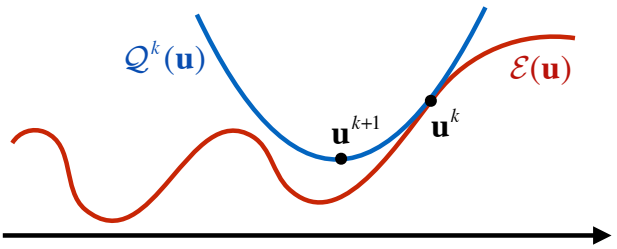


Fig. 3. Sketch of the majorize-minimization algorithm. At step k, a surrogate function $\mathcal{Q}^k$ is constructed given some estimate $\mathbf{u}^k$ of the minimum of $\mathcal{E}$, such that $\mathcal{Q}^k(\mathbf{u}^k) = \mathcal{E}(\mathbf{u}^k)$ and $\mathcal{Q}^k(\mathbf{u}) \geq \mathcal{E}(\mathbf{u})$. At step $k+1$, the next estimate $\mathbf{u}^{k+1}$ is obtained by minimizing $\mathcal{Q}^k$. These two steps are repeated until convergence.

3.3 Optimization algorithm

Let $\mathbf{f} = [f_i]_{N\times 1}$, $\mathbf{g} = [g_i]_{N\times 1}$, and $\mathbf{u} = [u_i]_{N\times 1}$ denote vectors representing the input image, static guidance and the output image (or dynamic guidance), respectively, where $N = |\mathcal{I}|$ is the size of the images. Let $\mathcal{W}_g = [\phi_\mu(g_i - g_j)]_{N\times N}$, $\mathcal{W}_u = [\phi_\nu(u_i - u_j)]_{N\times N}$, and $\mathcal{C} = diag([c_1, \ldots, c_N])$. $\mathcal{W}_g$ and $\mathcal{W}_u$ denote weight matrices of the 8-neighborhood system for static and dynamic guidances, respectively. We can rewrite our objective function in matrix/vector form as:

$$\mathcal{E}(\mathbf{u}) = (\mathbf{u} - \mathbf{f})^T \mathcal{C} (\mathbf{u} - \mathbf{f}) + \frac{\lambda}{\nu} \mathbb{1}^T (\mathcal{W}_g - \mathcal{W}) \mathbb{1}, \tag{6}$$

where $\mathcal{W} = \mathcal{W}_g \circ \mathcal{W}_u$, and $\circ$ denotes the Hadamard product of the matrices defined as the element-wise multiplication of their elements. $\mathbb{1}$ is a $N \times 1$ vector, where all the entries are 1. The diagonal entries c_i of $\mathcal{C}$ are confidence values for the pixels i in the input image.

3.3.1 The majorize-minimization algorithm

Our objective function is not convex, and thus it is non-trivial to minimize the objective function of (6). In this work, we compute a local minimum of (6) by iteratively computing a convex upper bound (surrogate function) and solving the corresponding convex optimization problems. Concretely, we use the majorize-minimization algorithm (Fig. 3) [43], [44], [45], which guarantees convergence to a local minimum of the nonconvex objective function [46]. This algorithm consists of two repeated steps: In the majorization step, given some estimate $\mathbf{u}^k$ of the minimum of $\mathcal{E}$ at step k, a convex surrogate function $\mathcal{Q}^k$ is constructed, such that

$$\begin{cases} \mathcal{Q}^k(\mathbf{u}^k) = \mathcal{E}(\mathbf{u}^k) \\ \mathcal{Q}^k(\mathbf{u}) \geq \mathcal{E}(\mathbf{u}) \end{cases}. \tag{7}$$

In the minimization step, the next estimate $\mathbf{u}^{k+1}$ is then computed by minimizing $\mathcal{Q}^k$. These steps are repeated until convergence.

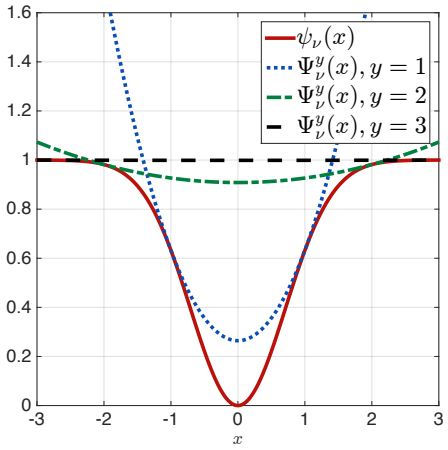


Fig. 4. Welsch's function ψ_ν and its surrogate functions Ψ_ν^y, when ν is set to 1. The convex surrogate function $\Psi_\nu^y(x)$ is an upper bound on $\psi_\nu(x)$ and coincides with $\psi_\nu(x)$ only when x is equal to y. **(Best viewed in colour.)**

3.3.2 *Implementation details*

The nonconvexity of our objective function comes from Welsch's function ψ_ν. As shown in **Appendix A**, a convex surrogate function Ψ_ν^y for ψ_ν, such that $\forall x, y$,

$$\begin{cases} \Psi_\nu^y(y) = \psi_\nu(y) \\ \Psi_\nu^y(x) \geq \psi_\nu(x) \end{cases}, \tag{8}$$

is given by

$$\Psi_\nu^y(x) = \psi_\nu(y) + (x^2 - y^2)(1 - \nu\psi_\nu(y)). \tag{9}$$

The surrogate function $\Psi_\nu^y(x)$ is an upper bound on $\psi_\nu(x)$ and coincides with $\psi_\nu(x)$ only when x is equal to y (Fig. 4).

Majorization step: Let us now use this result to compute the surrogate function $\mathcal{Q}^k$ for $\mathcal{E}$ by substituting ψ_ν with Ψ_ν^y in (2) as follows:

$$\begin{aligned} \mathcal{Q}^k(\mathbf{u}) = \mathbf{u}^T \left[\mathcal{C} + \lambda\mathcal{L}^k\right] \mathbf{u} - 2\mathbf{f}^T\mathcal{C}\mathbf{u} + \mathbf{f}^T\mathcal{C}\mathbf{f} \\ - \lambda\mathbf{u}^{kT}\mathcal{L}^k\mathbf{u}^k + \frac{\lambda}{\nu}\mathbb{1}^T\left(\mathcal{W}_g - \mathcal{W}^k\right)\mathbb{1}. \end{aligned} \tag{10}$$

$\mathcal{L}^k = \mathcal{D}^k - \mathcal{W}^k$ is a *dynamic* Laplacian matrix at step k, where $\mathcal{W}^k = \mathcal{W}_g \circ \mathcal{W}_{u^k}$ and $\mathcal{D}^k = diag\left(\left[d_1^k, \ldots, d_N^k\right]\right)$ where $d_i^k = \sum_{j=1}^N \phi_\mu(g_i - g_j)\phi_\nu(u_i^k - u_j^k)$.

Minimization step: We then obtain the next estimate $\mathbf{u}^{k+1}$ by minimizing the surrogate function $\mathcal{Q}^k$ w.r.t. $\mathbf{u}$ as follows[2]:

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}} \mathcal{Q}^k(\mathbf{u}) = (\mathcal{C} + \lambda\mathcal{L}^k)^{-1}\mathcal{C}\mathbf{f}. \tag{11}$$

The above iterative scheme decreases the value of $\mathcal{E}$ monotonically in each step, i.e.,

$$\mathcal{E}(\mathbf{u}^{k+1}) \leq \mathcal{Q}^k(\mathbf{u}^{k+1}) \leq \mathcal{Q}^k(\mathbf{u}^k) = \mathcal{E}(\mathbf{u}^k), \tag{12}$$

where the first and the second inequalities follow from (7) and (11), respectively, and converges to a local minimum of $\mathcal{E}$ [46]. The static guidance affinity matrix $\mathcal{W}_g$ is fixed regardless of steps, while the dynamic guidance matrix $\mathcal{W}_{u^k}$ is iteratively updated. Thus, we jointly use the structure of static and dynamic guidance to compute the solution $\mathbf{u}$. Note that all previous joint image filters (e.g., [2], [3], [11], [25]) except recently proposed ones [31] determine the structure of the output image by the weight function of the static guidance image ($\mathcal{W}_g$) only. We overcome this limitation by using an additional weight function of the dynamic guidance image ($\mathcal{W}_{u^k}$) in each round of an intermediate output $\mathbf{u}^k$, thus reflecting the structures of both static and dynamic guidance images simultaneously during the optimization process.

The majorize-minimization algorithm generalizes other well-known optimization methods [47], e.g., the expectation-maximization (EM) algorithm [48], iteratively reweighted least-squares (IRLS) [49], the Weiszfeld's method [50], and the half-quadratic (HQ) minimization [51], [52]. Nikolova and Chan [53] have shown that the method where the solution of the objective function is computed by iteratively solving linear systems at each step as in our solver is equivalent to the HQ minimization (of multiplicative form). That is, both methods give exactly the same iterations. Thus, our solver gives an identical solution to the HQ minimization (of multiplicative form) for (6), if initializations are the same.

Algorithm 1 summarizes the optimization procedure.

Algorithm 1 The SD filter.

1: **Input:**
   $\mathbf{f}$ (an input image); $\mathbf{g}$ (a static guidance image);
   $\mathbf{u}^0$ (an initial estimate for a dynamic guidance image).
2: **Parameters:**
   μ (a bandwidth parameter for $\mathbf{g}$);
   ν (a bandwidth parameter for $\mathbf{u}$);
   λ (a regularization parameter);
   K (the maximum number of steps).
3: Compute a confidence matrix $\mathcal{C} = diag\left([c_1, \ldots, c_N]\right)$ where $c_i \geq 0$.
4: Compute an affinity matrix for the static guidance image $\mathbf{g}$, $\mathcal{W}_g = [\phi_\mu(g_i - g_j)]_{N\times N}$, where $\phi_\mu(x) = \exp(-\mu x^2)$.
5: **for** $k = 0, \ldots, K$ **do**
6: Compute an affinity matrix for the dynamic guidance image $\mathbf{u}^k$, $\mathcal{W}_{u^k} = \left[\phi_\nu(u_i^k - u_j^k)\right]_{N\times N}$.
7: Compute a dynamic Laplacian matrix $\mathcal{L}^k = \mathcal{D}^k - \mathcal{W}^k$ where $\mathcal{W}^k = \mathcal{W}_g \circ \mathcal{W}_{u^k}$ and $\mathcal{D}^k = diag\left(\left[d_1^k, \ldots, d_N^k\right]\right)$ where $d_i^k = \sum_{j=1}^N \phi_\mu(g_i - g_j)\phi_\nu(u_i^k - u_j^k)$.
8: Update $\mathbf{u}^{k+1}$ by solving $(\mathcal{C} + \lambda\mathcal{L}^k)\mathbf{u}^{k+1} = \mathcal{C}\mathbf{f}$.
9: **end for**
10: **Output:** $\mathbf{u}^K$ (a final estimate).

