

# Bi-Exponential Edge-Preserving Smoother

Philippe Thévenaz, Daniel Sage, and Michael Unser, *Fellow, IEEE*

**Abstract**—Edge-preserving smoothers need not be taxed by a severe computational cost. We present, in this paper, a lean algorithm that is inspired by the bi-exponential filter and preserves its structure—a pair of one-tap recursions. By a careful but simple local adaptation of the filter weights to the data, we are able to design an edge-preserving smoother that has a very low memory and computational footprint while requiring a trivial coding effort. We demonstrate that our filter (a bi-exponential edge-preserving smoother, or BEEPS) has formal links with the traditional bilateral filter. On a practical side, we observe that the BEEPS also produces images that are similar to those that would result from the bilateral filter, but at a much-reduced computational cost. The cost per pixel is constant and depends neither on the data nor on the filter parameters, not even on the degree of smoothing.

**Index Terms**—Bi-exponential filter, bilateral filter, nonlocal means, recursive filter.

## I. INTRODUCTION

**S**MOOTHING an image can serve many purposes. One of them is to remove its confounding details, with the goal of bringing out its main features. To do so, early works on image processing were concentrating on the straightforward theory of space-invariant linear smoothers. Unfortunately, such filters perform an indiscriminate blurring, whereas maintaining the sharpness of the main edges would instead be desirable. Thus, nonlinear filters were soon designed that preserved edges while still smoothing out uninformative details. Some of them, like the median filter, do not adapt to the data. Others, like the bilateral filter first presented in [1] and later attributed to [2], or the anisotropic-diffusion filter [3], do. It has been shown in [4] how those two filters, along with additional denoising approaches, such as weighted least squares and robust estimation, can be understood and bridged within a Bayesian framework. A precursor to the bilateral filter was described in [5], while a modern in-depth analysis is available from [6]. Moreover, the bilateral filter is also related to the nonlocal means introduced in [7] and accelerated in [8].

Nowadays, wavelets are often favored when it comes to denoising data. Notwithstanding, the bilateral filter remains attractive for data-simplification tasks, even though it is computationally heavy in its original form. Consequently, accelerating this filter has attracted the attention of researchers.

Manuscript received November 11, 2011; revised May 10, 2012; accepted May 12, 2012. Date of publication May 22, 2012; date of current version August 22, 2012. This work was supported in part by the Center for Biomedical Imaging of the Geneva-Lausanne Universities, the École polytechnique fédérale de Lausanne, and by the Leenaards and Louis-Jeannet Foundations. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Debargha Mukherjee.

The authors are with Biomedical Imaging Group, École polytechnique fédérale de Lausanne, Lausanne VD CH-1015, Switzerland (e-mail: philippe.thevenaz@epfl.ch; daniel.sage@epfl.ch; michael.unser@epfl.ch).

Digital Object Identifier 10.1109/TIP.2012.2200903

For instance, it was proposed in [9] that speedups could be obtained by relying on a strict quantization of data on finitely many intensity levels, in which case the bilateral filter can be implemented by the appropriate combination of a series of convolutions. This method was used for instance in [10] to homogenize the illumination of an image by compressing its dynamic range, and further refined in [11]–[13]. Substantial acceleration was also achieved in [14], again at the cost of some form of quantization. An alternative approach to accelerate the bilateral filter was proposed in [15], where the combination of a convolution in a higher-dimensional space with two simple nonlinearities was down-sampled to achieve the desired speedup. Performing fast adaptive filtering in a space-intensity manifold was similarly proposed in [16], where the strength of 1-D edges is determined by measuring the curvilinear length of the interpolated curve joining two data samples. Achieving constant-time processing was also proposed in [17] by refraining to actually operate directly on the data, relying instead on a learning approach.

The bilateral filter contains two essential ingredients: a range filter  $r$  and a spatial filter  $s$ . Some of the acceleration strategies cited above succeed by focusing on the range filter. This is also the case with the method proposed by some of us in [18] where, instead of taking advantage of quantization, it is shown that a trigonometric form of  $r$  can lead to time savings; meanwhile,  $s$  is left unconstrained. One of the several acceleration methods proposed in [13] follows a similar approach, by considering a polynomial form for  $r$ . In this paper, however, we take the opposite view and remove every restriction on  $r$ ; in return, we constrain  $s$  to take the form of a bi-exponential filter. This allows for the design of a substitute to the 1-D bilateral filter that runs as a pair of one-tap recursive filters. Such a construction is therefore very fast. A major contribution of this paper is to show that, under appropriate hypotheses, our filter has formal relations with the bilateral filter.

This paper is organized as follows. We initiate Section II with a forthright exposition of the bi-exponential edge-preserving smoother (BEEPS), which we then discuss in relation to the traditional bi-exponential filter at the core of our new method. We devote Section III to the theoretical relation between the BEEPS and the bilateral filter. The practical outcome of our proposal and the discussion of its computational load are provided in Section IV, along with an extensive experimental qualitative and quantitative comparison between the BEEPS and the bilateral filter. We give, in Section V, a glimpse on how to use the BEEPS in conditions that are unaccessible to the bilateral filter, and we conclude in Section VI.

## II. PROPOSED ALGORITHM

### A. Description

Let  $r : \mathbb{R}^2 \mapsto [0, 1]$  be an arbitrary bivariate function that measures the similarity of its arguments, taken to be a pair of image intensities. Like in the case of the bilateral filter, we shall call  $r$  a *range filter*. Moreover, let the parameter  $\lambda \in [0, 1]$  control the degree of smoothing of a convolutional *space filter* with impulse response  $s$ . The algorithm for our BEEPS then essentially consists of a pair of one-tap recursions.

The first recursion is progressive. Letting  $x[k]$  be the current sample of an input sequence  $x$  at location  $k \in \mathbb{Z}$ , we recursively compute the elements of an auxiliary sequence  $\varphi$  as

$$\varphi[k] = (1 - \varrho[k] \lambda) x[k] + \varrho[k] \lambda \varphi[k - 1] \quad (1)$$

where

$$\varrho[k] = r(x[k], \varphi[k - 1]). \quad (2)$$

The second recursion is regressive and very similar to the first one, except for a reversal of the order in which the indices are traversed. We recursively compute a second auxiliary sequence  $\phi$  as

$$\phi[k] = (1 - \rho[k] \lambda) x[k] + \rho[k] \lambda \phi[k + 1] \quad (3)$$

where

$$\rho[k] = r(x[k], \phi[k + 1]). \quad (4)$$

We complete our algorithm by merging the resulting progressive sequence  $\varphi$  and regressive sequence  $\phi$  to produce the samples of the output sequence  $y$  as

$$y[k] = \frac{\varphi[k] - (1 - \lambda) x[k] + \phi[k]}{1 + \lambda}. \quad (5)$$

### B. Initialization of the Recursions

The algorithm of Section II-A requires a small, finite number of operations per pixel. In return, the forward and backward recursions must be properly initialized. In practice, we often know the sequence  $x$  only through  $K$  samples indexed by  $k \in [0, \dots, K - 1]$ . Thus, we now face the task of choosing  $\varphi[0]$  and  $\phi[K - 1]$ , which depend on the unknown values  $x[-1]$  and  $x[K]$ , respectively. We propose the trivial choice

$$\varphi[0] = x[0] \quad (6)$$

$$\phi[K - 1] = x[K - 1] \quad (7)$$

which offers the convenience of speed and simplicity.

### C. Synthetic Whirl

To provide a preliminary illustration of the effect of the BEEPS, we suggest to apply it to a synthetic image  $w$  that contains a variety of frequencies and contrasts, to which we have applied the BEEPS with  $\lambda = 0.9$  and  $s(u, v) = \exp(-(u - v)^2/(2\sigma^2))$ , where  $\sigma = 50$ . This resulted in the image shown at the bottom of Fig. 1. There, we see that the features of  $w$  that had a low contrast (bottom-half) have been smoothed away, unless their frequency content was low. All the same, the features of  $w$  that had a high-frequency content

