Lazy sliding window implementation of the bilateral filter on parallel architectures

Michael M. Bronstein, *Member, IEEE,*

***Abstract*—Bilateral filter is one of the state-of-the-art methods for noise reduction in images. The plausible visual result the filter produces makes it a common choice for image and video processing applications, yet, its high computational complexity makes a real-time implementation a challenging task. Presented here is a parallel version of the bilateral filter using a lazy sliding window, suitable for SIMD-type architectures.**

***Index Terms*—bilateral filter, lazy sliding window, retinex, noise reduction.**

I. Introduction

Noise reduction is an important task in many image or video processing applications. Here, we understand the term "noise" in a broad sense, referring to random noise present in analog sources, digital image and video artifacts resulting from compression (high-frequency quantization and blocking), false contouring artifacts resulting from quantization, etc. Reducing noise in images, in addition to improving the visual quality, allows improving the compressibility of the image. In video sequences, noise reduction facilitates block matching in motion-adaptive video processing algorithms.

Bilateral filter is a class of method for non-linear content adaptive image processing, introduced in [1] (for an earlier work introducing the same principle in the formulation of a PDE processing referred to as *Beltrami flow* see [2], and an extension referred to as *non-local means* see [3]). The plausible visual result the filter produces, in particular, the edge preservation property, makes it a common choice in image and video processing applications. In addition to direct applications related to noise reduction, bilateral filter is the core of many adaptive dynamic range extension (*retinex*) algorithms [4], [5].

For a grayscale image $I_{k,j}$, $k = 1,...,N; j = 1,...,M$ with intensity levels in the range $[0, 255]$, the filter computes each output pixel $\hat{I}_{k,j}$ as a weighted average of its neighbors in the window around $I_{k,j}$,

$$\hat{I}_{k,j} = \frac{1}{w_{k,j}} \sum_{m=-P}^{P} \sum_{n=-P}^{P} w_{k,j,m,n} I_{k-m,j-n}. \tag{1}$$

The weights are inversely related to the spatial and radiometric distance between the pixels,

$$w_{k,j,m,n} = e^{-\frac{m^2+n^2}{2\sigma_s^2}} e^{-\frac{(I_{k,j}-I_{k-m,j-n})^2}{2\sigma_r^2}},$$

and

$$w_{k,j} = \sum_{m=-P}^{P} \sum_{n=-P}^{P} w_{k,j,m,n}$$

is a normalization guaranteeing that all the filter coefficients add up to one. σ_s and σ_r are the spatial and radiometric variance parameters governing the filter behavior (roughly, the larger the variance, the smoother is the result). The window size is $(2P + 1) \times (2P + 1)$ pixels, with P typically varying between 5 to 15.

Straightforward computation of the bilateral filter (1) is performed using a sliding window. For each pixel in the raster scan order, neighbor pixels within a window around it are taken and used to compute the filter output; the window is moved right by one pixel and so on (Figure 1, left).

Due to the large amount of computations in such an approach, a real-time implementation of the filter, especially in high-definition images and videos is extremely challenging. Several accelerations have been proposed for the bilateral filter. Durand *et al.* [6], [7] showed an approximation by a sum of linear filters. Pham and van Vliet [8] showed a separable version of the bilateral filter. Weiss [9] proposed a method based on an efficient histogram computation.

This paper deals with an efficient implementation of the bilateral filter on parallel architectures of digital signal processors. The fact that the bilateral filter applies the same processing at every pixel makes it especially suitable for SIMD (single instruction multiple data) type processors, such as many modern DSPs and multimedia extensions in many general purpose CPUs (e.g., Intel SSE). We propose a special type of raster scan referred to as the *lazy sliding window*, which allows performing bilateral filtering in a manner efficient both in storage and the number of computations.

The rest of the paper is organized as follows. Section 2 describes the Durand-Dorsey acceleration. In Section III, we describe an efficient parallel implementation of the Durand-Dorsey scheme using the lazy sliding window approach. Section IV shows simulation results. Section V discusses possible extensions and applications, and Section VI concludes the paper.

II. Durand-Dorsey acceleration

If the intensity $I_{i,j}$ is assumed a constant c, the non-linear bilateral filter (1) can be expressed as a linear filter,

$$\begin{aligned}\hat{I}^c_{k,j} &= \frac{1}{w_{k,j}} \sum_{m=-P}^{P} \sum_{n=-P}^{P} e^{-\frac{m^2+n^2}{2\sigma_s^2}} e^{-\frac{(c-I_{k-m,j-n})^2}{2\sigma_r^2}} I_{k-m,j-n} \\ &= \frac{1}{w_{k,j}} \sum_{m=-P}^{P} \sum_{n=-P}^{P} e^{-\frac{m^2+n^2}{2\sigma_s^2}} g^c_{k-m,j-n}, \end{aligned} \tag{2}$$

where $g^c_{i,j} = e^{-\frac{(c-I_{i,j})^2}{2\sigma_r^2}} I_{i,j}$ is computed by applying a Gaussian non-linearity and $w_{k,j}$ are normalization factors defined by

$$w_{k,j} = \sum_{m=-P}^{P} \sum_{n=-P}^{P} e^{-\frac{m^2+n^2}{2\sigma_s^2}} e^{-\frac{(c-I_{k-m,j-n})^2}{2\sigma_r^2}}.$$

