Machine Learning for Adaptive Bilateral Filtering

Iuri Frosio[a], Karen Egiazarian [a,b], and Kari Pulli [a]

[a] NVIDIA Research, 2701 San Tomas Expressway, Santa Clara, CA, USA;
[b] Tampere University of Technology, Department of Signal Processing, Tampere, Finland

ABSTRACT

We describe a supervised learning procedure for estimating the relation between a set of local image features and the local optimal parameters of an adaptive bilateral filter. A set of two entropy-based features is used to represent the properties of the image at a local scale. Experimental results show that our entropy-based adaptive bilateral filter outperforms other extensions of the bilateral filter where parameter tuning is based on empirical rules. Beyond bilateral filter, our learning procedure represents a general framework that can be used to develop a wide class of adaptive filters.

Keywords: Denoising, adaptive bilateral filter, machine learning, training, optimization.

1. INTRODUCTION

Bilateral filtering (BF) smooths an image while preserving strong edges.[1] It has been used in applications ranging from image denoising[2] to edge enhancement,[3] exposure correction,[4] and tone mapping.[5] To preserve edges and reduce noise, BF uses the weighted sum of the pixels in a local neighborhood, where weights depend on both the spatial and intensity distance. More specifically, the output of the filter in position $\mathbf{x}$, $d(\mathbf{x})$ is given by:

$$d(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in N(\mathbf{x})} e^{-\frac{||\mathbf{y}-\mathbf{x}||^2}{2\sigma_d^2}} e^{-\frac{||t(\mathbf{y})-t(\mathbf{x})||^2}{2\sigma_r^2}} t(\mathbf{x})}{\sum_{\mathbf{y}\in N(\mathbf{x})} e^{-\frac{||\mathbf{y}-\mathbf{x}||^2}{2\sigma_d^2}} e^{-\frac{||t(\mathbf{y})-t(\mathbf{x})||^2}{2\sigma_r^2}}}, \quad (1)$$

where $t(\mathbf{x})$ is the noisy image, $N(\mathbf{x})$ is the neighborhood of $\mathbf{x}$, and σ_d and σ_r are the filter parameters that control the fall-off of the weights in the spatial and intensity domains. Improvements to basic BF include Adaptive Bilateral Filter (ABF) that adapts the filter parameters to the global[2] or local characteristics of the image.[2,6]

Based on the experimental results obtained on a large set of natural images, Zhang and Gunturk[2] reported that the optimal σ_d value is relatively insensitive to the noise standard deviation σ_n and it is generally in the range $[1.5 - 2.1]$, while the optimal σ_r is approximately linearly related to σ_n. This allows setting a global, reasonable σ_r value for an image corrupted by noise with known standard deviation σ_n, but the same authors suggested that future research should look for spatially adaptive parameters selection. In the same paper and based on their empirical rule for setting σ_r and σ_d, Zhang and Gunturk[2] combine wavelet-based multiresolution analysis and bilateral filter to get an adaptive filter, whose computational complexity is however larger than that of the original BF. Furthermore, the optimal value for σ_d is likely to be a function of the image size and resolution, although this has not been deeply investigated by the authors. Another interesting attempt to develop an ABF is described by Qi et al.,[6] where σ_d is fixed as suggested by Zhang and Gunturk,[2] while σ_r is modulated across the image as a function of the global noise variance and the local signal standard deviation. The intuition behind this empirical rule is that close to the edges the intensity domain of the filter should be small to collect only those data that belong to the same grey level cluster, thus preserving the local edge. In this sense, the local variance of the image can be regarded as a local feature that guides the behavior of the filter. Similar approaches have been described also by Zhang and Allebach,[3] where the characteristics of the filter are locally tuned to suppress noise and enhance edges, as well as by Wong,[7] where local phase coherency is measured and used to roughly identify edges and consequently modulate σ_d and σ_r across the image.

Author contact: ifrosio@nvidia.com

In all the cases described above, empirical rules derived from intuition and experience are employed to determine the "sub-optimal" relation between the local characteristics of the image and the parameters of ABF. This is, however, an error-prone procedure, because of the unavoidable approximations in the signal and noise models, and because of the difficulty of catching the complex relations between large sets of parameters, image features, and the filter output.

Machine learning can be used to avoid these issues and learn the relation between the characteristics of a given image and the optimal filter parameters at a local scale. Such approach has already been employed by Hammond and Simoncelli,[8] where a spatially varying decision function learned from the image data (minimizing a weighted least-squares error function) was used to properly mix the outputs of two sliding wavelet denoising filters, achieving an image quality superior to either method. Similarly, in Zhu and Milanfar,[9] spatially variant filter parameters are modulated based on a training procedure. We developed a learning procedure which is characterized by a far more general formulation with respect to these approaches; this makes it potentially useful for the development of a large class of adaptive filters. The method builds a set of entropy-based local image descriptors, extracted from the noisy image and used to modulate σ_r and σ_s across the image. A supervised learning procedure is then used to estimate the parameters of a logistic function that relates the local image characteristics with the optimal filter parameters. We provide a detailed validation of the learning framework, applied to the specific case of BF. The filter developed with the proposed framework will be indicated as Entropy-based Adaptive Bilateral Filter (EABF). Experiments performed at different noise levels demonstrate that this learning procedure produces significant improvement at any noise level, with the most significant improvements achieved for high noise levels. In Sec. 4, we analyze in detail the difference between the parameter modulation strategy learned within our framework and the empirical ones proposed by Qi et al.[6] and by Zhang and Gutturnuk.[2] Finally, we perform an extensive comparison of the filter output demonstrating that EABF outperforms both traditional BF and some of its recently proposed variants.