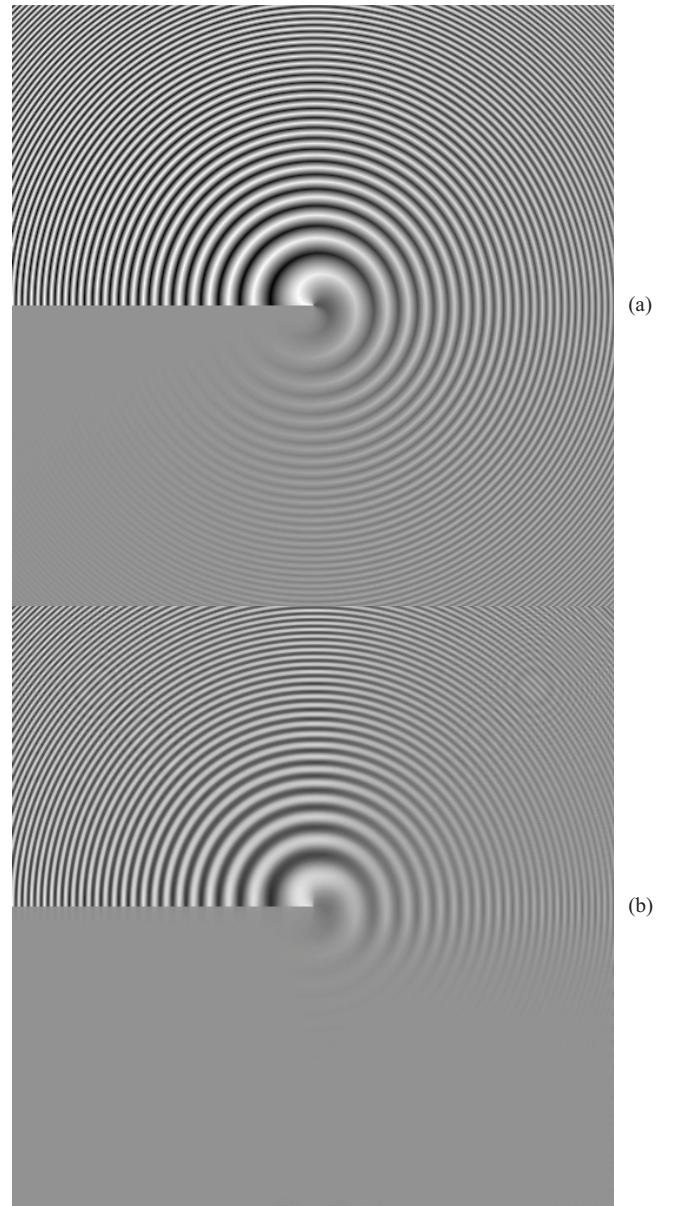


Fig. 1. Whirl. (a) Original image  $w$  displaying a variety of frequencies and contrasts. (Details about the synthesis of this image are discussed in the appendix.) (b) BEEPS suppresses features of jointly low contrast and high frequency.

were also smoothed away (top-right), unless their contrast was strong (top-left). Therefore, the general behavior of the BEEPS is that of an edge-preserving smoother, akin to what would have resulted from a bilateral filter.

### D. Origin of the Proposed Algorithm

Since the range filter  $r$  is arbitrary, in general, the BEEPS output  $y$  is made to depend on the data in nonlinear fashion through (2) and (4). Thus, to simplify our initial understanding of the BEEPS, we start by removing this nonlinearity and temporarily set  $r(u, v) = 1$  for all  $u, v \in \mathbb{R}$ . Then, Recursions (1) and (3) become recursive filters with constant coefficients since  $\varrho = 1 = \rho$ . The progressive and regressive  $z$ -transforms  $P(z) = \sum_{k \in \mathbb{Z}} \varphi[k] z^{-k}$  and  $R(z) = \sum_{k \in \mathbb{Z}} \phi[k] z^{-k}$  now exist

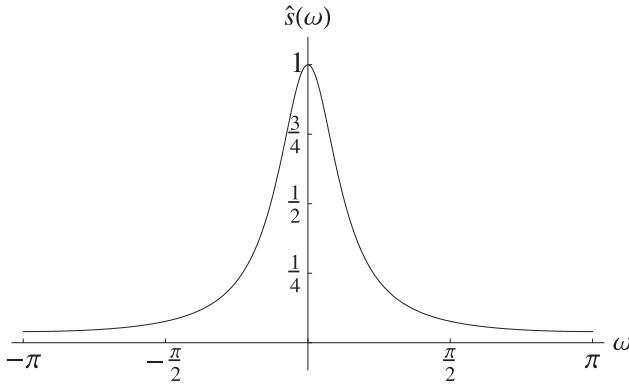


Fig. 2. Frequency response  $\hat{s}$  for  $\lambda = (2/3)$ .

for  $z \in \mathbb{C} \setminus \{0\}$  and for  $\lambda \in (-1, 1)$ . They are given by

$$P(z) = \frac{1-\lambda}{1-\lambda z^{-1}} X(z) \quad (8)$$

$$R(z) = \frac{1-\lambda}{1-\lambda z} X(z) \quad (9)$$

where  $X(z) = \sum_{k \in \mathbb{Z}} x[k] z^{-k}$  is the  $z$ -transform of the input sequence  $x$ . Thus, from (5) with the simplification  $r = 1$ , the BEEPS behaves like a linear filter with the transfer function

$$S(z) = \sum_{k \in \mathbb{Z}} s[k] z^{-k} = \frac{(1-\lambda)^2}{(1-\lambda z^{-1})(1-\lambda z)}. \quad (10)$$

In an imaging context, the filter described by (10) is known as the *bi-exponential filter*. It is a classical convolutional filter with an even impulse response  $s$  that has an infinite support. In the 1-D discrete domain,  $s$  is given by the samples at  $k \in \mathbb{Z}$  of an exponential that decays away from the origin like

$$s[k] = \Lambda \lambda^{|k|} \quad (11)$$

where the contra-decay  $\lambda$  controls the decay  $(1-\lambda)$  of the impulse response and where  $\Lambda$  is the normalization factor

$$\Lambda = \frac{1-\lambda}{1+\lambda}. \quad (12)$$

By convention, we shall assume that  $s[k] = \delta[k]$  when  $\lambda = 0$ . (There,  $\delta[\cdot]$  represents the unit sample.) Various decompositions of the bi-exponential filter, either as a cascade [19] or as a sum [20] of two recursive filters, have been proposed in the literature.

The zeroth-order moment of  $s$  is unity because

$$\sum_{k \in \mathbb{Z}} s[k] = S(e^{j\omega}) \Big|_{\omega=0} = 1. \quad (13)$$

Likewise, the  $\ell_1$  norm  $\|s\|_{\ell_1}$  is also unity for a nonnegative contra-decay, since then  $s = |s|$ . Therefore, the filter is stable in the bounded-input, bounded-output sense—albeit the filter is noncausal. Moreover, the first-order moment vanishes since  $s[k] = s[-k]$ . Finally, the variance of  $s$  is finite for nonnegative contra-decays because

$$\sum_{k \in \mathbb{Z}} k^2 s[k] = j^2 \frac{d^2 S(e^{j\omega})}{d\omega^2} \Big|_{\omega=0} = \frac{2\lambda}{(1-\lambda)^2}. \quad (14)$$

As illustrated in Fig. 2, it turns out that the bi-exponential filter is lowpass for  $\lambda \in [0, 1)$ , the discrete-time Fourier transform  $\hat{s}$  of its impulse response  $s$  being given by

$$\hat{s}(\omega) = S(e^{j\omega}) = \frac{1}{1 + \frac{4\lambda}{(1-\lambda)^2} \sin^2 \frac{\omega}{2}}. \quad (15)$$

It is a remarkable fact that the bi-exponential filter exhibits no ripples in the frequency domain, which makes it better suited than the box filter that is at the core of many histogram-based versions of the bilateral filter [9]–[13]. Moreover, histograms typically require quantization of some sort, while no quantization is required with the BEEPS.

Finally, our proposed algorithm originates from—and closely follows the structure of—a bi-exponential filter, except that the BEEPS has space-varying coefficients. By setting  $r = 1$ , we just showed that the BEEPS is able to replicate a bi-exponential filter for any value of its parameter  $\lambda$ . Moreover, for the special case  $\lambda = 0$ , we leave to the reader to check that the restriction  $r = 1$  can be lifted: in the absence of smoothing, the BEEPS acts as the identity, irrespectively of the choice of the range filter  $r$ . Furthermore, in the special case, when the range filter is chosen to satisfy  $r(u, v) = 0$  for  $u \neq v$  and  $r(u, u) = 1$ , then it can be seen that the BEEPS again falls back to identity, this time irrespectively of the choice of the smoothing parameter  $\lambda$  of the space filter  $s$ . Taken together, these properties conspire at making the BEEPS a well-behaved method that gracefully accommodates widely different range filters, from  $r = 1$  (the variance of  $r$  is infinite) to  $r(u, v) = 0$  for  $u \neq v$  (the variance of  $r$  is zero).

### III. EDGE PRESERVATION

#### A. BEEPS

We consider now the general use of the BEEPS, with some nonvanishing degree of smoothing  $0 < \lambda < 1$  and a nontrivial range filter  $r$ . Although  $r$  can be chosen freely, it is customary to assume that it takes the shape of a centered bump function. In particular, a prototypical instance is the de-normalized Gaussian function

$$r(u, v) = e^{-\frac{(u-v)^2}{2\sigma^2}} \quad (16)$$

where  $\sigma$  is a parameter that controls the width of the bump.