3.3.3 *Initialization*

Our solver finds a local minimum, and thus different initializations for $\mathbf{u}^0$ (dynamic guidance at $k = 0$) may give different solutions. In this work, we compute the initial estimate $\mathbf{u}^0$ by minimizing the objective function of (1) and using an upper bound on Welsch's function in the regularizer (Fig. 5). Two functions are used for initialization.

l_2 **initialization.** Welsch's function is upper bounded by an l_2 regularizer, i.e., $\forall x$, $x^2 \geq \psi_\nu(x)$. This can be easily shown by using the inequality, $\exp(x) \geq 1 + x$. The initial estimate $\mathbf{u}^0$ is computed by minimizing the objective function in (1) with a weighted l_2 norm as a regularizer, i.e., using the WLS framework [23]:

$$\Omega_{l_2}(u, g) = \sum_{i,j\in\mathcal{N}} \phi_\mu(g_i - g_j)(u_i - u_j)^2. \tag{13}$$

2. In case of a color image, the linear system is solved in each channel.

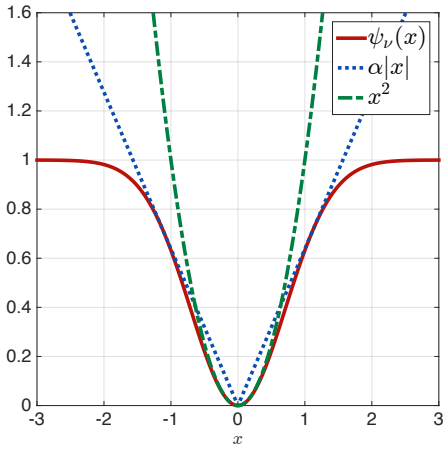


Fig. 5. Upper bounds on Welsch's function ψ_ν: an l_2 regularizer x^2 and an l_1 regularizer $\alpha|x|$ (a tight upper bound on ψ_ν), when ν is set to 1. **(Best viewed in colour.)**

The l_2 initialization is simple, but may yield a slow convergence rate as illustrated by Fig. 6(a).

l_1 **initialization.** A good initial solution accelerates the convergence of our solver. We propose to use the following regularizer to compute the initial estimate $\mathbf{u}^0$:

$$\Omega_{l_1}(u,g) = \sum_{i,j\in\mathcal{N}} \phi_\mu(g_i - g_j)\alpha|u_i - u_j|, \qquad (14)$$

where α is set to a positive constant, chosen so $\alpha|x|$ is a much tighter upper bound on Welsch's function than the l_2 regularizer, as shown in Fig. 5. This regularizer is convex, and thus the global minimum[3] is guaranteed [49].

4 Analysis

In this section, we analyze the properties of the SD filter, including runtime and scale adjustment, and illustrate its convergence rate and filtering behavior with different initializations. A connection to other filtering methods is also presented.

4.1 Runtime

The major computational cost of the SD filter comes from solving the linear system of (11) repeatedly. We use a local 8-neighborhood system, resulting in the sparse matrix $(\mathcal{C} + \lambda\mathcal{L}^k)$. This matrix has a positive diagonal and four sub-diagonals on each side. These sub-diagonals are not adjacent, and their separation is proportional to the image width. It is therefore not banded with a fixed bandwidth. Yet, it is sparse and positive semidefinite, thus (11) can be solved efficiently using sparse Cholesky factorization. We use the `CHOLMOD` solver [54] of `MATLAB` in our implementation. Running $k = 5$ steps of our algorithm from an l_1 or l_2 initialization takes about 2.5 seconds for an image of size 500×300 on a 4.0GHz CPU[4]. The l_1 and l_2 initializations take about 1.2 and 0.4 seconds, respectively, on average for the same image size.

3. We use IRLS [49] to find the solution.

4. The SD filter of (11) applies the very popular WLS filter [23] iteratively with a fixed input image, allowing us to use many acceleration techniques for WLS filtering [29], [55]. When using `MEX` implementation of the fast WLS algorithm in [55], we obtain the filtering result with 0.1 seconds for the same image size.

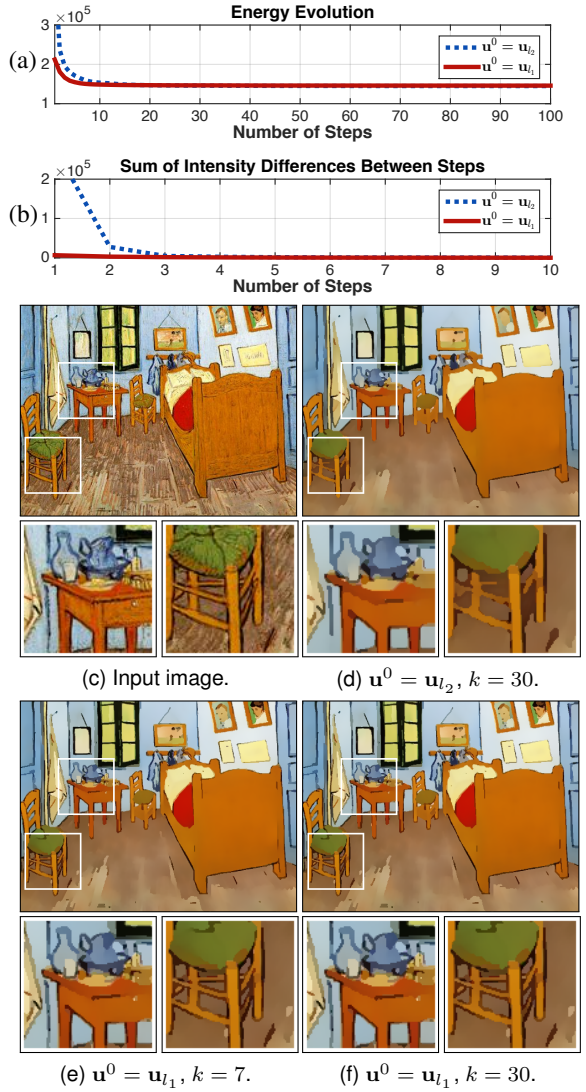


Fig. 6. Examples of (a) energy evolution of (6) and (b) a sum of intensity difference between successive steps (i.e., $\|\mathbf{u}^k - \mathbf{u}^{k+1}\|_1$), given (c) the input image. Our solver converges in fewer steps with the l_1 initialization ($\mathbf{u}^0 = \mathbf{u}_{l_1}$) than with the l_2 one ($\mathbf{u}^0 = \mathbf{u}_{l_2}$), with faster overall speed. It also guarantees a meaningful solution in the steady-state: (d) $\mathbf{u}^0 = \mathbf{u}_{l_2}$, $k = 30$, (e) $\mathbf{u}^0 = \mathbf{u}_{l_1}$, $k = 7$, and (f) $\mathbf{u}^0 = \mathbf{u}_{l_1}$, $k = 30$. In this example, for removing textures, static guidnce is set to the Gaussian filtered version (standard deviation, 1) of the input image ($\lambda = 50$, $\mu = 5$, $\nu = 40$). See Section 5.2 for details.

4.2 Influence of initialization

As explained earlier, the majorize-minimization algorithm is guaranteed to converge to a local minimum of $\mathcal{E}$ in (6) [46]. In this section, we show the convergence rate of (11) as the step index k increases, and observe its behavior with different initializations, using $\mathbf{u}_{l_2}$ and $\mathbf{u}_{l_1}$, the global minima of (1) using Ω_{l_2} and Ω_{l_1} as regularizers, respectively. Figure 6 shows how (a) the energy $\mathcal{E}$ of (6) and (b) the intensity differences (i.e., $\|\mathbf{u}^k - \mathbf{u}^{k+1}\|_1$) evolve over steps for a sample input image (Fig. 6(c)). In this example, our solver converges in fewer steps with the l_1 initialization ($\mathbf{u}^0 = \mathbf{u}_{l_1}$) than with the l_2 one ($\mathbf{u}^0 = \mathbf{u}_{l_2}$), with faster overall speed, despite the overhead of the l_1 minimization: Our solver with the l_2 and l_1 initializations converges in 30 and 7 steps (Fig. 6(d) and (e)), taking 45 and 20 seconds (including initialization), respectively. Although our solver with the l_2 initialization converges more slowly, the per-

pixel intensity difference still decreases monotonically in (b). Note that after few steps (typically from 5), the solver gives almost the same filtering results. For example, after 5 steps, the average and maximum values of the per-pixel intensity difference are 9.4×10^{-5} and 1.7×10^{-3}, respectively, with the l_2 initialization, and 4.3×10^{-5} and 8.7×10^{-4} with the l_1 initialization[5]. It should also be noted that most filtering methods, except the recently proposed RGF [16], eventually give a constant-value image, if they are performed repeatedly, regardless of whether the filters have implicit or explicit forms (e.g., BF [22] and AD [17]). In contrast, the SD filter converges to a meaningful image no matter how many steps are performed (Fig. 6(e) and (f)).

4.3 Scale adjustment

There are two approaches to incorporating scale information in image filtering [16], [17], [23]. In implicit regularization methods, an intuitive way to adjust the degree of smoothing is to use an isotropic Gaussian kernel. Due to the spatially invariant properties of the kernel, this approach regularizes both noise and features evenly without considering image structure [17]. RGF addresses this problem in two phases: Small-scale structures are removed by the Gaussian kernel, and large-scale ones are then recovered [16]. Since RGF [16] is based on the Gaussian kernel, it inherits its limitations. This leads to poor edge localization at coarse scales, with rounded corners and shifted boundaries. The regularization parameter is empirically used to adjust scales in explicit regularization methods [23]. It balances the degree of influence between fidelity and regularization terms in such a way that a large value leads to more regularized results than a small one.

