LLSURE: Local Linear SURE-Based Edge-Preserving Image Filtering

Tianshuang Qiu, Aiqi Wang, Nannan Yu, and Aimin Song

***Abstract*— In this paper, we propose a novel approach for performing high-quality edge-preserving image filtering. Based on a local linear model and using the principle of Stein's unbiased risk estimate as an estimator for the mean squared error from the noisy image only, we derive a simple explicit image filter which can filter out noise while preserving edges and fine-scale details. Moreover, this filter has a fast and exact linear-time algorithm whose computational complexity is independent of the filtering kernel size; thus, it can be applied to real time image processing tasks. The experimental results demonstrate the effectiveness of the new filter for various computer vision applications, including noise reduction, detail smoothing and enhancement, high dynamic range compression, and flash/no-flash denoising.**

***Index Terms*— Edge-preserving image filtering, high dynamic range (HDR) compression, local linear model, Stein's unbiased risk estimate (SURE).**

I. Introduction

FILTERING is perhaps the most important operation of image processing and computer vision, and it is used extensively in a wide range of applications, including image smoothing and sharpening, noise removal, resolution enhancement and reduction, feature extraction, and edge detection. The simplest filtering should be explicit linear translation-invariant (LTI) filtering, which can be implemented using a convolution mask. For example, box filter, also known as "moving average," is implemented by a local averaging operation where the value of each pixel is replaced by the average of all the values in the local neighborhood. Box filter is the quickest blur algorithm, but its smoothing effect is often not sufficient. Another widely used LTI filter is Gaussian filter with the weights chosen according to the shape of a Gaussian function. Gaussian filter is a very good filter for removing noise drawn from a normal distribution. And the multi-scale space representation of an image can be obtained easily by Gaussian smoothing with increasing variance. Although LTI filtering is very simple and is used extensively in early vision processing, it also has some disadvantages. LTI filtering not only smooths the noise but also blurs important structures along with noise, and outliers exert large influence on filtered output.

To reduce these undesirable effects of linear filtering, a variety of edge-preserving filtering techniques have been proposed over the past few years. Since taking into account local structures and statistics during the filtering process, edge-preserving filtering is non-linear and can preserve the image details and local geometries while removing the undesirable noise. Most of popular filtering techniques in this class have been developed based on partial differential equations (PDE's) and variational models. For example, non-linear/anisotropic diffusions (AD) [1], as well as regularization methods based on the total variation (TV) [2], are most popular and widely used non-linear filtering methods in signal and image processing. In general, an initial image is progressively approximated by filtered versions which are smoother or simpler in some sense. Actually, this process introduces a hierarchy into the image structures, thus one can use a scale-space representation for extracting semantically important information. These methods are very effective tools for edge preserving filtering, however they are implemented as an iterative process which is usually slow and may raise issues of stability.

As a good alternative to the iterative algorithm, the bilateral filter was first termed by Tomasi and Manduchi [3] based on the work [4], [5], and later modified and improved in [6]. Since its formulation is simple, and method is non-iterative which achieving satisfying results with only a single pass, bilateral filtering has been proven to be a valuable tool in a variety of areas of computer vision and image processing [7]–[11]. However the direct implementation of bilateral filter is known to be slow. Although several techniques [12]–[15] are proposed to speed up the evaluation of the bilateral filter, its fast implementation is still a challenging problem. And it has recently been noticed that bilateral filter may have the gradient reversal artifacts in detail decomposition and high dynamic range (HDR) compression [16].

Recently, some novel edge-preserving smoothing filters have been proposed, including weighted least squares filter (WLS) [16], edge avoiding wavelets (EAW) [17], and domain transform (DT) method [18] to approximate geodesic distance by iterating 1D-filtering operations. In particular, based on a local linear model, He *et al.* [19] proposed a new filtering method - guided filter that can perform effective

Manuscript received February 23, 2012; revised June 18, 2012; accepted August 2, 2012. Date of publication August 20, 2012; date of current version December 20, 2012. This work was supported in part by the National Natural Science Foundation of China, under Grant 61172108 and Grant 61139001 and the National Science and Technology Support Program, under Grant 2012 BAJ18B06. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hitoshi Kiya.

T. Qiu, N. Yu, and A. Song are with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: qiutsh@dlut.edu.cn; yunannan1981@yahoo.com.cn; songaimin@djtu.edu.cn).

A. Wang is with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China, and also with the School of Mathematics and Physics, Dalian Jiaotong University, Dalian 116028, China (e-mail: aiqi.wang79@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2012.2214052

edge-preserving smoothing by considering the content of a guidance image. To avoid a trivial solution, He *et al.* introduced a regularization parameter which determines the amount of smoothing.

Although edge-preserving smoothing filters are wildly used as useful tools for a variety of image editing and manipulation tasks, most of them are originally proposed to remove noise while preserving fine details and geometrical structures in the original image. It is well known that the denoising performance of an algorithm is often measured in terms of peak signal-to-noise ratio (PSNR). A higher PSNR would normally indicate that the reconstruction is of higher quality. To maximize the PSNR, an alternative approach is to minimize the mean square error (MSE) which can be estimated accurately by Stein's unbiased risk estimate (SURE) from the noisy image only. As it does not depend on a priori knowledge of the unknown signal, SURE has already turn out to be a flexible and effective tool which can be applied by directly parameterizing the estimator and finding the optimal parameters that minimize the MSE estimate. The best-known use of the SURE for wavelet denoising is Donoho's SureShrink algorithm [20]. Recently, an analytical form of SURE for the NLM algorithm has been derived [21] and further extended and studied [22]. In particular, Luisier *et al.* [23], [24] have proposed a very appealing denoising algrithm - Stein's unbiased risk estimator-linear expansion of thresholds (SURE-LET) and later been extended to color images, video and mixed Poisson-Gaussian noise condition [25], [26]. Similar idea has been early and independently proposed by Pesquet and his collaborators [27], [28].

Inspired by the SURE-LET method and He's guided filter, we present a novel edge-preserving smoothing filter, called LLSURE filter which is based on a local linear model and the principle of Stein's unbiased risk estimate (SURE). In our case, the filtered output in a local window are considered as a very simple affine transform of input signal in the same window, and the optimal transform coefficients are determined by minimizing the SURE. The LLSURE filter has the edge-preserving smoothing property that can filter out noise while preserving edges and fine-scale details. Moreover, it is very simple and has an exact linear-time algorithm which can be applied to various image processing tasks.

The paper is organized as follows. In Section II, we introduce some notations and briefly review the SURE principle. In Section III, we first show how this principle can be exploited to build an efficient edge-preserving smoothing filter by using a local linear model, and then extend the approach to a joint filter. Finally, the multi-scale edge-preserving decomposition is constructed and the computational complexity of the proposed algorithms is discussed. Next, in Section IV, the experimental results and various applications are shown to demonstrate the effectiveness and flexibility of the LLSURE filter. Some concluding remarks are given in Section V.

II. Notation and Problem Formulation

A. Problem Setting

We consider the measurement model

$$y_i = x_i + n_i, \quad i = 1, \ldots, N \tag{1}$$

where x_i is the underlying latent signal of interest at a position i, y_i is the noisy measured signal (pixel value), and n_i is the corrupting zero-mean white Gaussian noise with variance σ^2. The standard simplified denoising problem is to find a reasonably good estimate $\hat{\mathbf{x}}$ of $\mathbf{x} = [x_1, \ldots, x_N]^T$ from the corresponding data set $\mathbf{y} = [y_1, \ldots, y_N]^T$. To restate the problem more concisely, the complete measurement model in vector notation is given by

$$\mathbf{y} = \mathbf{x} + \mathbf{n}. \tag{2}$$

B. Mean Squared Error and Stein's Unbiased Risk Estimate

The mean squared error (MSE) of the denoised image with respect to its noise-free version is

$$\mathrm{MSE}(\hat{\mathbf{x}}) = \frac{1}{N}\|\mathbf{x} - \hat{\mathbf{x}}\|^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{3}$$

where $\|\cdot\|^2$ is the Euclidean norm.