Focusing on the progressive recursion (1) and (2) in which  $\varphi$  and  $\varrho$  are determined, respectively, we can check that the hypothetical case  $x[k] \approx \varphi[k-1]$  would lead to  $\varrho[k] \approx 1$  in (2) and consequently, to  $\varphi[k] \approx x[k]$  in (1). Additional implications would then be  $\varphi[k] \approx \varphi[k-1]$  and  $x[k] \approx x[k-1]$ , which shows that this hypothetical case corresponds to that of a sequence  $x$  that has slow local variations. In other words, the hypothesis  $x[k] \approx \varphi[k-1]$  is valid away from edges. Conversely, as an edge is encountered during the progressive recursion  $k \rightarrow (k+1)$ , the hypothesis  $x[k] \approx \varphi[k-1]$  must be abandoned. Then, because of (2) and (16) we have that  $\varrho[k] \approx 0$ , which loosens the dependence of  $\varphi[k]$  on  $\varphi[k-1]$  through (1) and reinitializes the recursion like in (6). Consequently, when  $r$  is a bump-like function, we observe that  $\varrho$  measures the degree of monotony experienced during the recursion: unless an edge is found,  $\varrho$

remains close to unity. In the presence of an edge between indices  $(k-1)$  and  $k$ , however, the value of  $\varrho[k]$  drops and  $\varphi[k]$  is properly reinitialized before the recursion is resumed. A similar reasoning applies to  $\rho$  and  $\phi$  in the context of the regressive recursion.

Finally, the effect of our proposed algorithm is to smooth the data sandwiched between edges, and to maintain the edges themselves. The function  $r$  provides the tool to gradually discriminate between edge and nonedge. Smoothing is provided by a bi-exponential filter  $s$  controlled by the contral-decay parameter  $\lambda$ . We chose to name  $r$  a range filter and  $s$  a space filter to suggest a link between our BEEPS and the bilateral filter. In the next sections, we investigate this link further.

### B. Bilateral Filter

The bilateral filter has been proposed as another heuristic solution to the preservation of edges while smoothing an image [4], [6]. The governing principle is to locally adapt the coefficients of a linear smoothing filter to perform smoothing only over neighboring data of similar photometry (i.e., intensity). In its discrete version, the value  $y_0$  at coordinates  $\mathbf{k} \in \mathbb{Z}^d$  that results from the application of a bilateral filter to the image  $x$  is given by

$$y_0[\mathbf{k}] = \frac{\sum_{\mathbf{n} \in \mathbb{Z}^d} r(x[\mathbf{k}], x[\mathbf{k} - \mathbf{n}]) s[\mathbf{n}] x[\mathbf{k} - \mathbf{n}]}{\sum_{\mathbf{n} \in \mathbb{Z}^d} r(x[\mathbf{k}], x[\mathbf{k} - \mathbf{n}]) s[\mathbf{n}]} \quad (17)$$

There, the bivariate function  $r$  is used to measure the degree of photometric similarity of its arguments, while the spatial filter of impulse response  $s$  determines the spatial extent of the bilateral filter. By convention, it is customary to assume that the values taken by the range filter  $r$  satisfy  $r \in [0, 1]$ , so that a small  $r$  forbids any sort of smoothing, and a large  $r$  authorizes the smoothing provided by the spatial filter  $s$ . By design, the denominator of (17) ensures a proper normalization—observe that, through  $r$ , this normalization depends on data in a nonlinear fashion and must therefore be computed anew at every coordinate  $\mathbf{k}$ .

For the purpose of discussion, we assume 1-D data, with  $d = 1$ . Moreover, we take  $s$  to be the bi-exponential filter (11). This leads us to rewrite (17) as

$$\begin{aligned} y_0[k] &= \sum_{n=1}^{\infty} \frac{r_{k,k-n} \Lambda}{B_0[k]} \lambda^n x[k-n] + \frac{r_{k,k} \Lambda}{B_0[k]} x[k] \\ &\quad + \sum_{n=1}^{\infty} \frac{r_{k,k+n} \Lambda}{B_0[k]} \lambda^n x[k+n] \end{aligned} \quad (18)$$

where the normalization factor  $B_0$  of the bilateral filter is

$$B_0[k] = \sum_{n=1}^{\infty} r_{k,k-n} \Lambda \lambda^n + r_{k,k} \Lambda + \sum_{n=1}^{\infty} r_{k,k+n} \Lambda \lambda^n \quad (19)$$

and where we have taken advantage of the short-hand notation

$$\forall m, n \in \mathbb{Z} : r(x[m], x[n]) = r_{m,n}. \quad (20)$$

### C. Link Between the BEEPS and the Bilateral Filter

To discover a relation between (18) and the algorithm of Section II, we suggest first to rewrite (1) and (3) in a way that conceals the explicit dependence of  $\varphi[k]$  on  $\varphi[k-1]$  and of  $\phi[k]$  on  $\phi[k+1]$ . By expanding the recursions, we obtain

$$\begin{aligned} \varphi[k] &= \sum_{n=1}^{\infty} \left( \prod_{p=0}^{n-1} \varrho[k-p] \right) (1 - \varrho[k-n] \lambda) \lambda^n x[k-n] \\ &\quad + (1 - \varrho[k] \lambda) x[k] \end{aligned} \quad (21)$$

$$\begin{aligned} \phi[k] &= \sum_{n=1}^{\infty} \left( \prod_{p=0}^{n-1} \rho[k+p] \right) (1 - \rho[k+n] \lambda) \lambda^n x[k+n] \\ &\quad + (1 - \rho[k] \lambda) x[k]. \end{aligned} \quad (22)$$

These expressions will come useful for analysis purpose only.

Next, we adopt the hypothesis of weak smoothing whereby  $0 \leq \lambda \ll 1$ . This allows us to use (21) and (22) to write one sample of the output (5) of the BEEPS as

$$\begin{aligned} y[k] &= \frac{\varrho[k] (1 - \varrho[k-1] \lambda)}{1 + \lambda} \lambda x[k-1] + \frac{1 - \varrho[k] \lambda}{1 + \lambda} x[k] \\ &\quad + \frac{\rho[k] (1 - \rho[k+1] \lambda)}{1 + \lambda} \lambda x[k+1] + \frac{1 - \rho[k] \lambda}{1 + \lambda} x[k] \\ &\quad - \Lambda x[k] + \mathcal{O}(\lambda^2) \\ &= \underbrace{\frac{\varrho[k] \lambda}{1 + \lambda}}_{ap} x[k-1] + \underbrace{\frac{\rho[k] \lambda}{1 + \lambda}}_{ar} x[k+1] \\ &\quad + \underbrace{\left( 1 - \frac{(\varrho[k] + \rho[k]) \lambda}{1 + \lambda} \right)}_a x[k] + \mathcal{O}(\lambda^2). \end{aligned} \quad (23)$$

We apply the same hypothesis to write the normalization factor of the bilateral filter as

$$B_0[k] = r_{k,k-1} \Lambda \lambda + r_{k,k} \Lambda + r_{k,k+1} \Lambda \lambda + \mathcal{O}(\lambda^2). \quad (24)$$

In the weak-smoothing regime, we also establish the outcome of the bilateral filter (18) as

$$\begin{aligned} y_0[k] &= \underbrace{\frac{r_{k,k-1} \lambda}{r_{k,k+1} \lambda + r_{k,k} \lambda + r_{k,k-1} \lambda + \mathcal{O}(\lambda^2)}}_{a_{0P}} x[k-1] \\ &\quad + \underbrace{\frac{r_{k,k}}{r_{k,k+1} \lambda + r_{k,k} \lambda + r_{k,k-1} \lambda + \mathcal{O}(\lambda^2)}}_{a_0} x[k] \\ &\quad + \underbrace{\frac{r_{k,k+1} \lambda}{r_{k,k+1} \lambda + r_{k,k} \lambda + r_{k,k-1} \lambda + \mathcal{O}(\lambda^2)}}_{a_{0R}} x[k+1] \\ &\quad + \mathcal{O}(\lambda^2). \end{aligned} \quad (25)$$

We are now equipped to perform a term-by-term comparison of (23) and (25). Given the definitions of  $\{a_R, a, a_P\}$  and  $\{a_{0R}, a_0, a_{0P}\}$  provided in (23) and (25), respectively, the ratio

of the coefficients of matched terms is

$$\begin{aligned} \frac{a_P}{a_{0P}} &= \frac{\varphi[k] \lambda (r_{k,k+1} \lambda + r_{k,k} + r_{k,k-1} \lambda + \mathcal{O}(\lambda^2))}{r_{k,k-1} \lambda (1 + \lambda)} \\ &= \frac{\varphi[k] \lambda r_{k,k} + \mathcal{O}(\lambda^2)}{r_{k,k-1} \lambda + \mathcal{O}(\lambda^2)} \end{aligned} \quad (26)$$

$$\begin{aligned} \frac{a}{a_0} &= \frac{\left(1 - \frac{(\varphi[k] + \varphi[k])\lambda}{1 + \lambda}\right)(r_{k,k+1} \lambda + r_{k,k} + r_{k,k-1} \lambda + \mathcal{O}(\lambda^2))}{r_{k,k}} \\ &= 1 + \frac{r_{k,k+1} - \frac{\varphi[k] + \varphi[k]}{1 + \lambda} r_{k,k} + r_{k,k-1}}{r_{k,k}} \lambda + \mathcal{O}(\lambda^2) \end{aligned} \quad (27)$$

$$\begin{aligned} \frac{a_R}{a_{0R}} &= \frac{\varphi[k] \lambda (r_{k,k+1} \lambda + r_{k,k} + r_{k,k-1} \lambda + \mathcal{O}(\lambda^2))}{r_{k,k+1} \lambda (1 + \lambda)} \\ &= \frac{\varphi[k] \lambda r_{k,k} + \mathcal{O}(\lambda^2)}{r_{k,k+1} \lambda + \mathcal{O}(\lambda^2)}. \end{aligned} \quad (28)$$