Now, we will show how the regularization parameter λ controls scales in the SD filter. It follows from (11) that

$$(\mathcal{C}+\lambda\mathcal{D}^k)\mathbf{u}^{k+1}-\lambda\mathcal{W}^k\mathbf{u}^{k+1}=\mathcal{C}\mathbf{f}. \tag{15}$$

Let us define diagonal matrices $\mathcal{A}$ and $\mathcal{A}'$ as

$$\mathcal{A}=(\mathcal{C}+\lambda\mathcal{D}^k)^{-1}\mathcal{C}, \tag{16}$$

and

$$\mathcal{A}'=\lambda(\mathcal{C}+\lambda\mathcal{D}^k)^{-1}\mathcal{D}^k, \tag{17}$$

such that $\mathcal{A}+\mathcal{A}'=\mathbf{I}$. By multiplying the left- and right-hand sides of (15) by $(\mathcal{C}+\lambda\mathcal{D}^k)^{-1}$, we obtain

$$\begin{aligned}\mathbf{u}^{k+1}&=(\mathbf{I}-\underbrace{\lambda(\mathcal{C}+\lambda\mathcal{D}^k)^{-1}\mathcal{D}^k}_{\mathcal{A}'}\mathcal{P}^k)^{-1}\underbrace{(\mathcal{C}+\lambda\mathcal{D}^k)^{-1}\mathcal{C}}_{\mathcal{A}=\mathbf{I}-\mathcal{A}'}\mathbf{f}\\&=\mathcal{A}(\mathbf{I}-\mathcal{A}'\mathcal{P}^k)^{-1}\mathbf{f},\end{aligned} \tag{18}$$

where $\mathcal{P}^k=(\mathcal{D}^k)^{-1}\mathcal{W}^k$. Note that this gives the exact same form as random walk with restart [56], [57], [58], except that $\mathcal{P}^k$ is updated at every step in our case. It has been shown that the restarting probability $\mathcal{A}$ controls the extent of smoothing in different scales in the image [56]. In our case, the regularization parameter λ determines the restarting probability, which shows that by varying this parameter, we can adjust the degree of smoothing.

5. We normalize the intensity range to [0, 1].

4.4 Connections with other filters

Let us consider the objective function of (1), but without the regularization term. We can implicitly impose joint regularization on the objective function by introducing a spatial weight function ϕ_s and static guidance weighting ϕ_μ to the fidelity term. The objective function is then

$$\mathcal{E}(u)=\sum_i\sum_{j\in\eta}\phi_s(i-j)\phi_\mu(g_i-g_j)(u_i-f_j)^2, \tag{19}$$

where s is a spatial bandwidth parameter and η is the local neighborhood of the pixel i. The solution of this objective function can be computed as follows:

$$u_i^{k+1}=\frac{\sum\limits_{j\in\eta}\phi_s(i-j)\phi_\mu(g_i-g_j)f_j}{\sum\limits_{j\in\eta}\phi_s(i-j)\phi_\mu(g_i-g_j)}. \tag{20}$$

This becomes equivalent to the joint BF [3] (or BF [22] by setting $g=f$). By further introducing Welsch's function ψ_ν to (19), we obtain the following objective function:

$$\mathcal{E}(u)=\sum_i\sum_{j\in\eta}\phi_s(i-j)\phi_\mu(g_i-g_j)\psi_\nu(u_i-f_j), \tag{21}$$

We can compute a local minimum of this objective function by the majorize-minimization algorithm as follows (see **Appendix B**):

$$u_i^{k+1}=\frac{\sum\limits_{j\in\eta}\phi_s(i-j)\phi_\mu(g_i-g_j)\phi_\nu(u_i^k-f_j)f_j}{\sum\limits_{j\in\eta}\phi_s(i-j)\phi_\mu(g_i-g_j)\phi_\nu(u_i^k-f_j)}. \tag{22}$$

This again becomes the joint BF [3] with $\phi_\nu(x)=1$. A variant of RGF [16] can be derived by setting $\phi_\mu(x)=1$.

Let us now consider the case when f, g, and u are continuous functions over a continuous domain. A continuous form of the objective function of (1) can then be written as [59]:

$$\int_{\mathcal{J}}\left\{c_i(u_i-f_i)^2+\lambda\phi_\mu\left(\|\nabla g\|\right)\psi_\nu\left(\|\nabla u\|\right)\right\}d\mathcal{J}, \tag{23}$$

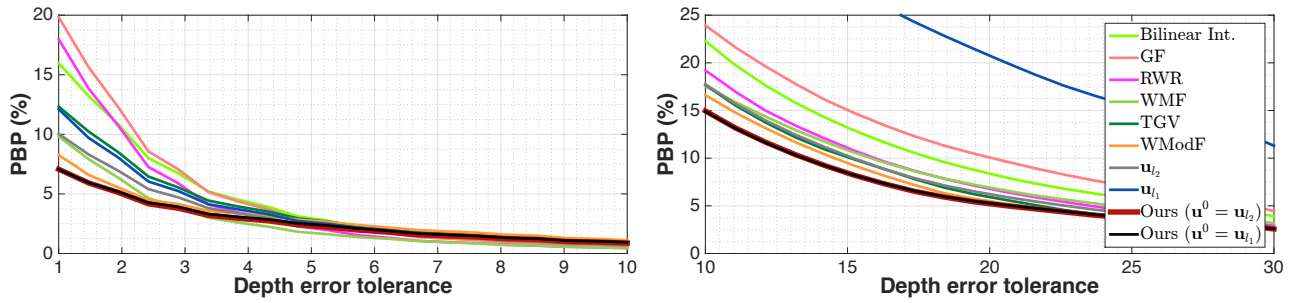
where $\mathcal{J}\subset\mathbb{R}_+^2$ and ∇ denotes the gradient operator. This function can be minimized via gradient descent using the calculus of variations [59] as follows:

$$\frac{\partial u}{\partial k}=c_i(u_i-f_i)+\lambda\text{div}\left[\phi_\mu\left(\|\nabla g\|\right)\phi_\nu\left(\|\nabla u\|\right)\nabla u\right], \tag{24}$$

where ∂k and div represent a partial derivative w.r.t. k and the divergence operator, respectively. This is a variant of the diffusion-reaction equation [17], [28] with an *additional term* $\phi_\mu\left(\|\nabla g\|\right)$ involving the static guidance g. The result of the SD filter is a steady-state solution of this scheme. Note that while being widely used in image filtering, nonconvex potentials have never been implemented before in the context of *joint* image filtering to the best of our knowledge. In our approach, nonconvex potentials make joint image filtering robust to outliers by effectively exploiting structural information from both guidance and input images. If one uses (24) with $\phi_\nu(x)=1$ (i.e., without dynamic guidance), the steady-state solution of (24) can be found in closed form, and is exactly same as the result obtained by the WLS framework [23]. With $\phi_\mu(x)=1$ (i.e., without static guidance), the steady-state solution of (24) is found by solving the following equation:

$$0=c_i(u_i-f_i)+\lambda\nabla\cdot\left[\phi_\nu\left(\|\nabla u\|\right)\nabla u\right]. \tag{25}$$

In this case, the solution cannot be computed in closed form, but it can be obtained by repeatedly solving (11) with $\mathcal{W}^k=\mathcal{W}_{u^k}$. This corresponds to a variant of RGF [16] with the fidelity term.



(a) Middlebury dataset [60]. (b) Graz dataset [10].

Fig. 7. Average PBP on (a) the Middlebury dataset [60] and (b) the Graz dataset [10] as a function of the depth error tolerance δ. **(Best viewed in colour.)**

TABLE 1
Average PBP ($\delta = 1$) on the Middlebury dataset [60] with varying the regularization parameter λ. PBP is defined as the percentage of bad pixels for all regions as in (26).

	$\mathbf{u}_{l_2}$	$\mathbf{u}_{l_1}$	Ours ($\mathbf{u}^0 = \mathbf{u}_{l_2}$)	Ours ($\mathbf{u}^0 = \mathbf{u}_{l_1}$)
λ	avg. $\pm$ std.	avg. $\pm$ std.	avg. $\pm$ std.	avg. $\pm$ std.
0.001	10.0±4.7	10.9±6.1	7.0±3.4	7.1±3.4
0.005	10.0±4.7	10.9±6.1	7.0±3.4	7.1±3.4
0.010	10.0±4.7	10.9±6.1	7.0±3.4	7.1±3.4
0.050	10.0±4.7	11.4±6.5	7.1±3.4	7.1±3.4
0.100	10.0±4.8	12.1±7.0	7.1±3.4	7.1±3.4
0.200	10.1±4.8	13.4±7.9	7.1±3.5	7.1±3.4
0.500	10.4±5.0	16.2±9.5	7.2±3.5	7.2±3.5
1.000	10.9±5.3	19.5±11.2	7.4±3.7	7.4±3.6
10.00	16.8±9.8	35.0±19.4	10.3±6.0	10.3±6.0

TABLE 2
Average PBP ($\delta = 1$) and processing time on the Middlebury dataset [60] with varying the number of steps k.

	Ours ($\mathbf{u}^0 = \mathbf{u}_{l_2}$)		Ours ($\mathbf{u}^0 = \mathbf{u}_{l_1}$)	
k	avg. $\pm$ std.	*time (s)*	avg. $\pm$ std.	*time (s)*
1	10.0±4.8	0.7	7.6±3.5	4.9
5	7.2±3.5	2.7	7.1±3.4	7.0
10	7.1±3.4	5.4	7.1±3.4	9.6
30	7.1±3.4	15.5	7.1±3.4	19.6
35	7.0±3.4	18.1	7.1±3.4	22.3
100	7.0±3.4	51.9	7.1±3.4	55.7

5 Applications

We apply the SD filter to depth upsampling, scale-space filtering, texture removal, flash/non-flash denoising, and RGB/NIR denoising. In each application, we compare the SD filter to the state of the art. The results for the comparison have been obtained from source codes provided by the authors, and all the parameters for all compared methods have been empirically set to yield the best average performance through extensive experiments.

