

Adaptive and Dynamic Regularization for Rolling Guidance Image Filtering

M. Fukatsu^{1,2}  S. Yoshizawa²  H. Takemura¹  H. Yokota² 

¹Department of Mechanical Engineering, Graduate school of Science and Technology, Tokyo University of Science, Japan

²Image Processing Research Team, RAP, RIKEN, Japan

Abstract

Separating shapes and textures of digital images at different scales is useful in computer graphics. The Rolling Guidance (RG) filter, which removes structures smaller than a specified scale while preserving salient edges, has attracted considerable attention. Conventional RG-based filters have some drawbacks, including smoothness/sharpness quality dependence on scale and non-uniform convergence. This paper proposes a novel RG-based image filter that has more stable filtering quality at varying scales. Our filtering approach is an adaptive and dynamic regularization for a recursive regression model in the RG framework to produce more edge saliency and appropriate scale convergence. Our numerical experiments demonstrated filtering results with uniform convergence and high accuracy for varying scales.

CCS Concepts

• **Computing methodologies** → **Computational photography; Image processing;**

1. Introduction

Textures in digital images are composed of various scales ($\sigma \in \mathbb{R}_{>0}$). Thus, the Rolling Guidance (RG) filter [ZSXJ14b], which removes structures smaller than σ while preserving salient edges, has various applications in computer graphics. The RG filter recursively applies a nonlinear convolution called a joint filter (e.g., the Bilateral Filter (BF) [ED04, PSA*04], Domain Transform (DT) [GO11], and Guided Filter (GF) [HST13]) that reflects the previous filtering result as a guide to the same input image every time. In contrast to naive iterations of these joint filters (e.g., an iterative BF where it is difficult to control the iteration number to achieve the target σ), RG only updates its guidance and stably converges to the target σ (see Fig. 1 for an example of RG-based filtering results).

The GF-based joint filters are popular and have been widely used in practice [LZZ*14, KCWL15, DYT*19, LYB17, SCG*21, GDANW21] because fast and accurate approximation of a joint BF (and also a RG-BF) are not trivial, and RG-DT has the problem of non-uniform convergence [YY21] in elongated texture regions. However, RG-GF does not generate the expected level of smoothness (i.e., does not achieve the target σ as demonstrated in Fig. 2(b)) when σ is large [HST13, ZSXJ14b], and the sharpness of salient edges is also low compared with RG-BF/DT. In addition, the fast box-kernel averaging [SKM98] that is often employed in GF and RG-GF may produce undesirable artifacts [YY14].

In this paper, we propose a novel scale-aware image filter based on the RG-GF framework that restores salient edges with high con-

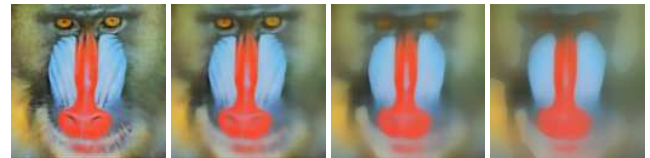


Figure 1: The filtering results of our proposed method with varying $\sigma \in \{4, 8, 16, 32\}$ (left to right), where $\epsilon = 0.01$ and $t = 20$.

vergence for various scales in order to tackle the above-mentioned problems. The main idea behind our approach is very simple and inspired by the idea of weighted GFs [LZZ*14, KCWL15], which have employed an adaptive regularization parameter (controlling edge-awareness) via statistical information (local variance of a guidance image). Instead of using a constant or local variance in the conventional filters, we directly model our regularization such that the regions of salient edges (small structures) become closer to the input (target scale) image. More precisely, our adaptive and dynamic regularization function consists of the normalized-difference between RG-GF filtered and target scale images that resemble σ to 2σ scale band-pass effects. We also adapt the Domain-Splitting (DS) technique [YY14] to our filter, which approximates L^1 Gaussian convolution quickly and very accurately instead of using conventional box-kernel averaging to avoid undesirable artifacts. Our numerical experiments show that the filter has high approximation accuracy and convergence rates, and the sharpness and smoothness of our filtering results are desirable at various scales (Fig. 2).

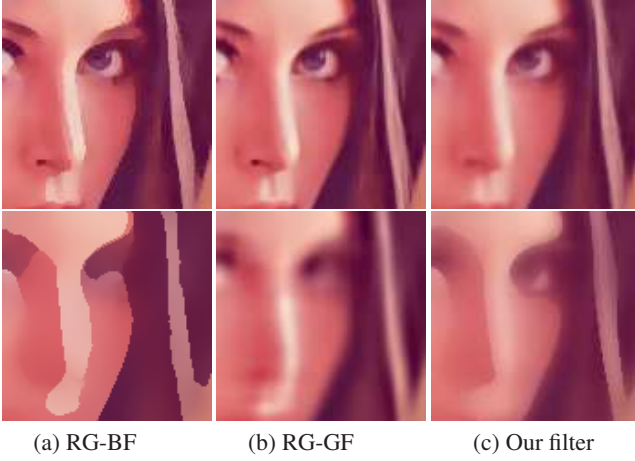


Figure 2: RG-BF [ZSXJ14a] (a), RG-GF [ZSXJ14b] (b), and our filtering (c) results, where $\sigma_r = 20$ (a) is the edge-awareness of joint BF, $\varepsilon = \{0.001, 0.01\}$ (b,c), $t = 20$, and $\sigma \in \{2, 8\}$ (top and bottom, respectively). All terms are defined in the text.