In denoising applications, the performance is often measured in terms of peak signal-to-noise ratio (PSNR), which can be defined as follows:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}(\mathbf{x}^2)}{\mathrm{MSE}(\hat{\mathbf{x}})}\right). \tag{4}$$

The higher the PSNR is, the better the performance of denoising algorithm. Since $\mathbf{x}$ is the noise-free signal which does not affect the value of PSNR in any algorithm, maximizing PSNR is equivalent to minimizing MSE. However, one can not approximate MSE without the original signal $\mathbf{x}$. Thanks to Stein's unbiased risk estimate (SURE) provides a means for unbiased estimation of the true MSE, it is possible to replace MSE by SURE without any assumptions on the original signal. SURE is specified by the following analytical expression [22], [29]:

$$\mathrm{SURE}(\hat{\mathbf{x}}) = \frac{1}{N}\|\mathbf{y} - \hat{\mathbf{x}}\|^2 + \frac{2\sigma^2}{N}\mathrm{div}_{\mathbf{y}}\{\hat{\mathbf{x}}\} - \sigma^2 \tag{5}$$

where $\mathrm{div}_{\mathbf{y}}\{\hat{\mathbf{x}}\}$ is the divergence of the output estimate with respect to the measurements

$$\mathrm{div}_{\mathbf{y}}\{\hat{\mathbf{x}}\} = \sum_{i=1}^{N}\frac{\partial \hat{x}_i}{\partial y_i} \tag{6}$$

which needs to be well defined in the weak sense. The derivation of SURE relies on the additive white Gaussian noise hypothesis and assumes the knowledge of the noise variance σ^2. In practice, σ^2 can be easily estimated from the measured data (e.g., using the median of absolute deviation).

III. Method

A. LLSURE-Local Linear SURE-Based Filter

The expression of the estimate $\hat{\mathbf{x}}$ plays a crucial role in SURE-based denoising methods. For example, D. Van De Ville and M. Kocher [21], [22] used the filtered result of NLM algorithm as an estimate to derive the divergence term, and then plugged the divergence into the SURE expression to obtain an analytical form of SURE-based NLM which can

be used to optimize the parameters of the NLM algorithm. Different to them, Luisier *et al*. [23], [24] parameterized directly the estimate $\hat{\mathbf{x}}$ using a linear combination of derivatives of Gaussians (DOG) of the measured signal $\mathbf{y}$ and minimized the MSE estimate to determine the parameters.

Similar to SURE-LET, we also assume that the filtered output $\hat{\mathbf{x}}$ is a linear combination of measured signal $\mathbf{y}$, but our method is based on a local linear model [19]. Firstly, in a local neighborhood around every position, we assume that the filtered output image patch is a very simple affine transform of input image patch in the same position. Secondly, for every local neighborhood, we determine the optimal transform coefficients by minimizing the MSE estimate (SURE). Finally, all filtered output image patches are averaged together to obtain the final filtered result. The proposed procedure works as follows.

Let w_i be a local window around the position i, $\mathbf{y}_w$ and $\hat{\mathbf{x}}_w$ be an input image patch and a filtered output image patch respectively corresponding to the window w_i. Here, for simplicity, we consider w_i as a square window of pixels with a fixed size. Let

$$\hat{\mathbf{x}}_{w_i} = a_i \mathbf{y}_{w_i} + b_i, \quad a_i, b_i \in \mathbb{R}, \quad a_i \geq 0 \tag{7}$$

where a_i and b_i are some affine transform coefficients assumed to be constants in window w_i. Thus, for any $j \in w_i$, we have $\hat{x}_j = a_i y_j + b_i$. Notice that we restrict a_i to be non-negative for preventing the filtered output results far from the input data.

Plugging (7) into the SURE expression (5), we have

$$\begin{aligned}
\mathrm{SURE}(a_i, b_i) =& \frac{1}{N_w}\|\mathbf{y}_{w_i} - (a_i \mathbf{y}_{w_i} + b_i)\|^2 \\
&+ \frac{2\sigma^2}{N_w}\mathrm{div}_{\mathbf{y}}\{a_i \mathbf{y}_{w_i} + b_i\} - \sigma^2 \\
=& \frac{1}{N_w}\|\mathbf{y}_{w_i} - (a_i \mathbf{y}_{w_i} + b_i)\|^2 \\
&+ \frac{2\sigma^2}{N_w} a_i N_w - \sigma^2 \\
=& \frac{1}{N_w}\|\mathbf{y}_{w_i} - (a_i \mathbf{y}_{w_i} + b_i)\|^2 + 2\sigma^2 a_i - \sigma^2
\end{aligned} \tag{8}$$

where $N_w = (2r+1)\times(2r+1)$ is the pixel number in window w_i, and r is the window radius.

In order to determine the optimal transform coefficients a_i and b_i corresponding to each local window w_i, we minimize the $\mathrm{SURE}(a_i, b_i)$. Taking the first derivatives of (8) with respect to a_i and b_i (see Appendix A), we have

$$\begin{cases} a_i = (\sigma_i^2 - \sigma^2)/\sigma_i^2 \\ b_i = (1 - a_i)\bar{y}_i \end{cases} \tag{9}$$

where $\bar{y}_i$ and σ_i^2 are the mean and variance respectively of input data in local window w_i. Since a_i is restricted as non-negative, the optimal transform coefficients are determined finally as follows

$$\begin{cases} a_i^* = \max(\sigma_i^2 - \sigma^2, 0)/(\sigma_i^2 + \varepsilon) \\ b_i^* = (1 - a_i^*)\bar{y}_i \end{cases} \tag{10}$$

where ε is a small constant number avoiding the denominator equal to zero.

According to (7) and (10), we obtain all filtered output in local window w_i

$$\hat{x}_j^i = a_i^* y_j + b_i^*, \quad j \in w_i. \tag{11}$$

In the following, if we move the local window in the entire image and compute the corresponding optimal transform coefficients according to (10), we can get the result similar to (11) at every position of the entire image. However, due to the overlap for each other moved windows, pixel j is involved in all the windows w_k that contain j, so the value of $\hat{x}_j^i$ in (11) may not be the same when it is computed in different windows. As a result, for the pixels lying in the overlapping regions, we obtain multiple estimates. These competing estimates must be fused or aggregated into the one final estimate. Aggregation, as an important part for image denoising using patch-based methods, is widely studied in [30]–[34].