Using this observation, Durand and Dorsey [6] proposed an approximation to the bilateral filter by dividing the dynamic range of the image intensity levels into $L+1$ equal bands using a linear filter approximation (2) in each band. The results are then merged in the

M. M. Bronstein is with the Institute of Computational Science, Faculty of Informatics, Università della Svizzera Italiana, via G. Buffi 3, Lugano 6904, Switzerland. e-mail: michael.bronstein@usi.ch

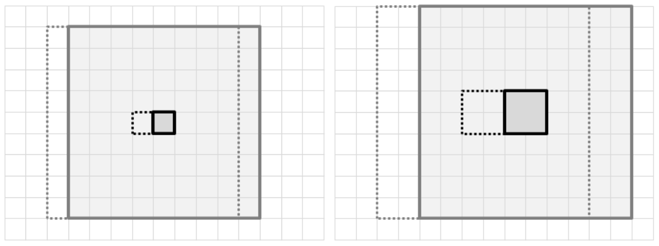

Fig. 1. Filtering using sliding window (left) and lazy sliding window (right). Shown in solid and dotted lines are two subsequent positions of the window.

following way:

$$\hat{I}_{i,j} \approx \sum_{l=0}^{L} \mu_{i,j} \cdot \hat{I}_{i,j}^{l\Delta},$$

where $\Delta = 255L^{-1}$ is the intensity band width and $\mu_{i,j} = \max\{1-\Delta^{-1}|I_{i,j} - l\Delta|, 0\}$ is a soft mask (triangular window) determining whether the (i,j)-th pixel belongs to the l-th band.

The Durand-Dorsey approach offers a significant acceleration compared to the straightforward bilateral filter computation. The number of intensity levels L (typically, between 10 and 50) is a parameter for performance-quality tradeoff. Furthermore, since for a typical selection of the window size P and spatial variance σ_s, the spatial part of the filter is approximately constant in the window, the Gaussian blur can be replaced by simple averaging [9]. Finally, a further acceleration can be achieved by first reducing the resolution of $I_{i,j}$ and then performing the averaging; in this case, a smaller window is needed. The merging is performed after upscaling the results to the original resolution.

The linear filters in each band can be computed efficiently using convolution operations on parallel architectures. The disadvantage is, however, that merging the bands (involving a non-linear operation) requires the storage of all the convolution results. A sliding window approach computing the filter result at each pixel in raster-scan order is advantageous in the sense that it does not require significant storage, but does not exploit the main advantage of SIMD architectures, which is the ability to compute several results (filtered pixels) at once. In the following section, we address this problem.

III. Lazy sliding window

In order to take advantage of the SIMD parallelism, we introduce the *lazy sliding window* approach. Instead of looking on one central pixel and a window around it, we have a block of $K \times K$ pixels (*central block*) and slide the window in increments of K. The window is the same for all the pixels in the central block (Figure 1, right). The window size is $K(2P+1) \times K(2P+1)$, where P now denotes the window size in blocks of size $K \times K$ (for simplicity, we assume hereinafter that M, N are divisible by K). Moving the window horizontally in a row requires the load of only $2P+1$ next vertically adjacent blocks of size $K \times K$. When P is sufficiently large, the lazy sliding window approach is (asymptotically) identical to the traditional sliding window.

The bilateral filter with the Durand-Dorsey acceleration can be implemented using this approach as shown in Algorithm 1. The original Durand-Dorsey algorithm is a particular case of $K = 1$. Note that Stages 3–7 are mostly element-wise operations with $K \times K$ matrices $\mathbf{I}^{i,j}, \mathbf{G}^{i,j}$, and $\mathbf{M}^{i,j}$ and can be carried out efficiently on a SIMD-type digital signal processor capable of performing K^2 single-instruction multiple-data operations. Furthermore, with an addition of a minimum (VMIN) and maximum (VMAX) operation on the block pixels, Steps 3–7 can be skipped for values of l for blocks in which $\max_{m,n}\{\mathbf{I}_{m,n}\} < (l-1)\Delta$ and $\max_{m,n}\{\mathbf{I}_{m,n}\} > (l+1)\Delta$.

```
Input: parameters $K, P, L$ and $\sigma_r$; input image $I$ divided into
       $K \times K$ blocks, $\mathbf{I}^{i,j}$, $i = 1, ..., M/K$, $j = 1, ..., N/K$.
Output: filtered image $\hat{I}$.
1 for $l = 0, ..., L$ do
2     for every block $\mathbf{I}^{i,j}$ in raster scan order do
3         Compute the non-linearity for the neighbor blocks in a
          pixel-wise manner,
          $\mathbf{G}_{m,n}^{i+k,j+q} = e^{-\frac{\left(l\Delta - \mathbf{I}_{m,n}^{i+k,j+q}\right)^2}{2\sigma_r^2}}$,  $k, q = -P, ..., +P$.
4         Weight $h = \sum_{m,n=1}^{K} \sum_{k,q=-P}^{P} \mathbf{G}_{m,n}^{i+k,j+q} \cdot \mathbf{I}_{m,n}^{i+k,j+q}$.
5         Normalization $w = \sum_{m,n=1}^{K} \sum_{k,q=-P}^{P} \mathbf{G}_{m,n}^{i+k,j+q}$.
6         Mask $\mathbf{M}_{m,n}^{i,j} = \max\{1 - \Delta^{-1}|\mathbf{I}_{m,n}^{i,j} - l\Delta|, 0\}$.
7         Merge $\hat{\mathbf{I}}_{m,n}^{i,j} \leftarrow \hat{\mathbf{I}}_{m,n}^{i,j} + \mathbf{M}_{m,n}^{i,j} h/w$.
8     end
9 end
```