2. METHOD

2.1 A general framework for developing adaptive filters

An adaptive version of BF, ABF, is obtained using different values of the filter parameters for each pixel, i.e., by defining two maps $\sigma_d = \sigma_d(x, y)$ and $\sigma_r = \sigma_r(x, y)$. We illustrate a general supervised learning method for developing an adaptive filter, that in the specific case of EABF permits map estimation using the local properties of the image in position (x, y). We adopt the following notation:

- For each pixel (x, y), $\mathbf{f}_{x,y} = \left[f^0_{x,y}\ f^1_{x,y}\ ...\ f^{F-1}_{x,y}\right]^T$ is a vector with F scalar features describing the local image characteristics; the first element of $\mathbf{f}_{x,y}$ is equal to one, $f^0_{x,y} = 1$. The actual features used for EABF are described in Sec. 2.3.
- The vector $\mathbf{p}_{x,y} = \left[p^0_{x,y}\ p^1_{x,y}\ ...\ p^{P-1}_{x,y}\right]^T$ contains the P parameters of the adaptive filter in (x, y); for EABF, $P = 2$, $p^0_{x,y} = \sigma_r(x, y)$ and $p^1_{x,y} = \sigma_d(x, y)$.
- A logistic function relates $p^p_{x,y}$ and $\mathbf{f}_{x,y}$ as:

$$p^p_{x,y} = p^p_{Min} + \frac{p^p_{Max} - p^p_{Min}}{1 + e^{-\boldsymbol{\theta}^T_p \mathbf{f}_{x,y}}}, \tag{2}$$

  where $\boldsymbol{\theta}_p = \left[\theta^0_p\ \theta^1_p\ ...\ \theta^{F-1}_p\right]^T$ contains F unknown elements and each $p^p_{x,y}$ takes a value between p^p_{Min} and p^p_{Max}. The complete vector is $\boldsymbol{\theta} = \left[\boldsymbol{\theta}_0\ \boldsymbol{\theta}_1\ ...\ \boldsymbol{\theta}_{P-1}\right]^T$.
- $\{\mathbf{t}^j\}_{j=0..M-1}$ is a set of M noise-free training images; $\{\mathbf{t}^j_n\}_{j=0..M-1}$ is the corresponding set of noisy images, obtained by corrupting each $\mathbf{t}^j$ by Additive Gaussian White Noise (AWGN) with standard deviation σ_n. Notice that here and in the following we use the vectorized, 1D representation of a 2D image.
- The EABF filter produces a denoised image $\mathbf{d}^j = \mathbf{d}^j\left(\mathbf{t}^j_n, \boldsymbol{\theta}\right)$ from $\mathbf{t}^j_n$; the filtering parameters $\sigma_d(x, y)$ and $\sigma_r(x, y)$ for the pixel (x, y) are obtained from $\mathbf{f}_{x,y}$, $\boldsymbol{\theta}$, and Eq. 2.

- $E = E\left(\{\mathbf{t}^j\}, \{\mathbf{t}_n^j\}, \boldsymbol{\theta}\right)$ is a cost function, typically related to the quality of the denoised images. We define $E = -\sum_{j=0}^{M-1} PSNR(\mathbf{d}^j, \mathbf{t}^j)$, where $PSNR(\mathbf{u}, \mathbf{v})$ is the Peak Signal-to-Noise Ratio of $\mathbf{u}$ and $\mathbf{v}$.

The supervised learning procedure consists of the minimization of $E = E\left(\{\mathbf{t}^j\}, \{\mathbf{t}_n^j\}, \boldsymbol{\theta}\right)$ with respect to $\boldsymbol{\theta}$. After this step, the estimated $\boldsymbol{\theta}$ and Eq. 2 can be used to compute $\mathbf{p}_{x,y}$ for each pixel of any image, out of the training set.

2.2 Optimization

Because of the non-linearities in E as well as in the filtering procedure, a closed-form solution to identify the value of $\boldsymbol{\theta}$ associated to the minimum of E is not available and an iterative optimization algorithm is required. We use the steepest-descent method,[10] where the value of $\boldsymbol{\theta}$ at the k^{th} iteration, $\boldsymbol{\theta}^k$, is updated as:

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k - \mu \frac{\partial E}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^k}, \tag{3}$$

where $0 < \mu < 1$ is a small damping parameter. However, analytical computation of the derivatives of E with respect to $\boldsymbol{\theta}$ is generally hard or even not feasible. To overcome this issue, we numerically approximate $\partial E/\partial \boldsymbol{\theta}|_{\boldsymbol{\theta}^k}$ by sampling E in $E\left(\{\mathbf{t}^j\}, \{\mathbf{t}_n^j\}, \boldsymbol{\theta}^k + \boldsymbol{\delta\theta}^k\right)$ for PF different small values of $\boldsymbol{\delta\theta}^k$.[10] Notice that this is generally computationally intensive, as it requires applying the filter and computing the cost function PF times for each of the M images of the training set, for each iteration. To avoid overfitting, we employed an additional set of N images for validation, $\{\mathbf{v}^j\}_{j=0..N-1}$. At each iteration, the cost function $E\left(\{\mathbf{v}^j\}, \{\mathbf{v}_n^j\}, \boldsymbol{\theta}^k\right)$ is evaluated on this set of images; at the end of the learning procedure, the $\boldsymbol{\theta}^k$ associated to the lowest $E\left(\{\mathbf{v}^j\}, \{\mathbf{v}_n^j\}, \boldsymbol{\theta}^k\right)$ is selected. This prevents identifying $\boldsymbol{\theta}^k$ which is optimal for the training set, but does not generalize well to other images.

2.3 Entropy-based local image descriptors

Local image descriptors are necessary to properly guide the modulation of the filtering parameters across the image. Entropy-based image descriptors have already been applied for texture classification,[11] edge detection,[12] and development of adaptive filters based on local image content.[13] We introduce two entropy-based features that can be used in $\mathbf{f}_{x,y}$ for EABF. Let us first introduce the Shannon's entropy of a vector $\mathbf{i}$ of integers (in the range $[0, B]$):

$$e_{\mathbf{i}} = -\sum_{h=0}^{B} log_2(p(h))\, p(h), \tag{4}$$

where $p(h)$ is the probability of h in $\mathbf{i}$. The entropy is minimal ($e_{\mathbf{i}} = 0$) when all the elements share the same value h (i.e., $p(h) = 1$ for that h), and it is maximal ($e_{\mathbf{i}} = log_2(B+1)$) when $p\,(h)$ has a uniform distribution. Entropy can be used to describe the local characteristics of an image $i\,(x, y)$ as described below.

