Interpreting and Extending the Guided Filter via Cyclic Coordinate Descent

Longquan Dai, *Member, IEEE,* Mengke Yuan, Liang Tang, Yuan Xie, *Member, IEEE,*
Xiaopeng Zhang, *Member, IEEE,* and Jinhui Tang, *Senior Member, IEEE*

Abstract—**The Guided Filter (GF) is a widely used smoothing tool in computer vision and image processing. However, to the best of our knowledge, few papers investigate the mathematical connection between this filter and the least squares optimization. In this paper, we first interpret the guided filter as the cyclic coordinate descent solver of a least squares objective function. This discovery implies an extension approach to generalize the guided filter since we can change the least squares objective function and define new filters as the first pass iteration of the cyclic coordinate descent solver of modified objective functions. In addition, referring to the iterative minimizing procedure of the cyclic coordinate descent, we can derive new rolling filtering schemes. So we are reasonable to say that our discovery not only reveals an approach to design new GF-like filters adapting to specific requirements of applications but also offers thorough explanations for two rolling filtering schemes of the guided filter as well as the method to extend them. Experiments prove our new proposed filters and rolling filtering schemes could produce state-of-the-art results.**

Index Terms—**The Guided Filter (GF), Cyclic Coordinate Descent (CCD), Rolling Filtering Schemes.**

I. Introduction

The most fundamental smoothing tools in image processing should be edge-aware filters. We can roughly divide them into two categories: the explicit filter and the implicit filter. The first kind of filters takes a mapping operator to transform inputs to outputs. The well-known Gaussian, bilateral and guided filters [1], [2], [3] are part of this type because we can formulate them as the convolution operators explicitly. In contrast, the mapping operator of the implicit filter is not given. Instead, the filtering output is considered as the minimizer of an objective function. Xu *et al.* [4] provide an instance of this kind of filters.

Each kind of filters has its merits and demerits. The explicit filter is easy to implement and often has low computational cost. But, it is not an easy task to design a new explicit filter and analyze its rolling filtering behavior as people usually define the mapping operator according to their intuition and this encumbers the mathematical analysis of the explicit filter. In order to achieve state-of-the-art results, the implicit filter spend considerable computational time. This is because the final results are yielded by the iterative solvers such as gradient descent [5]. Disregarding the above shortcoming, the implicit filter brings more convenience: designing a new implicit filter can be reduced to proposing an objective function and finding its solver, which has been well studied and owns a solid theoretical foundation.

It is possible to draw upon the strong points of one kind of filters to overcome the shortcomings of another kind of filters. Specifically, if we establish the connection between the mapping operator of the explicit filter and the iterative solver of the objective function of the implicit filter, the problem that cannot be addressed from the explicit/implicit perspective alone would be solved by uniting two filtering viewpoints. The benefits of this joint perspective are twofold:

- the filtering behavior of the explicit filter as well as its rolling filtering usage is described by the iterative solver. We thus not only can define new filters by modifying objective functions but also are able to disclose their rolling filtering usages from the minimizing procedure of their iterative solver.
- the explicit filter deepens our understanding of the implicit filter as it entitles each minimizing pass a filtering connotation other than its original optimization interpretation. This connection facilitates the intuitive understanding of each iteration as well as its functions in optimization.

Establishing the connection between the explicit filter and the implicit filter is not new. Let $\boldsymbol{q}$, $\boldsymbol{p}$, L and Λ be an $N \times 1$ output vector, constraint, $N \times N$ Laplacian matrix and diagonal matrix, respectively. He *et al.* [3] proved the output of the guided filter approximates to one Jacobi iteration in optimization (1). But He's discovery leaves much to be

$$\min_{\boldsymbol{q}} (\boldsymbol{q}-\boldsymbol{p})^T \Lambda (\boldsymbol{q}-\boldsymbol{p}) + \boldsymbol{q}^T L \boldsymbol{q} \tag{1}$$

desired because the guided filter and the iterative solver of optimization (1) are not strictly equal and therefore the

Longquan Dai is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094 China e-mail: dailongquan@njust.edu.cn (see https://dailongquan.github.io/).

Mengke Yuan is with University of Chinese Academy of Sciences and the National Laboratory of Pattern Recognition of Institute of Automation Chinese Academy of Sciences, Beijing, 100190, China. e-mail: mengke.yuan@nlpr.ia.ac.cn

Liang Tang is with CASA Environmental Technology Co., Ltd and CASA EM&EW IOT Research Center, 214024, Wuxi, China. e-mail: tangl@casaet.com

Yuan Xie is with Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China. e-mail: yuan.xie@ia.ac.cn

Xiaopeng Zhang is with the National Laboratory of Pattern Recognition of Institute of Automation Chinese Academy of Sciences, Beijing, 100190, China. e-mail: zhangxiaopeng@nlpr.ia.ac.cn

Jinhui Tang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094 China e-mail: jinhuitang@njust.edu.cn

Manuscript received April 19, 2005; revised August 26, 2015.

Jacobi algorithm cannot describe the behavior of guided image filtering with multiple times.

Considering the potential benefits of the joint perspective for the guided filter [3], we will disclose the equivalence between the guided filter and the cyclic coordinate descent solver of the least squares optimization. Further, the connection is exploited to extend the guided filter as well as its rolling filtering scheme. Our main contributions are threefold:

- We unveil that the guided filter equals to the cyclic coordinate descent solver of a least squares objective function and point out the rolling filtering usage of the guided filter can be viewed as the minimizing procedure for the objective function.
- We find a general framework to define new GF-like filters and develop novel instances of GF-like filters in this framework.
- We offer a mathematical foundation for two rolling filtering schemes of the guided filter and the approach to extend them.

II. Related Work

In the literature, a lot of efforts are devoted to disclosing the connection between the explicit filter and the implicit filter. Li *et al.* [6] reveal the median filter corresponds to the analytic form of the minimizer of the sum of the weighted absolute error, thus it has fully explainable connections to global energy minimization. However, this discovery between the mapping operator and the closed-form solution is not as useful as the relationship that connects the mapping operator of an explicit filter to the iterative solver of the implicit filter since the latter makes the two different filters be complementary.

Different from the way of Li, Elad [7] shows the bilateral filter (BF) [2] emerges from a Bayesian optimization, as a single iteration of the well-known Jacobi algorithm. But he did not prove the bilateral filter is the solver of the Bayesian optimization. Dong *et al.* [8] make progress by showing the bilateral filter equals to the iterative reweighting solver, which is a good approximation to the Newton's method [5]. Later, Caraffa *et al.* [9] introduce guidance information to the robust optimization and define the guided bilateral filter as the solver of the robust optimization. Imitating the iteratively minimizing procedure, they invent a rolling filtering scheme for the guided bilateral filter.

Since 2010, the guided filter has attracted much attention. In order to address its defect caused by the box window, Lu *et al.* [10] design a cross window for the cross-based local multipoint filter (CLMF). However, their geometric-adaptive window is still problematic. To solve the problem, Tan *et al.* [11] design a symmetrical window for their multipoint filter with local polynomial approximation and range guidance (MLPA). In addition, Tan *et al.* introduce a spatial regularization term to the guided filter and make their MLPA spatial-aware. But the ability is at the cost of increasing runtime drastically. This shortcoming is conquered by Dai *et al.* [12] who assemble the fully connected guided filter (FCGF) by introducing tree distance to the guided filter. Similarly, incorporating edge-aware weights into the guided filter, Li *et al.* [13] propose the weighted guided filter (WGF) to address the halo artifacts of the guided filter. Unlike Li, Qiu *et al.* [14] put forward local linear SURE-based edge-preserving filter (LLSURE) that exploits Stein's unbiased risk estimate as a predictor for the mean squared error adopted by the guided filter to filter out noise while preserving edges and fine-scale details. Different from previous work, Ham *et al.* [15] formulate the guided image filtering as a nonconvex optimization, which is solved by the majorize-minimization algorithm.

TABLE I: Lists of abbreviations.

CCD	Cyclic Coordinate Descent
BF	Bilateral Filter
CLMF	Cross-based Local Multipoint Filter
MLPA	Multipoint filter with Local Polynomial Approximation and range guidance
FCGF	Fully Connected Guided Filter
WGF	Weighted Guided Filter
LLSURE	Local Linear SURE-based Edge-preserving filter
GF	Guided Filter
IGF	Inverse Guided Filter
SGF	Spatial-aware Guided Filter
TVGF	Total Variation Guided Filter
CGF	Conservative Guided Filter
ICGF	Inverse Conservative Guided Filter
RGF	Rolling Guidance Filtering
RMSF	Rolling Mutual Structure Filtering
GF-RMSF	GF based RMSF
CGF-RMSF	CGF based RMSF
RFNF	Rolling Flash/No-Flash Filtering
SIMD	Single Instruction Multiple Data

Although the guided filter is designed as one pass, non-iterative filter, its rolling filtering scheme arouses extensive concerns recently. Seo *et al.* [16] propose an iterative guided filtering method, which is also adopted by Yelameli *et al.* [17], for robust flash denoising/deblurring. Unlike previous work, Zhang *et al.* [18] propose a novel Rolling Guidance Filtering (RGF) scheme with the complete control of detail smoothing under a scale measure. However, none of above work establishes the connection between the guided filter and some iterative solvers and thus is not able to benefit from uniting two filtering schemes.