Formally, the aggregation can be performed by a weighted average of the multiple estimates:

$$\hat{x}_j = \sum_{i \in w_j} \lambda_i \hat{x}_j^i. \tag{12}$$

The key idea of a fine aggregation procedure is how to choose the weight λ_i for getting the best estimate. The simplest method of aggregating such multiple estimates is to average them using equal weight. However, such naive averaging does not consider the risk of the estimate $\hat{x}_j$. Obviously, when a pixel lies on near the boundary, it should receive different weights from two sides of the boundary. In general, the risk of the estimate $\hat{x}_j$ can be defined as

$$R(\hat{x}_j, x_j) = \mathbb{E}[(\hat{x}_j - x_j)^2] \tag{13}$$

where the expectation $\mathbb{E}$ is with respect to the probability measure associate with the noise. Actually, the risk can be decomposed into bias squared and variance terms

$$\begin{aligned}
R(\hat{x}_j, x_j) &= \mathbb{E}[(\hat{x}_j - \mathbb{E}[\hat{x}_j] + \mathbb{E}[\hat{x}_j] - x_j)^2] \\
&= \mathbf{var}(\hat{x}_j) + \mathbf{bias}^2(\hat{x}_j).
\end{aligned} \tag{14}$$

Here, similarly to [32] and [35], we assume the estimates $\hat{x}_j^i$, $i \in w_j$, are independent noisy observations of original signal x_j. This means that they can be considered of zero bias, so is their combination. As a result, to minimize the risk $R(\hat{x}_j, x_j)$, we only need to minimize the variance

$$\mathbf{var}(\hat{x}_j) = \mathbf{var}\left(\sum_{i \in w_j} \lambda_i \hat{x_j^i}\right) = \sum_{i \in w_j} \lambda_i^2 \mathbf{var}(\hat{x}_j^i) \tag{15}$$

under the constraint that $\sum_{i \in w_j} \lambda_i = 1$. Using a Lagrangian, this leads to

$$\lambda_i = \frac{\mathbf{var}^{-1}(\hat{x_j^i})}{\sum_{k \in w_j} \mathbf{var}^{-1}(\hat{x_j^k})}. \tag{16}$$

This kind of aggregation procedure is called as Variance-based Weighted Average (WAV) that is also described in [36] and in references therein.

While the WAV-based method can improve the accuracy of estimation, it is more computationally complex than the simple uniform average. In practice, we can replace $\mathbf{var}(\hat{x}_j)$ by

var(y_i), i.e. σ_i^2 that is calculated in (9). Thus after determining (a_i^*, b_i^*) for every window w_i in the image, we obtained the final filtered output by

$$\begin{aligned}\hat{x}_j &= \sum_{i\in w_j} \lambda_i \hat{x}_j^i \\ &= \sum_{i\in w_j} \frac{\sigma_i^{-2}}{\sum_{k\in w_j}\sigma_k^{-2}}(a_i^* y_j + b_i^*) \\ &= \frac{1}{\sum_{k\in w_j}\sigma_k^{-2}} \sum_{i\in w_j}(\sigma_i^{-2}a_i^* y_j + \sigma_i^{-2} b_i^*) \\ &= \bar{a}_j y_j + \bar{b}_j \end{aligned} \tag{17}$$

where $\bar{a}_j = \frac{1}{W_j}\sum_{i\in w_j}\sigma_i^{-2}a_i^*$, $\bar{b}_j = \frac{1}{W_j}\sum_{i\in w_j}\sigma_i^{-2}b_i^*$ and $W_j = \sum_{i\in w_j}\sigma_i^{-2}$.

Before going further, let us explain intuitively the edge-preserving smoothing property of LLSURE filter here.

From (10) and (11) it can be easily verified that if the local variance at a position i is not greater than the noise variance, i.e., $\sigma_i^2 - \sigma^2 \le 0$, then $a_i^* = 0$ and $b_i^* = \bar{y}_i$, which shows the position i is in a "flat" region, and its value becomes the average of the pixels nearby. In contrast, if the local variance is much greater than the noise variance, then a_i^* becomes close to 1 while b_i^* is close to 0, which shows the location i is in a "high variance" region, and its value tends to remain unchanged. From above, we can get that the noise variance σ^2 is a crucial threshold that determines "what is an edge/a high variant regions that should be preserved."

Similarly, the use of different window sizes will also greatly affect the quality of processed images. If the window is too small, the algorithm may not work well and may lead to unwanted block-like artifacts. On the other hand, if the window is too large, the region containing "edge" will also become large, where there are relatively large variances. This will lead to the denoising performance of LLSURE filter is poor near by "edge." Our experiments indicate that $r = 2$ or $r = 3$ is a good choice.

B. Joint LLSURE Filter

When there are two undesirable images of same scene, one often expects to obtain a more satisfactory image by combining them. For instance, in a "flash/no-flash" scenario, a no-flash picture is smoothed from the flash one defining the edges to be preserved. This idea can be effectively performed using the joint (cross) bilateral filter which has been simultaneously discovered by Eisemann and Durand [37] and Petschnigg *et al.* [38] as a variant of the bilateral filter. Recently, He *et al.* [19] proposed also a novel explicit image filter - guided filter that is similar to joint (cross) bilateral filter and is wildly used in computer vision and image processing.

Here, we extend the LLSURE filter to joint LLSURE filter. In order to smooth the input signal $\mathbf{f}$ while relying on the guidance signal $\mathbf{y}$ to locate the edges to preserve, we rewrite (8) as

$$\begin{aligned}\text{J-SURE}(a_i, b_i) &= \frac{1}{N_w}\|\mathbf{f}_{w_i} - (a_i\mathbf{y}_{w_i} + b_i)\|^2 \\ &\quad + \frac{2\sigma^2}{N_w}\text{div}_{\mathbf{y}}\{a_i\mathbf{y}_{w_i} + b_i\} - \sigma^2 \\ &= \frac{1}{N_w}\|\mathbf{f}_{w_i} - (a_i\mathbf{y}_{w_i} + b_i)\|^2 + 2\sigma^2 a_i - \sigma^2 \end{aligned} \tag{18}$$

where $\mathbf{f}_{w_i}$ and $\mathbf{y}_{w_i}$ are filter input image patch and guidance image patch respectively corresponding to window w_i. Similar to (9) (see Appendix B), we can get

$$\begin{cases} a_i = \big(\text{cov}(\mathbf{f}_{w_i}, \mathbf{y}_{w_i}) - \sigma^2\big)/\sigma_i^2 \\ b_i = \bar{f}_i - a_i\bar{y}_i \end{cases} \tag{19}$$

where $\bar{f}_i$ is the mean of input signal $\mathbf{f}$ in local window w_i, $\bar{y}_i$ and σ_i^2 are the mean and variance of guidance signal $\mathbf{y}$ in local window w_i, and $\text{cov}(\mathbf{f}_{w_i}, \mathbf{y}_{w_i}) = (1/N_w)\sum_{j=1}^{N_w} f_j y_j - \bar{f}_i\bar{y}_i$ is the covariance between $\mathbf{f}_{w_i}$ and $\mathbf{y}_{w_i}$.

Similar to (10), we expect the σ^2 in (19) plays same role as a threshold whatever the covariance $\text{cov}(\mathbf{f}_{w_i}, \mathbf{y}_{w_i})$ is positive or negative. Therefore, we introduce the soft-threshold function to determine the final optimal transform coefficients as follows

$$\begin{cases} a_i^* = \text{soft}\left(\text{cov}(\mathbf{f}_{w_i}, \mathbf{y}_{w_i}), \sigma^2\right)/(\sigma_i^2 + \varepsilon) \\ b_i^* = \bar{f}_i - a_i\bar{y}_i \end{cases} \tag{20}$$

where $\text{soft}(x, a) = \text{sign}(x)\max(|x| - a, 0)$.

After determining a_i^* and b_i^*, the following process is analogous to that for getting (11) and (17), so the final output of the joint filter at each position is computed by

$$\hat{f}_j = \sum_{i\in w_j}\lambda_i(a_i^* y_j + b_i^*) = \bar{a}_j y_j + \bar{b}_j \tag{21}$$

where $\bar{a}_j = \frac{1}{W_j}\sum_{i\in w_j}\sigma_i^{-2}a_i^*$, $\bar{b}_j = \frac{1}{W_j}\sum_{i\in w_j}\sigma_i^{-2}b_i^*$ and $W_j = \sum_{i\in w_j}\sigma_i^{-2}$.