We then assume that the otherwise arbitrary range function  $r$  is constrained to reach unity when its two arguments take the same value, which is usually the case for the functions of interest. Formally, we demand that

$$r_{k,k} = 1. \quad (29)$$

We then conclude from (2), (4), and (20) that

$$\begin{aligned} \lim_{\lambda \rightarrow 0} \frac{a_P}{a_{0P}} &= \frac{r(x[k], \varphi[k-1])}{r(x[k], x[k-1])} \\ \lim_{\lambda \rightarrow 0} \frac{a}{a_0} &= 1 \\ \lim_{\lambda \rightarrow 0} \frac{a_R}{a_{0R}} &= \frac{r(x[k], \varphi[k+1])}{r(x[k], x[k+1])}. \end{aligned} \quad (30)$$

Finally, we observe from (1) and (3) that a vanishing  $\lambda$  would lead to  $\varphi[k] = x[k] = \phi[k]$ . This results in  $\lim_{\lambda \rightarrow 0} (a_P)/(a_{0P}) = 1 = \lim_{\lambda \rightarrow 0} (a_R)/(a_{0R})$ , which shows that the BEEPS and the bilateral filter are equivalent under the conditions described.

The equivalence established in this section can be summarized by observing the parallel between (23) and (25), like in

$$y[k] = a_P x[k-1] + a x[k] + a_R x[k+1] + \mathcal{O}(\lambda^2) \quad (31)$$

$$y_0[k] = a_{0P} x[k-1] + a_0 x[k] + a_{0R} x[k+1] + \mathcal{O}(\lambda^2). \quad (32)$$

In doing so, it is crucial to remember that the various coefficients  $\{a_P, a, a_R, a_{0P}, a_0, a_{0R}\}$  are not constant-valued but depend on the data. What we have shown is that this data dependence is identical for the bilateral filter and for the BEEPS when  $\lambda$  vanishes. For small  $\lambda$ , it follows that these two edge-preserving methods will yield nearly identical results. For larger  $\lambda$ , we shall see in Section IV that the two methods still behave similarly.

#### D. Adjustment of the Contra-Decay

While, in its original form, the bilateral filter (17) can accommodate any arbitrary range filter  $r$  and spatial filters  $s$ , it has nonetheless become customary to choose a Gaussian in each case. Since the BEEPS can accommodate any arbitrary range filter as well, it is natural to impose that the range filter of the BEEPS be proportional to the range filter of a

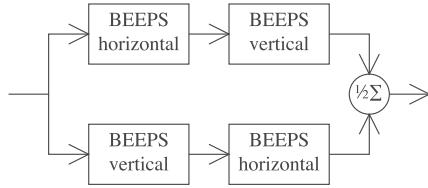


Fig. 3. Average of the row-first and column-first separable application of the BEEPS.

bilateral filter when it is desired that the former copies the behavior of the latter. However, the normalization found in the denominator of (17) is indeed explicit, while it is only implicit in the BEEPS. Thus, a de-normalized Gaussian taking the form (16) has to be used in the case of a BEEPS, while normalization or de-normalization is irrelevant in the case of the bilateral filter.

When as much similarity as possible is desired between the BEEPS and the bilateral filter while keeping the spatial filter a bi-exponential for the former and a Gaussian for the latter, the contra-decay  $\lambda$  of the BEEPS must be tuned to match the spatial standard deviation  $\sigma_s$  that characterizes the spatial Gaussian filter of the bilateral filter. To do so, we take advantage of (14), which suggests the relation

$$\lambda = 1 - \frac{\sqrt{2 \sigma_s^2 + 1} - 1}{\sigma_s^2}. \quad (33)$$

#### E. Application to Several Dimensions

The BEEPS proposed in Section II is 1-D, such as the bi-exponential filter it originates from. Likewise, it contains progressive and regressive recursions, which precludes a comprehensive extension of these two filters to higher-dimensions. Thus, the prevailing way to apply either filter in several dimensions is to design a separable implementation where successive 1-D operations are carried over various directions. This extension is natural with the bi-exponential filter because it is a linear filter and the order of exploration of the 1-D operations does not matter. However, one major issue with the BEEPS is that it is nonlinear. This implies that the order of exploration matters for the BEEPS. Along  $D$  directions there are  $D!$  possible ways to put separability to fruition, which leads to  $(D!)^d$  1-D processes in  $d$  dimensions. In the context of 2-D images, the standard practical choice consists in  $D = 2$  and  $D! = 2$ , which corresponds to the application of the BEEPS along rows and columns, in either order. The next practical choice for  $d = 2$  would be  $D = 4$  and  $D! = 24$ , where the main diagonals would be added to the mix.

In this paper, we choose to apply the 1-D BEEPS four times to process a 2-D image, with  $d = 2$  and  $D! = 2$ . More precisely, we apply first a horizontal BEEPS to the original image and submit the resulting intermediate image to a subsequent vertical application. These two sequential applications we call row-first. Separately, we also apply the opposite sequence of vertical then horizontal BEEPS to the original image, which we call column-first. We then average the row-first and column-first outcomes to obtain the final result. We give a sketch of our strategy in Fig. 3. This



Fig. 4. Gallery of results of the BEEPS applied to the GreekDome image. The spatial smoothing increases from top to bottom with  $\lambda \in \{0.8, 0.9, 0.95\}$ . From left to right, there is an increase in the standard deviation of the range filter  $\sigma \in \{20, 50, 100\}$ . The original image is visually close to the topmost, left-most image in the gallery. The computational cost was identical for all images.

strategy comes at the price of doubling our computational cost because we consider two branches in the sketch of Fig. 3, but improves over the early solution for the separability of the bilateral filter proposed in [21] where a single branch was followed. As discussed in [16], a complementary way to increase isotropy would be to iterate the separable 1-D filters  $n$  times. In that case, due to the additivity of the variance (14), the target compound variance  $\sigma_s$  is achieved after  $n$  applications of BEEPS, each of which with the contradeay  $\lambda_n = 1 - (\sqrt{2n\sigma_s^2 + n^2} - n)/\sigma_s^2$ .

Our proposed algorithm offers improvements over several aspects of the edge-aware recursive filtering that is briefly described in [16] and that also involves 1-D recursive filters. In particular, our range filter can be chosen and tuned freely, while the weight-adaptive method of [16] restricts itself to the use of curvilinear lengths in a space-intensity manifold. Moreover, we take advantage of the recursive feedback offered by (2) and (4) to make the adaptation of the range filter less local than that proposed in [16]. Finally, we apply the two recursions (1) and (3) in parallel and merge them in (5), which ensures that our filter is anisotropic in one dimension. By contrast, the formulation of [16] suggests that the progressive and the regressive recursions be applied in sequence instead. Suppose that the progressive recursion is applied first: this would imply that the interruption of the forward propagation chain mentioned in [16] would be smoothed out when the sub-

sequent regressive filter is applied, thus leading to anisotropy, even with 1-D data.

## IV. EXPERIMENTS

### A. Illustration

We now apply the BEEPS to the  $(876 \times 584)$  GreekDome image found in a publicly available collection devoted to the task of investigating bilateral filters.<sup>1</sup> To illustrate the impact of several combinations of the range parameter  $\sigma$  and of the spatial parameter  $\lambda$ , we organize Fig. 4 as a grid of images where  $\sigma$  increases from left to right and  $\lambda$  from top to bottom. More precisely, the spatial smoothing corresponds to a filter of standard deviation  $\{6.3, 13.4, 27.5\}$ , while the range filter takes a Gaussian shape and is characterized by  $\sigma \in \{20, 50, 100\}$ .

With  $\sigma = 100$ , the right-most part of Fig. 4 presents results that are close to those that would be achieved by the bi-exponential linear filter of Section II-D, because there  $r$  is flat and barely distinguishes low-contrast from high-contrast features. In particular, because of the initialization we chose to implement for  $\phi[0]$  and  $\phi[K-1]$ , the image that corresponds to  $\lambda \rightarrow 1$  will take a constant value that is the average of the four corner pixels of the input image. Meanwhile, with  $\sigma = 20$ , the left-most part of Fig. 4 presents results that take

<sup>1</sup> Available at <http://people.csail.mit.edu/sparis/bf/>.