Algorithm 1: Lazy sliding window bilateral filter

A. Pipelining

Since the lazy sliding window moves at each step by one block, the computations of Stages 3–6 from previous steps can be reused. If we store the previous values of $\mathbf{G}^{i,j}$, we have to perform Stage 3 for the new $2P+1$ blocks only. If in addition we store the sums $\sum_{m,n=1}^{K} \sum_{k=-P}^{P} \sum_{q=-P}^{P-1} \mathbf{G}_{m,n}^{i+k,j+q}$ and $\sum_{m,n=1}^{K} \sum_{k=-P}^{P} \sum_{q=-P}^{P-1} \mathbf{G}_{m,n}^{i+k,j+q} \cdot \mathbf{I}_{m,n}^{i+k,j+q}$ on Stages 5 and 4, respectively, after sliding the window, we have to add only the contributions $\sum_{m,n=1}^{K} \sum_{k=-P}^{P} \mathbf{G}_{m,n}^{i+k,j+P}$ and $\sum_{m,n=1}^{K} \sum_{k=-P}^{P} \mathbf{G}_{m,n}^{i+k,j+P} \cdot \mathbf{I}_{m,n}^{i+k,j+P}$ of the new blocks. This allows efficient pipelining.

In a pipelined version of the lazy sliding window bilateral filter, four buffers for storing $(2P+1) \times (2P+1)$ blocks of $\mathbf{I}^{i,j}$ and computation results $\sum_{m,n=1}^{K} \mathbf{G}_{m,n}^{i,j}$, $\sum_{m,n=1}^{K} \mathbf{G}_{m,n}^{i,j} \cdot \mathbf{I}_{m,n}^{i,j}$, and $\mathbf{M}^{i,j}$ are kept, denoted by $\mathbf{BI}^{k,q}$, $BW^{k,q}$, $BH^{k,q}$ and $\mathbf{BM}^{k,q}$ $(k, q = -P, ..., P)$, respectively. The total size of the buffers is $(2P+1)^2 \times (2K^2+2)$ pixels. Sliding the window shifts the buffers to the left, adding new values to the right (Algorithm 2).

B. Complexity analysis

For the purpose of complexity analysis, we assume that a SIMD processor is capable of performing single instruction on a register containing values of K^2 pixels (or alternatively, a $K \times K$ matrix of pixels) at once. The stages of Algorithm 2 have the following complexity:

- Stage 4 Load: $C_{LOAD} \times (2P+1)$ in the steady state.
- Stage 5 (element-wise nonlinearity): $C_{NL} \times (2P+1)$.
- Stage 6 (normalization): summation of pixels in each block: $C_{VADD} \times (2P+1)$.
- Stage 7 (filter): element-wise multiplication: $C_{MUL} \times (2P+1)$, summation of pixels in each block: $C_{VADD} \times (2P+1)$.
- Stage 8 (mask): element-wise maximum: $C_{MAX} \times (2P+1)$.
- Stage 9 (merge): element-wise multiplication: C_{MUL}, element-wise addition: C_{ADD}, element-wise division: C_{DIV}.