C. Edge-Preserving Decompositions Based on LLSURE

Multi-scale edge-preserving decomposition is often required in many applications of computational photography. The decomposition consists of a coarse piecewise smoothed version of the image, along with a sequence of difference images capturing detail at progressively finer scales [16].

Using the LLSURE filter, it is easy to construct such multi-scale edge-preserving decomposition. First, for convenience, the LLSURE method can be simply written as

$$\hat{\mathbf{x}} = \text{LLSURE}(\mathbf{y}, r, \sigma^2) \tag{22}$$

where $\mathbf{y}$ and $\hat{\mathbf{x}}$ are input signal and filtered output signal respectively, r is window radius, and σ^2 is noise variance. And then, let $\mathbf{y}_1, \ldots, \mathbf{y}_k$ denote progressively coarser versions of $\mathbf{y}$. If we denote $\mathbf{y}_0 = \mathbf{y}$, the k detail layers are defined as

$$\mathbf{d}_i = \mathbf{y}_{i-1} - \mathbf{y}_i, \qquad i = 1, \ldots, k. \tag{23}$$

Finally the original image $\mathbf{y}$ is easily recovered from this decomposition by simply adding up the coarsest base layer

TABLE I

PERFORMANCE OF VARIOUS APPROACHES AS MEASURED BY PSNR

σ	5	10	15	20	25	5	10	15	20	25	5	10	15	20	25
Input PSNR	33.88	28.30	25.31	23.13	21.86	33.88	28.30	25.31	23.13	21.86	33.88	28.30	25.31	23.13	21.86
Method	*Lena* 512 × 512					*Cameraman* 512 × 512					*Barb* 512 × 512				
FBF	36.51	32.92	30.96	29.02	27.43	37.51	33.74	31.44	29.67	28.50	35.29	30.98	28.28	27.10	26.15
GIF	36.88	32.91	30.87	29.45	28.53	37.81	33.48	31.97	29.69	28.35	35.74	31.20	28.51	27.01	25.36
SURE-LET	37.64	34.43	32.56	31.30	30.33	38.20	33.77	31.44	29.79	28.55	36.59	32.28	30.02	28.47	27.30
LLSURE	37.13	33.78	31.93	30.84	29.93	38.65	34.61	32.55	31.27	30.34	36.03	31.81	29.43	27.93	26.95
Method	*Man* 512 × 512					*Boat* 512 × 512					*Peppers* 512 × 512				
FBF	35.85	31.52	29.40	28.23	27.01	35.68	31.63	29.63	27.83	26.94	35.72	32.62	30.80	29.60	28.57
GIF	35.83	31.50	29.41	27.92	27.10	36.20	31.71	29.53	27.77	26.75	36.02	32.61	30.78	29.51	28.30
SURE-LET	36.49	32.41	30.47	29.23	28.52	37.07	33.21	31.17	29.85	28.93	36.38	33.38	31.84	30.81	30.03
LLSURE	36.12	32.25	30.35	29.11	28.31	36.62	32.61	30.60	29.35	28.34	36.15	33.20	31.78	30.73	29.88

and the detail layers

$$\mathbf{y} = \mathbf{y}_k + \sum_{i=1}^{k} \mathbf{d}_i. \tag{24}$$

Similar to [16], we have two methods for computing the progressive coarsening sequence $\mathbf{y}_1, \ldots, \mathbf{y}_k$. The first one is to filter the original image k times using LLSURE filter, each time increasing the value of the parameter σ^2. In other words

$$\mathbf{y}_i = \text{LLSURE}(\mathbf{y}, r, c^i\sigma^2) \tag{25}$$

for some initial value of σ^2 and some factor c.

The second method is to obtain each image in the sequence by applying LLSURE to the previous image:

$$\mathbf{y}_i = \text{LLSURE}(\mathbf{y}_{i-1}, r, c^i\sigma^2). \tag{26}$$

In the second method the image is repeatedly smoothed, and the resulting coarsened images tend more strongly towards piecewise constant regions separated by strong edges. We still increase σ^2 by factor c at each iteration, as this result in a more significant increase in smoothness in each iteration.

Although the process producing the sequence is different, however, we find the results are similar. So we only use the first method in our experiment.

D. $O(N)$ Time Exact Algorithm

$O(N)$ time implies that the time complexity is only dependent on the size of input image and yet independent of the window radius r [14], [19], thus we are free to use arbitrary kernel sizes in the applications. From (10) and (17) one can see that the major computational cost of LLSURE filter method depends on the computation of local mean and variance. A brute-force implementation will result an increasing in its computational complexity when the kernel becomes larger. In order to decrease this cost down to a constant time, an effective strategy is to calculate them by a box filter which can be easily implemented in constant time using integral image [39]. Therefore the LLSURE filter can also be fast implemented in constant time $O(N)$.

For color image, the LLSURE filter method is applied simply for each channel separately.

TABLE II

RELATION COMPUTATION TIME OF VARIOUS DENOISING APPROACHES (SECONDS)

Image size	Method			
	FBF	GIF	SURE-LET	LLSURE
128 × 128	0.1736	0.0145	0.1249	0.0151
256 × 256	0.4972	0.0481	0.2704	0.0514
512 × 512	1.8034	0.1691	0.8813	0.1973
1024 × 1024	6.9142	0.6411	3.4729	0.7872

IV. EXPERIMENTAL RESULTS AND APPLICATIONS

In this section, we apply the LLSURE filter to a great variety of computer vision and graphics applications.

A. Image Denoising

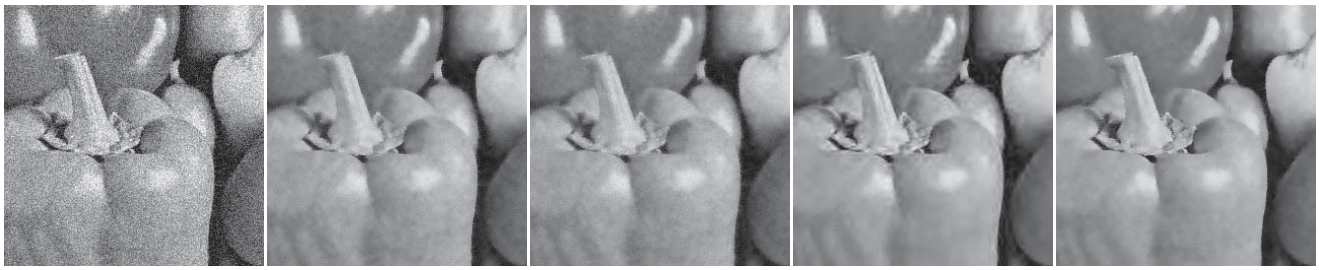
Image noise reduction without structure degradation is perhaps the most important task in low-level image processing. In this experiment, the proposed algorithm is evaluated and compared with many other existing techniques, including fast bilateral filter (FBF) [15], guided image filter (GIF) [19] and SURE-LET method [23]. The parameters of each method have been set according to the optimal performance produced in the experiment. Actually, as the noise variance σ^2 can be estimated from the measured data using the median of absolute deviation, our LLSURE method is completely automatic during denoising process if we set the window radius fixed ($r = 2$ in this experiment), contrary to the FBF and GIF which have to tune manually the parameters to get high quality filtered results. This reliable comparison was only possible thanks to the kindness of the various authors who have provided their respective Matlab codes on their personal websites.

The tested images are corrupted by simulated additive Gaussian white noise at five different power levels $\sigma \in [5, 10, 15, 20, 25]$. And the denoising process has been performed over ten different noise realizations for each standard deviation and the resulting PSNRs and computation times averaged over these ten runs.