Let us define the gradient image as $g\,(x, y) = \sqrt{i_x(x,y)^2 + i_y(x,y)^2}$, where $i_x(x, y)$ and $i_y(x, y)$ indicate the gradient of the image in x and y directions (these are computed here as $i_x(x, y) = i(x, y) - i(x-1, y)$ and $i_y(x, y) = i(x, y) - i(x, y-1)$). For the noise-free images in Fig. 1a - 1d, Table 1 shows the entropies computed for each image ($e_{\mathbf{i}}$) and for the corresponding gradient image ($e_{\mathbf{g}}$); $e_{\mathbf{g}} = 0$ for the flat and gradient area, where the norm of the gradient is constant, whereas $e_{\mathbf{i}} = 0$ in the flat area and $e_{\mathbf{i}} = 6.0$ in the gradient area. Both $e_{\mathbf{i}}$ and $e_{\mathbf{g}}$ are slightly larger in a textured area (Fig. 1c), where grey levels and gradient norms assume few (only two in this example) different values because of the regularity of the texture. In the complex image including edges and gradients (Fig. 1d), both $e_{\mathbf{i}}$ and $e_{\mathbf{g}}$ are high. These results suggest that the pair $[e_{\mathbf{i}}, e_{\mathbf{g}}]$ can be used to discriminate between flat, gradient, textured, and complex areas, at least in the absence of noise.

Let us assume now that the grey level image is corrupted by AWGN with variance σ_n^2. The entropies of the images in Fig. 1, after their corruption with AWGN, $\sigma_n = 5$, are reported in Table 1. Also in this case, $e_{\mathbf{g}}$ significantly increases for the gradient area with respect to the flat area, whereas both $e_{\mathbf{i}}$ and $e_{\mathbf{g}}$ slightly increase from the flat area to the textured one. The entropies of the complex areas remain the highest even in a presence

Table 1: Entropies for the flat (F), gradient (G), textured (T), and complex (C) images shown in Fig. 1, corrupted by AWGN with standard deviation σ_n, with and without grey level clipping.

σ_n	0				5				5			
Clipping	No				No				Yes			
Type	F	G	T	C	F	G	T	C	F	G	T	C
$e_{\mathbf{i}}$	0.0	6.0	1.0	5.6	4.4	6.2	5.4	6.2	1.8	4.9	4.1	3.3
$e_{\mathbf{g}}$	0.0	0.0	1.2	5.5	4.2	4.2	5.6	6.5	3.2	4.0	5.4	5.5

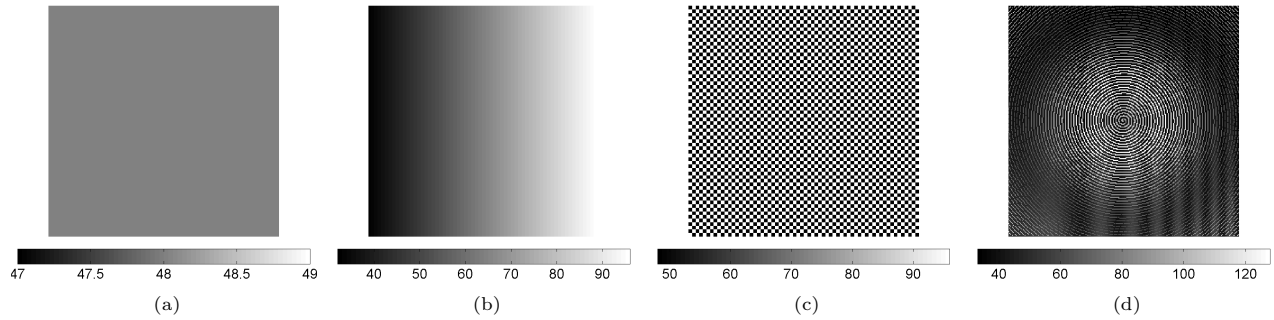


Figure 1: Panels (a-d) show a 128×128 flat, gradient, textured, and complex, noise-free area. Entropies for these images are reported in Table 1.

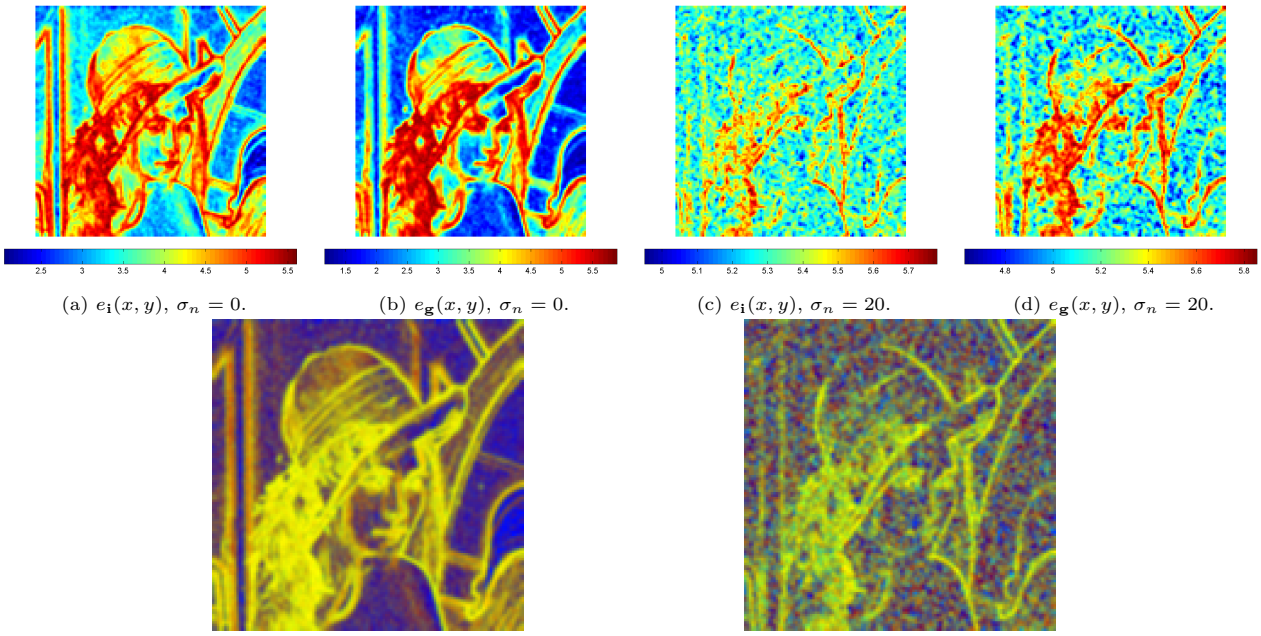
of noise. Beyond noise, grey level clipping due to the limited sensor dynamics may occur in the brightest and darkest areas of real images, leading to a loss of information content. Table 1 also shows $e_{\mathbf{i}}$ and $e_{\mathbf{g}}$ in the presence of clipping (achieved in this case by shifting the grey levels of the noisy images by -32 grey levels and clipping to $[0, 255]$).