TABLE I

SIZE ( $W_s \times W_s$ ) OF THE SPATIAL FILTER  $s$  OF THE BILATERAL FILTER

$\lambda$	0.25	0.50	0.80	0.90	0.95	0.98
$\sigma_s$	0.9	2.0	6.3	13.4	27.5	70.0
$W_s$	7	13	39	83	167	421

a cartoon-like appearance when the spatial smoothing becomes large enough, which is indeed the hallmark of edge-preserving smoothers.

### B. Comparison to the Bilateral Filter

Since we could establish in Section III-C a formal link between the BEEPS and the bilateral filter, and since practitioners were able to gain proficiency in tuning the parameters of the long-established bilateral filter, a direct practical comparison between these two filters is of relevance. At the same time, we feel the need to stress that the BEEPS is a filter in its own right, and that the bilateral filter is as much an approximation of the BEEPS as the BEEPS is an approximation of the bilateral filter.

The theoretical result of Section III-C is valid in 1-D and when the spatial filter of the bilateral filter is bi-exponential, which lead to (18). Thus, to maximize the fairness of the comparisons in the context of 2-D images, we have let  $s$  in (17) be the tensor product of bi-exponential filters as well. Further, to make its nonrecursive implementation practical, we have truncated this impulse response to a half width of three standard deviations. We indicate, in Table I, the size of the truncated filters for the series of contra-decays  $\lambda$  used in this paper, along with the corresponding standard deviation of the spatial filter. Furthermore, we have followed a brute-force approach to implement (17) in a nonseparable and straightforward—albeit excruciatingly slow—fashion. To compute the sums in (17) while letting the index  $\mathbf{n}$  run over the whole support of  $s$ , we have extended virtually the input image  $x$  in a way that is consistent with (6) and (7), as

$$x[k_1, k_2] = \begin{cases} x[0, k_2], & k_1 < 0 \\ x[K_1 - 1, k_2], & K_1 \leq k_1 \\ x[k_1, 0], & k_2 < 0 \\ x[k_1, K_2 - 1], & K_2 \leq k_2. \end{cases} \quad (34)$$

Our first comparison is visual. We give, in Fig. 5, the outcome of applying the bilateral filter [Fig. 5(a)] and the BEEPS [Fig. 5(b)] to the Rock image, for a series of range filters and for a fixed spatial filter. We qualitatively observe that the bilateral and the BEEPS results are very much alike in the sky region, which is characterized mostly by smooth edges. This is also true of the widely separated sharp edges found in the rocky area. In the tree area, characterized by fine-grained texture, the BEEPS may appear—on print—to maintain more details than the bilateral filter when the standard deviation of the range filter is strong. A full-scale zoom on the image (not shown) reveals, however, that the anisotropy of the BEEPS is defective in this region.

In our next comparison, we let the range filter  $r$  of the bilateral filter and the range filter of the BEEPS be fixed,

TABLE II

SIMILARITY OF THE BILATERAL FILTER WITH RESPECT TO THE BEEPS AT OPTIMAL AND MATCHED CONTRA-DECAYS

$\sigma_r$	2	5	10	20	50	100	200	
$\sigma_s$	$\lambda_{BF}$	Optimal $\lambda_{opt}$						
0.9	0.25	0.231	0.233	0.231	0.234	0.237	0.241	0.243
2.0	0.50	0.411	0.416	0.415	0.447	0.473	0.485	0.489
6.3	0.80	0.367	0.619	0.672	0.719	0.776	0.800	0.798
13.4	0.90	0.196	0.704	0.857	0.864	0.891	0.910	0.905
27.5	0.95	0.108	0.211	0.948	0.956	0.962	0.964	0.956
70.0	0.98	0.052	0.105	0.177	0.999	0.999	0.996	0.986
$\sigma_s$	$\lambda_{BF}$	$J_{opt}$ (PSNR) at optimal $\lambda_{opt}$						
0.9	0.25	74.2	66.5	60.0	55.7	57.0	63.4	68.8
2.0	0.50	63.9	56.3	49.9	45.5	46.7	53.8	62.5
6.3	0.80	57.6	49.6	43.4	38.7	38.7	45.3	53.9
13.4	0.90	56.7	47.5	41.5	36.7	35.6	40.4	47.1
27.5	0.95	57.1	46.8	40.3	35.3	33.1	36.2	41.7
70.0	0.98	58.3	47.2	39.6	33.7	30.7	33.2	38.2
$\sigma_s$	$\lambda_{BF}$	Similarity loss ( $J_{opt} - J$ ) at matching $\lambda = \lambda_{BF}$						
0.9	0.25	0.57	0.34	0.30	0.32	0.99	2.48	4.18
2.0	0.50	0.58	0.47	0.38	0.24	0.33	0.70	2.55
6.3	0.80	1.16	0.33	0.20	0.14	0.08	0.00	0.02
13.4	0.90	2.46	0.17	0.03	0.04	0.01	0.28	0.58
27.5	0.95	4.14	0.85	0.00	0.00	0.07	0.83	1.10
70.0	0.98	6.40	2.57	0.65	0.10	0.61	1.90	1.99

built according to (16). We then vary the contra-decay  $\lambda_{BF}$  associated to the bilateral filter; independently, we explore several (possibly mismatched) values of the contra-decay  $\lambda$  associated with the BEEPS and measure the mean-square difference between the images resulting from the application of these two edge-preserving smoothers. This difference is expressed in dB as the peak signal-to-noise ratio (PSNR)  $J$  defined by

$$J\{y_0, y\} = 10 \log_{10} \frac{255^2}{\frac{1}{|\Omega|-1} \sum_{\mathbf{k} \in \Omega} (y_0[\mathbf{k}] - y[\mathbf{k}])^2}. \quad (35)$$

We give, in Fig. 6, a few typical curves that correspond to this experiment for the Rock image. We see that the highest similarity between the bilateral filter and the BEEPS is achieved when the contra-decays  $\lambda$  and  $\lambda_{BF}$  approximately match each other. This provides us with an experimental verification of the prediction made in Section III-C, according to which the similarity  $J$  is maximized by letting  $\lambda = \lambda_{BF}$ . Moreover, the examination of Fig. 6 also confirms the prediction that the degree of similarity increases when the contra-decays get smaller. We confirm these results in Table II, which complements Fig. 6 by extending the domain of explored range variances and contra-decays for the Rock image. We conclude from this table that setting  $\lambda$  to  $\lambda_{BF}$  (instead of the objectively optimal  $\lambda_{opt}$ ) leads to a decrease in similarity that has no practical impact. We illustrate visually in Fig. 7, the results of the comparison at  $\sigma = 50$  and matching  $\lambda$ . Additional images lead us to the same conclusions (results not shown).

In our final comparison, we consider only the case  $\lambda = \lambda_{BF}$  and investigate how the bilateral filter and the BEEPS differ, on the basis of the series of images that we provide in Fig. 8. At first, we investigate the visual impact of both filters on the cutouts indicated in this figure. The results are provided at constant spatial smoothing corresponding to the

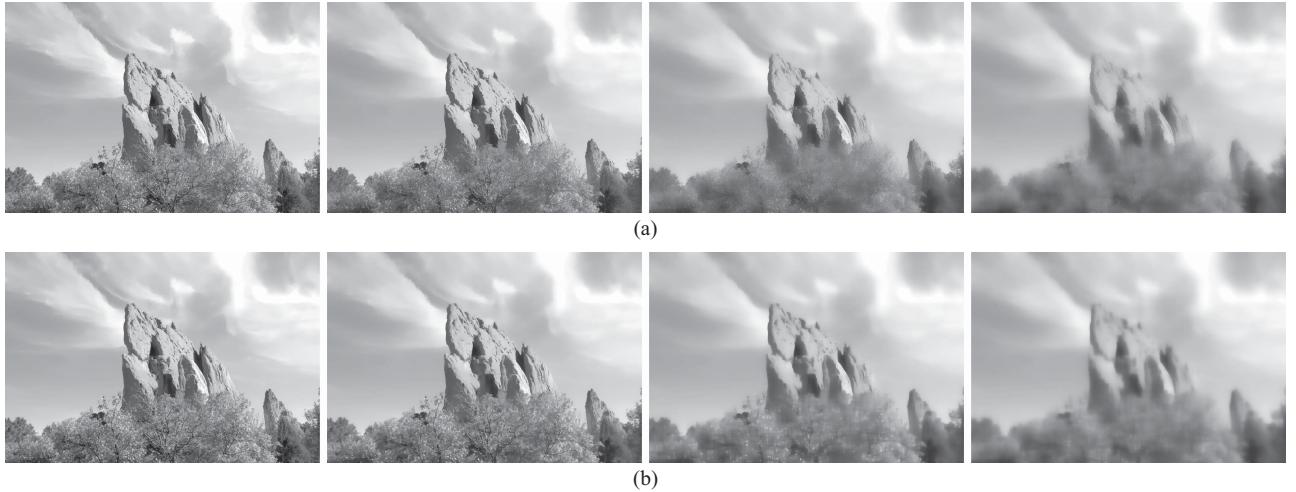


Fig. 5. (a) Bilateral filter and (b) BEEPS, as applied to the Rock image. The contra-decay is constant with  $\lambda = 0.9$ . The standard deviation of the range filter increases from left to right with  $\sigma \in \{10, 20, 50, 100\}$ .