Table I and Table II summarize the relative PSNR results and computation time of various methods considered in

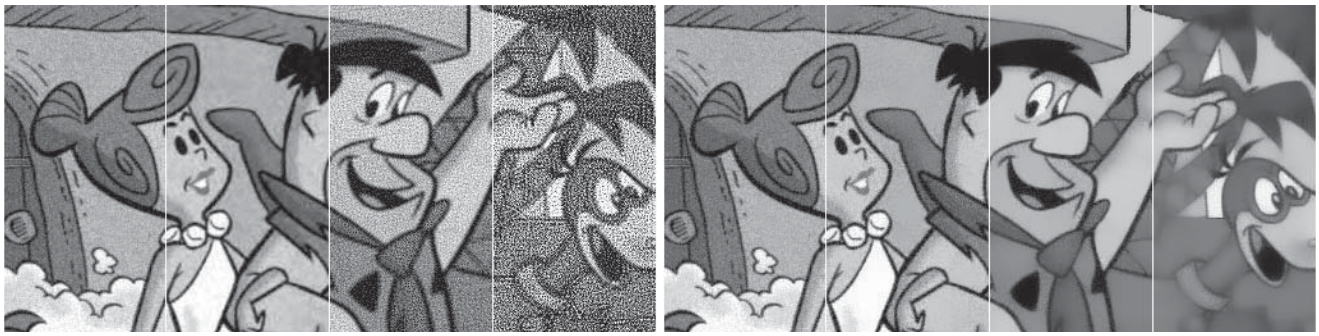
(a) (b) (c) (d) (e)

Fig. 1. Results for part of the 512×512 *Lena* test image corrupted with additive white noise of $\sigma = 15$. (a) Noisy image: PSNR $= 25.31$ dB. (b) FBF denoising result: PSNR $= 30.96$ dB. (c) GIF denoising result: PSNR $= 30.87$ dB. (d) SURE-LET denoising result: PSNR $= 32.56$ dB. (e) LLSURE denoising result: PSNR $= 31.93$ dB.



(a) (b) (c) (d) (e)

Fig. 2. Results for part of the 512×512 *Peppers* test image corrupted with additive white noise of $\sigma = 25$. (a) Noisy image: PSNR $= 21.70$ dB. (b) FBF denoising result: PSNR $= 28.57$ dB. (c) GIF denoising result: PSNR $= 28.30$ dB. (d) SURE-LET denoising result: PSNR $= 30.03$ dB. (e) LLSURE denoising result: PSNR $= 29.88$ dB.



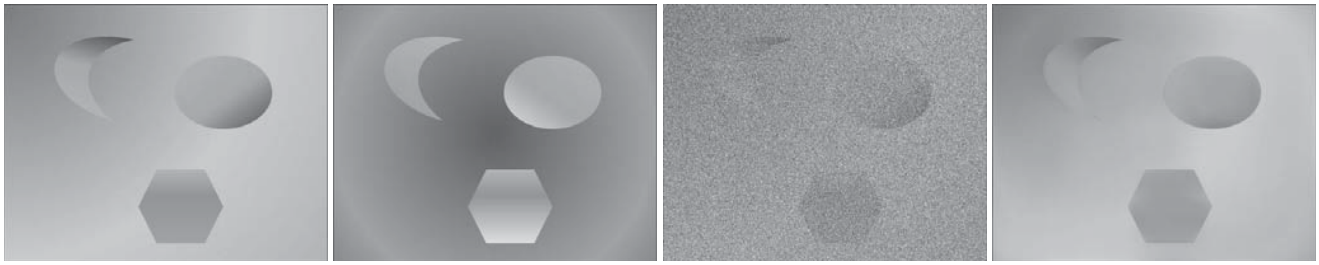
(a) (b)

Fig. 3. Denoised results with different estimated variances. (a) Denoised results using SURE-LET. (b) Denoised results using our LLSURE. The estimated values are 0.7σ, 1.0σ, 2.0σ, and 3.0σ, respectively.

this paper. From Table I, we can notice that our method (LLSURE) gives better PSNR results than those of FBF and GIF for all of the images. Although the SURE-LET gives best PSNR results for most of the image, it is slower than our method as shown in Table II. More precisely, the execution of our current un-optimized Matlab (R2009) implementation of the whole denoising task takes about 0.1973 s for 512×512 grayscale images and about 0.7872 s for 1024×1024 grayscale images on a computer with Pentium (R) Dual-Core CPU speed of 2.50 GHz and 2 GB of memory, whereas Luisier *et al.* SURE-LET takes 0.8813 s and 3.4729 s respectively on the same working conditions.

We also show some visual examples for the "lena" test image, Fig. 1 (a), corrupted with additive Gaussian noise of $\sigma = 15$, and the "peppers" test image, Fig. 2 (a), corrupted with noise of $\sigma = 25$. As we can notice that our method does not produce any artifacts, and at the same time is able to well preserve the features of the original image.

Finally, we would like to stress that contrary to FBF and GIF, our LLSURE method is completely automatic during denoising process since the noise variance σ^2 can be estimated from the measured data. Moreover, the estimated variance can be interpreted as a scale parameter in LLSURE method, that is, we can construct a multi-scale space representation of the initial image with different variance, whereas in Luisier *et al*. SURE-LET method, the estimated variance is only a crucial threshold which determines the denoising quality. This is illustrated in Fig. 3. As we can observe that several gradually simplified (smoothed) versions of the initial image is obtained by using our method with increasing variance, thus one can use this scale-space representation for extracting semantically important information.



(a) (b) (c) (d)

Fig. 4. Performance of the joint LLSURE. (a) Noise-free image. (b) Guidance image. (c) Noisy image with $\sigma = 80$. (d) Denoised result using joint LLSURE.

(a) (b) (c) (d)

Fig. 5. Flash/no-Flash denoising. (a) No-Flash image. (b) Flash image. (c) Denoised results using LLSURE with $\sigma = 0.1$. (d) Denoised results using joint LLSURE with $\sigma = 0.01$.

This strategy is used widely in various image processing tasks.

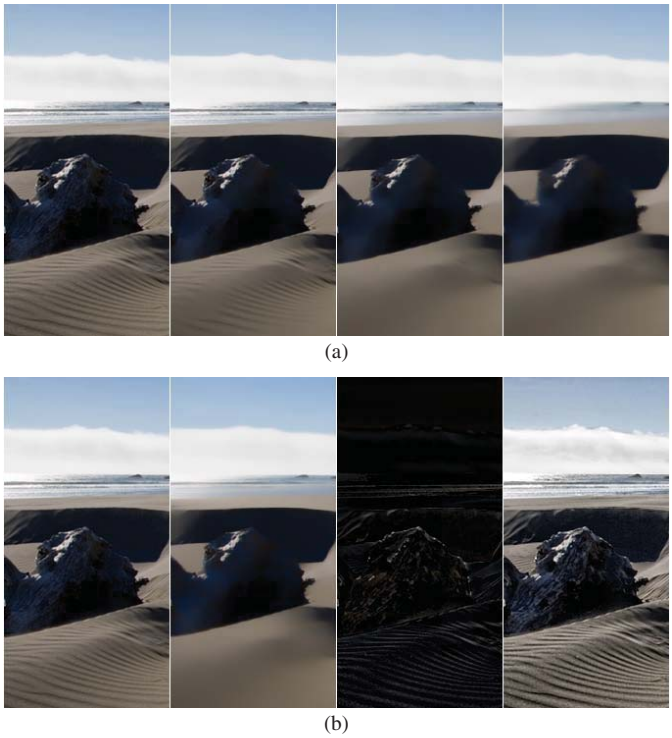
B. Joint Denoising

In Section III-B, the LLSURE filter is extended to the joint LLSURE filter. Here, we demonstrate the capability of the joint LLSURE filter to reduce noise and smooth a input image based on the edge information of the guidance image. First, a synthetic example is shown in Fig. 4, where Fig. 4 (a) is a synthetic noise-free image composed of objects of different shape and varying contrast, Fig. 4 (b) is the guidance image which has same edge information and dramatic difference in brightness, Fig. 4 (c) is a noisy variant of Fig. 4 (a) with noise $\sigma = 80$, and Fig. 4 (d) is the denoised result of using our joint LLSURE filter. As we can observe that, even in a case of a very noisy image, this algorithm works fairly well. Next, an example of flash/no-flash denoising is shown in Fig. 5. By comparing Fig. 5 (c) and Fig. 5 (d), we can notice that the joint LLSURE filter can provide much better result than the LLSURE filter in which only the no-flash image is used. This is because the joint filter can capture more significant edge information by considering the content of a guidance image.