Overall, Table 1 suggests that a vector $[e_{\mathbf{i}}, e_{\mathbf{g}}]$ can be used to discriminate between flat, gradient, textured, and complex images, even in the presence of grey level clipping and/or noise. To obtain a local description of an image in terms of entropies, we therefore propose to compute $e_{\mathbf{i}}$ and $e_{\mathbf{g}}$ on a sliding window of size $W \times W$ (here the patch size W is 8) with a shift between consecutive windows of 2 pixels in each direction. To limit the effect of noise, the image is pre-filtered with a moving average 3×3 filter before computing $e_{\mathbf{g}}$. For each pixel (x, y), the final values $e_{\mathbf{i}}(x, y)$ and $e_{\mathbf{g}}(x, y)$ are obtained by averaging the value of $e_{\mathbf{i}}$ and $e_{\mathbf{g}}$ over all the patches including the pixel (x, y). Figure 2 shows the computed entropy maps for the *lena* image, in absence and presence of noise. Notice that, although high correlation exists between $e_{\mathbf{i}}$ and $e_{\mathbf{g}}$, these also convey different information in some parts of the image even in presence of noise (e.g., the face details are more evident in $e_{\mathbf{g}}(x, y)$ than in $e_{\mathbf{i}}(x, y)$ in the noisy case).

2.4 Training, validation, and testing datasets

We built a training set $\{\mathbf{t}_n^j\}_{j=1..M}$ composed of fifteen images taken from the McMaster[14] dataset and twenty images from the Kodak[15] dataset, for a total of $M = 35$ training images. Three images from the McMaster and three from the Kodak dataset were used for validation during the training procedure. Overall, the training dataset included natural and artificial scenes, piecewise constant and texturized areas, as well as images characterized by dominant frequencies both in the low or high range.

Similar to the results presented by Zhang and Gunturk[2] and Qi et al.,[6] we used for testing a set of well-known images composed by *barbara*, *boats*, *goldhill*, *house*, *lena*, and *peppers*. Different levels of noise ($\sigma_n = \{5, 10, 20, 30, 40\}$) were considered, clipping the image grey levels between 0 and 255 after adding the noise. We trained our model using $F = 3$ features, (i.e., using the the image and edge entropy features introduced before) through 1000 iterations of the steepest-descent algorithm; at the end of the training procedure, only small changes in the value of $\boldsymbol{\theta}$ were found between two consecutive iterations. The minimum and maximum values of σ_d were set to 0.0 and 5.0, whereas for σ_r the minimum and maximum values were set to 1.0 and $6\sigma_n$.



(a) $e_{\mathbf{i}}(x,y)$, $\sigma_n = 0$. (b) $e_{\mathbf{g}}(x,y)$, $\sigma_n = 0$. (c) $e_{\mathbf{i}}(x,y)$, $\sigma_n = 20$. (d) $e_{\mathbf{g}}(x,y)$, $\sigma_n = 20$.

(e) $\sigma_n = 0$. (f) $\sigma_n = 20$.

Figure 2: Local features $e_{\mathbf{i}}(x,y)$ and $e_{\mathbf{g}}(x,y)$ for the *lena* image, in absence and presence of noise. Panels (e) and (f) show a false color representation of the same image, derived from $e_{\mathbf{i}}(x,y)$ and $e_{\mathbf{g}}(x,y)$, with and without noise.

3. RESULTS

We have compared the results achieved by the EABF (rightmost columns in Table 2) with different implementations of BF. In particular, we first considered the empirical rule suggested by Zhang and Gunturk[2] to set the parameters of BF, where σ_r is computed as a linear function of σ_n as $\sigma_r = 2\sigma_n$, whereas $\sigma_d = 1.8$ independent from the noise level. Results from this filter are reported in the second column of Table 2, and they represent the typical image quality achieved by BF when parameter setting is performed through an empirical approach. EABF achieves a PSNR which is on the average +1.02dB higher than that achieved by BF,[2] with a maximum improvement of 2.80dB for *lena* and $\sigma_n = 40$. The advantage of EABF with respect to BF[2] is generally higher for higher noise levels. Only for the *barbara* image and $\sigma_n = 10, 20, 30$ EABF achieves a PSNR which is lower (-0.69dB in the worst case) than BF.[2]

A more challenging test for EABF was obtained comparing its output with the best output achievable by BF. The best BF-filtered images were obtained by identification of the pair (σ_d, σ_r) that maximizes the PSNR for each image in the testing set and for each noise level (this can be done by an extensive search or, within the proposed learning framework, using $F = 1$ and a unique training image). The corresponding PSNRs are reported in the first column of Table 2: as expected, these are always significantly higher than the PSNRs achieved by BF,[2] thus suggesting that the empirical rule for setting σ_d and σ_r produces only sub-optimal results. Nevertheless, images filtered with EABF generally have a higher PSNR with respect to that produced by BF and optimal parameter setting, with the only exception of *barbara* image. The average advantage of EABF with respect to BF with optimal parameter setting is +0.33dB, with larger improvement generally obtained for higher noise levels. Overall, EABF produces an average PSNR which is higher than the maximum PSNR achievable by non-adaptive BF, thus demonstrating the large advantage of an adaptive filter over a non-adaptive one.