2. Proposed Filter

For a given image pixel $\mathbf{x} \in \mathbb{R}^2$, let $\mathbf{I} \equiv \{I_v = I_v(\mathbf{x})\}$ and $\mathbf{J}^t \equiv \{J_v^t = J_v^t(\mathbf{x})\}$ on \mathbb{R}^3 be the input and our t -th filtered image colors, respectively. Our filter is then recursively defined by

$$\mathbf{J}_v^{t+1}(\mathbf{x}) = GF(\mathbf{J}^t, I_v, \varepsilon^t) \equiv \mathbf{a}_v^T \mathbf{J}^t + b_v, \quad (1)$$

where $v \in \{r, g, b\}$ denotes each color channel, $t \in \mathbb{N} \cup \{0\}$ is an iteration number, $GF(\cdot)$ is our modified guided filter of the guidance \mathbf{J}^t and filtering I_v colors with a regularization function of $\varepsilon^t = \varepsilon^t(\mathbf{x}) \in \mathbb{R}$, T represents the transpose operation, and $\{\mathbf{a}_v = \mathbf{a}_v(\mathbf{x})\} \in \mathbb{R}^{3 \times 3}$ and $\{b_v = b_v(\mathbf{x})\} \in \mathbb{R}^3$ are the coefficients of the t -th $GF(\cdot)$. Here, the initial smoothed image $\mathbf{J}^0 \equiv f(\mathbf{I})$ is defined by the normalized L^1 Gaussian convolution

$$f(h) = \frac{\int G_\sigma(\mathbf{x} - \mathbf{y})h(\mathbf{y})d\mathbf{y}}{\int G_\sigma(\mathbf{x} - \mathbf{y})d\mathbf{y}}, \quad G_\sigma(\cdot) = \exp\left(-\frac{|\cdot|}{\sigma}\right), \quad (2)$$

at \mathbf{x} where $\mathbf{y} \in \mathbb{R}^2$, $h = h(\mathbf{x}) \in \mathbb{R}$ is its integrand (each element for the vector case), $|\cdot|$ denotes an absolute value, and $\exp(\cdot)$ is the exponential function. The L^1 Gaussian convolution never increases the number of extrema for the one-dimensional continuous case according to scale-space theory [Lin97] [§6.2.3]. On the other hand, box-kernel averaging (often used in conventional GF-based filters, including RG-GF [ZSXJ14b]) may produce undesired artifacts [YY21] because its Fourier domain image oscillates as a *sinc* function. In our filter, $f(\cdot)$ is implemented by using the DS [YY14] technique in order to avoid such artifacts.

For each $v \in \{r, g, b\}$ and t , minimizing the following functional (also known as a ridge regression model [HK70, HST13])

$$\sum_{\mathbf{y}} G_\sigma(\mathbf{x} - \mathbf{y}) \left((\mathbf{a}_v^T(\mathbf{y})\{J_v^t(\mathbf{x})\} + b_v(\mathbf{y}) - I_v(\mathbf{x}))^2 + \varepsilon^t(\mathbf{x})|\mathbf{a}_v(\mathbf{y})|^2 \right)$$

with respect to (\mathbf{a}_v, b_v) yields the GF coefficients as

$$\mathbf{a}_v = (\mathbf{C} + \varepsilon^t \mathbf{U})^{-1} (f(I_v) \mathbf{J}^t - f(I_v) f(\mathbf{J}^t)), \quad (3)$$

$$b_v = f(I_v) - \mathbf{a}_v^T f(\mathbf{J}^t), \quad (4)$$

where \mathbf{U} and \mathbf{C} on $\mathbb{R}^{3 \times 3}$ are the identity and following covariance (of \mathbf{J}^t with respect to $f(\cdot)$) matrices, respectively:

$$\mathbf{C} = \begin{pmatrix} f((J_r^t)^2) - (f(J_r^t))^2 & f(J_r^t J_g^t) - f(J_r^t) f(J_g^t) & f(J_r^t J_b^t) - f(J_r^t) f(J_b^t) \\ f(J_r^t J_g^t) - f(J_r^t) f(J_g^t) & f((J_g^t)^2) - (f(J_g^t))^2 & f(J_g^t J_b^t) - f(J_g^t) f(J_b^t) \\ f(J_r^t J_b^t) - f(J_r^t) f(J_b^t) & f(J_g^t J_b^t) - f(J_g^t) f(J_b^t) & f((J_b^t)^2) - (f(J_b^t))^2 \end{pmatrix}.$$

According to Eqs. (3) and (4), a smaller (greater) ε^t in Eq. (3) provides a sharper, i.e., closer to \mathbf{I} (smoother, i.e., closer to \mathbf{J}^0) filtering result. Also, \mathbf{J}^{t+1} is desirable for our purpose if \mathbf{J}^{t+1} becomes \mathbf{I} on the salient edges and if other regions converge to \mathbf{J}^0 . Because the position and the amount of the regularization magnitude can be estimated by the difference between \mathbf{J}^{t+1} and \mathbf{J}^0 , we model our adaptive and dynamic regularization function $\varepsilon^t(\mathbf{x})$ by the following normalized-difference magnitude:

$$\varepsilon^{t+1}(\mathbf{x}) = \varepsilon^0 \frac{\delta + |\mathbf{J}^{t+1}(\mathbf{x}) - \mathbf{J}^0(\mathbf{x})|}{\delta + f(|\mathbf{J}^{t+1}(\mathbf{x}) - \mathbf{J}^0(\mathbf{x})|)}, \quad \varepsilon^0 \equiv \varepsilon \mathcal{L}^2, \quad (5)$$

where δ is a small constant ($\delta = 10e-6$ was used in our experiments) to avoid numerical instability, ε is a user-specified edge-awareness parameter, and \mathcal{L} is the color range of \mathbf{I} .