C. Detail Smoothing and Enhancement

Detail smoothing and enhancement are basic and common image processing operations, available in most image editing software. Here we show that these operations can be implemented in a simple and efficient way using our LLSURE method.

Fig. 6(a) shows three gradually smoothed versions of the original image according to (25) with increasing variance,



(a)

(b)

Fig. 6. Detail smoothing and enhancement results. (a) Gradually smoothed results using our LLSURE. (b) Detail enhancement result using our LLSURE. From left to right: original image, smoothing result, detail layer, and enhancement result.

where window radius $r = 5$ is fixed. From it we can observe that the small scale details in the image are gradually removed while the strong edges are not significantly blurred, unlike what is done by Gaussian filter.

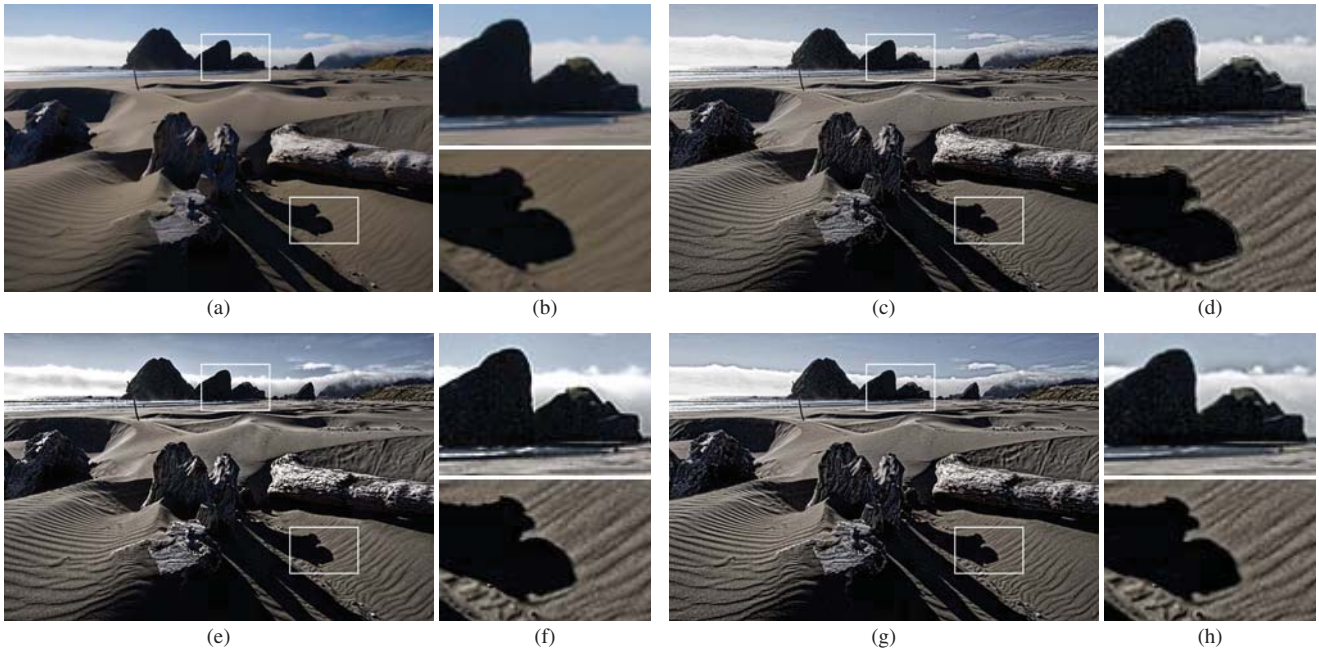


Fig. 7. Detail enhancement results and zoomed-in view of two different zones. (a) and (b) Original image and two zoomed-in parts. (c) and (d) FBF enhancing results and two zoomed-in parts. (e) and (f) WLS enhancing results and two zoomed-in parts. (g) and (h) LLSURE enhancing results and two zoomed-in parts. The detail layer is boosted×5.

In contrast, we can also enhance the small scale details without introducing obvious artifacts. Here let $\mathbf{y}_\sigma$ be the smoothed version of the original image $\mathbf{y}$ created by using LLSURE method with the value σ, and $\mathbf{d} = \mathbf{y} - \mathbf{y}_\sigma$ be the detail layer. The detail enhanced image is obtained as $\mathbf{y}_e = \mathbf{y} + a \times \mathbf{d}$, where a is the scale factor.

A detail enhancement example is shown in Fig. 6 (b), where the detail layer are boosted $\times 3$. In Fig 7, we compare our method with FBF [15] and WLS [16]. As we can notice that our results, as well as the one of Farbman *et al.*, is halo-free with an increased fine-scale detail and does not bring the gradient reversal artifacts seen in Fig. 7 (d). Moreover, our method is much faster than WLS. For the color test image, size 457×628, our method takes about 1.7288 s, whereas FBF and WLS take 3.5187 s and 3.9650 s respectively.

D. HDR Compression

High dynamic range images are widely used since they have wider intensity range than normal digital images, which fit the intensity range of real scene identified by human eyes. Displaying and printing these images, however, on conventional media requires reducing the range of their luminance component. Here we show it is easy to perform detail preserving compression of HDR images by using our LLSURE method.

First we convert RGB to the YUV space. And then we decompose the log-luminance channel (Y) into $k+1$ progressively coarser versions, $\mathbf{L}_0 = \log(\mathbf{Y}), \mathbf{L}_1, \ldots, \mathbf{L}_k$, according to (25), and obtain different detail layers according to (23). Finally we compute the compressed luminance channel simple by scaling the coarsest base layer and these detail layers and recombining them as follows

$$\log(\mathbf{Y}_C) = \beta_0 \mathbf{L}_k + \sum_{i=1}^{k} \beta_i \mathbf{d}_i \tag{27}$$

and convert it back to RGB to get the final result.