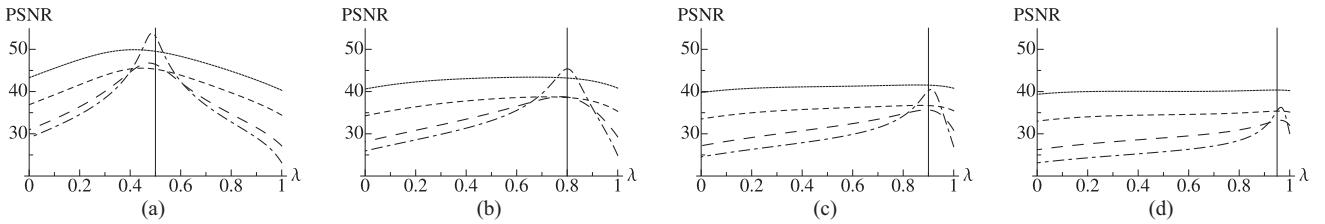


Fig. 6. PSNR versus contra-decay  $\lambda$  of the BEEPS, between the bilateral filter and the BEEPS applied to the Rock image. (a)–(d) Contra-decay associated to the bilateral filter is indicated by a vertical bar, with  $\lambda_{BF} \in \{0.5, 0.8, 0.9, 0.95\}$ . The dotted, dashed, long-dashed, and mixed curves correspond to the standard deviations  $\sigma \in \{10, 20, 50, 100\}$  of the range filter, respectively.

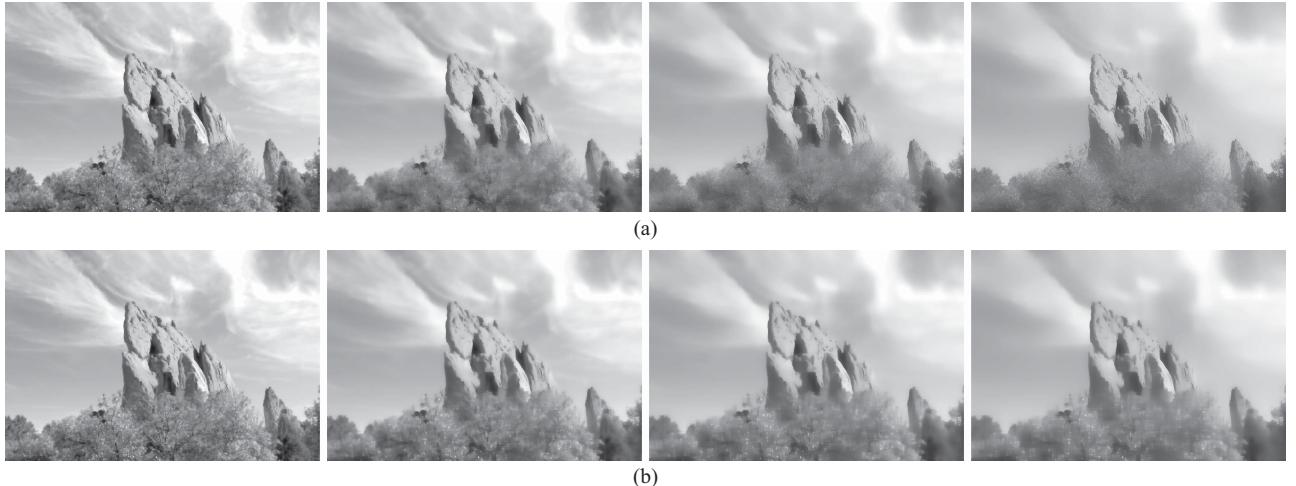


Fig. 7. (a) Bilateral filter and (b) BEEPS, as applied to the Rock image. The contra-decay increases from left to right with  $\lambda \in \{0.5, 0.8, 0.9, 0.95\}$ . The standard deviation of the range filter is constant with  $\sigma = 50$ .

contra-decay  $\lambda = 0.9$  in Fig. 9, and at constant range filter characterized by  $\sigma = 50$  in Fig. 10. At constant smoothing (Fig. 9), we observe that the BEEPS removes details earlier than the bilateral filter as the range filter becomes flatter. This is particularly true at  $\sigma = 50$  where edges of medium strength seem to resist simplification with the bilateral filter but do disappear with the BEEPS, like those found in the tunic of the eye (Dragon), in the texture of the scales and the

lower eyelid (Turtle), and the fine structure of stone (Rock). Instead, the BEEPS and the bilateral filter seem to be in agreement for weaker and stronger edges alike, for instance the weak edges found in the texture of the wall (GreekDome) and of the sky (Rock); this agreement extends to the strong edges of the lintel (GreekDome) and the contrast between stone and sky (Rock). At constant standard deviation of the range filter (Fig. 10), we observe that the bilateral filter yields



Fig. 8. Test images and cutouts. (a)  $(80 \times 182)$  cutout of the Dragon image encloses its eye and has high contrast, with round and wavy shapes. (b)  $(80 \times 60)$  cutout of the GreekDome image encloses the lintel of a window and contains both low- and high-frequencies. (c)  $(80 \times 60)$  cutout of the Turtle image encloses a part of its head and has a wide dynamic range. (d)  $(80 \times 60)$  cutout of the Rock image encloses an outline of the rock against a fuzzy texture in the sky.

sometimes unforeseen results. For instance, the outlines of the scale (Turtle) are attenuated when increasing  $\lambda$  from 0.5 to 0.8, which is expected; however, it is an unexpected result that they seem to get sharper as smoothing increases from  $\lambda = 0.8$  to  $\lambda = 0.95$ . Similarly, details in the corner of the mouth of the dragon are present with strong smoothing and absent with weak smoothing. This at-first counter-intuitive aspect of the bilateral filter is absent from the BEEPS, which makes this filter easier to understand: the image just gets simpler and less detailed as the smoothing is increased, all the while preserving edges in accordance with the prescriptions of the range filter.

We provide in Table III an objective comparison of the bilateral filter and the BEEPS, where we give the similarity  $J$  computed over the whole support of the images used in this paper, for various combinations of spatial smoothing and range filters. Again, at small  $\lambda$ , we verify the validity of the predictions of Section III-C. As  $\lambda$  increases, the two methods con-

sistently produce diverging results. The divergence is the most pronounced when the range filter takes the standard deviation  $\sigma = 50$ . For both smaller and larger values, however, the bilateral filter and the BEEPS tend to agree more with each other. In particular, the two methods produce very similar results for narrow range filters, which are difficult to handle by several of the acceleration techniques applied to the bilateral filter.

### C. Performance

A remarkable feature of the BEEPS is that it runs in constant time, irrespective of  $\lambda$  or  $\sigma$ . For instance, the powerful smoothing appearing in the bottom row of Fig. 4 is achieved as quickly as the unobtrusive smoothing found in the top row of the same figure. In this respect, the BEEPS differs markedly from methods like the bilateral filter, where the degree of smoothing is ultimately controlled by the support of  $s$ , which imposes the number of terms that participate