In contrast to the conventional guided filter [HST13] and its rolling guidance extension (RG-GF) [ZSXJ14b], which both consist of a constant regularization parameter ($\varepsilon^t = \text{const.}$ for any \mathbf{x} and t), our filter (Eq. 1) adaptively and dynamically changes the regularization magnitude for each \mathbf{x} and t as defined in Eq. (5). In addition, conventional GF-based filters [HST13, LZZZ*14, ZSXJ14b, KCWL15, LYB17, DYT*19] usually apply an averaging filter to their coefficients (\mathbf{a}_v, b_v) such as $f(\mathbf{a}_v)^T \mathbf{J}^t + f(b_v)$, instead of the procedure we use in Eq. (1), whereas our formulation reduces the total number of $f(\cdot)$ from $(3 + 33t)$ (conventional, e.g., [ZSXJ14b]) to $(3 + 19t)$ times with respect to t (despite the fact that Eq. (5) increases the number of $f(\cdot)$ by t). Algorithms 1 and 2 describe the pseudocodes of our filter. Note that \mathbf{J}^0 is only computed once in Algorithm 1 and then is re-used in Algorithm 2 as $\bar{\mathbf{I}} = f(\mathbf{I})$, which is required to compute the GF coefficients (\mathbf{a}_v, b_v) during t iterations.

Algorithm 1: Our Proposed Filter

Input : Pixels $\{\mathbf{x}\}$ and their corresponding input image colors $\mathbf{I} = \{I_v(\mathbf{x}), v \in \{r, g, b\}\}$, regularization ε and scale σ parameters, image range

$\mathcal{L} = |\max(\mathbf{I}) - \min(\mathbf{I})|$, and iteration number t .

Output: Filtered colors $\{\mathbf{J}^{t+1} = \{J_v^{t+1}(\mathbf{x}), v \in \{r, g, b\}\}$.

- 1 Initialize $f(\cdot)$; $\varepsilon^0 \leftarrow \varepsilon \times \mathcal{L}^2$; $\delta \leftarrow 10e-6$;
- 2 $\mathbf{J}^0 \leftarrow \bar{\mathbf{I}} \leftarrow f(\mathbf{I})$ for all \mathbf{x} and $v \in \{r, g, b\}$;
- 3 **for** $i \leftarrow 0$ **to** t **do**
- 4 $\mathbf{J}^{i+1} \leftarrow GF\text{-DS}(\mathbf{J}^i, \mathbf{I}, \bar{\mathbf{I}}, \{\varepsilon^i\})$;
 // Eq. (1) via Algorithm 2.
- 5 **if** $i < t$ **then**
- 6 $\varepsilon^{i+1}(\mathbf{x}) \leftarrow \varepsilon^0 \frac{\delta + |\mathbf{J}^{i+1}(\mathbf{x}) - \mathbf{J}^0(\mathbf{x})|}{\delta + f(|\mathbf{J}^{i+1}(\mathbf{x}) - \mathbf{J}^0(\mathbf{x})|)}$ for all \mathbf{x} ;
 // Eq. (5).
- 7 **end if**
- 8 **end for**
- 9 **return** $\{\mathbf{J}^{t+1}\}$;

Algorithm 2: GF-DS: Guided Filter via Domain-Splitting

Input : Pixels $\{\mathbf{x}\}$ and their corresponding guidance $\mathbf{h} = \{h_v(\mathbf{x})\}$, input $\mathbf{I} = \{I_v(\mathbf{x})\}$, and smoothed input $\bar{\mathbf{I}} = \{\bar{I}_v(\mathbf{x})\}$ colors ($v \in \{r, g, b\}$), and regularization $\{\gamma(\mathbf{x})\}$ and scale σ parameters, respectively.

Output: Filtered colors $\{\mathbf{q} = \{q_v(\mathbf{x}), v \in \{r, g, b\}\}$.

1 **Function** GF-DS($\mathbf{h}, \mathbf{I}, \bar{\mathbf{I}}, \{\gamma(\mathbf{x})\}$):

2 **forall** $v \in \{r, g, b\}$ **do**

3 $\{\bar{h}_v\} \leftarrow f(h_v)$; $\{\hat{h}_v\} \leftarrow f(h_v \times h_v)$;
 $\{corr_{h,I_v}\} \leftarrow \{f(h_r \times I_v), f(h_g \times I_v), f(h_b \times I_v)\}$;

4 **end forall**

5 $\{\bar{h}_{rg}\} \leftarrow f(h_r \times h_g)$; $\{\bar{h}_{rb}\} \leftarrow f(h_r \times h_b)$;
 $\{\bar{h}_{gb}\} \leftarrow f(h_g \times h_b)$;

6 **forall** \mathbf{x} **do**

7 $A_{11} \leftarrow \hat{h}_r - \bar{h}_r^2$; $A_{12} \leftarrow \bar{h}_{rg} - \bar{h}_r \times \bar{h}_g$;
 $A_{13} \leftarrow \bar{h}_{rb} - \bar{h}_r \times \bar{h}_b$;

8 $A_{22} \leftarrow \hat{h}_g - \bar{h}_g^2$; $A_{23} \leftarrow \bar{h}_{gb} - \bar{h}_g \times \bar{h}_b$;

9 $A_{33} \leftarrow \hat{h}_b - \bar{h}_b^2$;

10 $\mathbf{C} \leftarrow \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{12} & A_{22} & A_{23} \\ A_{13} & A_{23} & A_{33} \end{pmatrix}$;

 // The 3x3 covariance matrix \mathbf{C} of \mathbf{J}^t .