The remainder of this paper is organized as follows: in section III, we establish the connection between the guided filter and the cyclic coordinate descent solver of the least squares optimization. Employing this connection, we propose several new GF-like filters and rolling filtering schemes in section IV. In the last section, we conduct extensive experiments to disclose the smoothing ability of newly proposed filters and rolling filtering schemes. Final results prove that our filters and rolling filtering schemes are superior to other methods. At last we note that our paper involves lots of abbreviations.

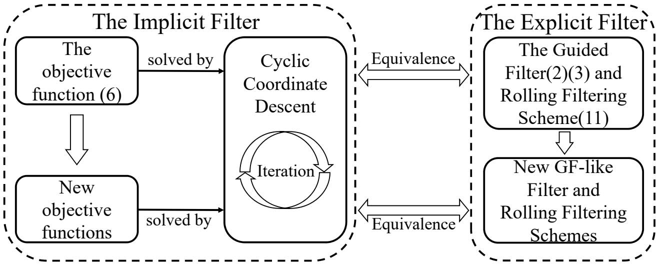

The Implicit Filter
The objective function (6)
solved by
Cyclic Coordinate Descent
Iteration
New objective functions
solved by
Equivalence
Equivalence
The Explicit Filter
The Guided Filter(2)(3) and Rolling Filtering Scheme(11)
New GF-like Filter and Rolling Filtering Schemes

Fig. 1: The cyclic coordinate descent interpretation for the guided filter and the method to extend it. The mapping operator (3) (4) of the guided filter equals to the cyclic coordinate descent solver (7) (9) of objective function (6). We can modify the objective function and derive new GF-like filters and rolling filtering schemes from its cyclic coordinate descent solver. More specifically, new GF-like filters can be derived from the first pass iteration of the cyclic coordinate descent solver of the modified objective functions and the minimizing procedure determined by the cyclic coordinate descent solver indicates a new rolling scheme.

For easy reference, we list all of them in Table I.

III. The Equivalence

The guided filter has a close connection with the cyclic coordinate descent solver of the least squares optimization. As outlined in Fig 1, we will devote this section to exhibit the equivalence between the mapping operator of the guided filter and the cyclic coordinate descent solver of the least squares optimization.

A. The Definition of the Guided Filter

Initially, the guided filter is defined in the two steps local multipoint filtering framework [19]:

1) *multipoint estimation:* calculating multipoint estimates q_i' for each pixel i in the image domain Ω according to the linear transform $q_i' = a_k I_i + b_k, \forall k \in \omega_i$ in a window ω_i centered at i, where I is the guidance image, (a_k, b_k) is the minimizer (3) of (2), p is the filtering input, ε is a constant, $E_{\omega_i}(x)$ and $D_{\omega_i}(x)$ denote the average and variance of x in the window ω_i.

$$\min_{a_k,b_k} \sum_{i\in\omega_k} ((a_k I_i + b_k - p_i)^2 + \varepsilon a_k^2) \tag{2}$$

$$\begin{aligned} a_k &= \frac{E_{\omega_k}(Ip) - E_{\omega_k}(I)E_{\omega_k}(p)}{D_{\omega_k}(I)+\varepsilon} \\ b_k &= E_{\omega_k}(p) - a_k E_{\omega_k}(I) \end{aligned} \tag{3}$$

2) *aggregation:* as each pixel i has a number of estimates indexed by $k \in \omega_i$, the filtering result q is defined as the average of these multipoint estimates.

$$q_i = E_{\omega_i}(q') = E_{\omega_i}(a)I_i + E_{\omega_i}(b) \tag{4}$$

Algorithm 1 The Guided Image Filtering Algorithm

```
procedure GF
    Inputs:
    input image $p$, guidance image $I$,
    regularization $\epsilon$
    $a_k = \frac{E_{\omega_k}(Ip) - E_{\omega_k}(I)E_{\omega_k}(p)}{D_{\omega_k}(I)+\epsilon}$
    $b_k = E_{\omega_k}(p) - a_k E_{\omega_k}(I)$
    $q_i = E_{\omega_i}(q') = E_{\omega_i}(a)I_i + E_{\omega_i}(b)$
end procedure
```

For clarity, we list above guided image filtering algorithm of the guided filter in Algorithm 1. Please refer to original paper [3] for detail information.

B. A Cyclic Coordinate Descent Interpretation

Mathematically, the cyclic coordinate descent is based on the idea that the minimizer of a multivariable function F can be obtained by minimizing it along one direction at a time. That is, in each iteration, for each index i of the problem, CCD

$$\boldsymbol{x}_i^{n+1} = \arg\min_{\boldsymbol{y}} F(\boldsymbol{x}_1^{n+1}, \ldots, \boldsymbol{x}_{i-1}^{n+1}, \boldsymbol{y}, \boldsymbol{x}_{i+1}^{n}, \ldots, \boldsymbol{x}_m^{n}) \tag{5}$$

algorithm cyclically solves it according to (5). where $\boldsymbol{x}_i^n$ and $\boldsymbol{y}$ are vectors. Thus, one may begin with an initial guess $\boldsymbol{x}^0$ and gets a sequence $\{\boldsymbol{x}^0, \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots\}$ that has $F(\boldsymbol{x}^0) \geq F(\boldsymbol{x}^1) \geq F(\boldsymbol{x}^2) \geq \ldots$.

Applying the cyclic coordinate descent algorithm to optimize the objective function (6), we can verify that (3) (4)

$$\min_{q,a,b} \sum_{k\in\Omega} \sum_{i\in\omega_k} ((a_k I_i + b_k - q_i)^2 + \varepsilon a_k^2) \tag{6}$$

are the closed-form solutions of (6) by cyclically minimizing q and (a, b). In the first step, let $q^0 = p$ and $\mathcal{P}_0(q^n, I, \varepsilon) = \sum_{i\in\omega_k} ((a_k I_i + b_k - q_i^n)^2 + \varepsilon a_k^2)$, the cyclic coordinate descent minimizes optimization (7). We can formulate the closed-

$$a_k^{n+1}, b_k^{n+1} = \arg\min_{a_k,b_k} \mathcal{P}_0(q^n, I, \varepsilon) \tag{7}$$

form solutions of a_k^{n+1}, b_k^{n+1} of (7) as (8). In the second step

$$\begin{aligned} a_k^{n+1} &= \frac{E_{\omega_k}(Iq^n) - E_{\omega_k}(I)E_{\omega_k}(q^n)}{D_{\omega_k}(I)+\varepsilon} \\ b_k^{n+1} &= E_{\omega_k}(q^n) - a_k^{n+1} E_{\omega_k}(I) \end{aligned} \tag{8}$$

with fixed a_k^{n+1}, b_k^{n+1}, the cyclic coordinate descent algorithm computes the minimizer of (9), where $\mathcal{P}_1(a^{n+1}, b^{n+1}, I) =$

$$q_i^{n+1} = \arg\min_{q_i} \mathcal{P}_1(a^{n+1}, b^{n+1}, I) \tag{9}$$

$\sum_{k\in\omega_i} (a_k^{n+1} I_i + b_k^{n+1} - q_i)^2$ and q_i^{n+1} can be formulated as (10).

$$q_i^{n+1} = E_{\omega_i}(a^{n+1})I_i + E_{\omega_i}(b^{n+1}) \tag{10}$$

We note that (8) (10) are same to (3) (4) except for an extra iteration index n. This implies the guided filter equals to the first cyclic coordinate descent iteration of (6) with an initial guess $q^0 = p$. Moreover, let $\text{GF}(q^n, I, \varepsilon)$ denote the filtering outputs of the guided filter with respect to the input q^n, the rolling filtering scheme (11) can be interpreted as the cyclic

$$q^{n+1} = \text{GF}(q^n, I, \varepsilon) \tag{11}$$

coordinate descent minimizing procedure of (6). The reason is that the filtering result of the guided filter with the input q^n obtained from the n^{th} cyclic coordinate descent iteration equals to q^{n+1} *i.e.* $n + 1^{\text{th}}$ cyclic coordinate descent iteration. To further broaden the understanding for above two equivalence, we outline this discovery and main idea in Fig 1. Specifically, employing the equivalence,

- we can modify the objective function and exploit the cyclic coordinate descent to define new GF-like filters due to the equivalence between the guided filter and one cyclic coordinate descent iteration.
- we can derive rolling filtering schemes from the cyclic coordinate descent algorithm because the iteratively minimizing procedure of the cyclic coordinate descent indicates new rolling filtering schemes.

IV. The Method for Extension

The equivalence between the guided filter and the cyclic coordinate descent algorithm exposes possible ways to extend the filter. In this section, we are going to develop new GF-like filters and their rolling filtering schemes according to the technical roadmap illustrated in Fig 1.

A. *New GF-like Filters*

New filters can be derived by modifying the objective function (6) and defining them as the first CCD pass of new objective functions. Following this roadmap, we propose five GF-like filters(*i.e.* SGF, TVGF, CGF, IGF, ICGF) and two rolling filtering schemes (*i.e.* CGF-RMSF and RFNF). Refer to Table I for details of these abbreviations.