5.1 Depth upsampling

Acquiring range data through an active sensor has been a popular alternative to passive stereo vision. Active sensors capture dense range data in dynamic conditions at video rate, but the acquired depth image may have low resolution, and be degraded by noise. These problems can be addressed by exploiting a registered high-resolution RGB image as a depth cue [10], [11], [12], [25], [61], [62]. We apply the SD filter to this task. We will show that, contrary to existing methods, the robust and nonconvex regularizer in the SD filter avoids smoothing depth edges, and gives a sharp depth transition at depth discontinuities.

Parameter settings. In our model, input and guidance images ($\mathbf{f}$ and $\mathbf{g}$) are set to sparse depth and dense color (or intensity) high-resolution images, respectively, and c_i is set to 1 if the pixel i of the sparse depth image $\mathbf{f}$ has valid data, and 0 otherwise. The bandwidths, μ and ν, and the step index k are fixed to 60, 30, and 10, respectively, in all experiments, unless otherwise specified. The regularization parameter λ is set to 0.1 for synthetic examples, and is set to 5 for real-world examples due to the huge amount of noise. For the quantitative comparison, we measure the percentage of bad matching pixels for all regions (PBP) defined as follows:

$$\text{PBP} = \frac{1}{N}\sum_i \left(\left|u_i - u_i^{gt}\right| > \delta\right), \tag{26}$$

where δ is a depth error tolerance [60]. u_i and u_i^{gt} represent upsampled and ground-truth depth images at the pixel i, respectively. We also measure the percentage of bad matching pixels near depth discontinuities (PBP*) using a ground-truth index map for discontinuous regions, provided by [60]. PBP* is measured only when the index map is available.

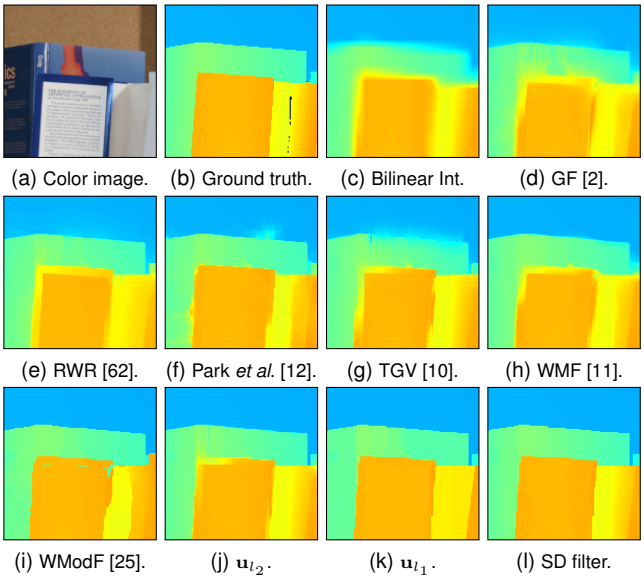
5.1.1 Synthetic data

We synthesize a low-resolution depth image by downsampling ($\times 8$) ground-truth data from the Middlebury benchmark dataset [60]: *Tsukuba*, *venus*, *teddy*, *cones*, *art*, *books*, *dolls*, *laundry*, *moebius*, and *reindeer*, and use the corresponding color image as static guidance.

Performance evaluation. We compare the average PBP ($\delta = 1$) of the results obtained by the SD filter with the l_2 and l_1 initializations and the corresponding initial estimates ($\mathbf{u}_{l_2}$ and $\mathbf{u}_{l_1}$) while varying the regularization parameter λ in Table 1. A first important conclusion is that the nonconvex regularizer in the SD filter shows a better performance than the convex ones. The table also shows that the results of the l_1 regularizer are strongly influenced by the regularization parameter, whereas the SD filter gives almost the same PBP regardless of the initialization. For example, when λ is set to 10, the SD filter reduces the average PBP from 16.8 and 35.0 for $\mathbf{u}_{l_2}$ and $\mathbf{u}_{l_1}$, respectively, to 10.3. Table 2 compares the average PBP ($\delta = 1$) and processing time of the SD filter with the l_2 and l_1 initializations while varying

TABLE 3
PBP ($\delta = 1$) comparison with the state of the art on the Middlebury dataset [60]. Superscripts indicate the rank of each method for each image.

Method	*Tsukuba*	*Venus*	*Teddy*	*Cones*	*Art*	*Books*	*Dolls*	*Laun.*	*Moeb.*	*Rein.*	avg. ± std.
Bilinear Int.	10.40^{10}	3.29^{10}	11.80^{10}	14.60^{10}	34.59^{9}	13.07^{7}	14.59^{6}	22.85^{10}	17.15^{8}	17.48^{8}	15.98 ± 8.29^{9}
GF [2]	9.87^{9}	2.74^{9}	15.50^{11}	15.50^{11}	45.31^{11}	19.14^{10}	19.64^{11}	26.89^{11}	22.22^{11}	21.70^{9}	19.85 ± 11.2^{11}
RWR [62]	9.63^{8}	1.36^{7}	11.40^{9}	11.70^{9}	41.07^{10}	17.67^{9}	18.36^{10}	21.74^{9}	21.28^{10}	25.71^{11}	17.99 ± 10.8^{10}
Park *et al.* [12]	–	–	10.10^{8}	9.71^{7}	27.81^{8}	11.18^{5}	15.23^{7}	18.27^{8}	13.82^{7}	12.35^{6}	14.81 ± 5.97^{8}
TGV [10]	5.40^{6}	1.31^{6}	9.82^{7}	10.20^{8}	23.59^{7}	13.14^{8}	16.14^{9}	15.12^{7}	18.26^{9}	10.42^{5}	12.34 ± 6.40^{7}
WMF [11]	6.14^{7}	1.03^{5}	7.88^{3}	8.10^{5}	22.13^{6}	$\mathbf{8.67}^{1}$	$\mathbf{9.58}^{1}$	14.26^{6}	10.52^{4}	10.04^{3}	9.84 ± 5.48^{4}
WModF [25]	4.35^{5}	0.61^{3}	9.51^{5}	9.43^{6}	12.39^{3}	10.68^{4}	11.48^{4}	9.91^{3}	$\mathbf{8.60}^{1}$	12.63^{7}	8.96 ± 3.76^{3}
$\mathbf{u}_{l_2}$	3.60^{4}	1.60^{8}	9.28^{4}	7.86^{4}	17.88^{4}	11.90^{6}	12.63^{5}	11.74^{4}	13.61^{6}	10.28^{4}	10.04 ± 4.78^{5}
$\mathbf{u}_{l_1}$	2.59^{3}	0.67^{4}	9.51^{5}	7.56^{3}	17.98^{5}	19.29^{11}	15.86^{8}	12.75^{5}	13.08^{5}	21.85^{10}	12.11 ± 7.03^{6}
Ours ($\mathbf{u}^0 = \mathbf{u}_{l_2}$)	$\mathbf{2.39}^{1}$	$\mathbf{0.52}^{1}$	$\mathbf{7.39}^{1}$	$\mathbf{5.24}^{1}$	10.04^{2}	9.79^{3}	10.84^{3}	$\mathbf{8.11}^{1}$	9.49^{2}	$\mathbf{6.93}^{1}$	$\mathbf{7.07\pm3.43}^{1}$
Ours ($\mathbf{u}^0 = \mathbf{u}_{l_1}$)	2.45^{2}	$\mathbf{0.52}^{1}$	7.42^{2}	5.36^{2}	$\mathbf{9.96}^{1}$	9.52^{2}	10.77^{2}	8.22^{2}	9.49^{2}	6.98^{2}	$\mathbf{7.07\pm3.38}^{1}$



(a) Color image. (b) Ground truth. (c) Bilinear Int. (d) GF [2].
(e) RWR [62]. (f) Park *et al.* [12]. (g) TGV [10]. (h) WMF [11].
(i) WModF [25]. (j) $\mathbf{u}_{l_2}$. (k) $\mathbf{u}_{l_1}$. (l) SD filter.

Fig. 8. Visual comparison of upsampled depth images on a snippet of the *books* sequence in the Middlebury dataset [60]: (a) a high-resolution color image, (b) a ground-truth depth image, (c) bilinear interpolation, (d) GF [2], (e) RWR [62], (f) Park *et al.* [12], (g) TGV [10], (h) WMF [11], (i) WModF [25], (j) $\mathbf{u}_{l_2}$, (k) $\mathbf{u}_{l_1}$, and (l) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_1}$). In contrast to the static guidance filters such as GF [2] and WMF [11], the SD filter interpolates the low-resolution depth image using the structure of both color and depth images, preserving sharp depth edges.

the number of steps k. This table shows that the results obtained by the l_1 initialization converges faster than with the l_2 one. The l_1 and l_2 initializations converge in 5 and 35 steps, respectively. Both initializations give almost the same error at the convergence, but the l_1 initialization takes less time (7s) until convergence than the l_2 one (18s). We fix the step index k to 10, regardless of the initialization.

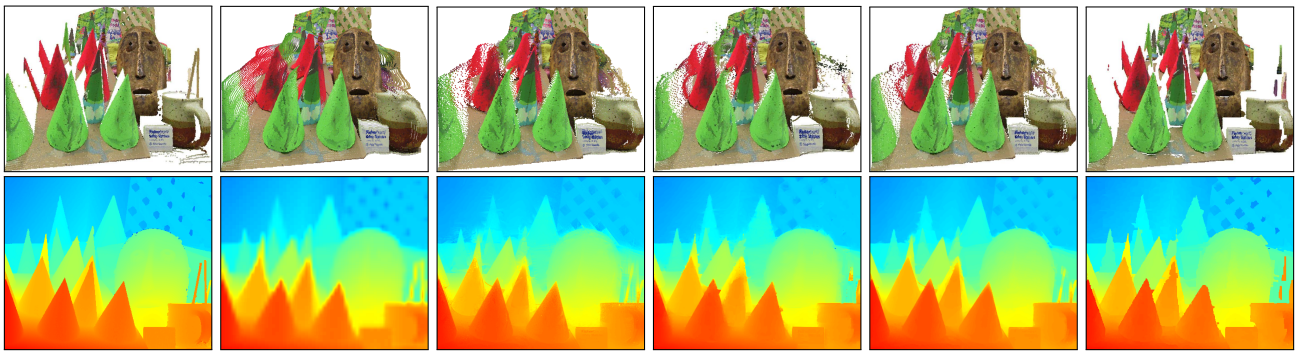
Comparison with the state of the art. We compare the average PBP ($\delta = 1$) of the SD filter and the state of the art in Table 3. Superscripts indicate the rank of each method in each column. This table shows that WMF [11] and WModF [25] give better quantitative results than other state-of-the-art methods, but that the SD filter outperforms all methods on average, regardless of the initialization. Figure 7(a) shows the variation of the average PBP as a function of the depth error tolerance δ. The SD filter outperforms other methods until the error tolerance δ reaches around 4, and after that all methods show the similar PBP performance. Figure 8 shows a visual comparison of the

TABLE 4
PBP* ($\delta = 1$) comparison on the Middlebury dataset [60].