The outcome are K^2 filtered pixels and the total complexity is $C = L \times (C_{LOAD} \times (2P+1) + C_{NL} \times (2P+1) + C_{MUL} \times (2P+$

```
1  for $l = 0, ..., L$ do
2      for every block $\mathbf{I}^{i,j}$ in raster scan order do
3          Shift the buffers to left $\mathbf{BI}^{k,q-1} \leftarrow \mathbf{BI}^{k,q}$,
           $\mathbf{BM}^{k,q-1} \leftarrow \mathbf{BM}^{k,q}$, $BW^{k,q-1} \leftarrow BW^{k,q}$, and
           $BH^{k,q-1} \leftarrow BH^{k,q}$ for $k = -P, ..., P$.
4          Load $2P+1$ image blocks to rightmost part of the
           buffer $\mathbf{BI}^{k,P} \leftarrow \mathbf{I}^{i+k,j+P}$; $k = -P, ..., P$.
5          Compute the non-linearity
           $\mathbf{G}^k_{m,n} = e^{-\frac{\left(l\Delta - \mathbf{BI}^{k,P}_{m,n}\right)^2}{2\sigma_r^2}}$; $k = -P, ..., P$.
6          Buffer $BW^{k,P} = \sum_{m,n=1}^{K} \mathbf{G}^k_{m,n}$; $k = -P, ..., P$.
7          Buffer $BH^{k,P} = \sum_{m,n=1}^{K} \mathbf{G}^k_{m,n}\mathbf{BI}^{k,P}_{m,n}$;
           $k = -P, ..., P$.
8          Buffer $\mathbf{BM}^{k,P}_{m,n} = \max\{1 - \Delta^{-1}|\mathbf{BI}^{i,P}_{m,n} - l\Delta|, 0\}$.
9          Merge $\hat{\mathbf{I}}^{i,j}_{m,n} \leftarrow \hat{\mathbf{I}}^{i,j}_{m,n} + \mathbf{BM}^{0,0}_{m,n} BH^{0,0}/BW^{0,0}$.
10     end
11 end
```

Algorithm 2: Pipelined lazy sliding window bilateral filter (for simplicity, boundary effects are ignored)

	Durand-Dorsey	Lazy sliding window ($K =$) 2	3	4	6	Non-linear
LOAD	46.69	5.98	1.63	0.78	0.22	41
NL	46.69	5.98	1.63	0.78	0.22	1681
MUL	47.69	6.23	1.74	0.85	0.25	1681
VADD	93.34	11.96	3.25	1.57	0.44	3362
MAX	46.69	5.98	1.63	0.78	0.22	-
DIV	1	0.25	0.11	0.06	0.03	1
ADD	1	0.25	0.11	0.06	0.03	-
Total	283.17	36.63	10.08	4.86	1.40	6766

TABLE I
NUMBER OF $K \times K$ SIMD INSTRUCTIONS PER PIXEL PER BAND, ASSUMING IMAGE OF SIZE 288×288, WINDOW OF APPROXIMATELY THE SAME SIZE 40×40, AND ACCOUNTING FOR BOUNDARY EFFECTS. FOR COMPARISON, THE COMPLEXITY OF NON-LINEAR BILATERAL FILTER (WITH CONSTANT SPATIAL COMPONENT, $\sigma_s \gg 1$) IS SHOWN.

$2) + 2C_{VADD} \times (2P+1) + C_{MAX} \times (2P+1) + C_{ADD} + C_{DIV})$ for each position of the window, excluding boundary effects. The window is slided $M/K \times N/K$ times, which gives the total complexity of C/K^2 per pixel. Most of the operations, excluding summation of vector elements on stages 6 and 7, are vertical SIMD operations.

Note that in order to obtain approximately the same window size ($K(2P+1) \times K(2P+1)$), we need a smaller P for larger values of K. Thus, the performance gain in the proposed approach is two-fold: first, from performing SIMD operations in parallel (giving a K^2 boost) and second, from reducing the overhead by using bigger blocks and thus having smaller P, which is similar to the gain from downsampling proposed in [7] (see Table I). The latter results in a significant complexity gain per se, which can be observed if we normalize the results in Table I by K^2 in order to compare the number of pixel-wise instructions: the gain from using our approach compared to Durand-Dorsey ranges from 1.9 ($K = 2$) to 5.6 ($K = 6$). It should be noted that the value of K depends on the size of the SIMD register, and thus the implementation performance is architecture-dependent.

IV. RESULTS

To test our approach, we compared different approximations of the bilateral filter on several test images. [1] In the first experiment, we compared the original non-linear bilateral filter (1) with the Gaussian filter replaced by averaging (corresponding to the case $\sigma_s \gg 1$), the Durand-Dorsey implementation, and the pipelined lazy sliding window on the problem of denoising the 288×288 Lena image contaminated by mild additive Gaussian noise (Figure 2, first column). In all the compared methods, window of approximately the same size (41×41) and $\sigma_r = 10$ were used. In the Durand-Dorsey and our method, we used $L = 10$; lazy sliding window was of sizes $K = 2, 3, 4$, and 6 with corresponding $P = 10, 6, 5$, and 3.

Figure 2 compares the outcome of different approaches, which are visual nearly identical. Table II compares the complexity and RMS/maximum error w.r.t. the non-linear bilateral filter. The proposed method offers significant speedup compared to the Durand-Dorsey approach, while being almost equally good approximation of the bilateral filter in terms of RMS error and even better in terms of maximum error.