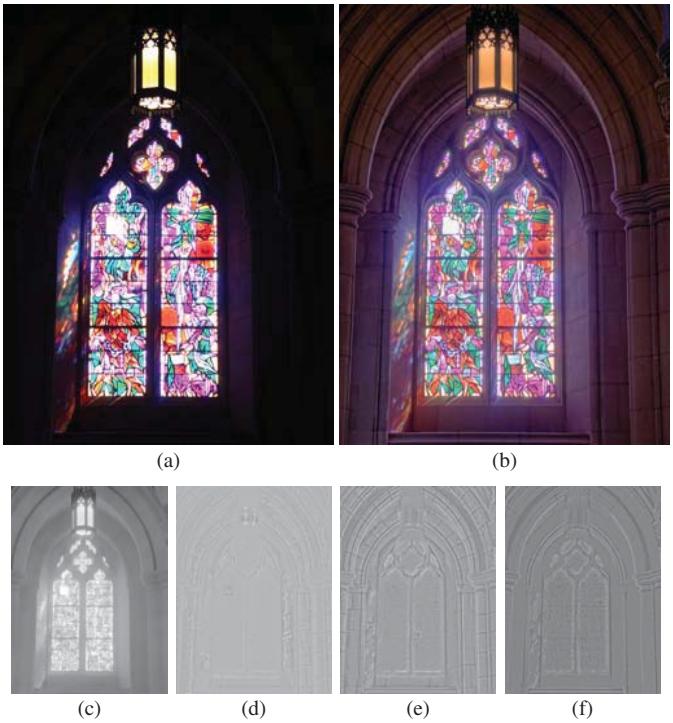


Fig. 8. HDR compression results using our LLSURE. (a) Original HDR image. (b) HDR compression result. (c) Base layer L_3. (d) Detail layer d_0. (e) Detail layer d_1. (f) Detail layer d_2.

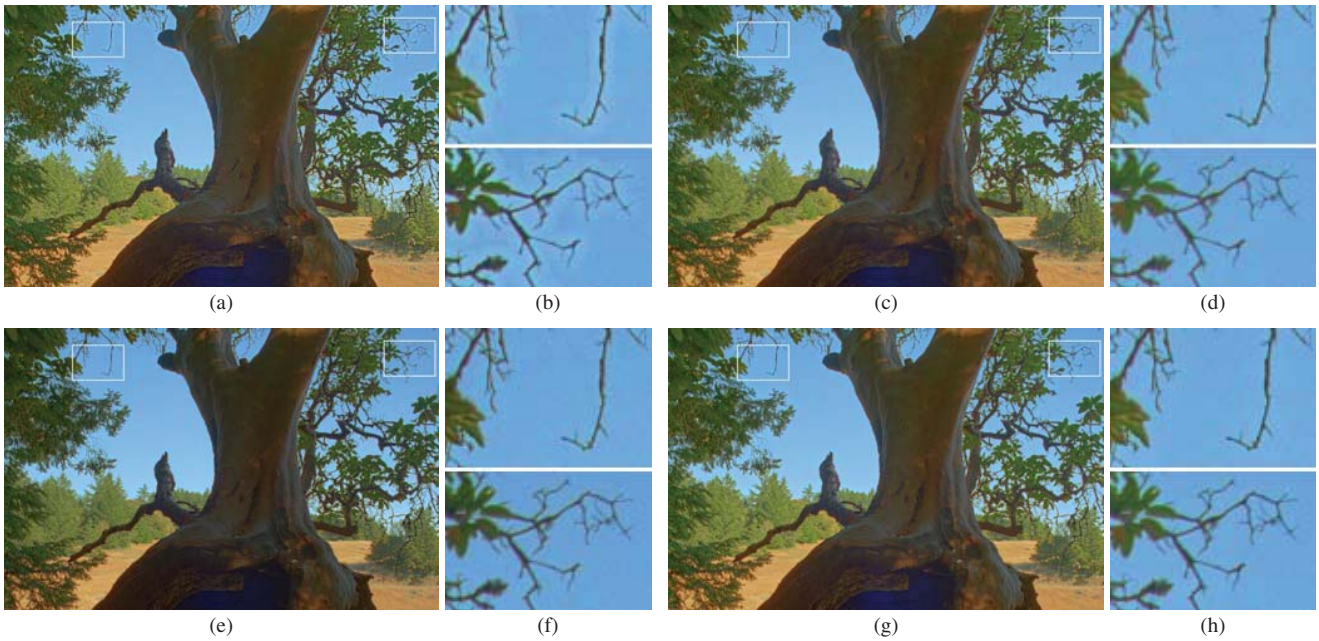


Fig. 9. HDR compression results and zoomed-in view of two different zones. (a) and (b) FBF compressing results and two zoomed-in parts. (c) and (d) GIF compressing results and two zoomed-in parts. (e) and (f) WLS compressing results and two zoomed-in parts. (g) and (h) LLSURE compressing results and two zoomed-in parts.

Fig. 8 illustrates the procedure to compress a HDR image, where three detail layers and the coarsest base layer also are shown. In Fig. 9 we compare our results to the ones obtained by current state-of-the-art methods FBF, GIF and WLS. As shown in the zoom-in patches, except for the FBF which leads to some halos visible around the thin twigs and the leave edges, other three methods are comparable and produce halo-free compression results. However, our method, as well as the GIF, is faster than the WLS.

V. Conclusion

We have presented a new approach for performing high-quality edge-preserving image filtering in real time. Our approach is based on a local linear model and uses the principle of Stein's unbiased risk estimate (SURE) to determine the optimal affine transform coefficients.

The proposed approach has several desirable features. First, since using a local linear model, it is simple and can be compute efficiently with arbitrary kernel sizes in constant time. Second, it has the nice property of edge-preserving smoothing which can removed noise while preserve fine details and geometrical structures in the original image. Third, it is completely automatic which doesn't need to tune manually the parameters during denoising process. Finally, it is flexible and can be widely applied in various image processing tasks, including noise reduction, detail smoothing/enhancement, HDR compression and flash/no-flash denoising. The experimental results show that our approach is robust and versatile.

For the future work, we would like to investigate more sophisticated schemes to overcome some of the shortcomings of the LLSURE filter method, for example, it may leave some noise in the vicinity of edges in denoising process. And we would also like to extend our approach to more application, including image matting, haze removal, and image colorization.

Appendix A

To derive (9), taking the first derivatives of (8) with respect to the a_i and b_i, we have

$$\begin{aligned}
&\frac{\partial}{\partial b_i}\mathrm{SURE}(a_i, b_i) = 0 \\
\Rightarrow\ &\frac{2}{N_w}\sum_{j=1}^{N_w}(a_i y_j + b_i - y_j) = 0 \\
\Rightarrow\ &N_w b_i = (1 - a_i)\sum_{j=1}^{N_w} y_j \\
\Rightarrow\ &b_i = (1 - a_i)\bar{y}_i
\end{aligned} \tag{28}$$

and

$$\begin{aligned}
&\frac{\partial}{\partial a_i}\mathrm{SURE}(a_i, b_i) = 0 \\
\Rightarrow\ &\frac{1}{N_w}\sum_{j=1}^{N_w}(a_i y_j + b_i - y_j)y_j + \sigma^2 = 0 \\
\Rightarrow\ &\frac{1}{N_w}\sum_{j=1}^{N_w}\left[(a_i - 1)y_j^2 + (1 - a_i)\bar{y}_i y_j\right] + \sigma^2 = 0 \\
\Rightarrow\ &(a_i - 1)\frac{1}{N_w}\sum_{j=1}^{N_w}(y_j - \bar{y}_i)y_j + \sigma^2 = 0 \\
&\Rightarrow (a_i - 1)\sigma_i^2 + \sigma^2 = 0 \\
&\Rightarrow a_i = (\sigma_i^2 - \sigma^2)/\sigma_i^2.
\end{aligned} \tag{29}$$

Consequently, we obtain (9).