11 **forall** $v \in \{r, g, b\}$ **do**

12 $cov_{h_v} \leftarrow corr_{h,I_v} - \bar{I}_v \times \bar{h}_v$;

13 $\mathbf{a}_v \leftarrow (\mathbf{C} + \gamma(\mathbf{x})\mathbf{U})^{-1} cov_{h_v}$; // Eq. (3)

14 $b_v \leftarrow \bar{I}_v - \mathbf{a}_v \times \bar{h}_v$; // Eq. (4)

15 $q_v \leftarrow \mathbf{a}_v \times \mathbf{h} + b_v$; // Eq. (1)

16 **end forall**

17 **end forall**

18 **return** $\{\mathbf{q}\}$;

2.1. Analysis of Regularization with ϵ^t

For simplicity, consider a grayscale case of an input and t filtered intensities $I(\mathbf{x})$ and $J^t(\mathbf{x})$ at \mathbf{x} , respectively. Substituting Eq. (4) into Eq. (1) then leads to the following numerator for Eq. (5):

$$|J^{t+1}(\mathbf{x}) - J^0(\mathbf{x})| = |a(\mathbf{x})(J^t(\mathbf{x}) - f(J^t(\mathbf{x})))|, \quad (6)$$

where $a(\mathbf{x})$ is a scalar GF coefficient corresponding to $\{\mathbf{a}_v\}$. If $t = 0$, then Eq. (6) is equivalent to $|a(\mathbf{x})\text{DoG}(I(\mathbf{x}))|$ where DoG is the difference of Gaussian functions of two different scales (σ and 2σ in our case), and DoG represents a band-pass filter.

Figure 3 illustrates ϵ^{t+1} for varying t and demonstrates the DoG band-pass effect, which is close to zero (white) on salient edges and smoothly attenuates from a high value (black) near the edges. Furthermore, the salient edge region (white lines) becomes thinner as t increases, which contributes to sharpening the salient edges and accelerating convergence. Here, the numerator (the top images of Fig. 3) of Eq. (5) on flat-regions that are far from salient edges is also close to zero because of DoG characteristics. Thus, a preferable distribution of ϵ^{t+1} is obtained, as shown in the bottom images of Fig. 3, via normalizing the difference $|J^{t+1}(\mathbf{x}) - J^0(\mathbf{x})|$ by its average (i.e., $f(|J^{t+1}(\mathbf{x}) - J^0(\mathbf{x})|)$).

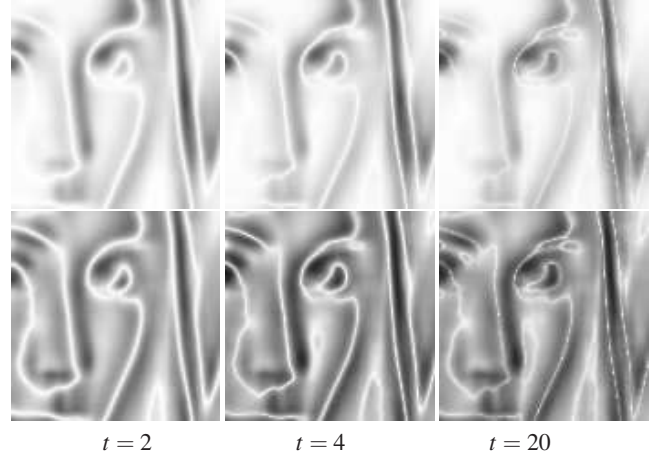


Figure 3: Visualization of $|J^{t+1}(\mathbf{x}) - J^0(\mathbf{x})|$ (top) and Eq. (5) (bottom) for varying t , where $\sigma = 4$ and $\epsilon = 0.01$.

3. Numerical Experiments

All experiments in this study were performed on a Ryzen 7 3700X CPU PC (3.6 GHz, 8 core, no parallelization) with 32 GB RAM and a 64-bit OS with a GNU g++9.3 compiler. The input images of our experiments are shown in Fig. 4.

We numerically compared the proposed filter with the conventional RG-based methods (RG-BF/GF/WGF/DT) and evaluated their computational speed, convergence, accuracy, and visual quality. The RG-BF is based on the permutohedral lattice [ABD10] implementation for fast approximation of its joint BF (C++ code was provided courtesy of [ZSXJ14a]). The moving average [SKM98] is employed for the fast box-kernel averaging for RG-GF/WGF, where RG-WGF consists of WGF [LZZ*14] as its joint filter. The RG-DT-DS [YY21] utilizes the DT [GO11] and DS [YY14] techniques for its joint filter.