	Durand-Dorsey	Lazy sliding window ($K =$) 2	3	4	6	Non-linear
Complexity	2831.7	366.3	100.8	48.86	14	6766
RMS error	0.81	1.05	1.04	1.09	1.09	-
Max error	18.64	11.4	11.21	11.52	11.78	-

TABLE II
COMPARISON OF DIFFERENT APPROXIMATIONS OF THE BILATERAL FILTER PERFORMED ON LENA IMAGE OF SIZE 256×256, USING WINDOW OF APPROXIMATELY THE SAME SIZE (41×41), $\sigma_r = 10$, $L = 10$, AND ACCOUNTING FOR BOUNDARY EFFECTS. COMPLEXITY IS GIVEN IN TOTAL NUMBER OF $K \times K$ SIMD INSTRUCTIONS PER PIXEL. ERROR IS W.R.T. THE NON-LINEAR BILATERAL FILTER.

Figure 3 show the influence of the number of bands L. Slight artifacts are visible in the form of artificial contours for small L (3 and 5). We found empirically that L between 10 and 20 provides plausible results and low error with respect to the non-linear bilateral filter, while keeping the complexity (which is proportional to L) reasonable.

In the second experiment, we compared different settings of the proposed method on the problem of denoising a 256×256 peppers image. Denoising quality was measured as peak signal-to-noise-ratio (PSNR) w.r.t. the clean image. Figure 4 shows denoising results by the proposed approach with different settings on the peppers image contaminated with different levels of noise.

V. EXTENSIONS AND APPLICATIONS

A. Color images

So far, we assumed I to be a grayscale image. In order to filter color images, the proposed scheme can be straightforwardly applied to each of the color channels individually. This increases the complexity by the factor of 3. A fast approximation is possible by applying the filter only to the achromatic channel of a Lab or YUV colospace, typically containing most information about the edges in the image. Figure 5 shows the result of bilateral filtering on an image contaminated by color noise in all channels. The filtering is performed in the YUV colorspace on each channel separately. Different values of σ_r can be used for each channel depending on the noise characteristics.

B. Dynamic range extension

In the problem of dynamic range extension, the image is decomposed into reflectance and illumination components, $I_{k,j} = R_{k,j} \cdot L_{k,j}$ ($L_{k,j} \geq I_{k,j}$ [4]), and then reassembled as $\tilde{I}_{k,j} = R_{k,j} \cdot L_{k,j}^{1/\gamma} = I_{k,j} \cdot L_{k,j}^{1/\gamma - 1}$, where the parameter γ controls the dynamic range enhancement.