APPENDIX B

To derive (19), taking the first derivatives of (18) with respect to the a_i and b_i, we have

$$\begin{aligned}
&\frac{\partial}{\partial b_i}\text{J-SURE}(a_i, b_i) = 0 \\
&\Rightarrow \frac{2}{N_w}\sum_{j=1}^{N_w}(a_i y_j + b_i - f_j) = 0 \\
&\Rightarrow N_w b_i = \sum_{j=1}^{N_w}(f_j - a_i y_j) \\
&\Rightarrow b_i = \bar{f}_i - a_i \bar{y}_i \qquad (30)
\end{aligned}$$

and

$$\begin{aligned}
&\frac{\partial}{\partial a_i}\text{J-SURE}(a_i, b_i) = 0 \\
&\Rightarrow \frac{1}{N_w}\sum_{j=1}^{N_w}(a_i y_j + b_i - f_j)y_j + \sigma^2 = 0 \\
&\Rightarrow \frac{1}{N_w}\sum_{j=1}^{N_w}\left[(a_i(y_j^2 - \bar{y}_i y_j) + \bar{f}_j y_j - f_i y_j\right] + \sigma^2 = 0 \\
&\Rightarrow a_i \frac{1}{N_w}\sum_{j=1}^{N_w}(y_j^2 - \bar{y}_i y_j) = \frac{1}{N_w}\sum_{j=1}^{N_w}(f_j - \bar{f}_j)y_j - \sigma^2 \\
&\Rightarrow a_i \sigma_i^2 = \frac{1}{N_w}\sum_{j=1}^{N_w}(f_j - \bar{f}_j)y_j - \sigma^2 \\
&\Rightarrow a_i = \frac{1}{\sigma_i^2}\left(\frac{1}{N_w}\sum_{j=1}^{N_w}(f_j - \bar{f}_j)y_j - \sigma^2\right). \qquad (31)
\end{aligned}$$

Consequently, we obtain (19).

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

REFERENCES

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

[2] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, Nov. 1992.

[3] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Bombay, India, Jan. 1998, pp. 839–846.

[4] V. Aurich and J. Weule, "Non-linear Gaussian filters performing edge preserving diffusion," in *Proc. DAGM Symp.*, Bielefeld, Germany, 1995, pp. 538–545.

[5] S. M. Smith and J. M. Brady, "SUSAN—A new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, May 1997.

[6] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1141–1151, Oct. 2002.

[7] C. Liu, W. T. Freeman, R. Szeliski, and S. Kang, "Noise estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2006, pp. 901–908.

[8] B. M. Oh, M. Chen, J. Dorsey, and F. Durand, "Image-based modeling and photo editing," in *Proc. Assoc. Comput. Mach. Special Interest Group Comput. Graph. Interact. Tech.*, 2001, pp. 433–442.

[9] S. Bae, S. Paris, and F. Durand, "Two-scale tone management for photographic look," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 637–645, Jul. 2006.

[10] D. DeCarlo and A. Santella, "Stylization and abstraction of photographs," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 769–776, Jul. 2002.

[11] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1221–1226, Jul. 2006.

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

[13] S. Paris and F. Durand, "A fast approximation of the bilateral filter using signal processing approach," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 568–580.

[14] F. Porikli, "Constant time O(1) bilateral filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 2008, pp. 1–8.

[15] Q. Yang, K. H. Tan, and N. Ahuja, "Real-time O(1) bilateral filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, Jun. 2009, pp. 557–564.

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

[17] R. Fattal, "Edge-avoiding wavelets and their applications," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–10, Jul. 2009.

[18] E. S. L. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–12, Jul. 2011.

[19] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 1–14.

[20] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Stat. Assoc.*, vol. 90, no. 432, pp. 1200–1224, Dec. 1995.

[21] D. Van De Ville and M. Kocher, "SURE-based non-local means," *IEEE Signal Process. Lett.*, vol. 16, no. 11, pp. 973–976, Nov. 2009.

[22] D. Van De Ville and M. Kocher, "Nonlocal means with dimensionality reduction and SURE-based parameter selection," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2683–2690, Sep. 2011.

[23] F. Luisier, T. Blu, and M. Unser, "A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 593–606, Mar. 2007.

[24] T. Blu and F. Luisier, "The SURE-LET approach to image denoising," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2778–2786, Nov. 2007.

[25] F. Luisier and T. Blu, "SURE-LET multichannel image denoising: Interscale orthonormal wavelet thresholding," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 482–492, Apr. 2008.

[26] F. Luisier, T. Blu, and M. Unser, "Image denoising in mixed poisson-Gaussian noise," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 696–708, Mar. 2011.

[27] A. Benazza-Benyahia and J.-C. Pequet, "Building robust wavelet estimators for multicomponent images using Stein's principle," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1814–1830, Nov. 2005.

[28] C. Choux, L. Duval, A. Benazza-Benyahia, and J.-C. Pequet, "A nonlinear Stein-based estimator for multichannel image denoising," *IEEE Trans. Image Process.*, vol. 56, no. 8, pp. 3855–3869, Aug. 2008.

[29] C. Stein, "Estimation of the mean of multivariate normal distribution," *Ann. Stat.*, vol. 9, no. 6, pp. 1135–1151, Nov. 1981.

[30] O. Guleryuz, "Weighted averaging for denosing with overcomplete dictionaries," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 3020–3034, Dec. 2007.

[31] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.

[32] J. Salmon and Y. Strozecki, "From patches to pixels in non-local methods: Weighted-average reprojection," in *Proc. IEEE Int. Conf. Image Process.*, Hong Kong, Sep. 2010, pp. 1929–1932.

[33] P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1635–1649, Apr. 2012.

[34] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising," *Int. J. Comput. Vis.*, vol. 86, no. 1, pp. 1–32, Jan. 2010.

[35] J. Salmon and Y. Strozecki, "Patch reprojections for non-local methods," *Signal Process.*, vol. 92, no. 2, pp. 477–489, Feb. 2012.

[36] C.-V. Deledalle, V. Duval, and J. Salmon, "Non-local methods with shape-adaptive patches (NLM-SAP)," *J. Math. Imag. Vis.*, vol. 43, no. 2, pp. 103–120, Jun. 2012.
[37] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 673–678, Aug. 2004.
[38] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 664–672, Aug. 2004.
[39] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

Tianshuang Qiu received the B.S. degree from Tianjin University, Tianjin, China, the M.S. degree from the Dalian University of Technology, Dalian, China, and the Ph.D. degree from Southeastern University, Nanjing, China, all in electrical engineering, in 1983, 1993, and 1996, respectively.

He was a Research Scientist with the Dalian Institute of Chemical Physics, Chinese Academy of Sciences, Dalian, from 1983 to 1996. He was a Staff Member with the Faculty of Electrical Engineering, Dalian Railway Institute, Dalian, China, in 1996. He was a Post-Doctoral Researcher with the Department of Electrical Engineering, Northern Illinois University, DeKalb. He is currently a Professor with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. His current research interests include non-Gaussian and nonstationary signal processing, radio frequency signal processing, and biomedical signal processing.

Aiqi Wang received the B.S. degree from Liaoning Normal University, Dalian, China, in 2002, and the M.S. degree from the Dalian University of Technology, Dalian, in 2005, where he is currently pursuing the Ph.D. degree in signal and information processing.

His current research interests include pattern recognition, image processing, and computer vision.

Nannan Yu received the B.S. and M.S. degrees from the Harbin Institute of Technology, Harbin, China, in 2003 and 2006, respectively. She is currently pursuing the Ph.D. degree in signal and information processing from the Dalian University of Technology, Dalian, China.

She is currently an Assistant Professor with Northeast Petroleum University, Daqing, China. Her current research interests include image fusion, biomedical signal processing, and sparse representation.

Aimin Song received the B.S. degree from Liaoning Normal University, Dalian, China, in 2002, and the M.S. degree from the Dalian University of Technology, Dalian, in 2005, where he is currently pursuing the Ph.D. degree in signal and information processing.

His current research interests include non-Gaussian and nonstationary signal processing, and time delay estimation theory.