Because the above filters use different averaging kernels, such as box (RG-GF/WGF) and Gaussian functions of L^1 (ours and RG-DT-DS) and L^2 (RG-BF) norms, the actual scale parameters used in the filters are transformed from a given σ by the following function in our experiments, except for RG-BF (σ is directly used in RG-BF):

$$\text{RG-GF/WGF/DT and our : } \sqrt{2\pi}\sigma/2,$$

which is derived by matching the integral over \mathbb{R} of the normalized L^2 Gaussian function with other kernels as follows:

$$\int_{-\infty}^{\infty} G_{\varphi}(x) dx = \int_{-\varphi}^{\varphi} \text{box}_{\varphi}(x) dx = 2\varphi, \quad \varphi \in \mathbb{R}_{>0},$$

$$\int_{-\infty}^{\infty} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx = \sqrt{2\pi}\sigma, \quad x \in \mathbb{R},$$

where $\text{box}_{\sigma}(x) = 1$ if $x \leq \sigma$ otherwise $\text{box}_{\sigma}(x) = 0$ and $G_{\sigma}(\cdot)$ is the L^1 Gaussian function defined in Eq. (2). Also, our initial regularization parameter ϵ in Eq. (5) is manually chosen such that its filtering result matches RG-BF visually, where the recommended value is 10 to 100 times ϵ in RG-GF/WGF. The timings in Sec. 3.1 and accuracy in Sec. 3.2 were measured by using the 24 combinations of $\sigma \in \{4, 8, 16, 32\}$ and $\epsilon \in \{0.5, 0.01, 0.05, 0.001, 0.005, 0.0001\}$.






					
	Mandrill	Lena	Castle	Wave	Snack
w	512	512	1536	256	1368
h	512	512	2048	256	912

Figure 4: Input images used in this study, where w (width) and h (height) stand for the numbers of row and column pixels, respectively.

3.1. Timings

Figure 5 shows the averaged timings of our filter and conventional filters with respect to varying image size (generated by magnifying Mandrill in Fig. 4) for $t \in \{4, 20\}$. In the case of $t = 20$, RG-BF was unable to process a pixel number greater than 1024^2 because of memory error. Here, four iterations ($t = 4$) have been recommended for fast results [ZSXJ14b], and 20 iterations ($t = 20$) are enough for high-quality results. The filtering speed performance with respect to t is summarized in Table 1. Although our filter is a little bit slower than the conventional methods, its linear computational complexity with respect to the number of pixels is numerically confirmed (Fig. 5) and provides practical computational speed.

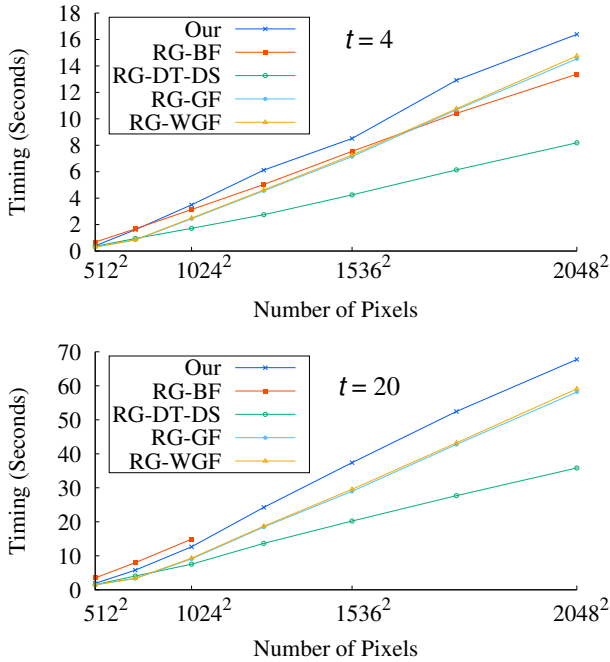


Figure 5: Timings (seconds) of our filter and conventional filters with respect to image size (number of pixels).

Table 1: Speed (megapixels per second) via the iteration number t for our filter and conventional filters.

RG-BF	RG-GF	RG-WGF	RG-DT-DS	Our filter
$1.27/t$	$1.30/t$	$1.30/t$	$2.19/t$	$1.08/t$

3.2. Approximation Accuracy

Table 2 shows the peak signal-to-noise ratio (PSNR) and maximum error (E_{max}) [YY14] of our filter and the RG-GF filter for 100 randomly generated images, where each was examined by using the above-mentioned 24 parameter sets with $t = 20$. The size of the random input image is equal to 64×64 (i.e., $w, h = 64$), and the color values are in the $[0, 1]$ range. We also compared our filter with RG-GF-Deriche which uses the popular recursive approximation [Der93] for its L^2 Gaussian convolution instead of box-kernel averaging. Finite Impulse Response (FIR) [Get13] with $10e-15$ accuracy was employed as the correct filtering results for both the L^1 (ours) and the L^2 (RG-GF and RG-GF-Deriche) Gaussian convolutions. Our filter achieved very accurate filtering results thanks to DS approximating $f(\cdot)$ in Eq. (2). Although there is a trade-off between computational speed and accuracy, as shown in Sec. 3.1, our accurate filtering avoids some undesired artifacts (Fig. 6(b)), whereas the box averaging in RG-GF generates phantom edges and rectangular shapes that do not exist in the input image.

Table 2: Accuracy comparison for our filter and RG-GF filters via PSNR and E_{max} .