Method	*Tsukuba*	*Venus*	*Teddy*	*Cones*
Bilinear Int.	46.30^{10}	37.10^{10}	35.30^{10}	35.80^{11}
GF [2]	43.20^{9}	26.50^{9}	37.50^{11}	34.40^{10}
RWR [62]	39.50^{8}	12.30^{6}	27.50^{8}	25.40^{9}
Park *et al.* [12]	–	–	25.40^{6}	19.90^{7}
TGV [10]	23.90^{6}	14.40^{7}	29.00^{9}	23.60^{8}
WMF [11]	28.00^{7}	10.10^{5}	22.20^{3}	19.20^{5}
WModF [25]	20.20^{5}	5.73^{4}	23.70^{5}	19.20^{5}
$\mathbf{u}_{l_2}$	15.50^{4}	15.10^{8}	26.30^{7}	18.90^{4}
$\mathbf{u}_{l_1}$	11.40^{3}	5.48^{3}	22.80^{4}	15.10^{3}
Ours ($\mathbf{u}^0 = \mathbf{u}_{l_2}$)	$\mathbf{10.40}^{1}$	5.10^{2}	20.30^{2}	$\mathbf{12.40}^{1}$
Ours ($\mathbf{u}^0 = \mathbf{u}_{l_1}$)	10.80^{2}	$\mathbf{4.83}^{1}$	$\mathbf{20.00}^{1}$	$\mathbf{12.40}^{1}$

upsampled depth images on a snippet of the *books* sequence, and it clearly shows the different behavior of static guidance and joint static/dynamic guidance models: The upsampled depth image has a gradient similar to that of the color image in static guidance methods such as GF [2] and WMF [11], which smooths depth edges and causes jagged artifacts in scene reconstruction (Fig. 9). In contrast, the SD filter interpolates the low-resolution depth image using joint static and dynamic guidance, and provides a sharp depth transition by considering the structure of both color and depth images (Fig. 8(l)). The upsampled depth image of $\mathbf{u}_{l_1}$ also preserves depth discontinuities, since the l_1 regularizer is robust to outliers. The l_1 regularizer, however, favors a piecewise constant solution. This is problematic in regions that violate the fronto-parallel surface assumption (e.g., a slanted surface), and every pixel in this surface has the same depth value (Fig. 8(k)). TGV [10] addresses this problem using a piecewise linear model, but this approach still smooths depth edges in low-contrast regions on the color image (Fig. 8(g)). Note that bilinear interpolation gives better quantitative results (Table 3) than the GF [2] and RWR [62], although its subjective quality looks worse than these methods (e.g., depth edges in Fig. 8(c-e)). This is because PBP does not differentiate depth discontinuities from other regions.

Table 4 shows the average PBP* ($\delta = 1$) measured near depth discontinuities, and it clearly shows the superior performance of the SD filter around these discontinuities. This can be further verified by observing the statistical distribution of the upsampled depth image as shown in Fig. 10. Bilinear interpolation loses most of the high-frequency information. Static guidance filtering meth-



(a) Ground truth. (b) Bilinear Int. (c) GF [2]. (d) TGV [10]. (e) WMF [11]. (f) SD filter.

Fig. 9. Examples of (top) point cloud scene reconstruction using (bottom) depth images computed by (a) the ground truth, (b) bilinear interpolation, (c) GF [2], (d) TGV [10], (e) WMF [11], and (f) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$). Current depth upsampling methods smooth depth edges, causing jagged artifacts.

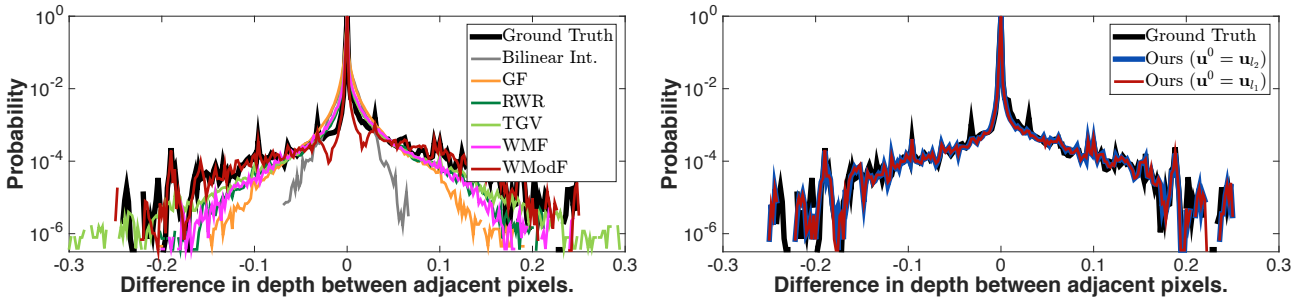


Fig. 10. Log_{10} of the normalized histograms (probability) of relative depths between adjacent pixels. Each histogram is computed using depth gradients along x- and y-axis, and then averaged on the Middlebury dataset [60]. **(Best viewed in colour.)**

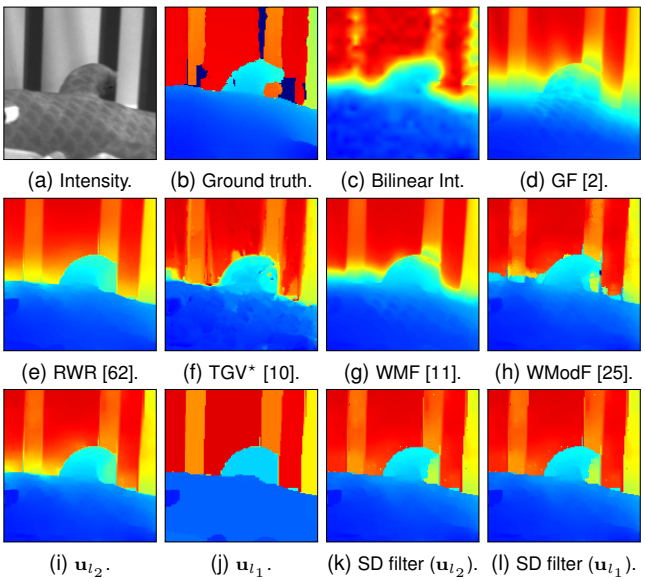
TABLE 5
The Kullback-Leibler divergence between the normalized histograms of depth gradients computed from the ground truths and upsampled depth images.

Bilinear Int.	230.73^{10}
GF [2]	108.19^{9}
RWR [62]	96.43^{8}
TGV [10]	22.43^{5}
WMF [11]	86.33^{7}
WModF [25]	13.11^{4}
$\mathbf{u}_{l_2}$	25.36^{6}
$\mathbf{u}_{l_1}$	9.72^{3}
Ours ($\mathbf{u}^0 = \mathbf{u}_{l_2}$)	$\mathbf{4.19}^{1}$
Ours ($\mathbf{u}^0 = \mathbf{u}_{l_1}$)	$\mathbf{4.63}^{2}$

ods show different statistical distributions from the ground-truth depth image. In contrast, the SD filter considers characteristics of depths as well, and delivers almost the same distribution as the ground truth depths. For evaluating this quantitatively, we also measure the Kullback-Leibler divergence between the statistical distributions of depth gradients obtained from the ground-truth and upsampled depth images in Table 5. From this table, we can see that the methods using a robust convex regularizer such as TGV [10] give better quantitative results than other methods, and the SD filter outperforms all methods. This verifies once more the importance of using a robust nonconvex regularizer and using both static and dynamic guidance in joint image filtering.

5.1.2 Real data

Recently, Ferstl *et al*. [10] have introduced a benchmark dataset that provides both low-resolution depth images captured by a ToF camera and highly accurate ground-truth depth images acquired with structured light. We have performed a quantitative evaluation using this dataset [10] in Fig. 7(b): We measure the average PBP of upsampled depth images with varying the depth error tolerance



(a) Intensity. (b) Ground truth. (c) Bilinear Int. (d) GF [2].
(e) RWR [62]. (f) TGV* [10]. (g) WMF [11]. (h) WModF [25].
(i) $\mathbf{u}_{l_2}$. (j) $\mathbf{u}_{l_1}$. (k) SD filter ($\mathbf{u}_{l_2}$). (l) SD filter ($\mathbf{u}_{l_1}$).

Fig. 11. Visual comparison of upsampled depth images on a snippet of the *shark* sequence in the Graz dataset [10]: (a) a high-resolution intensity image, (b) a ground-truth depth image, (c) bilinear interpolation, (d) GF [2], (e) RWR [62], (f) TGV* [10], (g) WMF [11], (h) WModF [25], (i) $\mathbf{u}_{l_2}$, (j) $\mathbf{u}_{l_1}$, (k) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$), and (l) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_1}$). Note that the l_1 regularizer in (j) favors a piecewise constant solution, and this is problematic in a slanted surface. TGV* denotes the results provided by the authors of [10].

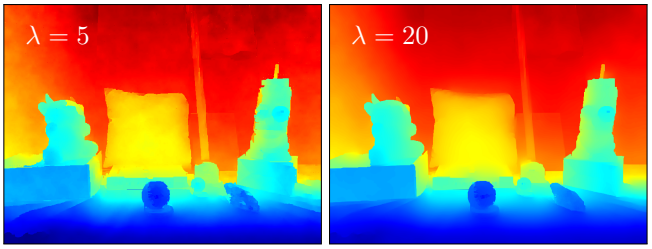


Fig. 12. Influence of the regularization parameter λ on the upsampled depth image. The results become smooth as the parameter λ increases.