[1] Code will be available at `www.inf.usi.ch/bronstein`

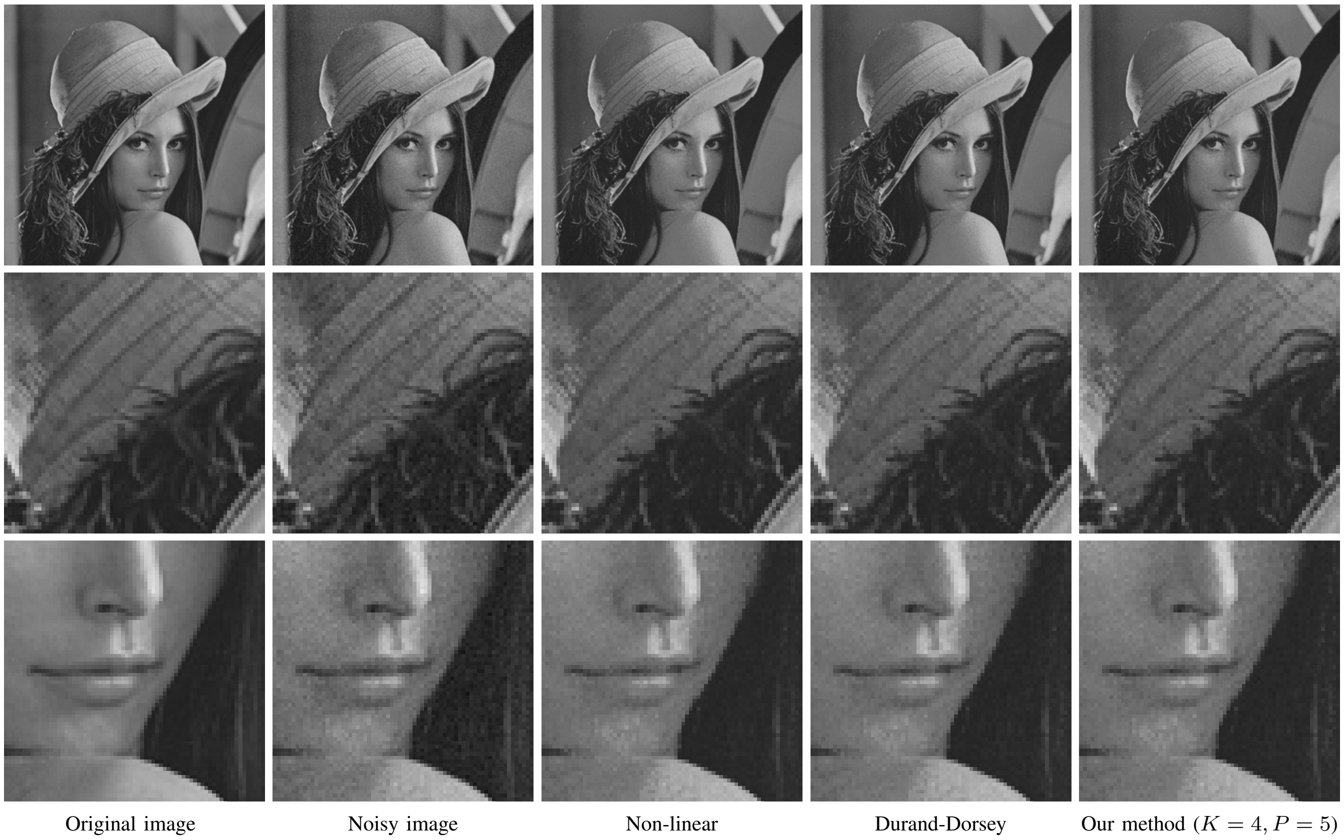

Fig. 2. First row: Filtering the Lena image with Gaussian synthetic noise of standard deviation 5. In all filters, $\sigma_r = 10$ and window of approximately 41×41 pixels was used. In Durand-Dorsey and lazy sliding window, $L = 10$ was used. Second row: texture details. Third row: geometry detail.

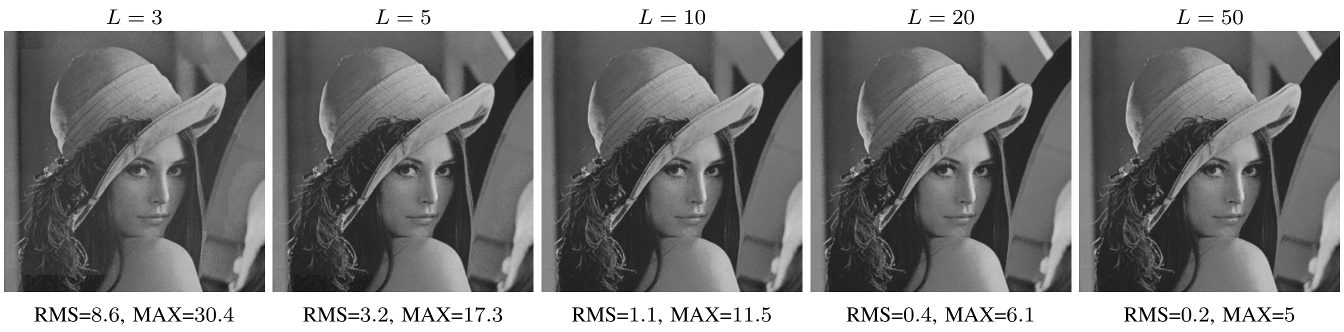

Fig. 3. Influence of the number of bands L in lazy sliding window approach on the approximation quality. Parameters $K = 4, P = 5$, and $\sigma_r = 10$ were used. The error is w.r.t. the non-linear bilateral filter.

Elad [5] proposed estimating these components in the logarithmic domain $\tilde{I}_{k,j} = \tilde{R}_{k,j} + \tilde{L}_{k,j} = \log(R_{k,j}) + \log(L_{k,j})$ using a pair of bilateral filters. With this approach, $\tilde{L}$ is estimated by applying the *enveloped* version of the bilateral filter to $\tilde{I}_{k,j} = \log(I_{k,j})$,

$$\tilde{L}_{k,j} = \frac{1}{w_{k,j}} \sum_{m=-P}^{P} \sum_{n=-P}^{P} w_{k,j,m,n} \tilde{I}_{k-m,j-n},$$

where

$$\begin{aligned} w_{k,j,m,n} &= e^{-\frac{m^2+n^2}{2\sigma_s^2}} e^{-\frac{(I_{k,j}-\tilde{I}_{k-m,j-n})^2}{2\sigma_r^2}} \theta(\tilde{I}_{k-m,j-n} - \tilde{I}_{k,j}), \\ w_{k,j} &= \sum_{m=-P}^{P} \sum_{n=-P}^{P} w_{k,j,m,n}, \end{aligned}$$

and $\theta(x) = 1$ if $x > 0$ and zero otherwise (the step function $\theta(\tilde{I}_{k-m,j-n} - \tilde{I}_{k,j})$ ensures that $L_{k,j} \geq I_{k,j}$ due to monotonicity of the log). $\tilde{R}$ is then estimated by applying the standard bilateral filter to $\tilde{I} - \tilde{L}$. Both filter results provide the reflectance and illumination components $R_{k,j} = e^{\tilde{R}_{k,j}}$ and $L_{k,j} = e^{\tilde{L}_{k,j}}$, from which the new image is composed (for additional details and derivations, the reader is referred to [5]).