	RG-GF	RG-GF-Deriche [Der93]	Our filter
PSNR	45.0	55.2	291.3
E_{max}	$5.58e-02$	$1.19e-01$	$6.04e-13$

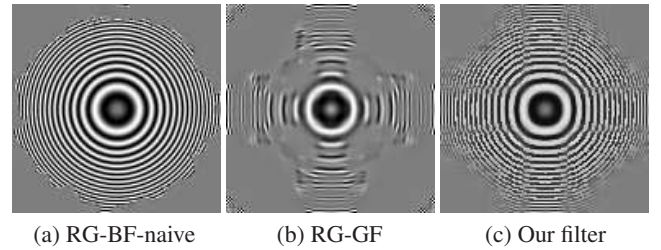


Figure 6: Comparison of RG-BF-naive (averaging via the exact L^2 Gaussian kernel) (a), RG-GF (b), and our filtering (c) results, where $\sigma = 4$, $\sigma_r = 4.7$ (a), $\epsilon = 0.0001$ (b), $\epsilon = 0.01$ (c), and $t = 20$.

3.3. Convergence

Figure 7 gives a typical example of the normalized mean absolute error (MAE) [ZSXJ14b] and maximum error (ME), (i.e., the maximum value of color differences for all \mathbf{x}) for varying t , where MAE and the difference are measured between \mathbf{J}^{t+1} and \mathbf{J}^t (the corresponding filtering results are shown in Fig. 11). At first glance, all filters converge nicely according to MAE. However, the MEs of the conventional filters in Fig. 7 oscillate, especially in RG-DT-DS (probably because of the DT zigzag process). Also, the ME of RG-BF is excluded in Fig. 7 because of its high value. Conversely, our filtering result nicely and quickly converges in both MAE and ME because our regularization function (Eq. 5) automatically and dynamically adapts the position and amount of the recursive filtering process. Figure 8 illustrates an example of convergence processes of our filter and the RG-DT-DS filter, where RG-DT-DS shows non-uniform convergence behavior in elongated texture regions, whereas our filter converges uniformly in terms of its texture structure.

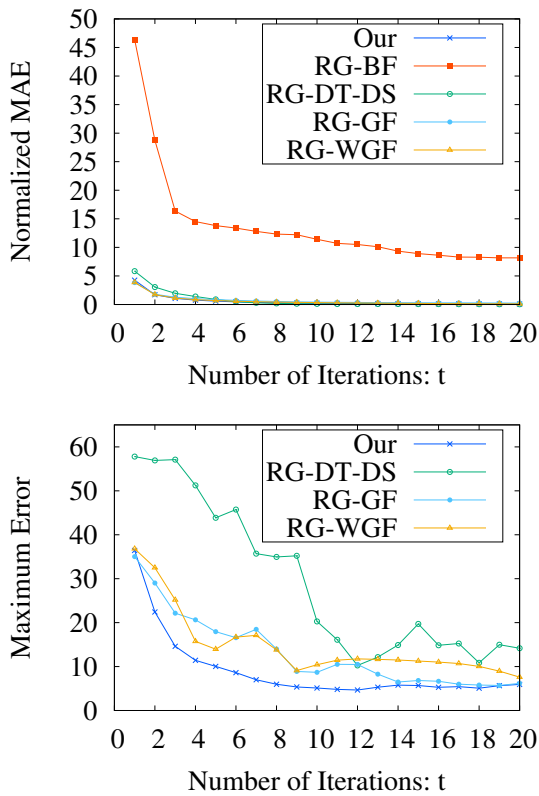


Figure 7: Convergence comparisons with respect to the iteration number t via the normalized MAE [ZSXJ14b] (top) and ME (bottom); RG-BF is not shown because of its high error value). The corresponding filtering results are shown in Fig. 11.

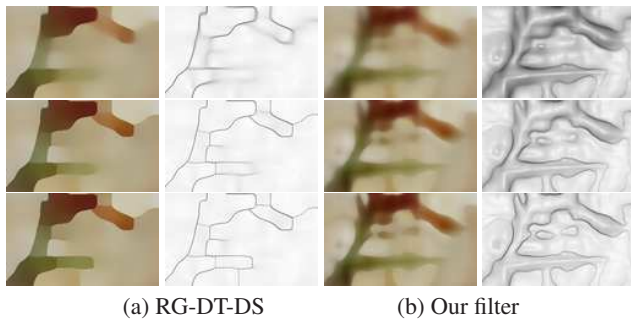


Figure 8: Convergence processes of the RG-DT-DS [YY21] filter (a) and our filter (b), where the images on the right correspond to the gradient magnitude, $t \in \{4, 10, 20\}$ (top to bottom), $\sigma = 12$, $\phi = 1.5$ is the edge-awareness of RG-DT-DS [YY21] (a), and $\varepsilon = 0.01$ (b).

3.4. Visual Quality

Figures 2, 6, and 8-11 illustrate visual comparisons of our filter and conventional filters. In Figs. 2 and 11, RG-GF does not attain the target σ (small texture details remain), whereas the smoothness and sharpness of our results are much closer to RG-BF and RG-DT-DS. Similar to the RG-GF case described in Fig. 6 and Sec. 3.2, RG-

BF and RG-WGF also produce some undesired artifacts around the salient edges, as demonstrated in Figs. 9 and 10, respectively. For example, some phantom edges are generated at the castle roofs in Fig. 11(e), the boundary between the hair and background in Fig. 9(a), and the highlight of the hat in Fig. 10(a), whereas our filter does not produce such artifacts. An example of non-uniform convergence via RG-DT-DS is visualized in Fig. 8 and explained in Sec. 3.3.