1) The Spatial-aware Guided Filter (SGF): Both Tan [11] and Dai [12] complain that the guided filter only considers the color similarity. To address the problem, Tan introduces a spatial regularization term into his multipoint filter and Dai appends a spatial tree weight into the fully connected guided filter. However, both methods will increase the runtime significantly. In addition, the fully connected guided filter is apt to yield artifacts in the rolling guidance filtering [18] as the minimal spanning tree used by the fully connected guided filter often fails to represent the geometric structure of the iteratively refined guidance. In this regard, we derive the spatial-aware guided filter, a GF-like counterpart for the bilateral filter used in the rolling guidance filtering, from the cyclic coordinate descent solver of (12), where σ is a constant controlling the spatial similarity. Our spatial-aware guided filter not only takes the Gaussian spatial similarity $w_{ki} = \exp(-\|k-i\|^2/\sigma)$ into account but also can be integrated with the rolling guidance filtering scheme [18] satisfactorily in terms of computational efficiency and smoothing quality.

$$\min_{q,a,b} \sum_{k\in\Omega} \sum_{i\in\omega_k} (w_{ki}(a_k I_i + b_k - q_i)^2 + \varepsilon a_k^2) \tag{12}$$

The minimizer of (12) can be found by iteratively computing (13) (14) with an initial guess $q^0 = p$, where $E^w_{\omega_i}(x) = \sum_{j\in\omega_i} w_{ij} x_j / \sum_{j\in\omega_i} w_{ij}$ and $D^w_{\omega_i}(x) = E^w_{\omega_i}(x^2) - (E^w_{\omega_i}(x))^2$ denote weighted average/variance in the window ω_i with respect to the gaussian weight w_{ij}. Following the cyclic coordinate descent interpretation for the guided filter, we define the spatial-aware guided filter as (13) (14) with $n = 0$ and $q^0 = p$.

$$\begin{aligned} a_k^{n+1}, b_k^{n+1} &= \text{SGF}^c(q^n, I, \varepsilon)_k \\ &= \arg\min_{a_k, b_k} \sum_{i\in\omega_k} (w_{ki}(a_k I_i + b_k - q_i^n)^2 + \varepsilon a_k^2) \\ a_k^{n+1} &= \frac{E^w_{\omega_k}(Iq^n) - E^w_{\omega_k}(I) E^w_{\omega_k}(q^n)}{D^w_{\omega_k}(I) + \varepsilon} \\ b_k^{n+1} &= E^w_{\omega_k}(q^n) - a_k^{n+1} E^w_{\omega_k}(I) \end{aligned} \tag{13}$$

$$\begin{aligned} q_i^{n+1} &= \text{SGF}(q^n, I, \varepsilon)_i \\ &= \arg\min_{q_i} \sum_{i\in\omega_k} w_{ki}(a_k^{n+1} I_i + b_k^{n+1} - q_i)^2 \\ &= E^w_{\omega_i}(a^{n+1}) I_i + E^w_{\omega_i}(b^{n+1}) \end{aligned} \tag{14}$$

2) The Total Variation Guided Filter (TVGF): The guided filter has no idea about what kind of the output is preferred because the cost function (6) of the guided filter only considers the constraint between the guidance I and output q. In order to produce the most favorite noise free result, we assemble a new cost function (15) by appending a Total Variation

$$\min_{q,a,b} \sum_{k\in\Omega} (\lambda \text{TV}^2(q_k) + \sum_{i\in\omega_k} ((a_k I_i + b_k - q_i)^2 + \varepsilon a_k^2)) \tag{15}$$

(TV) regularization term $\text{TV}(g_i) = \sqrt{\partial_x^2 g_i + \partial_y^2 g_i}$ to the cost function (6). Putting $\mathcal{P}_2(a^{n+1}, b^{n+1}, I) = \sum_{k\in\Omega}(\lambda \text{TV}^2(q_k) + \sum_{i\in\omega_k}((a_k^{n+1} I_i + b_k^{n+1} - q_i)^2)$ and $q^0 = p$, we can achieve the minimizer by iteratively calculating (16) (17) according

$$a_k^{n+1}, b_k^{n+1} = \arg\min_{a_k, b_k} \mathcal{P}_0(q^n, I, \varepsilon) \tag{16}$$

$$q_i^{n+1} = \arg\min_{q_i} \mathcal{P}_2(a^{n+1}, b^{n+1}, I) \tag{17}$$

to the cyclic coordinate descent algorithm. The optimal solutions of a_k^{n+1}, b_k^{n+1} are expressed as (18) which is same to the definition (7) of the guided filter. Different from (9),

$$\begin{aligned} a_k^{n+1} &= \frac{E_{\omega_k}(Iq^n) - E_{\omega_k}(I) E_{\omega_k}(q^n)}{D_{\omega_k}(I^n) + \varepsilon} \\ b_k^{n+1} &= E_{\omega_k}(q^n) - a_k^{n+1} E_{\omega_k}(I) \end{aligned} \tag{18}$$

the minimizer q_i^{n+1} of (17) equals to (19), where $f_i^{n+1} = \sum_{i\in\omega_k} a_k^{n+1} I_i + b_k^{n+1}$, $\mathcal{D} = \mathcal{F}^*(\partial_x)\mathcal{F}(\partial_x) + \mathcal{F}^*(\partial_y)\mathcal{F}(\partial_y)$, $\mathcal{F}$

$$q_i^{n+1} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(f^{n+1})}{|\omega_k|\mathcal{F}(1) + \lambda\mathcal{D}}\right)_i \tag{19}$$

is the Fast Fourier Transform (FFT) operator and $\mathcal{F}^*$ denotes the complex conjugate. $\mathcal{F}(1)$ is the Fourier Transform of the delta function and $|\omega_k|$ denotes the pixel number in the window ω_k. Moreover, the plus, multiplication and division in (19) are all component-wise operators.

We define the total variation guided filter as the first cyclic coordinate descent pass of (18) (19). Above cyclic coordinate descent minimization can be written in the rolling filtering form (20), where $q = \text{TVGF}(p, I, \varepsilon, \lambda)$ denotes the filtering output q of the total variation guided filter.

$$q^{n+1} = \text{TVGF}(q^n, I, \varepsilon, \lambda) \tag{20}$$

3) The Conservative Guided Filter (CGF): We found that the minimizers of objective functions (6) (15) are trivial zeros. This discovery indicates that both guided filter and total variation guided filter will consume the "energy" of images at each iteration and therefore are dissipative. Note that previous GF-like filters such as the cross-based local multipoint filter [10], the multipoint filter with local polynomial approximation and range guidance [11], the weighted guided filter [13], the fully connected guided filter [12] and the local linear SURE-based edge-preserving filter [14] are also dissipative filters. Readers can verify this from their trivial results of multiple time filtering. We consider that an ideal filter should be conservative, *i.e.* the rolling filtering result must converge to a nontrivial solution. To achieve this goal, we build the cost function (21) and exploit its cyclic coordinate descent iteration to derive the conservative guided filter. Here g denotes a reference image used to constrain the derivation between the output q and g via the data term $(q_i - g_i)^2$. In practice, we usually set it as the input p. So, the solution of (21) must be nontrivial.

$$\min_{q,a,b}\sum_{k\in\Omega}(\sum_{i\in\omega_k}((a_kI_i + b_k - q_i)^2 + \varepsilon a_k^2) + \lambda(q_k - g_k)^2) \tag{21}$$

According to the cyclic coordinate descent algorithm, we can achieve the minimal point of the objective function (21) by iteratively computing (22) (23) with an initial guess $q^0 = p$, where $\mathcal{P}_3(a^{n+1}, b^{n+1}, I, g, \lambda) = \sum_{k\in\omega_i}(a_k^{n+1}I_i + b_k^{n+1} - q_i)^2 + \lambda(q_i - g_i)^2$. Note that the solution of (22) is same to (18). Putting $\alpha = \frac{\lambda}{|\omega_i|+\lambda}$, we can reformulate the solution of (23) as

$$a_k^{n+1}, b_k^{n+1} = \arg\min_{a_k,b_k} \mathcal{P}_0(q^n, I, \varepsilon) \tag{22}$$

$$q_i^{n+1} = \arg\min_{q_i} \mathcal{P}_3(a^{n+1}, b^{n+1}, I, g, \lambda) \tag{23}$$

$$q_i^{n+1} = (1-\alpha)\text{GF}(q^n, I, \varepsilon)_i + \alpha g_i \tag{24}$$

Similar to the definition of the total variation guided filter, the conservative guided filter is defined as the first pass iteration of (22) (23). We thus have $q = \text{CGF}(p, I, g, \varepsilon, \lambda) = (1-\alpha)\text{GF}(p, I, \varepsilon) + \alpha g$. Further, the cyclic coordinate descent minimizing procedure for the objective function (21) can be reformulated in the rolling filtering form (25).