δ. This shows that the SD filter outperforms all methods for all ranges of δ, regardless of the initialization. A qualitative comparison can be found in Fig. 11. Although most methods reduce sensor noise relatively well, they smooth depth edges as well. In contrast, the SD filter preserves sharp depth discontinuities, while suppressing sensor noise. The amount of noise in low-resolution depth images may change. We can deal with this by adjusting the regularization parameter that determines the degree of smoothing in the upsampled depth image, as shown in Fig. 12.

5.2 Scale-space filtering and texture removal

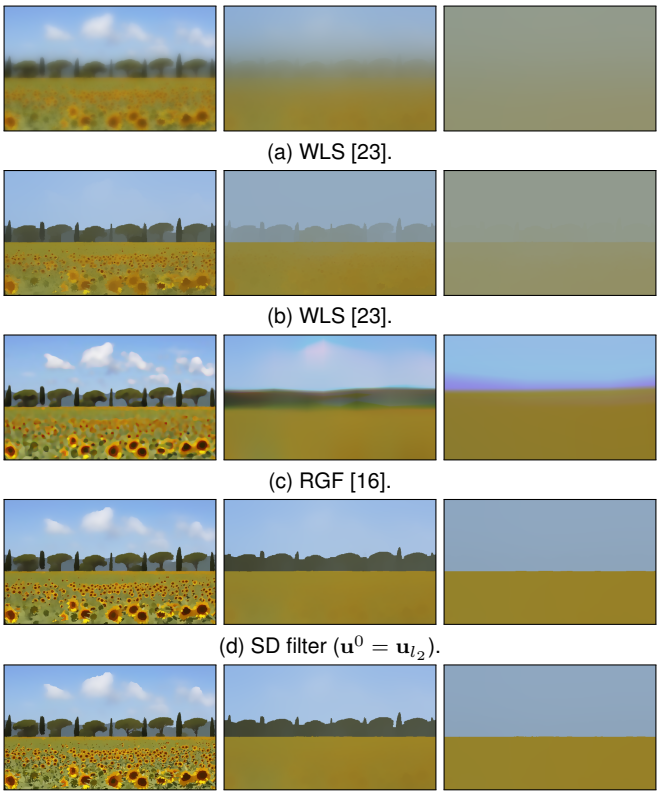
In natural scenes, various objects appear in different sizes and scales [63], and they contain a diversity of structural information from small-scale (e.g., texture and noise) to large-scale structures (e.g., boundaries). Extracting this hierarchy is an important part of understanding the relationship between objects, and it can be done by smoothing a scene through image filtering. We apply the SD filter to two kinds of scene decomposition tasks: scale-space filtering and texture removal.

Parameter settings. For scale-space filtering, the input image $\mathbf{f}$ is guided by itself ($\mathbf{g} = \mathbf{f}$). The regularization parameter λ varies from 5×10 to 5×10^4. In texture removal, the static guidance image is set to a Gaussian-filtered version of the input image, $\mathbf{g} = \mathbf{G}_\sigma \mathbf{f}$ where $\mathbf{G}_\sigma$ is the Gaussian kernel with standard deviation σ. The regularization parameter λ and σ vary from 5×10 to 2×10^3 and from 1 to 2, respectively. The bandwidths, μ and ν, the step index k, and the confidence value c_i are fixed to 5, 40, 5, and 1, respectively, for both applications.

5.2.1 Scale-space filtering

We obtain a scale-space representation by applying the SD filter while increasing the regularization parameter, as shown in Fig. 13. The SD filter preserves object boundaries well, even at coarse scales, and it is robust to global intensity shift.

Comparison with the state of the art. We compare the scale-space representation of WLS [23], RGF [16], and the SD filter in Fig. 13. The WLS framework [23], a representative of static guidance filtering, alters the scale of image structure by varying the regularization parameter. It suffers from global intensity shifting [2] (Fig. 13(a-b)), and may not preserve image structures at coarse scales (Fig. 13(a)). RGF [16] is the state of the art in scale-space filtering, and dynamic guidance in RGF could alleviate these problems. However, this filter does not use the structure of the input image, and controls the scale by the isotropic Gaussian kernel, leading to poor boundary localization at coarse scales (Fig. 13(c)). The SD filter uses the structure of input and desired output images, and the scale depends on the regularization parameter, providing well localized boundaries



(a) WLS [23].

(b) WLS [23].

(c) RGF [16].

(d) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$).

(e) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_1}$).

Fig. 13. Examples of scale-space filtering. (a) WLS [23] (from left to right, $\lambda = 5 \times 10, 2 \times 10^3, 5 \times 10^4$, $\mu = 5$), (b) WLS [23] (from left to right, $\lambda = 5 \times 10^3, 2 \times 10^5, 5 \times 10^6$, $\mu = 40$), (c) RGF [16] (from left to right, $\sigma_s = 5, 5 \times 10, 2.5 \times 10^2$, $\sigma_r = 0.05$, $k = 5$), (d) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$, and from left to right, $\lambda = 5 \times 10, 2 \times 10^3, 5 \times 10^4$), (e) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_1}$, and from left to right, $\lambda = 5 \times 10, 2 \times 10^3, 5 \times 10^4$). In these examples, we manually pick the scale parameter λ of the SD filter between 5×10 to 5×10^4. The parameters for other methods are similarly chosen, such that all methods have similar smoothing levels.

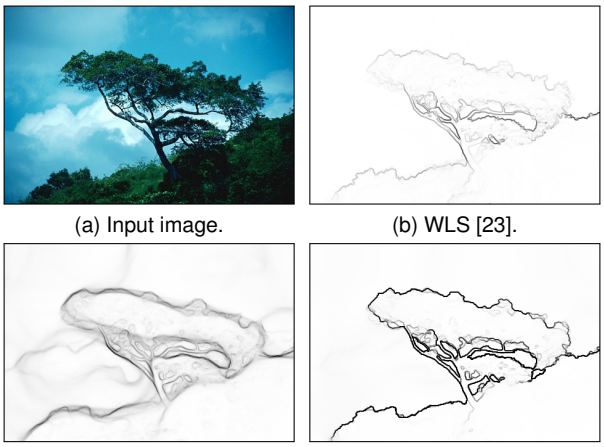
TABLE 6
Evaluation of boundary localization accuracy on the BSDS300 database [64]. Scale-space is constructed by WLS [23] ($\mu = 40$), RGF [16] ($\sigma_r = 0.05$, $k = 5$), and the SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$ and $\mathbf{u}^0 = \mathbf{u}_{l_1}$), with varying scale parameters.

Mathod	ODS	OIS	ODS*	OIS*
WLS [23]	0.51[4]	0.52[4]	0.51[4]	0.52[4]
RGF [16]	0.55[3]	0.60[3]	0.60[3]	0.63[3]
Ours ($\mathbf{u}^0 = \mathbf{u}_{l_2}$)	0.61[2]	**0.63**[1]	0.62[2]	**0.65**[1]
Ours ($\mathbf{u}^0 = \mathbf{u}_{l_1}$)	**0.62**[1]	**0.63**[1]	**0.63**[1]	**0.65**[1]

even at coarse scales. Moreover, it is robust to global intensity shift (Fig. 13(d-e)).

For a good scale-space representation, object boundaries should be sharp and coincide well with the meaningful boundaries at each scale [17]. There is no universally accepted evaluation method for scale-space filtering. Motivated by the evaluation protocol for boundary detection, we measure ODS and OIS [65] with the BSDS300 database [64], and evaluate boundary localization accuracy of scale-space filtering in Table 6. ODS is the F-measure[6] at a fixed contour threshold across the entire dataset, while OIS refers to the per-image best F-measure. For all images in the dataset, ODS and OIS are measured using the gradient magnitudes

6. The F-measure is the harmonic mean of precision and recall.



(a) Input image. (b) WLS [23]. (c) RGF [16]. (d) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$).

Fig. 14. Examples of the gradient magnitude averaged over scale-space: Given (a) an input image, scale-space is obtained by (b) WLS [23] ($\mu = 40$), (c) RGF [16] ($\sigma_r = 0.05$, $k = 5$), and (d) the SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$) with varying scale parameters. For each method, we set the maximum scale parameters manually, such that all the filtered images have similar smoothing levels. The gradient magnitude is then computed at each scale and averaged over scale-space.

of filtered images, each of which is averaged *over* scale-space, as shown in Fig. 14. We obtain the filtered images from the input image while varying scale parameters, i.e., λ in WLS [23] and the SD filter, and σ_s in RGF [16]. The maximum scale parameter is set manually for each method, such that all the filtered images are smoothed with similar levels. Similarly, we also measure ODS* (resp. OIS*) with the gradient images of filtering results, but using a fixed scale parameter that provides maximum ODS (resp. OIS) for each image. That is, ODS* (resp. OIS*) is the maximum value of ODS (resp. OIS) over scale-space. In both cases, the SD filter outperforms other filtering methods.

5.2.2 Texture removal

For removing texture while maintaining other high-frequency image structures, we need a guidance image that does not have the texture, but contains large image structures. Since it is hard to get such an image, we set the static guidance image to the Gaussian-filtered version of the original image with standard deviation σ. This removes the textures of scale σ, but it also smooths structural edges (e.g., boundaries). Our dynamic guidance and fidelity term reconstruct smoothed boundaries as shown in Fig. 15.

Comparison with the state of the art. As in scale-space filtering, to the best of our knowledge, there is no universally accepted evaluation protocol for texture removal. Filtering examples of (top) regular and (bottom) irregular textures are shown in Fig. 15. Most methods extract prominent image structures while filtering out textures. The Cov. M1 method [14] adopts a weighted average process to eliminate textures. The weight is computed using the covariance of features in a local patch to encode the repetitive patterns of textures. This method, however, smooths object boundaries and corners (Fig. 15(b)), and needs lots of computational cost for weight computation. RTV [15] optimizes an objective function similar to the SD filter, but with relative total variation. Although this method uses a nonconvex regularizer, it penalizes large gradients similar to the total variation regularizer, which smooths structural edges (Fig. 15(c)). As pointed out in [15], RTV cannot perfectly extract image structures that have similar scale and appearance to the underlying textures. RGF [16] can control the scale of image structure to be extracted, and preserves object boundaries well. However, it does not protect corners, possibly due to the inherent limitation of the Gaussian kernel used in RGF, and color artifacts are observed (Fig. 15(d)). On the other hand, the SD filter removes textures without artifacts, and maintains small, high-frequency, but important structures to be preserved such as corners (Fig. 15(e-f)). We can see that the SD filter with the l_1 initialization better preserves edges and corners than with the l_2 one.