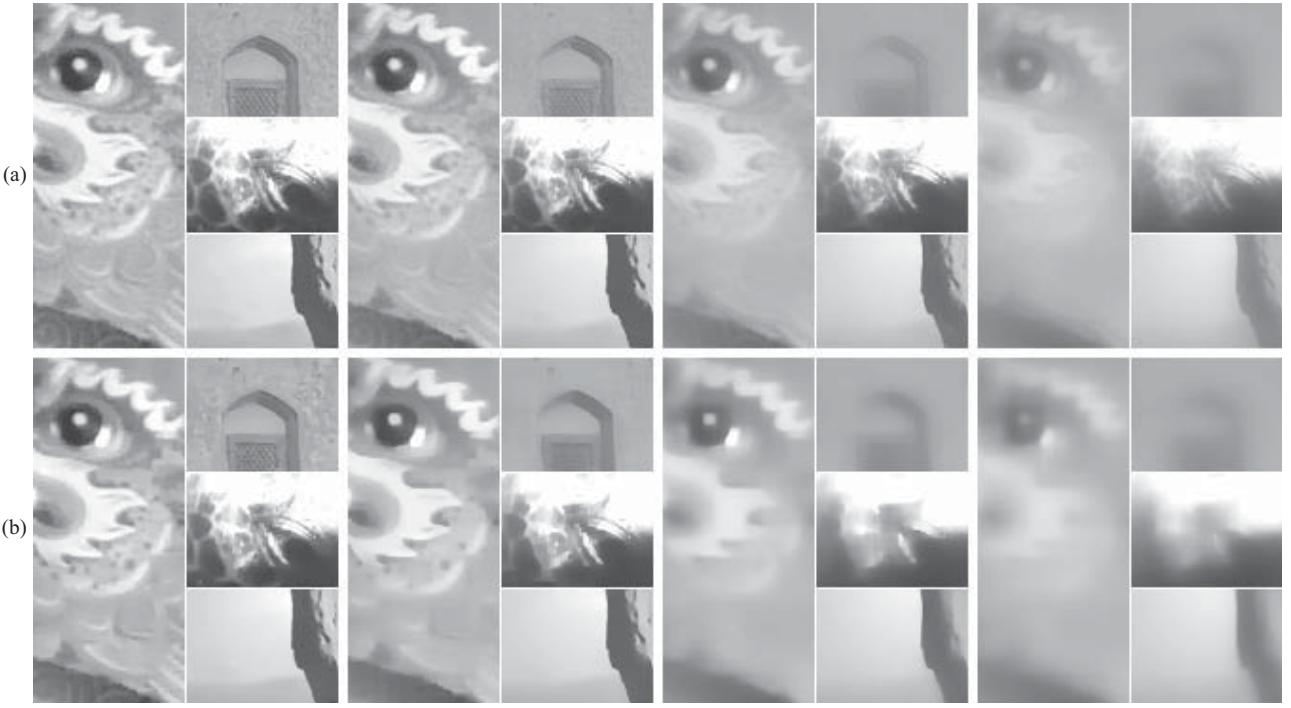


Fig. 9. (a) Application of the bilateral filter and (b) BEEPS to the cutouts from Fig. 8. The contra-decay of the bi-exponential spatial filter is kept constant, with  $\lambda = 0.9$ . The Gaussian range filter takes the standard deviation  $\sigma \in \{10, 20, 50, 100\}$ , from left to right.



Fig. 10. (a) Application of the bilateral filter and (b) BEEPS to the cutouts from Fig. 8. The bi-exponential spatial filter takes the contra-decay  $\lambda \in \{0.5, 0.8, 0.9, 0.95\}$ , from left to right. The standard deviation of the Gaussian range filter is kept constant, with  $\sigma = 50$ .

in both the numerator and denominator sums found in (17). For instance, even in [22], the computational cost of exploring  $s$  depends on the perimeter of the support of  $s$ .

Many proposals have been put forth to accelerate traditional edge-preserving smoothers. Most of those that reach real-time performance, however, rely on technological aids, such as

graphic cards [23] or FPGAs [24]. In this paper, we propose an algorithmic solution that we have implemented in a general-purpose language (Java), and that runs on a general-purpose processor. We found its execution is so fast that we have no need to seek further acceleration, all the while keeping a minimalistic coding effort, and at a negligible memory footprint.

TABLE III  
PSNR WITH RESPECT TO THE BILATERAL FILTER

$\sigma_r$		2	5	10	20	50	100	200
$\sigma_s$	$\lambda$	Whirl						
0.9	0.25	76.6	68.5	61.2	54.3	47.8	47.0	50.2
2.0	0.50	66.0	58.0	51.0	44.6	38.5	38.6	44.6
6.3	0.80	59.1	51.3	44.8	38.9	32.7	30.2	38.1
13.4	0.90	58.4	49.9	43.6	37.4	30.7	28.0	36.4
27.5	0.95	57.9	49.2	42.8	36.6	29.7	27.1	35.8
70.0	0.98	56.7	48.1	41.7	35.7	29.1	27.0	35.8
$\sigma_s$	$\lambda$	Dragon						
0.9	0.25	71.2	63.7	60.1	58.2	59.0	61.4	62.6
2.0	0.50	60.9	53.4	49.8	47.9	49.7	54.6	57.4
6.3	0.80	54.7	46.6	42.2	39.4	40.3	45.9	54.1
13.4	0.90	53.3	44.7	39.8	36.1	35.4	39.2	46.9
27.5	0.95	52.6	43.6	38.3	33.9	31.6	33.7	40.0
70.0	0.98	52.2	42.8	37.1	32.1	28.2	29.0	33.7
$\sigma_s$	$\lambda$	GreekDome						
0.9	0.25	72.3	66.2	62.9	61.6	63.8	67.7	69.5
2.0	0.50	62.6	56.0	52.7	51.5	54.6	61.0	64.7
6.3	0.80	56.5	49.4	45.6	43.7	45.8	52.4	60.8
13.4	0.90	54.9	47.6	43.2	40.3	40.8	45.2	53.1
27.5	0.95	53.7	46.5	41.6	37.7	36.4	38.9	45.0
70.0	0.98	52.0	45.1	39.8	35.1	31.9	33.3	38.2
$\sigma_s$	$\lambda$	Turtle						
0.9	0.25	72.0	64.7	60.2	56.9	55.8	58.0	60.5
2.0	0.50	61.7	54.6	50.0	46.6	45.7	49.4	54.5
6.3	0.80	55.2	47.7	42.7	38.6	36.6	39.9	47.2
13.4	0.90	53.5	45.6	40.2	35.6	32.5	34.1	39.6
27.5	0.95	52.6	44.4	38.7	33.7	29.4	29.3	33.1
70.0	0.98	52.0	43.4	37.4	31.9	26.5	24.6	26.8
$\sigma_s$	$\lambda$	Rock						
0.9	0.25	73.6	66.2	59.7	55.4	56.0	60.9	64.6
2.0	0.50	63.3	55.9	49.5	45.2	46.4	53.1	59.9
6.3	0.80	56.4	49.3	43.2	38.5	38.6	45.3	53.8
13.4	0.90	54.2	47.3	41.5	36.6	35.6	40.2	46.5
27.5	0.95	52.9	45.9	40.3	35.3	33.1	35.4	40.6
70.0	0.98	51.9	44.6	39.0	33.6	30.1	31.3	36.2

Nonetheless, such acceleration remains easy to achieve, for the BEEPS lends itself well to parallelization. The following strategies are compatible with one another; each one brings additional time savings.

- 1) The top and bottom branches of Fig. 3 can be executed concurrently, which can potentially double the speed.
- 2) Each line (whether a row or a column) can be processed independently of the other lines, which can potentially multiply the speed by  $K$  for an image of size  $(K \times K)$ .
- 3) Within a line, the progressive recursion and the regressive recursion of Section II can proceed in parallel, which can multiply the speed by a factor about 3/2.

TABLE IV  
PERFORMANCE OF THE BEEPS

Size	[Mpixel]	Time [s]	Frame rate [Hz]
256 × 256	0.1	0.014	69.4
512 × 512	0.3	0.039	25.5
1024 × 768	0.8	0.165	6.0
1280 × 720	0.9	0.196	5.1
1024 × 1024	1.0	0.205	4.8
1920 × 1080	2.1	0.426	2.3

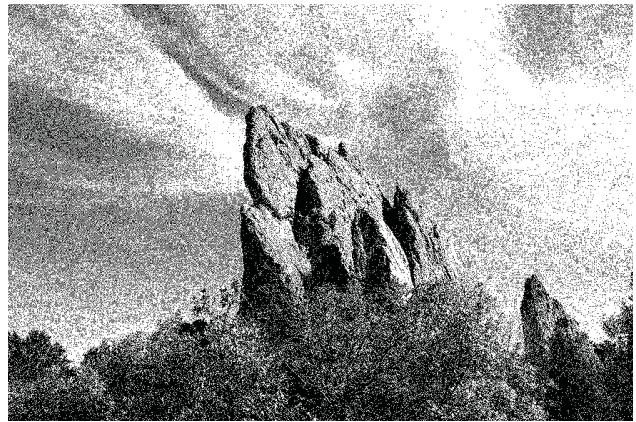


Fig. 11. Our algorithm can generate a dithered image.

We give in Table IV, the time spent processing images of different sizes. (The processor was a 2 × 2.8 GHz Quad-Core Intel Xeon.) Under these conditions, we were able to measure a consistent throughput of about  $5 \times 10^6$  pixel-per-second. The code is freely available as a plugin for ImageJ.<sup>2</sup> We finally observe that the BEEPS allows for high-throughput processing without recourse to specialize hardware.

## V. POSSIBLE EXTENSIONS

In some respects, the BEEPS is very much similar to the bilateral filter. In that capacity, it can be used as its potential substitute, for instance in some popular applications like the compression of high-dynamic-range images [9]. However, it also allows for the emergence of additional capabilities, as we are now going to demonstrate.

Definition (17) of the bilateral filter involves a ratio of terms. With uncooperative data, it could happen that the numerator differs from zero while the denominator vanishes, provided at least one of the range filter  $r$  or the spatial filter  $s$  is allowed to take negative values. For this reason, the conditions  $0 \leq r$  and  $0 \leq s$  are enforced in the context of the bilateral filter.

By contrast, the BEEPS described in Section II-A suffers no such limitation, which offers new intriguing opportunities. For instance, the spatial filter can be made to promote high frequencies by choosing  $-1 < \lambda < 0$ . We illustrate in Fig. 11, the application of the BEEPS with  $\lambda = -0.9$  and  $\sigma = 10$  to the Rock image, with subsequent black-and-white thresholding. The result is a dithered version of the

<sup>2</sup>Available at <http://bigwww.epfl.ch/thevenaz/beeps/>.