The third column in Table 2 reports the PSNR for the ABF filter proposed in by Qi et al.,[6] where the value of $\sigma_r(x,y)$ is computed for each pixel from σ_n and the local variance of the signal, $\sigma_s(x,y)$ (the exact equation is reported in Table 2). EABF provides in this case a PSNR which is on the average 0.41dB higher (also in this case the only exception is the *barbara* image), in this case independently from the noise level. These results

suggest that the proposed learning procedure can catch more complex relations between the filter parameters (than the one adopted by Qi et al.[6]) and, therefore, better modulate them across the filtered images.

A more visual comparison of the results is illustrated in Fig. 3, which shows the ground truth and noisy *lena* image, for $\sigma_n = 20$, and the same image, filtered with BF,[2] $\sigma_d = 1.8$, $\sigma_r = 2\sigma_n$, with ABF,[6] $\sigma_d = 1.8$, $\sigma_r = \sigma_n^2/(0.3\sigma_s)$ and with EABF. Figure 3 also shows the residual noise for each filter, together with a zoom on the eye area. For all the filters, the larger residual error is close to the edges of the image. For non-adaptive BF,[2] the residual error (Fig. 3f) is higher in the flat areas (like the background or the shoulder skin) with respect to EABF (Fig. 3l). This is explained considering that, for BF,[2] the spatial extent of the Gaussian of the bilateral filter is constant across the image ($\sigma_d = 1.8$), whereas EABF produces a smoother image in the flat areas by adoption of a larger σ_d (approximately comprised between 1.8 and 2.6, see Fig. 4d). Comparison of the zooms in the eye area shows more noise at high frequency for BF[2] with respect to EABF.

Figures 3g-h show the output of ABF[6] obtained with $\sigma_d = 1.8$ and $\sigma_r = \sigma_n^2/(0.3\sigma_s)$. Comparison with EABF shows a larger error close to the edges for this algorithm with respect to the proposed one. This is evident in Fig. 3h, where high-frequency noise is present in the eyebrow area, whereas this is filtered out by EABF (Fig. 3k).

Visual inspection of Fig. 4 highlights the different modulation of the filter parameters operated by Qi et al.'s ABF[6] and our EABF. It shows the $\sigma_d(x, y)$ and $\sigma_r(x, y)$ maps adopted by EABF and the $\sigma_r(x, y)$ used by Qi et al. Following the empirical rule proposed by Zhang et al., σ_r increases with the noise level for both EABF and for ABF of Qi et al., which uses values of σ_r that are largely different in the same image, with lower σ_r in the edge areas. On the other hand, the modulation strategy learned using our framework uses largely different values of σ_d (which is generally smaller in the edge areas and larger in the flat zones), whereas σ_r is only slightly reduced in the edge areas with respect to the flat ones. Overall, the learned parameter modulation strategy is more complex than the empirical ones reported by Zhang and Gunturk[2] or by Qi et al.,[6] but also more effective as demonstrated by the PSNRs reported in Table 2.

Figure 4 furthermore shows that the EABF modulation of $\sigma_d(x, y)$ and $\sigma_r(x, y)$ across the image produces similar patterns independent from the noise level, although some of the image details are more evident for low noise levels. This fact can be explained considering that high level of noise may significantly alter the vector of features $\mathbf{f}_{x,y}$ (see Fig. 2), leading to a less reliable description of the local image characteristics and therefore to a less precise map for the filter parameters. Despite of this, the improvement for EABF with respect to BF[2] and ABF[6] remains consistent even for high noise levels.

4. DISCUSSION

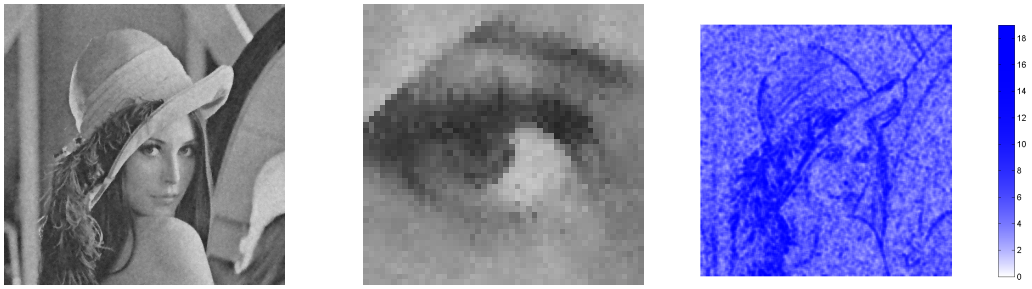
We have introduced a general framework for developing an adaptive filter, and we have successfully applied it to the bilateral filter. The proposed framework is, however, more general, and it can be applied to any filter that can be expressed in the form described in Sec. 2. We have illustrated the framework using BF because of the wide applicability and simplicity of this filter.

The framework allows learning the relation between the local image characteristics, described by the feature vector $\mathbf{f}_{x,y}$, and the "optimal" filter parameters for the pixel (x, y), $\mathbf{p}_{(x,y)}$. Optimality is defined by the user-defined cost function $E = E(\{\mathbf{t}^j\}, \{\mathbf{t}_n^j\}, \boldsymbol{\theta})$ and it, therefore, represents a quite general concept in our framework. Traditionally, the objective is the maximization of the image quality; this leads to a cost function similar to the one based on PSNR we adopted to develop EABF. On the other hand, in applications such as mobile computing it may be preferable to identify a trade-off between image quality and optimal usage of the limited hardware resources (e.g., energy or bandwidth). Our framework supports this by a slight modification of the cost function:

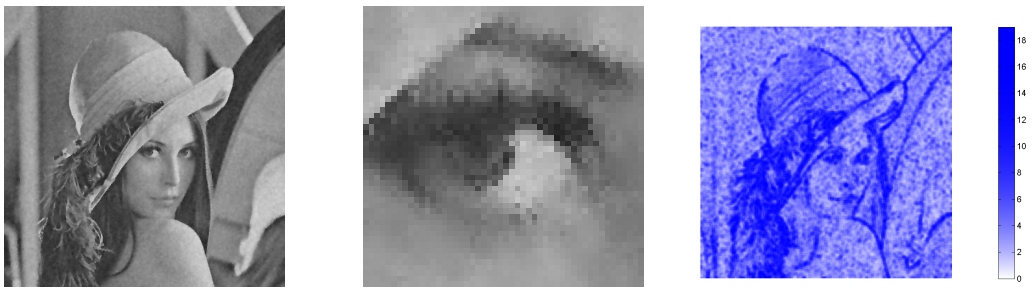
$$E = E\left(\{\mathbf{t}^j\}, \{\mathbf{t}_n^j\}, \boldsymbol{\theta}\right) = -\sum_{j=0}^{M-1} PSNR(\mathbf{d}^j, \mathbf{t}^j) + \lambda P\left(\mathbf{t}_n^j, \boldsymbol{\theta}\right), \tag{5}$$

where $P\left(\mathbf{t}_n^j, \boldsymbol{\theta}\right)$ is a penalty term for filtering the image $\mathbf{t}_n^j$ using the filter parameters computed from $\boldsymbol{\theta}$ and Eq. 2, and it can represent computational cost as well as bandwidth use, memory utilization and so on, whereas λ is a regularization parameter that fixes the weight of P with respect to the image quality term.

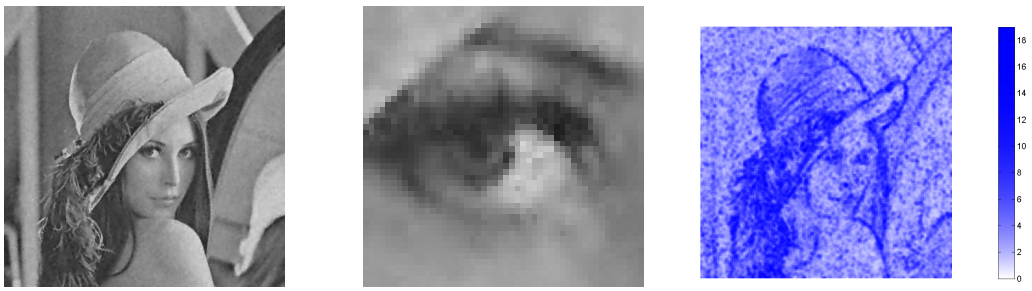
(a) Ground truth. (b) Ground truth—detail. (c) Noisy, $\sigma_n = 20$.



(d) BF,[2] $\sigma_d = 1.8$, $\sigma_r = 2\sigma_n$, PSNR = 30.02dB. (e) BF[2]—detail. (f) Absolute residual error, BF.[2]

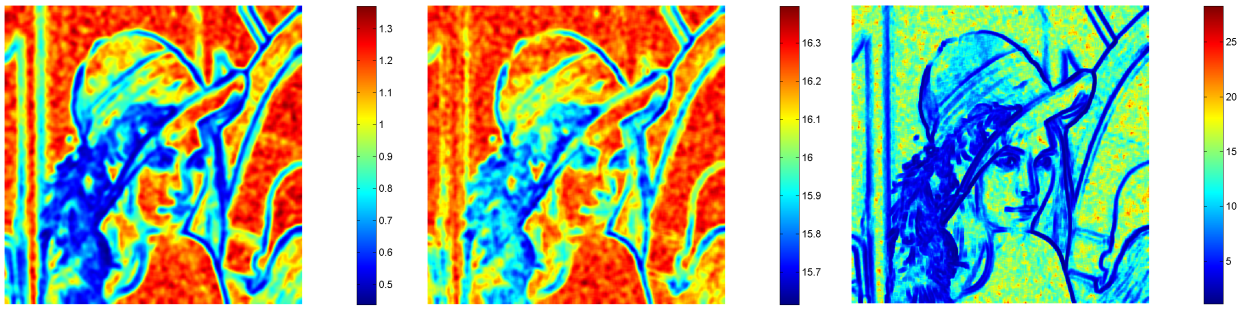


(g) ABF,[6] $\sigma_d = 1.8$, $\sigma_r = \sigma_n^2/(0.3\sigma_s)$, PSNR = 31.05dB. (h) ABF[6]—detail. (i) Absolute residual error, ABF.[6]

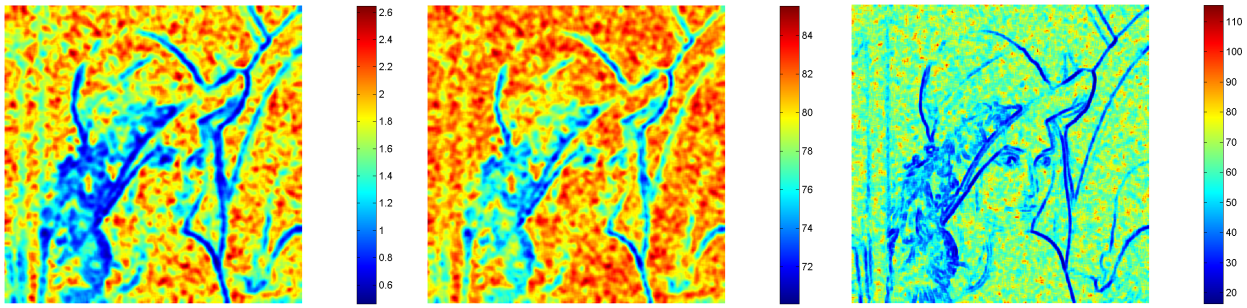


(j) EABF, PSNR = 32.08dB. (k) EABF—detail. (l) Absolute residual error, EABF.