Our approach can be straightforwardly adapted to be used in this problem. The enveloped version of the bilateral filter is computed as in Algorithm 2 with two modifications. First, Step 5 has a different nonlinearity now including the step function,

$$\mathbf{G}^k_{m,n} = e^{-\frac{\left(l\Delta - \mathbf{BI}^{k,P}_{m,n}\right)^2}{2\sigma_r^2}} \theta(l\Delta - \mathbf{BI}^{k,P}_{m,n}).$$

Second, the mask computation in Step 8 has one-sided interpolation, $\mathbf{BM}^{k,P}_{m,n} = \theta(\mathbf{BI}^{i,P}_{m,n} - l\Delta)\theta((l+1)\Delta - \mathbf{BI}^{i,P}_{m,n})$, which ensures that the constraint $L_{k,j} \geq I_{k,j}$ is satisfied. Figure 6 shows an example of dynamic range extension by means of two bilateral filters

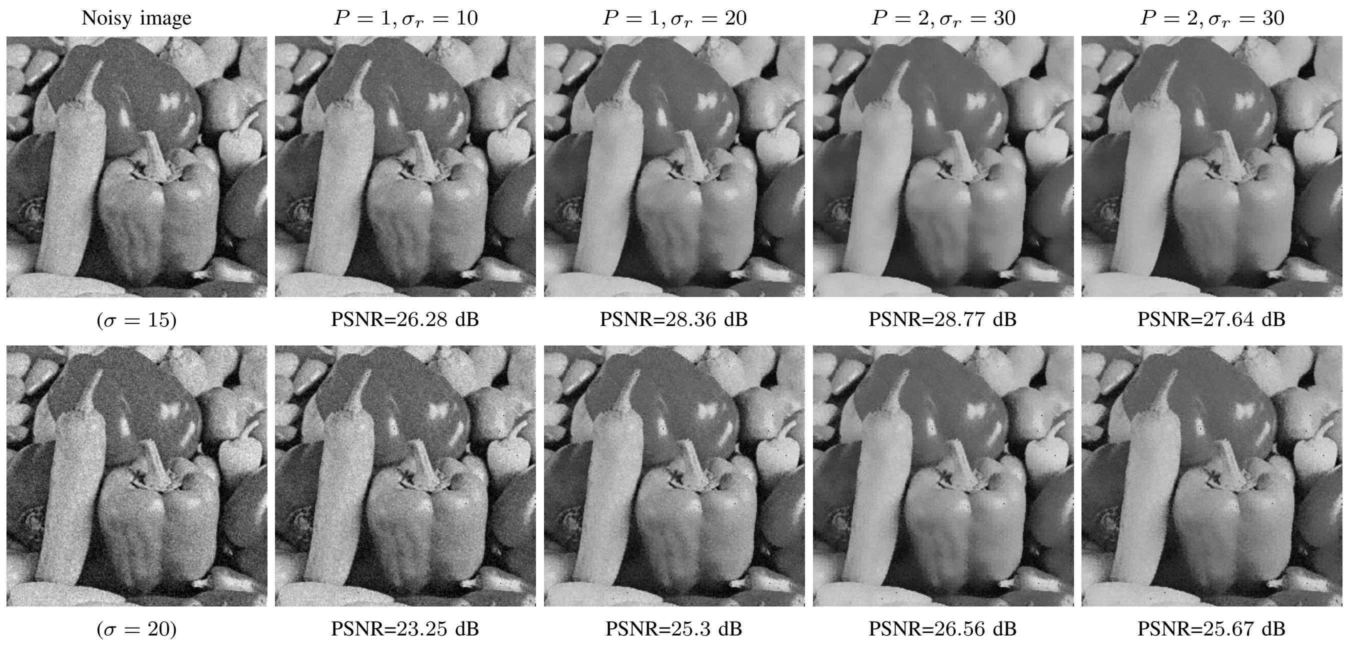

Fig. 4. Denoising using the lazy sliding window approach with different settings. $K = 4, L = 20$ were fixed; σ_r and P determining the spatial and radiometric variance of the filter varied. Denoising quality is given by PSNR w.r.t. the original image.

implemented using the proposed method.

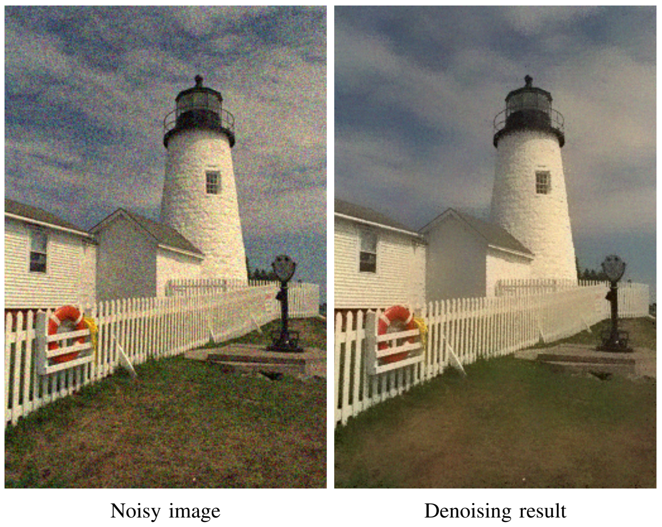

Fig. 5. Denoising of a color image using the proposed approach with $K = 4, P = 2, \sigma_r = 20$, and $L = 10$ (best viewed in color).