Fig. 12. Our algorithm can fulfill creative drives—here, a silver lining.

original where detailed elements, such as tree branches, are well preserved. The BEEPS can also accommodate range filters that the bilateral filter cannot. For instance, we have created a silver-lining effect by combining  $\lambda = -0.5$  with a range filter  $r$  in the shape of a signed sigmoid curve, as realized by a hyperbolic tangent. We give the outcome in Fig. 12.

## VI. CONCLUSION

We have proposed an edge-preserving smoother that is essentially a version of the bi-exponential filter with adaptive weights. We have designed the mechanism of weight adaptation to ensure that our algorithm falls back to standard smoothing in limit cases. We were able to show that, with the appropriate hypotheses, it has formal links with the bilateral filter. Contrarily to several methods already available to accelerate the bilateral filter, ours never quantizes data, neither in terms of grayscale intensity nor in terms of range filter. In our case, the choice of the range filter enjoys complete freedom, at no penalty whatsoever. Making use of separability, we have extended our proposed filter beyond a single dimension and suggested ways to weaken the resulting lack of isotropy. We have implemented our algorithm in Java and discussed its performance on a general-purpose processor. The simplicity of our algorithm makes it effortless to code. No pre-computations were involved, and its memory footprint is negligible. We have conducted numerous experiments, qualitative and quantitative, and performed an extensive comparison of the results of our method with those of the bilateral filter. We concluded that ours offers an inexpensive way to perform edge-preserving smoothing, while producing results that closely mimic those of the bilateral filter.

## APPENDIX

We have built the whirl image  $w : \mathbb{R}^2 \mapsto [0, 255]$  as

$$w(A, \theta) = \frac{255}{2} + \frac{255}{2} \left(1 - \sin \frac{\theta}{2}\right) \frac{1}{\pi} \arcsin(-\cos(2\pi A^2 - \theta)). \quad (36)$$

We have rasterized  $w$  so as to build an  $(N \times N)$  Cartesian array indexed by  $\mathbf{n} \in [0 \dots N - 1]^2$ , with  $x_{1,2} =$

$(2W(n_{1,2})/(N-1) - W)$  and  $A \cos \theta = x_1$ ,  $A \sin \theta = x_2$ . In practice, we have set  $N = 512$  and  $W = 5$ . We show the original  $w$  at the top of Fig. 1.

## ACKNOWLEDGMENT

The authors would like to thank S. Paris and F. Durand for kindly providing some of the images used in this paper.

## REFERENCES

- [1] V. Aurich and J. Weule, “Non-linear Gaussian filters performing edge preserving diffusion,” in *Proc. Deutsche Arbeitsgemein. Mustererkennung Symp. Pattern Recognit.*, Bielefeld, Germany, Sep. 1995, pp. 538–545.
- [2] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. 6th Int. Comput. Vision Conf.*, Bombay, India, Jan. 1998, pp. 839–846.
- [3] 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.
- [4] 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.
- [5] J.-S. Lee, “Digital image smoothing and the sigma filter,” *Comput. Vis. Graph. Image Process.*, vol. 24, no. 2, pp. 255–269, Nov. 1983.
- [6] S. Paris, P. Kornprobst, J. Tumblin, and P. Durand, “Bilateral filtering: Theory and applications,” *Found. Trends Comput. Graph. Vis.*, vol. 4, no. 1, pp. 1–73, 2008.
- [7] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Proc. IEEE Comput. Soc. Comput. Vis. Pattern Recognit. Conf.*, San Diego, CA, Jun. 2005, pp. 60–65.
- [8] N. Dowsen and O. Salvado, “Hashed nonlocal means for rapid image filtering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 485–499, Mar. 2011.
- [9] 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.
- [10] M. Elad, “Retinex by two bilateral filters,” in *Proc. 5th Int. Scale Space PDE Methods Comput. Vis.*, vol. 3459. Hofgeismar, Germany, Apr. 2005, pp. 217–229, doi: 10.1007/11408031\_19.
- [11] G. Guarneri, S. Marsi, and G. Ramponi, “Fast bilateral filter for edge-preserving smoothing,” *Electron. Lett.*, vol. 42, no. 7, pp. 396–397, Mar. 2006.
- [12] B. Weiss, “Fast median and bilateral filtering,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 519–526, Jul. 2006.
- [13] F. Porikli, “Constant time  $\mathcal{O}(1)$  bilateral filtering,” in *Proc. IEEE Comput. Soc. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 2008, pp. 1–8.
- [14] Q. Yang, K.-H. Tan, and N. Ahuja, “Real-time  $\mathcal{O}(1)$  bilateral filtering,” in *Proc. IEEE Comput. Soc. Comput. Vis. Pattern Recognit. Workshops Conf.*, Miami FL, Jun. 2009, pp. 557–564.
- [15] S. Paris and F. Durand, “A fast approximation of the bilateral filter using a signal processing approach,” *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 24–52, Jan. 2009.
- [16] E. Gastal and M. Oliveira, “Domain transform for edge-aware image and video processing,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–11, Jul. 2011.
- [17] Q. Yang, S. Wang, and N. Ahuja, “SVM for edge-preserving filtering,” in *Proc. IEEE Comput. Soc. Comput. Vis. Pattern Recognit. Conf.*, San Francisco, CA, Jun. 2010, pp. 1775–1782.
- [18] K. Chaudhury, D. Sage, and M. Unser, “Fast  $\mathcal{O}(1)$  bilateral filtering using trigonometric range kernels,” *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3376–3382, Dec. 2011.
- [19] J. Shen and S. Castan, “An optimal linear operator for edge detection,” in *Proc. IEEE Comput. Soc. Comput. Vis. Pattern Recognit.*, Miami Beach, FL, Jun. 1986, pp. 109–114.
- [20] R. Deriche, “Fast algorithms for low-level vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 78–87, Jan. 1990.
- [21] T. Pham and L. van Vliet, “Separable bilateral filtering for fast video preprocessing,” in *Proc. IEEE Int. Multimedia Expo Conf.*, Amsterdam, The Netherlands, Jul. 2005, pp. 454–457.

- [22] M. Igarashi, M. Ikebe, S. Shimoyama, K. Yamano, and J. Motohisa, “ $\mathcal{O}(1)$  bilateral filtering with low memory usage,” in *Proc. 17th IEEE Int. Image Process. Conf.*, Hong Kong, Sep. 2010, pp. 3301–3304.
- [23] H. Winnemöller, S. Olsen, and B. Gooch, “Real-time video abstraction,” in *Proc. 33rd ACM Int. Exhibit. Comput. Graph. Interact. Technol. Conf.*, Boston, MA, Jul.–Aug. 2006, pp. 1221–1226.
- [24] H. Zhu, W. Sun, M. Wu, G. Guan, and Y. Guan, “Pre-processing of X-ray medical image based on improved temporal recursive self-adaptive filter,” in *Proc. 9th Int. Young Comput. Scientists Conf.*, Hunan, China, Nov. 2008, pp. 758–763.



**Philippe Thévenaz** received the Diploma degree in microengineering from École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland, and the Ph.D. degree from the Institute of Microtechnology (IMT), University of Neuchâtel, Neuchâtel, Switzerland, in 1986 and 1993, respectively.

He was a Visiting Fellow with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD, from 1993 to 1998, where he developed splines and multiresolution signal representations, geometric image transformations, and biomedical image registration. Since 1998, he has been at EPFL as a Senior Researcher.



**Daniel Sage** received the M.S. and Ph.D. degrees in control and signal processing from Institut National Polytechnique de Grenoble, Grenoble, France, in 1986 and 1989, respectively.

He was a Consulting Engineer with the Industrial Vision Department, Attexor S.A., Écublens VD, Switzerland, from 1989 to 1998. During his career, he has developed vision systems oriented to quality control in the industrial sector. In 1998, he joined the Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland, as the Head of software development. He is in charge of both the coordination of software development and of setting down the computing infrastructure of the group. He is also involved in the development of bioimaging software and methods for computer-assisted teaching.



**Michael Unser** (M’89–SM’94–F’99) is a Professor and Director of the Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland. He has a strong interest in sampling theories, multiresolution algorithms, wavelets, and the use of splines for image processing. From 1985 to 1997, he was with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD, conducting research on bioimaging and heading the Image Processing Group. He has published about 200 journal papers on those topics. His current research interests include biomedical image processing.

Dr. Unser is an EURASIP Fellow in 2009 and a member of the Swiss Academy of Engineering Sciences. He was the recipient of several international prizes, including three IEEE-SPS Best Paper Awards and two Technical Achievement Awards from the IEEE SPS in 2008 and the IEEE EMBS in 2010.



GitHub: <https://github.com/EatonL/BEEPS>

CSDN: <https://blog.csdn.net/bazhidao0031/article/details/81450815>