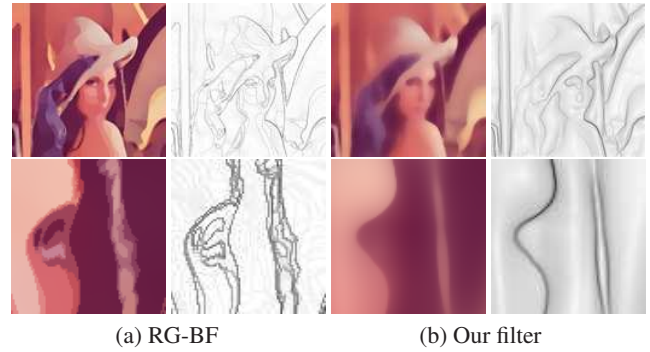


Figure 9: Quality comparison of the RG-BF filter [ZSXJ14b, ABD10] (a) and our filtering (b) results, where the right images correspond to the gradient magnitude, $\sigma = 16$, $\sigma_r = 10$ (a), and $\varepsilon = 0.01$ (b).

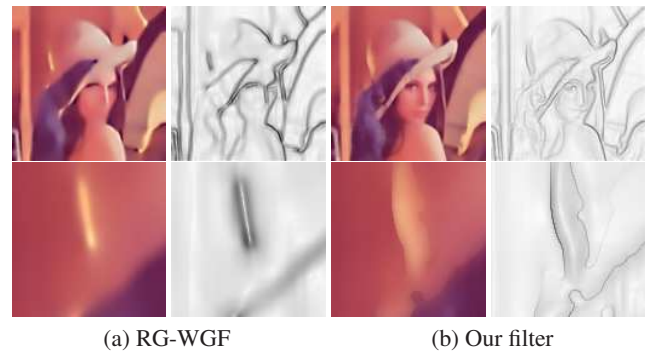


Figure 10: Quality comparison of the RG-WGF filter [ZSXJ14b, LZZ*14] (a) and our filtering (b) results, where the right images correspond to the gradient magnitude, $\sigma = 8$, $\varepsilon = 0.001$ (a), and $\varepsilon = 0.01$ (b).

4. Conclusion

We have proposed a novel scale-aware image filter with stable quality for use at varying scales. Our filter is based on an adaptive and dynamic regularization for a recursive regression model of the RG framework. We compared our filter numerically in terms of speed, accuracy, convergence, and visual quality with popular conventional methods. Our filter achieved more edge saliency, fewer artifacts, and appropriate scale convergence compared with conventional filters.

Future work includes improving the regularization function to obtain the same degree of edge sharpness as that achieved in RG-BF and RG-DT-DS and applying our filter to graphics applications such as feature extraction, detail enhancement, stylization, and image matting.

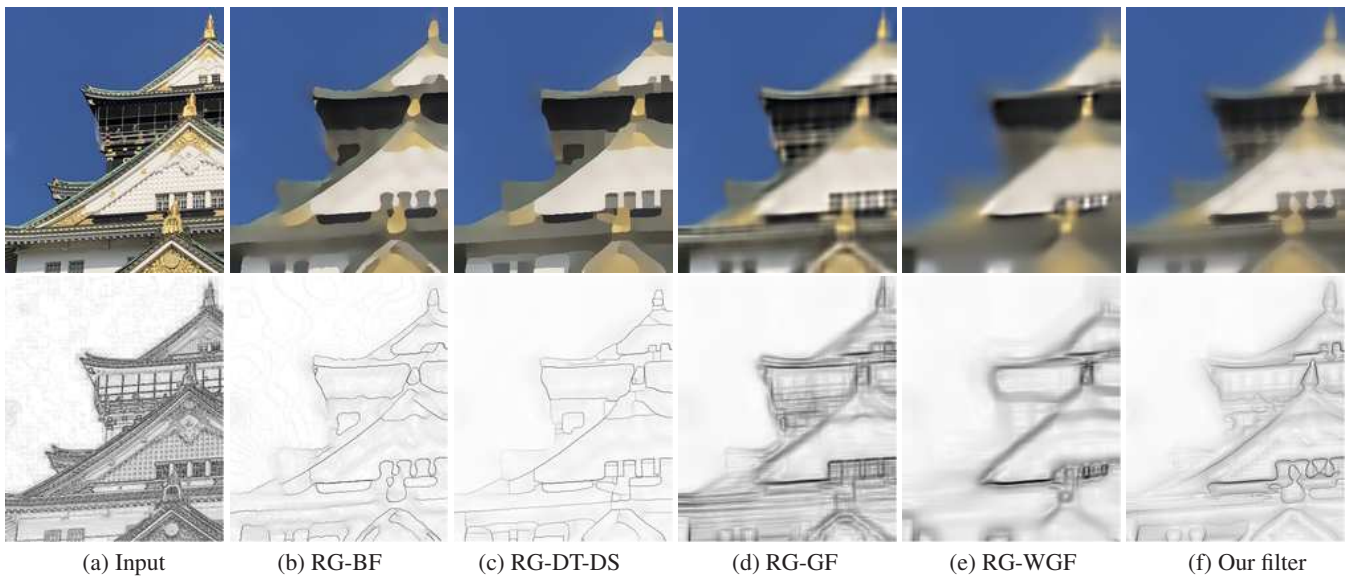


Figure 11: Input (a), RG-BF [ZSXJ14b, ABD10] (b), RG-DT-DS [YY21] (c), RG-GF/WGF [ZSXJ14b, LZZ* 14, SKM98] (d,e), and our filtering (f) results corresponding to Fig. 7, where $\sigma = 8$, $\sigma_r = 30$ (b), $\varepsilon = 0.002$ (d,e), $\varepsilon = 0.01$ (f), $\phi = 1.0$ (c), and $t = 20$.