5.3 Other applications

The SD filter can be applied to joint image restoration tasks. We apply it to RGB/NIR and flash/non-flash denoising problems as shown in Figs. 16 and 17. In RGB/NIR denoising, the color image $\mathbf{f}$ is regularized with the flash NIR image $\mathbf{g}$. Similarly, the non-flash image $\mathbf{f}$ is regularized with the flash image $\mathbf{g}$. Since there exist structural dissimilarities between static guidance and input images ($\mathbf{g}$ and $\mathbf{f}$), the results might have artifacts and unnatural appearance. For example, static guidance filtering (e.g., GF [2]) cannot deal with gradient reversal in flash NIR images [4], resulting in smoothed edges. The SD filter handles structural differences between guidance and input images, and gives qualitative results comparable to the state of the art [4].

6 Discussion

We have presented a robust joint filtering framework that is widely applicable to computer vision and computational photography tasks. Contrary to static guidance filtering methods, we leverage dynamic guidance images, and can exploit the structural information of the input image. Although our model does not have a closed-form solution, the corresponding optimization problem can be solved by an efficient algorithm that converges rapidly to a local minimum. The simple and flexible formulation of the SD filter makes it applicable to a great variety of applications, as demonstrated by our experiments.

Appendix A

Proposition 1. *Ψ_ν^y is a surrogate function of ψ_ν.*

Proof. Let us consider the following function:

$$f_\nu(x) = (1 - \exp(-\nu x))/\nu. \tag{27}$$

Since $f_\nu''(x) < 0$, this function is strictly concave, i.e.,

$$\forall x, y, \quad f_\nu(x) \leq f_\nu(y) + (x - y) f_\nu'(y), \tag{28}$$

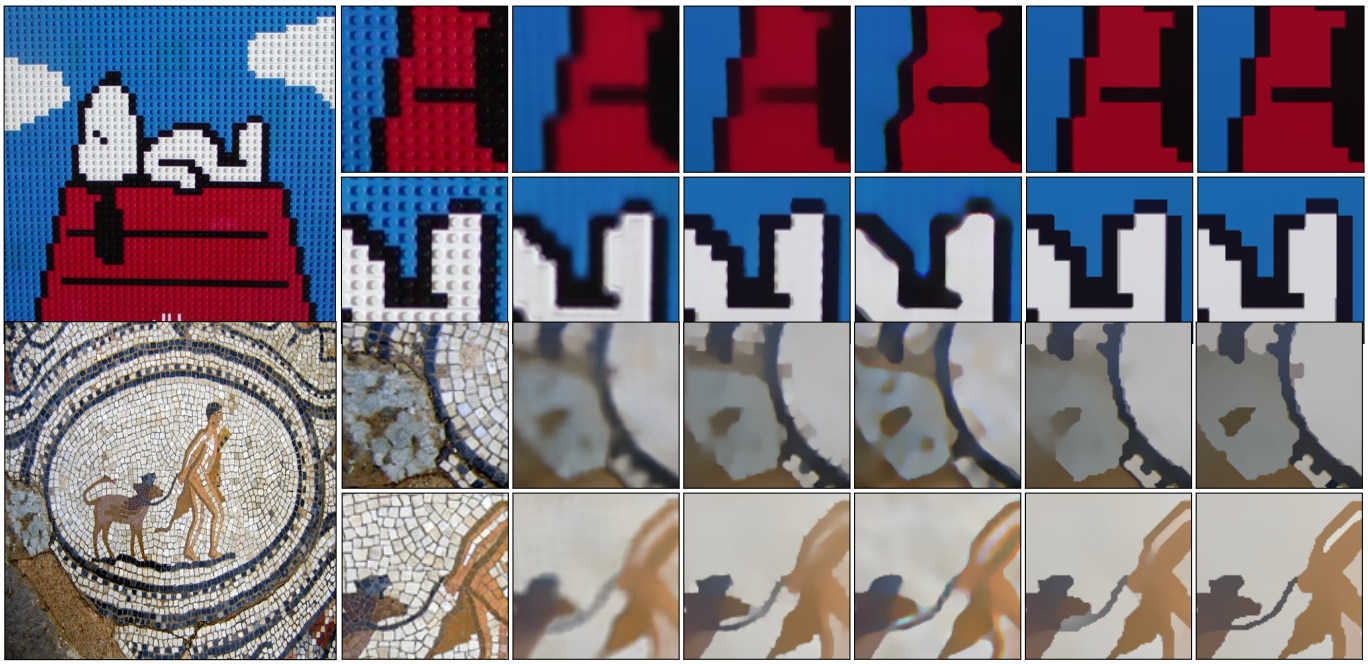
with equality holding when x is equal to y. Thus, a surrogate function f_ν^y is

$$\begin{aligned} f_\nu^y(x) &= f_\nu(y) + (x - y) f_\nu'(y) \\ &= f_\nu(y) + (x - y)(1 - \nu f_\nu(y)). \end{aligned} \tag{29}$$

It follows that

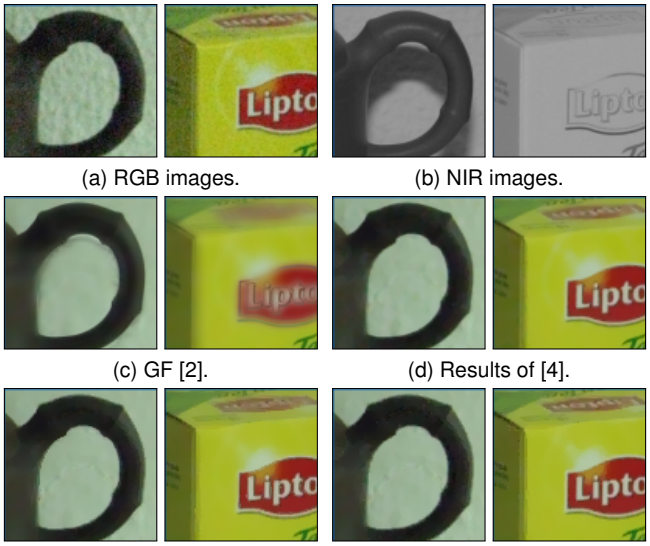
$$\begin{aligned} \psi_\nu(x) = f_\nu(x^2) &\leq f_\nu^{y^2}(x^2) \\ &= \psi_\nu(y) + (x^2 - y^2)(1 - \nu\psi_\nu(y)) = \Psi_\nu^y(x). \end{aligned} \tag{30}$$

□



(a) Input image and snippets. (b) Cov. M1 [14]. (c) RTV [15]. (d) RGF [16]. (e) SD filter. (f) SD filter.

Fig. 15. Visual comparison of texture removal for (top) regular and (bottom) irregular textures. (a) Input image, (b) Cov. M1 [14] ($\sigma = 0.3$, $r = 10$), (c) RTV [15] ($\lambda = 0.01$, $\sigma = 6$), (d) RGF [16] ($\sigma_s = 5$, and from top to bottom, $\sigma_r = 0.1, 0.05$, $k = 5$), (e) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$, and from top to bottom, $\lambda = 1 \times 10^3, 1 \times 10^2, \sigma = 2$), (f) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_1}$, and from top to bottom, $\lambda = 2 \times 10^3, 3 \times 10^2, \sigma = 2$). We set all parameters through extensive experiments for the best qualitative performance.



(a) RGB images. (b) NIR images.

(c) GF [2]. (d) Results of [4].

(e) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$). (f) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_1}$).

Fig. 16. RGB and flash NIR image restoration. (a) RGB images, (b) NIR images, (c) GF [2] ($r = 3$, $\varepsilon = 4^{-4}$), (d) results of [4], (e) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$, $\lambda = 10$, $\mu = 60$, $\nu = 30$, $k = 5$), (f) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_1}$, $\lambda = 10$, $\mu = 60$, $\nu = 30$, $k = 5$). The results of (d) are from the project webpage of [4].

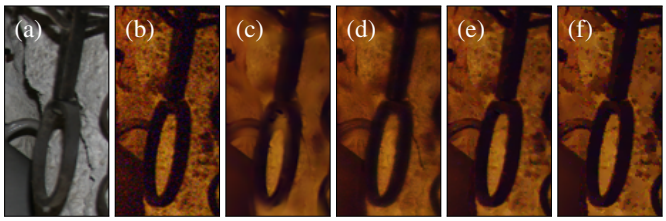


Fig. 17. Flash and non-flash image restoration. (a) Flash image, (b) non-flash image, (c) GF [2] ($r = 3$, $\varepsilon = 4^{-4}$), (d) result of [13], (e) result of [4], (f) SD filter ($\mathbf{u}^0 = \mathbf{u}_{l_2}$, $\lambda = 15$, $\mu = 60$, $\nu = 30$, $k = 5$). The results of (d) and (e) are from the project webpages of [13] and [4], respectively.