$$q^{n+1} = \text{CGF}(q^n, I, g, \varepsilon, \lambda) \tag{25}$$

4) The Inverse Guided Filters: In the filtering scheme of the guided filter, the guidance image is used to compute the smoothing result. People may raise the following question naturally: can we inverse the filtering procedure by estimating the guidance G from a smoothing result q? Luckily, the answer is positive. We employ the objective function (26) with the initial guess $G^0 = I$ to formulate the Inverse Guided Filter (IGF).

$$\min_{G,a,b}\sum_{k\in\Omega}\sum_{i\in\omega_k}((a_kG_i + b_k - q_i)^2 + \varepsilon a_k^2) \tag{26}$$

Applying the cyclic coordinate descent algorithm to the cost function (26), we iteratively calculate following two subproblems (27) (28), where $\mathcal{P}_0(p, G^n, \varepsilon) = \sum_{i\in\omega_k}((a_kG_i^n + b_k - q_i)^2 + \varepsilon a_k^2)$ and $\mathcal{P}_4(a^{n+1}, b^{n+1}, q) = \sum_{k\in\omega_i}(a_k^{n+1}G_i + b_k^{n+1} - p_i)^2$. Minimizing $\mathcal{P}_0(q, G^n, \varepsilon)$ in (27), we can formulate the closed-form solution of a_k^{n+1}, b_k^{n+1} as (29). Solving the least squares optimization (28), we have (30) and define the inverse guided filter as the first cyclic coordinate descent pass of (29) (31) with $G^0 = I$ and denote its output G as (31).

$$a_k^{n+1}, b_k^{n+1} = \arg\min_{a_k,b_k} \mathcal{P}_0(q, G^n, \varepsilon) \tag{27}$$

$$G_i^{n+1} = \arg\min_{G_i} \mathcal{P}_4(a^{n+1}, b^{n+1}, q) \tag{28}$$

$$\begin{aligned} a_k^{n+1} &= \frac{E_{\omega_k}(G^n q) - E_{\omega_k}(G^n)E_{\omega_k}(q)}{D_{\omega_k}(G^n) + \varepsilon} \\ b_k^{n+1} &= E_{\omega_k}(q) - a_k^{n+1}E_{\omega_k}(G^n) \end{aligned} \tag{29}$$

$$G_i^{n+1} = \frac{E_{\omega_i}(a^{n+1})q_i - E_{\omega_i}(a^{n+1}b^{n+1})}{E_{\omega_i}(a^{n+1}a^{n+1})} \tag{30}$$

$$G = \text{IGF}(q, I, \varepsilon) \tag{31}$$

Similarly, the Inverse Conservative Guided Filter (ICGF) $G = \text{ICGF}(q, I, g, \varepsilon, \lambda)$ is defined as the first cyclic coordinate descent pass of (32). Here g controls the derivation between the output G and g. In practice, we usually set it as the input guidance I. In the first step, we solve $a_k^{n+1}, b_k^{n+1} = \arg\min_{a_k,b_k} \mathcal{P}_0(q, G^n, \varepsilon)$ and obtain (33). In the second step, we optimize $\min_G \sum_{k\in\Omega}\sum_{i\in\omega_k}((a_kG_i + b_k - q_i)^2 + \varepsilon a_k^2) + \lambda(G_k - g_k)^2$ and thus have (34).

$$\min_{G,a,b}\sum_{k\in\Omega}\sum_{i\in\omega_k}((a_kG_i + b_k - q_i)^2 + \varepsilon a_k^2) + \lambda(G_k - g_k)^2 \tag{32}$$

$$\begin{aligned} a_k^{n+1} &= \frac{E_{\omega_k}(G^n q) - E_{\omega_k}(G^n)E_{\omega_k}(q)}{D_{\omega_k}(G^n) + \varepsilon} \\ b_k^{n+1} &= E_{\omega_k}(q) - a_k^{n+1} E_{\omega_k}(G^n) \end{aligned} \tag{33}$$

$$G_i^{n+1} = \frac{\sum_{k\in\omega_i}(a_k^{n+1} q_i - a_k^{n+1} b_k^{n+1}) + \lambda g_i}{\sum_{k\in\omega_i}(a_k^{n+1})^2 + \lambda} \tag{34}$$

Note that it is unwise to use the inverse guided filter or the inverse conservative guided filter alone because they usually do not produce visually meaningful results. We instead compose inverse guided filters with their guided filtering counterparts to perform mutual structure filtering in section IV-B1.

B. Rolling Filtering Schemes

Although the guided filter is designed as a non-iterative filter, its rolling filtering usages still have important applications. We devote this section to the theoretical explanation and improvement for these rolling filtering schemes.

1) Rolling Mutual Structure Filtering (RMSF): The guided filter assumes the geometric structure of both guidance and input coincides with each other. But it is a strong assumption and will cause that the guided filter yields texture mapping artifacts. One way to deal with the inconsistent structure between the guidance and the input is to estimate their mutual structures. Based on the discovery of Shen *et al.* [20], we minimize (35) to obtain textureless results, where $\mathcal{E}(q,a,b,G,c,d) = \sum_{k\in\Omega}\sum_{i\in\omega_k}((a_k G_i + b_k - q_i)^2 + \varepsilon a_k^2) + ((c_k q_i + d_k - G_i)^2 + \epsilon c_k^2)$.

$$\min_{q,a,b,G,c,d} \mathcal{E}(q,a,b,G,c,d) \tag{35}$$

The optimization can be solved by iteratively computing four subproblems (36)-(39) with $q^0 = p$ and $G^0 = I$, where $\alpha_i(x) = \frac{1}{1+E_{\omega_i}(x^2)}$. (36) (37) are used to estimate the linear coefficients used in the guided filter and the inverse guided filter. (38) (39) calculate the linear combination of the guided filter and the inverse guided filter. So we say that we disclose the filtering explanation for the rolling mutual structure filtering successfully and thus we call above procedure the guided filter based rolling mutual structure filtering. For clarity, we list the guided filter based rolling mutual structure filtering algorithm in Algorithm 2.

$$a_k^{n+1}, b_k^{n+1} = \arg\min_{a_k,b_k} \mathcal{P}_0(q^n, G^n, \varepsilon) \tag{36}$$

$$c_k^{n+1}, d_k^{n+1} = \arg\min_{c_k,d_k} \mathcal{P}_0(G^n, q^n, \epsilon) \tag{37}$$

$$\begin{aligned} q_i^{n+1} &= \arg\min_{q_i} \mathcal{P}_1(a_k^{n+1}, b_k^{n+1}, G^m) + \mathcal{P}_4(c_k^{n+1}, d_k^{n+1}, G^n) \\ &= \alpha_i(c^{n+1})\mathrm{GF}(q^n, G^n, \varepsilon)_i \\ &\quad + (1 - \alpha_i(c^{n+1}))\mathrm{IGF}(G^n, q^n, \epsilon)_i \end{aligned} \tag{38}$$

$$\begin{aligned} G_i^{n+1} &= \arg\min_{G_i} \mathcal{P}_1(c_k^{n+1}, d_k^{n+1}, q^n) + \mathcal{P}_4(a_k^{n+1}, b_k^{n+1}, q^n) \\ &= \alpha_i(a^{n+1})\mathrm{GF}(G^n, q^n, \epsilon)_i \\ &\quad + (1 - \alpha_i(a^{n+1}))\mathrm{IGF}(q^n, G^n, \varepsilon)_i \end{aligned} \tag{39}$$

Algorithm 2 The GF based RMSF Algorithm (The guided filter based rolling mutual structure filtering algorithm)

```
procedure GF-RMSF
    Inputs:
    $p, I, \varepsilon, \epsilon, N$
    Initialize:
    $q^0 = p, G^0 = I$
    for $n = 0$ to $N$ do
        $a_k^{n+1}, b_k^{n+1} = \arg\min_{a_k,b_k} \mathcal{P}_0(q^n, G^n, \varepsilon)$
        $c_k^{n+1}, d_k^{n+1} = \arg\min_{c_k,c_k} \mathcal{P}_0(G^n, q^n, \epsilon)$
        $q_i^{n+1} = \alpha_i(c^{n+1})\,\mathrm{GF}(q^n, G^n, \varepsilon)_i + (1 - \alpha_i(c^{n+1}))\,\mathrm{IGF}(G^n, q^n, \epsilon)_i$
        $G_i^{n+1} = \alpha_i(a^{n+1})\,\mathrm{GF}(G^n, q^{n+1}, \epsilon)_i + (1 - \alpha_i(a^{n+1}))\,\mathrm{IGF}(q^{n+1}, G^n, \varepsilon)_i$
    end for
end procedure
```

Note that the filtering pair, the guided filter and the inverse guided filter, play important roles in the guided filter based rolling mutual structure filtering algorithm according to (38) (39). Specifically, when the guidance G^0 and input q^0 are equal, the mutual structure for the same image $G^0 = q^0$ will be itself. Intuitively, we may consider that the output should be same to the input. In fact, the output and input are not equal. The reason is that in the rolling minimizing procedure (38) (39), the guided filter smooths out details but the inverse guided filter plays the role of preserving the major structure from a smoothed input. Thanks to the two antagonistic terms, the rolling mutual structure filtering can preserve the major structure and suppress details/textures in final results. Due to the same reason, the output cannot be same to the input.