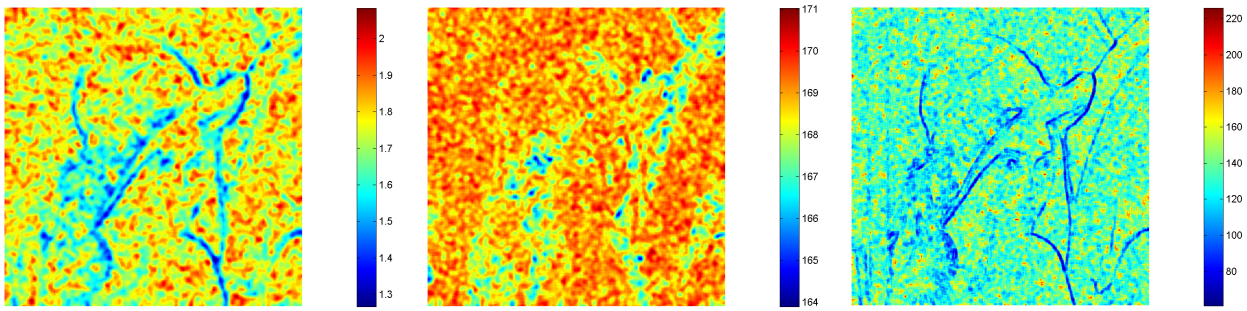
Figure 3: Panel (a) shows the ground truth *lena* image and panel (c) shows the same image corrupted by AWGN. The rectangles in panel (a) (better visualized in electronic version) indicate the area used for additional analysis of the modulation of the filter parameters (see Fig. 5). A zoomed detail in the eye area of the ground truth image is shown in panel (b). The remaining panels in the leftmost column show the image filtered with BF,[2] ABF,[6] and EABF, with zooms in the central column. Panels in the rightmost column show the corresponding absolute residual errors, filtered with a 7×7 moving average filter to better highlight the error distribution across the different areas of the image (better visualized in electronic version).



(a) $\sigma_d(x, y)$, EABF, $\sigma_n = 5$. (b) $\sigma_r(x, y)$, EABF, $\sigma_n = 5$. (c) $\sigma_r(x, y)$, ABF,[6] $\sigma_n = 5$.

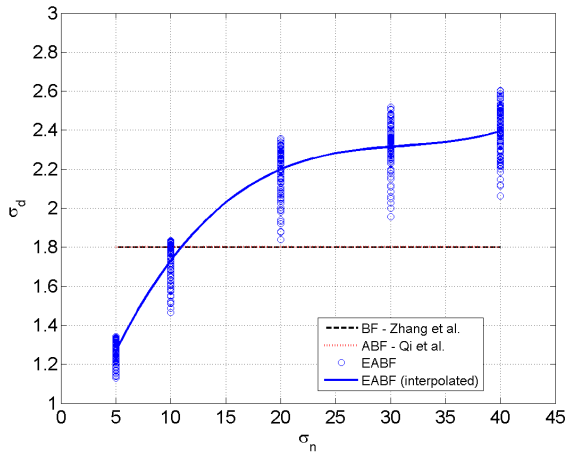


(d) $\sigma_d(x, y)$, EABF, $\sigma_n = 20$. (e) $\sigma_r(x, y)$, EABF, $\sigma_n = 20$. (f) $\sigma_r(x, y)$, ABF,[6] $\sigma_n = 20$.

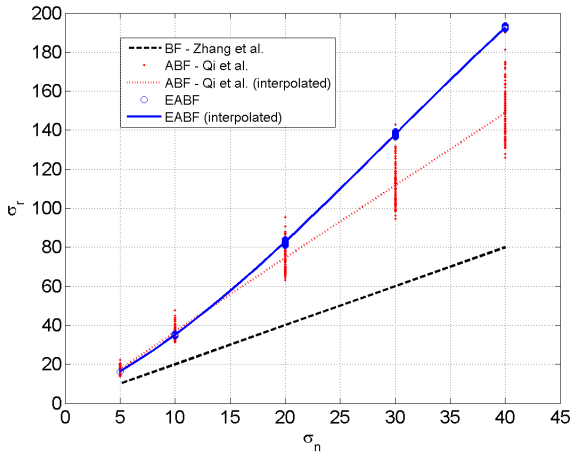


(g) $\sigma_d(x, y)$, EABF, $\sigma_n = 40$. (h) $\sigma_r(x, y)$, EABF, $\sigma_n = 40$. (i) $\sigma_r(x, y)$, ABF,[6] $\sigma_n = 40$.

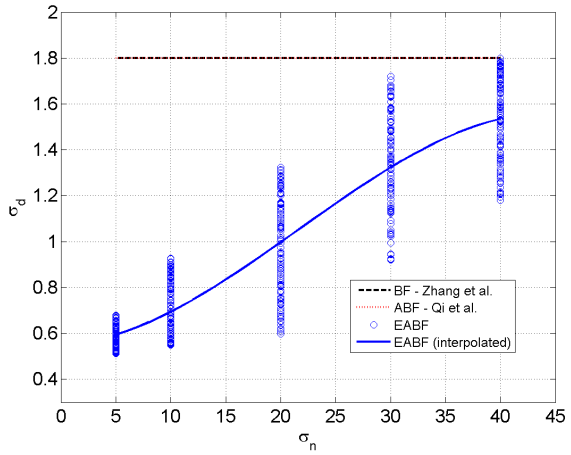
Figure 4: $\sigma_d(x, y)$ and $\sigma_r(x, y)$ for EABF and ABF,[6] for the *lena* image and different noise levels. For EABF, σ_r generally increases with σ_n, as empirically suggested by Zhang and Gunturk for BF.[2] However, the parameter modulation learned with our method prefers large variations of σ_d over changes of σ_r, differently from the rule suggested by Qi et al. in their ABF.[6]



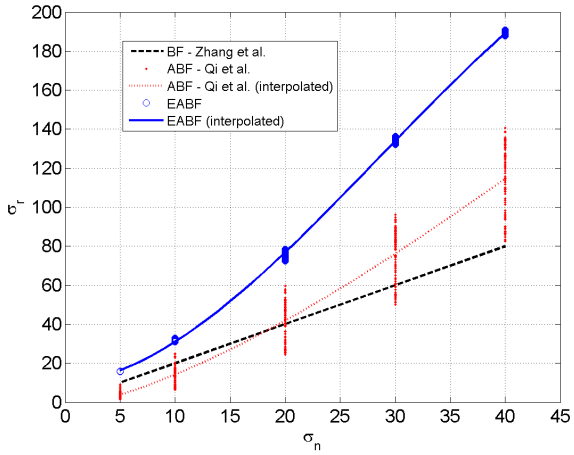
(a) $\sigma_d(x, y)$ for the pixels in the rectangle in the background area.



(b) $\sigma_r(x, y)$ for the pixels in the rectangle in the background area.



(c) $\sigma_d(x, y)$ for the pixels in the rectangle in the eye area.



(d) $\sigma_r(x, y)$ for the pixels in the rectangle in the eye area.