References

- [ABD10] ADAMS A., BAEK J., DAVIS M. A.: Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum* 29, 2 (2010), 753–762. doi:10.1111/j.1467-8659.2009.01645.x. 3, 5, 6
- [Der93] DERICHE R.: *Recursively implementing the Gaussian and its derivatives*. Tech. Rep. RR-1893, INRIA, 1993. 4
- [DYT*19] DAI L., YUAN M., TANG L., XIE Y., ZHANG X., TANG J.: Interpreting and extending the guided filter via cyclic coordinate descent. *IEEE Transactions on Image Processing* 28, 2 (2019), 767–778. doi:10.1109/TIP.2018.2869720. 1, 2
- [ED04] EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics* 23, 3 (2004), 673–678. doi:10.1145/1015706.1015778. 1
- [GDANW21] GALETTO F. J., DENG G., AL-NASRAWI M., WAHEED W.: Edge-aware filter based on adaptive patch variance weighted average. *IEEE Access* 9 (2021), 118291–118306. doi:10.1109/ACCESS.2021.3106907. 1
- [Get13] GETREUER P.: A survey of Gaussian convolution algorithms. *Image Process. Line* 3 (2013), 276–300. doi:10.5201/ipol.2013.87. 4
- [GO11] GASTAL E. S., OLIVEIRA M. M.: Domain transform for edge-aware image and video processing. *ACM Transactions on Graphics* 30, 4 (2011), 69:1–69:12. doi:10.1145/2010324.1964964. 1, 3
- [HK70] HOERL A. E., KENNARD R. W.: Ridge regression: Applications to nonorthogonal problems. *Technometrics* 12, 1 (1970), 69–82. doi:10.1080/00401706.1970.10488635. 2
- [HST13] HE K., SUN J., TANG X.: Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 6 (2013), 1397–1409. doi:10.1109/TPAMI.2012.213. 1, 2
- [KCWL15] KOU F., CHEN W., WEN C., LI Z.: Gradient domain guided image filtering. *IEEE Transactions on Image Processing* 24, 11 (2015), 4528–4539. doi:10.1109/TIP.2015.2468183. 1, 2
- [Lin97] LINDBERG T.: On the axiomatic foundations of linear scale-space. In *Gaussian Scale-Space Theory*. Springer, 1997, pp. 75–97. doi:10.1007/978-94-015-8802-7_6. 2
- [LYB17] LU K., YOU S., BARNES N.: Double-guided filtering: Image smoothing with structure and texture guidance. In *International Conference on Digital Image Computing* (2017), IEEE, pp. 1–8. doi:10.1109/DICTA.2017.8227425. 1, 2
- [LZZ*14] LI Z., ZHENG J., ZHU Z., YAO W., WU S.: Weighted guided image filtering. *IEEE Transactions on Image Processing* 24, 1 (2014), 120–129. doi:10.1109/TIP.2014.2371234. 1, 2, 3, 5, 6
- [PSA*04] PETSCHNIG G., SZELISKI R., AGRAWALA M., COHEN M., HOPPE H., TOYAMA K.: Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics* 23, 3 (2004), 664–672. doi:10.1145/1015706.1015777. 1
- [SCG*21] SHI Z., CHEN Y., GAVVES E., METTES P., SNOEK C. G.: Unsharp mask guided filtering. *IEEE Transactions on Image Processing* 30 (2021), 7472–7485. doi:10.1109/TIP.2021.3106812. 1
- [SKM98] SOCHEN N., KIMMEL R., MALLADI R.: A general framework for low level vision. *IEEE Transactions on Image Processing* 7, 3 (1998), 310–318. doi:10.1109/83.661181. 1, 3, 6
- [YY14] YOSHIZAWA S., YOKOTA H.: Fast L^1 Gaussian convolution via domain splitting. In *IEEE International Conference on Image Processing* (2014), IEEE-SP, pp. 2908–2912. doi:10.1109/ICIP.2014.7025588. 1, 2, 3, 4
- [YY21] YOSHIZAWA S., YOKOTA H.: Fast and faithful scale-aware image filters. *The Visual Computer* 37, 12 (2021), 3051–3062. doi:10.1007/s00371-021-02249-5. 1, 2, 3, 5, 6
- [ZSXJ14a] ZHANG Q., SHEN X., XU L., JIA J.: RG-BF C++ code, 2014. <https://www.cse.cuhk.edu.hk/leojia/projects/rollguidance>. 2, 3
- [ZSXJ14b] ZHANG Q., SHEN X., XU L., JIA J.: Rolling guidance filter. In *European Conference on Computer Vision* (2014), D. Fleet T. Pajdla B. S., Tuytelaars T., (Eds.), LNCS, Springer, pp. 815–830. doi:10.1007/978-3-319-10578-9_53. 1, 2, 4, 5, 6