The objective function proposed by Shen [20] can be reduced to (35) if Shen's parameters $\lambda = \beta = 0$. However, the equivalence does not imply our filtering interpretation for the rolling mutual structure filtering is trivial. One major contribution of our work is that we first disclose the filtering explanation for each iteration step. In addition, we can disclose things that are not revealed by Shen. For instance, Shen reports the rolling filtering scheme (40) cannot produce mutual structure filtering results. However, he does not provide an explanation for its filtering behavior. Here we employ the guided filter based rolling mutual structure filtering stated above to illustrate the reason. Comparing (38) (39) with (40), we can find that (38) (39) have two extra inverse guided filter terms which learn guidance from input. So, the mutual structure filtering result is the linear combination of the guided filter and the inverse guided filter. In contrast, (40) only considers the guided filter part which will wipe out details without the help of the inverse guided filter.

$$\begin{aligned} q_i^{n+1} &= \mathrm{GF}(q^n, G^n, \varepsilon)_i \\ G_i^{n+1} &= \mathrm{GF}(G^n, q^n, \varepsilon)_i \end{aligned} \tag{40}$$

Another contribution of our filtering interpretation is that we can employ the rolling mutual structure filtering interpretation to define the inverse conservative guided filter based rolling mutual structure filtering. This is because we can substitute

Algorithm 3 The CGF based RMSF Algorithm (The conservative guided filter based rolling mutual structure filtering algorithm)

```
procedure CGF-RMSF
    Inputs:
    $p$,$I$,$\varepsilon$,$\epsilon$,$\lambda$,$\beta$,$N$
    Initialize:
    $q^0 = p, G^0 = I$
    for $n = 0$ to $N$ do
        $a_k^{n+1}, b_k^{n+1} = \arg\min_{a_k, b_k} \mathcal{P}_0(q^n, G^n, \varepsilon)$
        $c_k^{n+1}, d_k^{n+1} = \arg\min_{c_k, c_k} \mathcal{P}_0(G^n, q^n, \epsilon)$
        $\alpha_i(c^{n+1}) = \frac{1}{1+E_{\omega_i}((c^{n+1})^2)}, \alpha_i(a^{n+1}) = \frac{1}{1+E_{\omega_i}((a^{n+1})^2)}$
        $q_i^{n+1} = \alpha_i(c^{n+1})\,\mathrm{CGF}(q^n, G^n, p, \varepsilon, \lambda)_i + (1 - \alpha_i(c^{n+1}))\,\mathrm{ICGF}(G^n, q^n, I, \epsilon, \beta)_i$
        $G_i^{n+1} = \alpha_i(a^{n+1})\,\mathrm{CGF}(G^n, q^{n+1}, I, \epsilon, \beta)_i + (1 - \alpha_i(a^{n+1}))\,\mathrm{ICGF}(q^{n+1}, G^n, p, \varepsilon, \lambda)_i$
    end for
end procedure
```

the filter pair $(GF(p, I, \varepsilon),\ IGF(p, I, \varepsilon))$ in the guided filter based rolling mutual structure filtering with $(CGF(p, I, \varepsilon, \lambda),$ $ICGF(p, I, \varepsilon, \lambda))$ to assemble the inverse conservative guided filter based rolling mutual structure filtering which is illustrated in Algorithm 3 and has better filtering results.

C. Rolling Flash/No-Flash Filtering (RFNF)

To enhance the quality of flash/no-flash image pairs, Seo *et al.* [16] take the guided filter to synthesize a new image which composes a base image B and a detail image D computed from the flash/no-flash image pairs $(I^{\mathrm{f}}, I^{\mathrm{n}})$ and offer a spectral analysis to illustrate why the rolling usage (41) with $q^0 =$

$$q^{k+1} = \underbrace{\mathrm{GF}(q^k, I^{\mathrm{f}}, \varepsilon)}_{\text{the base image } B} + \lambda \underbrace{(I^{\mathrm{f}} - \mathrm{GF}(I^{\mathrm{f}}, I^{\mathrm{f}}, \varepsilon))}_{\text{the detail image } D} \tag{41}$$

I^{n} can yield better result. Differently, we will interpret (41) as an approximation for the cyclic coordinate descent solver of the objective function (42), where $I^{\mathrm{e}} = \mathrm{GF}(I^{\mathrm{f}}, I^{\mathrm{f}}, \varepsilon) +$

$$\min_{q,a,b} \sum_{k\in\Omega} (\sum_{i\in\omega_k} ((a_k I_i + b_k - q_i)^2 + \varepsilon a_k^2) + \lambda (q_k - I_k^{\mathrm{e}})^2) \tag{42}$$

$\tau(I^{\mathrm{f}} - \mathrm{GF}(I^{\mathrm{f}}, I^{\mathrm{f}}, \varepsilon))$. Let g be an alias of I^{e}, the objective function is same to (21). Hence we can achieve the minimizer by iteratively computing (43) with $\alpha = \frac{\lambda}{|\omega_i| + \lambda}$.

$$q_i^{n+1} = (1 - \alpha)\mathrm{GF}(q^n, I^{\mathrm{f}}, \varepsilon)_i + \alpha I_i^{\mathrm{e}} \tag{43}$$

If $\alpha \approx 0$ and $\tau = \frac{\lambda}{\alpha}$, we have $(1 - \alpha) \approx 1$ and $\alpha I^{\mathrm{e}} \approx \lambda(I^{\mathrm{f}} - \mathrm{GF}(I^{\mathrm{f}}, I^{\mathrm{f}}, \varepsilon))$. In addition, q^{n+1} in (43) reduces to $\mathrm{GF}(q^n, I^{\mathrm{f}}, \varepsilon) + \lambda(I^{\mathrm{f}} - \mathrm{GF}(I^{\mathrm{f}}, I^{\mathrm{f}}, \varepsilon))$ which has the same form with (41). This discovery convinces that the rolling filtering scheme of Seo is just an approximation for a special case of (43). We therefore can generalize (41) to (43). More importantly, the generalization produces much better results in motion deblurring.

V. Comparison and Experiments

We now demonstrate that our new filters and rolling filtering schemes are capable to generate state-of-the-art results for different applications, where all filters are implemented in C++ without SIMD (single instruction multiple data) optimization on an i7 CPU with 4GB memory and five GF-like filters, including the cross-based local multipoint filter (CLMF) [10], the multipoint filter with local polynomial approximation and range guidance (MLPA) [11], the weighted guided filter (WGF) [13], the fully connected guided filter (FCGF) [12] and the local linear SURE-based edge-preserving filter (LLSURE) [14], are used to perform comparison.

A. Computational Complexity

Similar to the guided filter, the spatial-aware guided filter can be computed in linear time as it only substitutes the average operator $E_{\omega_i}(x)$ used in the guided filter for the weighted average operator $E_{\omega_i}^w(x)$. The computational complexity of the spatial-aware guided filter therefore is determined by $E_{\omega_i}^w(x)$ of which the computational complexity is same to the Gaussian convolution $\sum_{j\in\omega_i} w_{ij}x_j$. As is known to us, $\sum_{j\in\omega_i} w_{ij}x_j$ can be computed in linear time via the acceleration algorithms [1]. Table II reports the runtime of the spatial-aware guided filter to filter one-megapixel image. The speed of the spatial-aware guided filter is almost same to the guided filter and significantly faster than the multipoint filter of Tan [11] and the fully connected guided filter of Dai [12].

As for the total variation guided filter, it no longer can be computed in linear time as the computational complexity of FFT operator $\mathcal{F}$ is $O(n \log n)$. However, it does not mean that the total variation guided filter cannot be computed efficiently because the implementation of FFT is highly optimized on modern hardware [21]. We can verify this in Table II as it does not increase the runtime very much.

The computational complexity of the conservative guided filter, the inverse guided filter and the inverse conservative guided filter is same to the guided filter which is linear computational complexity because all of them only involve point-wise arithmetic calculations and the average operator $E(x)$. Table II reports the runtime of eleven filters to filter one-megapixel image. The speed of our conservative guided filter, inverse guided filter and inverse conservative guided filter is almost same to the guided filter and significantly faster than the cross-based local multipoint filter, the multipoint filter with local polynomial approximation and range guidance and the fully connected guided filter.

B. Texture Smoothing and Enhancement

Fig 2 plots the texture smoothing and enhancement results produced by the bilateral filter (BF) [2] and GF-like filters: the guided filter (GF) [3], the cross-based local multipoint filter (CLMF) [10], the multipoint filter with local polynomial approximation and range guidance (MLPA) [11], the weighted guided filter (WGF) [13] [13], the fully connected guided filter (FCGF) [12], the spatial-aware guided filter (SGF) in the Rolling Guidance Filtering (RGF) [18] scheme. From the

TABLE II: Running time comparison of eleven GF-like filters for one megapixels filtering, where GF, CLMF, MLPA, WGF, FCGF, LLSURE, SGF, TVGF and CGF denote the guided filter [3], the cross-based local multipoint filter [10], the multipoint filter with local polynomial approximation and range guidance [11], the weighted guided filter [13], the fully connected guided filter [12], the local linear SURE-based edge-preserving filter [14], ours spatial-aware guided filter, total variation guided filter and conservative guided filter, respectively.