VI. Conclusions

We presented an efficient implementation of the bilateral filter on parallel SIMD-type architectures. Our implementation can be advantageous in applications requiring real-time performance of filters in large resolution images, such as high-definition video processing. The bilateral filter can also be used for color image filtering and for adaptive dynamic range extension.

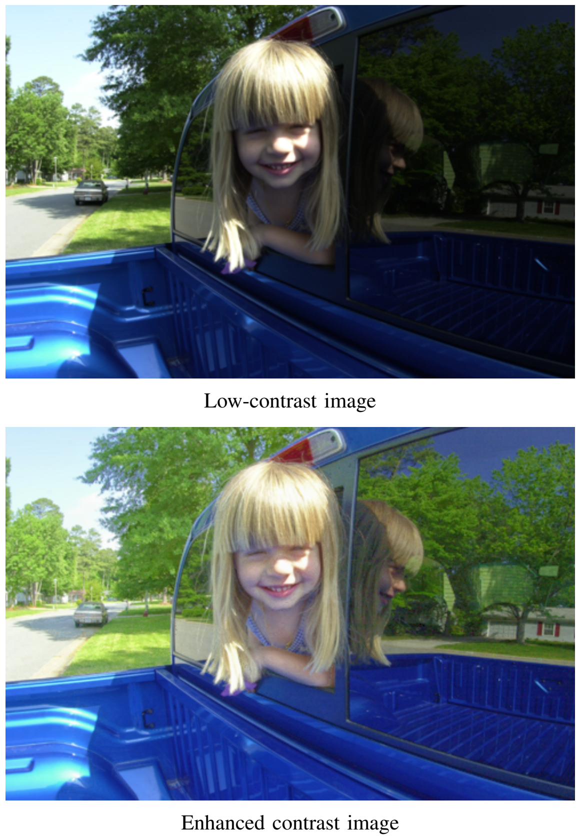

Fig. 6. Dynamic range extension using the approach of Elad [5]. Parameters $K = 4, P = 3, \sigma_r = 0.3, L = 28$ are used for the first (enveloped) filter; and $K = 4, P = 1, \sigma_r = 0.3, L = 14$ for the second (standard) filter; $\gamma = 3$ (best viewed in color).

References

[1] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. ICCV*, pages 839–846, 1998.

[2] N. Sochen, R. Kimmel, and R. Malladi. Geometrical framework for low level vision. *IEEE Trans. on Image Processing*, 7(3):310–318, 1998.

[3] A. Buades, B. Coll, and J.M. Morel. A non-local algorithm for image denoising. In *Proc. CVPR*, 2005.

[4] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel. A variational

framework for retinex. *IJCV*, 51(1):7–23, 2003.
[5] M. Elad. Retinex by two bilateral filters. In *Proc. Scale Space*, pages 217–229, 2005.
[6] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high dynamic-range images. In *Proc. SIGGRAPH*, pages 257–266, 2002.
[7] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *IJCV*, 81(1):24–52, 2009.
[8] T. Q. Pham and L. J. van Vliet. Separable bilateral filtering for fast video preprocessing. In *Proc. ICME*, 2005.
[9] B. Weiss. Fast median and bilateral filtering. In *Proc. SIGGRAPH*, 2006.

Michael M. Bronstein (M'02–) is an assistant professor in the Institute of Computational Science at the Faculty of Informatics, Università della Svizzera Italiana (USI), Lugano, Switzerland. Previously to joining USI, he held visiting appointments at Politecnico di Milano (2008), Stanford university (2009), INRIA (2009), and the University of Verona (2010). His main research interests are theoretical and computational methods in metric geometry and their application to problems in computer vision, pattern recognition, shape analysis, computer graphics, image processing, and machine learning. Michael Bronstein received the B.Sc. *summa cum laude* (2002) from the Department of Electrical Engineering and Ph.D. with distinction (2007) from the Department of Computer Science, Technion – Israel Institute of Technology. He has authored over 60 publications in leading journals and conferences, over a dozen of patents and the book "Numerical geometry of non-rigid shapes" (published by Springer Verlag). His research was recognized by numerous awards and was featured in CNN, SIAM News, and Wired. Prof. Bronstein was the co-chair of the Workshop on Non-rigid shapes and deformable image alignment (NORDIA) in 2008-2010 and has served on review and program committees of major conferences in computer vision and pattern recognition. In addition to academic activities, in 2004-2009 he served as a scientist and Vice President of video technology at the Silicon Valley start-up company Novafora Inc, leading a team of researchers and engineers developing Internet-scale computer vision and video analysis applications. Technology developed by Prof. Bronstein has been used in the foundation of several start-up companies, in which he has been involved as advisor and co-founder.