APPENDIX B

Let us consider the following objective function:

$$\mathcal{E}(u) = \sum_i \sum_{j \in \eta} \phi_s(i-j)\phi_\mu(g_i - g_j)\psi_\nu(u_i - f_j). \tag{31}$$

The surrogate function of (31) can be found using the inequality in (9) as follows:

$$\mathcal{Q}^k(u) = \sum_{i,j} w_{ij} w^g_{ij} \phi_\nu(u^k_i - f_j)(u_i - f_j)^2 + \tag{32}$$

$$\frac{1}{\nu} \sum_{i,j} w_{ij} w^g_{ij} \left\{ 1 - \phi_\nu(u^k_i - f_j) - \nu \phi_\nu(u^k_i - f_j)(u^k_i - f_j)^2 \right\}.$$

where $w_{ij} \equiv \phi_s(i-j)$ and $w^g_{ij} \equiv \phi_\mu(g_i - g_j)$. Then,

$$u^{k+1}_i = \arg\min_u \sum_{j \in \eta} w_{ij} w^g_{ij} \phi_\nu(u^k_i - f_j)(u_i - f_j) \tag{33}$$

$$= \frac{\sum_{j \in \eta} \phi_s(i-j)\phi_\mu(g_i - g_j)\phi_\nu(u^k_i - f_j) f_j}{\sum_{j \in \eta} \phi_s(i-j)\phi_\mu(g_i - g_j)\phi_\nu(u^k_i - f_j)}.$$

ACKNOWLEDGMENTS

The authors would like to thank Francis Bach for helpful discussions. This work was supported in part by the ERC grant VideoWorld and the Institut Universitaire de France.

REFERENCES

[1] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 298–311, 1997.

[2] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.

[3] J. Kopf, M. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 96–100, 2007.

[4] Q. Yan, X. Shen, L. Xu, S. Zhuo, X. Zhang, L. Shen, and J. Jia, "Cross-field joint image restoration via scale map," *in Proc. Int. Conf. Comput. Vis.*, pp. 1537–1544, 2013.

[5] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 504–511, 2013.

[6] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650–656, 2006.

[7] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1164–1172, 2015.

[8] B. Ham, M. Cho, C. Schmid, and J. Ponce, "Proposal flow," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[9] T. Zhou, Y. J. Lee, X. Y. Stella, and A. A. Efros, "Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1191–1200, 2015.

[10] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," *in Proc. Int. Conf. Comput. Vis.*, pp. 993–1000, 2013.

[11] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," *in Proc. Int. Conf. Comput. Vis.*, pp. 49–56, 2013.

[12] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, "High quality depth map upsampling for 3d-TOF cameras," *in Proc. Int. Conf. Comput. Vis.*, pp. 1623–1630, 2011.

[13] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graphic.*, vol. 23, no. 3, pp. 664–672, 2004.

[14] L. Karacan, E. Erdem, and A. Erdem, "Structure-preserving image smoothing via region covariances," *ACM Trans. Graphic.*, vol. 32, no. 6, pp. 176:1–176:11, 2013.

[15] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graphic.*, vol. 31, no. 6, pp. 139:1–139:10, 2012.

[16] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," *in Proc. Eur. Conf. Comput. Vis.*, pp. 815–830, 2014.

[17] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.

[18] T. Brox, O. Kleinschmidt, and D. Cremers, "Efficient nonlocal means for denoising of textural patterns," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1083–1092, 2008.

[19] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics-Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[20] W. Liu, P. P. Pokharel, and J. C. Príncipe, "Correntropy: properties and applications in non-gaussian signal processing," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5286–5298, 2007.

[21] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 4823–4831, 2015.

[22] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *in Proc. Int. Conf. Comput. Vis.*, pp. 839–846, 1998.

[23] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graphic.*, vol. 27, no. 3, pp. 67:1–67:10, 2008.

[24] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l_0 gradient minimization," *ACM Trans. Graphic.*, vol. 30, no. 6, pp. 174:1–174:12, 2011.

[25] D. Min, J. Lu, and M. Do, "Depth video enhancement based on weighted mode filtering," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1176–1190, 2012.

[26] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graphic.*, vol. 23, no. 3, pp. 689–694, 2004.

[27] M. Lang, O. Wang, T. Aydin, A. Smolic, and M. H. Gross, "Practical temporal consistency for image-based graphics applications," *ACM Trans. Graphic.*, vol. 31, no. 4, pp. 34:1–34:8, 2012.

[28] J. Weickert, "A review of nonlinear diffusion filtering," *International Conference on Scale-Space Theories in Computer Vision*, pp. 1–28, 1997.

[29] H. Badri, H. Yahia, and D. Aboutajdine, "Fast edge-aware processing via first order proximal approximation," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 6, pp. 743–755, 2015.

[30] X. Shen, C. Zhou, L. Xu, and J. Jia, "Mutual-structure for joint filtering," *in Proc. Int. Conf. Comput. Vis.*, pp. 3406–3414, 2015.

[31] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep joint image filtering," *in Proc. Eur. Conf. Comput. Vis.*, pp. 1–16, 2016.

[32] P. Kohli, L. Ladicky, and P. H. Torr, "Robust higher order potentials for enforcing label consistency," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–8, 2008.

[33] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *Adv. Neural Inf. Process. Syst*, 2011.

[34] C. Liang-Chieh, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *International Conference on Learning Representations*, 2015.

[35] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," *in Proc. Int. Conf. Comput. Vis.*, pp. 1529–1537, 2015.

[36] J. T. Barron and B. Poole, "The fast bilateral solver," *in Proc. Eur. Conf. Comput. Vis.*, 2016.

[37] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille, "Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[38] D. Chan, H. Buisman, C. Theobalt, S. Thrun *et al.*, "A noise-aware filter for real-time depth upsampling," *in Proc. Eur. Conf. Comput. Vis. Workshops*, 2008.

[39] O. Mac Aodha, N. D. Campbell, A. Nair, and G. J. Brostow, "Patch based synthesis for single depth image super-resolution," *in Proc. Eur. Conf. Comput. Vis.*, pp. 71–84, 2012.

[40] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson, "Crisp boundary detection using pointwise mutual information," *in Proc. Eur. Conf. Comput. Vis.*, pp. 799–814, 2014.

[41] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257–266, 2002.

[42] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011.

[43] G. R. Lanckriet and B. K. Sriperumbudur, "On the convergence of the concave-convex procedure," *Advances in Neural Information Processing Systems*, pp. 1759–1767, 2009.

[44] J. Mairal, "Incremental majorization-minimization optimization with application to large-scale machine learning," *arXiv preprint arXiv:1402.4419*, 2014.

[45] Z. Zhang, J. T. Kwok, and D.-Y. Yeung, "Surrogate maximization/minimization algorithms for adaboost and the logistic regression model," *in Proc. Int. Conf. Machine Learning*, p. 117, 2004.

[46] C. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of statistics*, pp. 95–103, 1983.

[47] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[48] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[49] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least-squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.

[50] R. Chandrasekaran and A. Tamir, "Open questions concerning weiszfeld's algorithm for the fermat-weber location problem," *Mathematical Programming*, vol. 44, no. 1-3, pp. 293–295, 1989.

[51] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 3, pp. 367–383, 1992.

[52] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM journal on scientific computing*, vol. 27, no. 3, pp. 937–966, 2005.

[53] M. Nikolova and R. H. Chan, "The equivalence of half-quadratic minimization and the gradient linearization iteration," *IEEE Trans. on Image Process.*, vol. 16, no. 6, pp. 1623–1627, 2007.

[54] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate," *ACM Trans. Math. Softw.*, vol. 35, no. 3, pp. 22:1–22:14, 2008.

[55] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. Do, "Fast global image smoothing based on weighted least squares," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5638–5653, 2014.
[56] T. Kim, K. Lee, and S. Lee, "Generative image segmentation using random walks with restart," *in Proc. Eur. Conf. Comput. Vis.*, pp. 264–275, 2008.
[57] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, "Automatic multimedia cross-modal correlation discovery," *in Proc. ACM KDD*, pp. 653–658, 2004.
[58] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," *In Proc. IEEE Conf. Data Mining*, 2006.
[59] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 421–432, 1998.
[60] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1, pp. 7–42, 2002.
[61] M.-Y. Liu, O. Tuzel, and Y. Taguchi, "Joint geodesic upsampling of depth images," *in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 169–176, 2013.
[62] B. Ham, D. Min, and K. Sohn, "A generalized random walk with restart and its application in depth up-sampling and interactive segmentation," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2574–2588, 2013.
[63] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," *in Proc. Int. Conf. Comput. Vis.*, pp. 1879–1886, 2011.
[64] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," *in Proc. Int. Conf. Conput. Vis.*, vol. 2, pp. 416–423, 2001.
[65] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

Bumsub Ham is an an Assistant Professor of Electrical and Electronic Engineering at Yonsei University in Seoul, Korea. He received the B.S. and Ph.D. degrees in Electrical and Electronic Engineering from Yonsei University in 2008 and 2013, respectively. From 2014 to 2016, he was Post-Doctoral Research Fellow with Willow Team of INRIA Rocquencourt, École Normale Supérieure de Paris, and Centre National de la Recherche Scientifique. His research interests include computer vision, computational photography, and machine learning, in particular, regularization and matching, both in theory and applications.

Minsu Cho is an Assistant Professor of computer science and engineering at POSTECH in Pohang, South Korea. He obtained his PhD degree in Electrical Engineering and Computer Science from Seoul National University in 2012. Before joining POSTECH in 2016, he worked as an Inria starting researcher in the ENS/Inria/CNRS Project team WILLOW at cole Normale Superiure, Paris, France. His research lies in the areas of computer vision and machine learning, especially in the problems of object discovery, weakly-supervised learning, and graph matching.

Jean Ponce is a Professor at École Normale Supérieure (ENS) and PSL Research University in Paris, France, where he leads a joint ENS/INRIA/CNRS research team, WILLOW, that focuses on computer vision and machine learning. Prior to this, he served for over 15 years on the faculty of the Department of Computer Science and the Beckman Institute at the University of Illinois at Urbana-Champaign. Dr. Ponce is the author of over 150 technical publications, including the textbook "Computer Vision: A Modern Approach", in collaboration with David Forsyth. He is a member of the Editorial Boards of Foundations and Trends in Computer Graphics and Vision, the International Journal of Computer Vision, and the SIAM Journal on Imaging Sciences. He was also editor-in-chief of the International Journal on Computer Vision (2003-2008), an Associate Editor of the IEEE Transactions on Robotics and Automation (1996-2001), and an Area Editor of Computer Vision and Image Understanding (1994-2000). Dr. Ponce was Program Chair of the 1997 IEEE Conference on Computer Vision and Pattern Recognition and served as General Chair of the year 2000 edition of this conference. In 2003, he was named an IEEE Fellow for his contributions to Computer Vision, and he received a US patent for the development of a robotic parts feeder. In 2008, he served as General Chair for the European Conference on Computer Vision.