	GF [3]	CLMF [10]	MLPA [11]	WGF [13]	FCGF [12]	LLSURE [14]	SGF	TVGF	CGF	IGF	ICGF
Time	820ms	1780ms	3520ms	840ms	1810ms	830ms	860ms	870ms	830ms	810ms	820ms

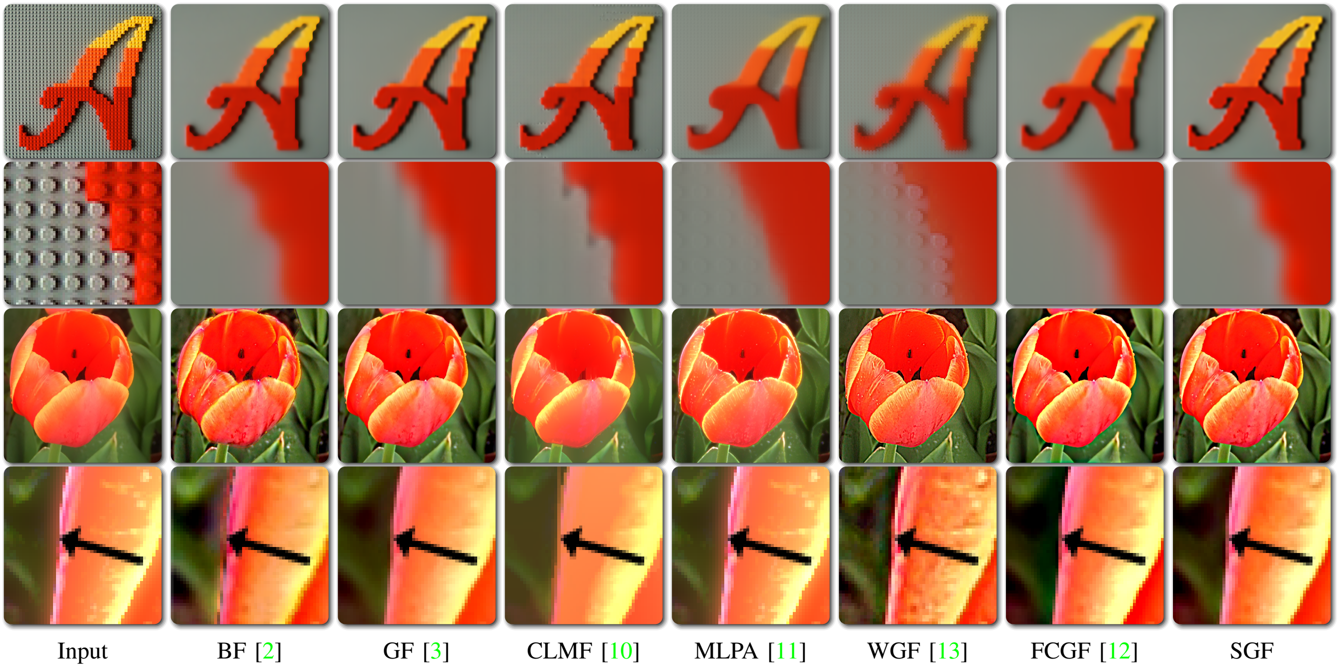


Fig. 2: Texture Smoothing and Enhancement results. The images in the first two rows are texture smoothing results and their close-ups. Similarly, the images in the last two rows are detail enhancement results and their close-ups. From left to right, the images are input and results yielded by BF($\sigma_s = 5, \sigma_r = 0.2$), GF($r = 10, \varepsilon = 0.001$), CLMF($\tau = 0.196, \varepsilon = 0.04$), MLPA($k = 0.1/255, \varepsilon_s = 0.005^2, \varepsilon_r = 1$), WGF($r = 5, \varepsilon = 0.04$), FCGF($\sigma = 0.5, \varepsilon = 1$), SGF($r = 10, \varepsilon = 0.001, \sigma = 5$) in the Rolling Guidance Filtering scheme [18].

TABLE III: Quantitative comparison for seven denoisng methods in terms of PSNR, MSE and SSIM. The method with best denoisng ability usually receives large PSNR and SSIM values and a small MSE index.

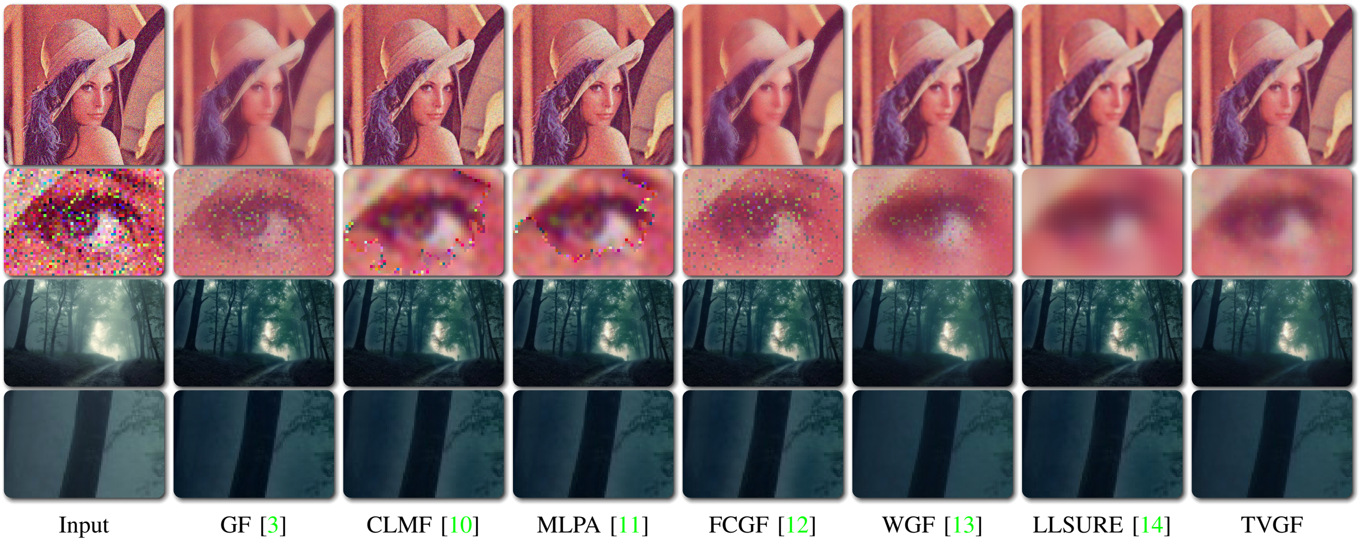
	GF [3]	CLMF [10]	MLPA [11]	FCGF [12]	WGF [13]	LLSURE [14]	TVGF
PSNR	21.8370	22.6815	22.3029	23.4069	23.1650	23.7239	**24.5191**
MSE	0.0066	0.0054	0.0060	0.0048	0.0050	0.0041	**0.0035**
SSIM	0.8878	0.8940	0.8908	0.9102	0.9059	0.9261	**0.9384**

texture smoothing results and their close-ups in the first two rows, we can observe that the result of SGF is similar to BF. Contrastively, GF, CLMF and WGF fail to remove all textures because they do not consider the spatial similarity which is the key to perform textures-aware smoothing in the RGF scheme. Similar to SGF, MLPA and FCGF take the spatial similarity into account too and therefore produce acceptable results. However, the biggest problem of them is the running cost: MLPA and FCGF spend more time than SGF. Although BF is comparable with SGF in texture smoothing, it suffers from gradient reversal artifacts in image enhancement as illustrated in the third row and the close-up in the fourth row. In contrast, all GF-like filters including our SGF do not have the problem according to the experiment results.

C. Noise and Haze Removal

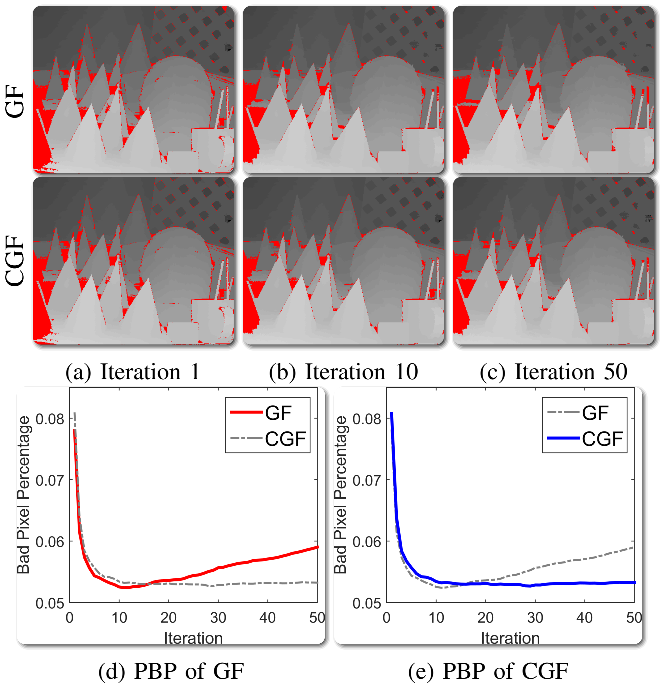
Benefiting from the total variation regularization [23], [24], the total variation guided filter (TVGF) is able to reduce noise without structure degradation. For qualitative comparison, we demonstrate the denoising results of seven filtering methods in the first two rows of Fig 3. Visually, only the results of FCGF and LLSURE are comparable with our TVGF. Table III adopts three indices including PSNR [25], MSE [26] and SSIM [27] to estimate the denoisng quality. Our TVGF ranks first on all three indices.

The total variation regularization also empowers TVGF the halo artifacts suppression ability, which is the selling point of WGF. The second row in Fig 3 and its close-ups show an instance from the single image haze removal experiment. From this figure, it is not difficult to find that only the results



Input GF [3] CLMF [10] MLPA [11] FCGF [12] WGF [13] LLSURE [14] TVGF

Fig. 3: Noise and Haze Removal. The images in the first two rows are denoising results and their close-ups. The images in the last two rows are haze removal results and their close-ups which illustrate halo artifacts. From left to right, the images are input and results yielded by GF($r = 10, \varepsilon = 0.1$), CLMF($\tau = 0.196, \varepsilon = 0.04$), MLPA($k = 0.1/255, \varepsilon_s = 0.005^2, \varepsilon_r = 1$), FCGF($\sigma = 0.5, \varepsilon = 1$), WGF($r = 20, \varepsilon = 0.001$), LLSURE($r = 5$), TVGF($r = 10, \varepsilon = 0.01, \lambda = 45$).



(a) Iteration 1 (b) Iteration 10 (c) Iteration 50

(d) PBP of GF (e) PBP of CGF

Fig. 4: Stereo Matching Comparison. From Fig a to Fig c, the images are the results of the guided filter (a typical dissipative filters) and the conservative guided filter (our conservative filter) respectively, where the images are stereo matching results produced by the guided filter and the conservative guided filter with different iteration numbers and red pixels indicate bad matching pixels. (d) and (e) illustrate PBP performance curve with respect to different iteration numbers. Here PBP denotes the Percentage of Bad matching Pixels [22].

of TVGF and WGF do not suffer from halo artifacts.

D. Multiple Time Filtering for Stereo Matching

In the stereo matching framework [28], the guided filter is employed to smooth each slice of the cost volume one pass. However “Is it optimal to filter each slice one pass?” We find that the answer is negative. The reason is that the PBP performance curve of the guided filter, illustrated in the first row of Fig 4, is a parabola-like line. We owe this to the dissipation property of the guided filter. Specifically speaking, the PBP performance increases with the iteration number N on an interval $[0, \bar{N})$ since the noise is removed gradually without degrading image edges very much. Here $\bar{N}$ denotes the optimal filtering number and usually is greater than one. With the filtering time increasing (*i.e.* $N \in (\bar{N}, \infty]$), the structure information of inputs will be depleted by dissipative filters, so the PBP performance will decrease on the interval $(\bar{N}, \infty]$. But it is very hard to decide the optimal filtering pass $\bar{N}$ in advance. Unlike the dissipative guided filter, the PBP curve of the conservative guided filter is monotonically decreasing because the conservative guided filter is conservative and can preserve the structure information of input no matter how many times filtering are applied. The property implies an easy way to choose N for the conservative guided filter (CGF): within an admissive computational burden, we make N as large as possible. The ability of the conservative guided filter brings us much convenience as we no longer need to tweak N carefully as we do for the guided filter.

E. Major Structure Extraction

We compare the ability of the guided filter based rolling mutual structure filtering (40) and the methods of Xu [4], Zhang [18], Shen [20] in extracting the major structure and illustrate results in Fig 5. The rolling filtering scheme (40) does not yield satisfactory result as it only considers the smoothing part of Rolling Mutual Structure Filtering (RMSF)

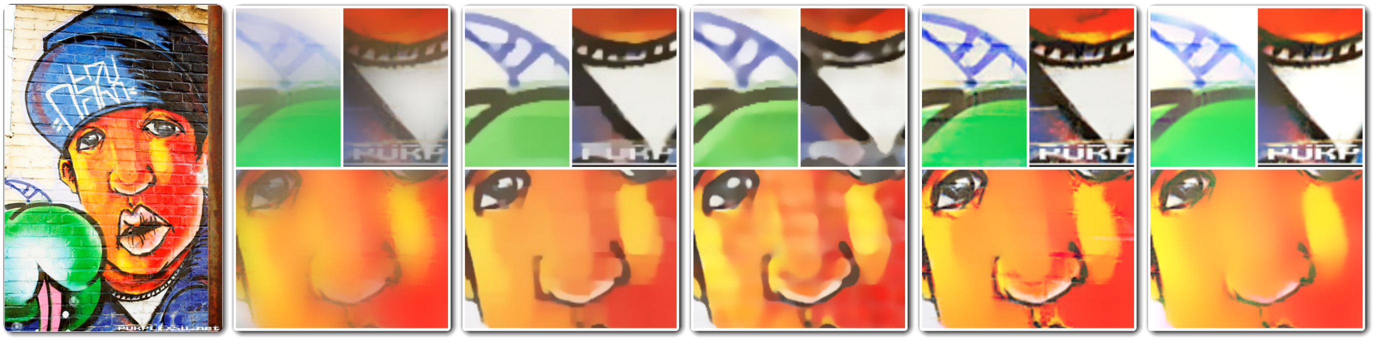


Fig. 5: Major Structure Extraction. The image of Xu and Zhang are subscribed to authors. Other results are yielded by Shen($\varepsilon_1 = \varepsilon_2 = 0.001, \lambda = \beta = 5$), (40)($r = 6, \varepsilon = 0.01$), CGF($r = 6, \varepsilon = 0.001, \lambda = 0.01$).

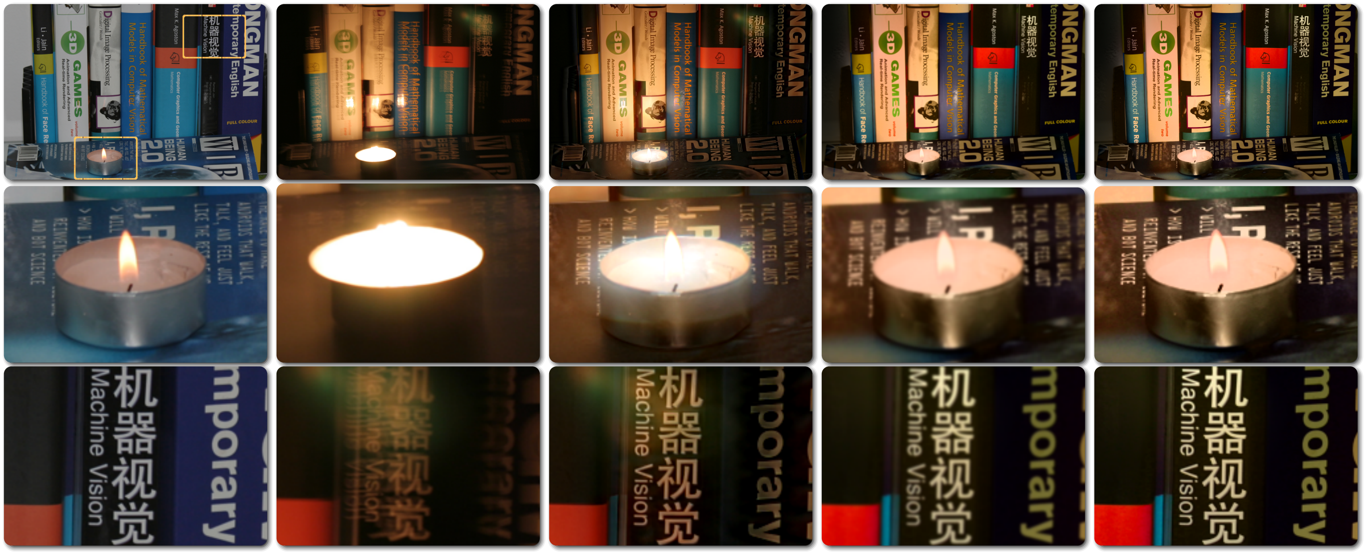


Fig. 6: Flash/No-flash Deblurring. From left to right, the images are flash image, non-flash image and results yield by Zhou, Seo and RFNF, where the image of Zhou suffers from over-saturated regions in the blurred image.

[20]. The results of Xu and Zhang are much better than (40). This is because Xu designs a relative total variation term to extract main structures and suppress textures. Different from Xu, Zhang designs a new filter and employs it to filter images with the complete control of detail smoothing under a scale measure. However, taking a close look at their results, we can easily find that the two methods could not distinguish major structures from textures very well and thus not only leave details in the major structures but also blur some important major structures. To solve this problem, Shen proposes mutual-structure for joint filtering. Although the method of Shen can produce the same results as the GF based RMSF, our CGF based RMSF derived from the GF based RMSF are able to remove textures more clearly. More importantly, we provide a more elegant and general way to deal with this task.

F. Flash/No-Flash Deblurring

Motion blur due to camera shake is an annoying problem while taking pictures. Our generalized Rolling Flash/No-Flash Filtering (RFNF) can be applied to flash/no-flash deblurring. The method of Zhou *et al.* [29] as well as the method (41) of Seo [16] are used to perform comparison. Fig 6 illustrates the results of three methods, where no-flash images suffer from mild noise and strong motion blur. As shown in the close-ups, our method outperforms the method of Zhou by obtaining much finer details with better color contrast even though our method does not estimate a blur kernel at all. In addition, compared with our method, Zhou's method also suffers from over-saturated regions in the blurred image. Unlike Zhou, the results of Seo are rather satisfactory. However, their edges are not as sharp as ours.

VI. Conclusion and Future Work

The major contribution of our work is to disclose the equivalence between the guided filter and the cyclic coordinate descent solver of a least squares optimization. The equivalence provides us new insight on how to extend the guided filter as well as the rolling filtering scheme. Specifically, employing the equivalence, we define the guided filter as the first pass iteration of the cyclic coordinate descent solver and derive new rolling filtering scheme from the minimizing procedure. In addition, we modify the objective function of the guided filter and obtain several GF-like filters and two rolling filtering usages from new objective functions.

In this paper, totally five GF-like filters(*i.e.* SGF, TVGF, CGF, IGF, ICGF) and two rolling filtering schemes (*i.e.* CGF-RMSF and RFNF) are proposed. We conduct extensive experiments to disclose/verify the smoothing property/ability of these filters and rolling filtering schemes. Experimental results prove their advantages.

However, we think that these achievements should not be the limit of the power of the connection between the guided filter and the cyclic coordinate descent. Our future work will be devoted to deriving more GF-like filters as well as their rolling schemes according to new objective functions adapted to various tasks in computer vision and graphics. Other than the filtering quality, the smoothing efficiency is another critical problem. We will incorporate with the acceleration technique [30] to speed up proposed GF-like filters and rolling filtering schemes.

Acknowledgment

This work was supported by the National Natural Science Foundation of China Youth Fund (Grant No. 61701235) and the Fundamental Research Funds for the Central Universities (Grant No. 30917011323).

References

[1] P. Getreuer, "A Survey of Gaussian Convolution Algorithms," *Image Processing On Line*, vol. 3, pp. 286–310, 2013. 1, 7

[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *IEEE International Conference on Computer Vision*, 1998, pp. 839–846. 1, 2, 7, 8

[3] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013. 1, 2, 3, 7, 8, 9

[4] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Transactions on Graphics*, vol. 31, no. 6, pp. 139:1–139:10, 2012. 1, 9, 10

[5] J. A. Snyman, *Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms*, New York, 2005. 1, 2

[6] Y. Li and S. Osher, "A new median formula with applications to pde based denoising," *Communications in Mathematical Sciences*, vol. 7, no. 3, pp. 741–753, 2009. 2

[7] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1151, 2002. 2

[8] G. Dong and S. Acton, "On the convergence of bilateral filter for edge-preserving image smoothing," *IEEE Signal Processing Letters*, vol. 14, no. 9, pp. 617–620, 2007. 2

[9] L. Caraffa, J.-P. Tarel, and P. Charbonnier, "The guided bilateral filter: When the joint/cross bilateral filter becomes robust," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1199–1208, 2015. 2

[10] J. Lu, K. Shi, D. Min, L. Lin, and M. Do, "Cross-based local multipoint filtering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 430–437. 2, 5, 7, 8, 9

[11] X. Tan, C. Sun, and T. Pham, "Multipoint filtering with local polynomial approximation and range guidance," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2941–2948. 2, 4, 5, 7, 8, 9

[12] L. Dai, M. Yuan, F. Zhang, and X. Zhang, "Fully connected guided image filtering," in *IEEE International Conference on Computer Vision*, 2015, pp. 352–360. 2, 4, 5, 7, 8, 9

[13] Z. Li, J. Zheng, Z. Zhu, W. Yao, and S. Wu, "Weighted guided image filtering," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 120–129, 2015. 2, 5, 7, 8, 9

[14] T. Qiu, A. Wang, N. Yu, and A. Song, "Llsure: Local linear sure-based edge-preserving image filtering," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 80–90, 2013. 2, 5, 7, 8, 9

[15] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 192–207, jan 2018. 2

[16] H. J. Seo and P. Milanfar, "Robust flash denoising/deblurring by iterative guided filtering," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, p. 3, 2012. 2, 7, 10

[17] M. Yelameli, P. Shanthi, and M. K. Subramanian., "Flash/no-flash sensor denoising and deblurring using guided image filter," *International Journal of Scientific & Engineering Research*, vol. 6, 2013. 2

[18] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *European Conference on Computer Vision*, 2014, pp. 815–830. 2, 4, 7, 8, 9, 10

[19] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising," *International Journal of Computer Vision*, vol. 86, no. 1, pp. 1–32, jul 2009. 3

[20] X. Shen, C. Zhou, L. Xu, and J. Jia, "Mutual-structure for joint filtering," in *IEEE International Conference on Computer Vision*, 2015. 6, 9, 10

[21] Y. Ogata, T. Endo, N. Maruyama, and S. Matsuoka, "An efficient, model-based cpu-gpu heterogeneous fft library," in *IEEE International Symposium on Parallel and Distributed Processing*, 2008, pp. 1–10. 7

[22] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002. 9

[23] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992. 8

[24] D. Strong and T. Chan, "Edge-preserving and scale-dependent properties of total variation regularization," *Inverse Problems*, vol. 19, no. 6, p. S165, 2003. 8

[25] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, 2010, pp. 2366–2369. 8

[26] E. Lehmann and G. Casella, *Theory of Point Estimation*. Springer Verlag, 1998. 8

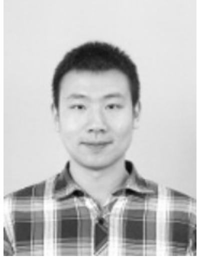
[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. 8

[28] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2013. 9

[29] S. Zhuo, D. Guo, and T. Sim, "Robust flash deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2440–2447. 10

[30] L. Dai, M. Yuan, Z. Li, X. Zhang, and J. Tang, "Hardware-efficient guided image filtering for multi-label problem," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jul 2017. 11

Longquan Dai is currently an Assistant Professor in Intelligent Media Analysis Group (IMAG) at School of Computer Science and Engineering, Nanjing University of Science and Technology (NJUST). He received his B.A., M.S. and Ph.D. degrees from Henan University of Technology (HAUT) in 2006, Shantou Univeristy (STU) in 2010 and National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA) in 2016, respetively. His current research interests lie in computer graphics, computer vision, and optimization-based techniques for image analysis and synthesis.

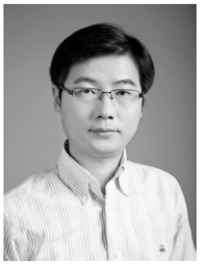


Mengke Yuan received the B.S. degrees of Applied Mathematics and M.S. degrees of Computational Mathematics from Zhengzhou University, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree in computer science with the Institute of Automation, Chinese Academy of Sciences. His research interests lie in computer vision, machine learning, optimization-based techniques for image analysis and synthesis, approximation theory based acceleration of image filtering technique.

Liang Tang received Ph.D. degrees from Institute of Automation, Chinese Academy of Sciences in 2015. He is currently the vice-president in charge of technology in CASA Environmental Technology Co., Ltd and CASA EM&EW IOT Research Center. His current research interests lie in image processing and signal processing techniques for industrial product development.

Xiaopeng Zhang (M’11) received the B.S. and M.S. degrees in mathematics from Northwest University, Xi’an, China, in 1984 and 1987, respectively, and the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 1999. He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His main research interests are computer graphics and computer vision.

Yuan Xie (M’12) received the Ph.D. degree in Pattern Recognition and Intelligent Systems from the Institute of Automation, Chinese Academy of Sciences (CAS), in 2013. He received his master degree in school of Information Science and Technology from Xiamen University, China, in 2010. He is currently with Visual Computing Laboratory, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, and also with the Research Center of Precision Sensing and Control, Institute of Automation, CAS. His research interests include image processing, computer vision, machine learning and pattern recognition. Dr. Xie received the Hong Kong Scholar Award from the Society of Hong Kong Scholars and the China National Postdoctoral Council in 2014.



Jinhui Tang received the B.E. and Ph.D. degrees from the University of Science and Technology of China in 2003 and 2008, respectively. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. From 2008 to 2010, he was a Research Fellow with the School of Computing, National University of Singapore. He visited the School of Information and Computer Science, UC Irvine from 2010 to 2010, as a Visiting Research Scientist. From 2011 to 2012, he visited Microsoft Research Asia, as a Visiting Researcher. His current research interests include large-scale multimedia search, social media mining, and computer vision. He has authored over 100 journal and conference papers in these areas. He serves as an Editorial Board Member of Pattern Analysis and Applications, Multimedia Tools and Applications, Information Sciences, Neurocomputing. He was a recipient of ACM China Rising Star Award 2014, and a co-recipient of the Best Paper Award in ACM Multimedia 2007, PCM 2011, and ICIMCS 2011. He is a member of ACM.