Figure 5: $\sigma_d(x, y)$ and $\sigma_r(x, y)$ vs. $\sigma_n(x, y)$ for the pixels in the rectangles in the background and eye areas in Fig. 3a, for BF,[2] ABF[6] and EABF. For both BF[2] and ABF,[6] σ_d is assumed to be constant, whereas our learned modulation strategy for this parameter is described by a complex curve the shape of which changes at different positions within the image. In the case of σ_r, the learned modulation strategy is characterized by an evident non-linearity, which is not present in BF[2] and less evident in ABF.[6]

Defining the training procedure in the context of machine learning offers a significant advantage over a more traditional approach, where human intuition and experience lead to empirical rules for adapting the filter parameters to the local image characteristics. Comparison with the works of Zhang and Gunturk[2] and Qi et al.[6] offers good examples of such advantage.

Based on the average results obtained by filtering 200 images with BF and different parameter sets, Zhang and Gunturk[2] first suggested that a good range for σ_d is roughly within the [1.5-2.1] interval. The $\sigma_d = \sigma_d(x, y)$ maps obtained with our method and illustrated in Fig. 4 for various noise levels suggest that such intuition is mainly correct, although different areas of the images benefit from a choice of σ_d which is well outside these boundaries (for instance, lower σ_d values should be used close to edges). Lower σ_d values also seem to be more appropriate for very low noise levels (see Fig. 4a, as well as the optimal σ_d values reported for BF in Table 2). Zhang and Gunturk additionally suggest that, once σ_d has been fixed, σ_n and the optimal value of σ_r are linearly related, although the slope of the line relating these two parameters depends on σ_d. The analysis of the filtering parameters obtained by the EABF in the rectangles in Fig. 3a, in the background and eye areas, confirms that

also this intuition is mainly correct, but EABF has the important advantage of automatically identifying the proper slope (Fig. 5b) as well as catching the non-linear relation (Fig. 5d) between σ_r and σ_n. Figures 5a and 5c also show that assuming a constant value for σ_d over the entire image represents an excessive simplification, as EABF uses significantly smaller values of σ_d in the eye area; furthermore, the relation of the optimal σ_d with respect to σ_n is in this case non-linear and hardly identifiable using intuition. Overall, this analysis shows how the proposed framework can identify complex, non-linear relations between the local image characteristics and the optimal filtering parameters; accomplishing the same task using human intuition and experience appears hard or even impossible, especially when filters more complex than BF are taken into consideration.

Another example of the advantages offered by the proposed learning procedure is evident by comparison with the ABF filter proposed by Qi et al.,[6] where σ_d is fixed and an empirical rule to modulate σ_r across the image is used. This leads to the adoption of a $\sigma_r(x, y)$ that increases with the image noise and it is generally much smaller close to the edges than in the flat areas. Analysis of Fig. 4 shows that EABF learned a significantly different parameter modulation strategy, where $\sigma_r(x, y)$ largely increases with the noise level, but it only slightly decreases in proximity of the edges, whereas $\sigma_d(x, y)$ has large variations across the image and in particular is smaller close to the edge and larger in the flat areas. The results in Table 2 demonstrate the superiority of the learned parameters modulation strategy with respect to the empirical one proposed by Qi et al. This furthermore shows the capability of the proposed learning procedure to catch complex relations between the filtering parameters and the local image characteristics.

A deeper visual inspection of Fig. 4 finally reveals that the local variance computed as by Qi et al.[6] is characterized by a higher-frequency content with respect to the parameter maps obtained with the proposed method. In the future we are therefore going to investigate the local image variance as additional feature that can contribute to a finer modulation of the filtering parameters.

In our experiments, we used only AWGN to corrupt the images. It is worth noticing that, for several, common distributions, the noise variance can be made constant through the application of a proper variance stabilizing transform.[16] In some cases, like for Poisson distribution, the noise distribution is also reasonably approximated as AWGN after variance stabilization; on the other hand, other noise distributions, like for instance impulse noise,[17] cannot be described by this model either before or after variance stabilization. Nevertheless, the proposed framework is general and it can be used with any noise distribution (e.g., Poisson,[18] mixture of Gaussian and Poisson with grey level clipping,[16] salt and pepper, and so on). To this aim, it is sufficient to use a different noise generator during the creation of the dataset $\{\mathbf{t}_n^j\}_{j+0..M-1}$.

In our experiments, we used a training set composed of $M = 35$ images. Because of the limited size of the dataset, the estimated $\boldsymbol{\theta}$ vector may be biased, leading to a decreased performance compared to those achievable using a larger training set, where images with widely varying characteristics can be used. This may explain why EABF produces inferior results on the *barbara* image which contains a lot of textured areas at different frequencies not present in the training dataset. The use of massive data requires, however, a very efficient implementation of the learning procedure. At present, only EABF has been implemented in parallel using CUDA to run on a GPGPU, whereas the rest of the learning procedure was written in C++ and run sequentially on a standard CPU. The typical training time for 1000 iterations of the approximate steepest-descent algorithm was 90 minutes on a Dell Alienware equipped with a Intel Core i7-4700MQ CPU @ 2.4GHz, 8G RAM, and an NVIDIA GeForce GT750M with 4G RAM. We are currently working on a full GPU implementation of the framework, that will avoid data transfer between the RAM and the GPU memory, leading to a significant speed-up and therefore offering the possibility to use a larger training set of images and to finally reduce the bias in the estimate of $\boldsymbol{\theta}$.

It is finally important to notice that the high computational cost of the learning procedure is mainly due to the high number (PF at each iteration) of evaluations of the cost function E (and therefore filtering of the M images in the training set), which is required to numerically compute $\partial E/\partial \boldsymbol{\theta}$. For simple filters like BF, analytically computing $\partial E/\partial \boldsymbol{\theta}$ is actually feasible, which could lead to a significant speed-up of the training procedure. On the other hand, this would make the proposed learning procedure less general, since it could not be applied to more complex filters where $\partial E/\partial \boldsymbol{\theta}$ cannot be computed analytically (like for instance in BM3D,[19] where some of the filter parameters are discrete, or in Total Variation denoising,[18] which is an iterative filtering procedure). Moreover, the development of any new adaptive filter would require explicitly computing $\partial E/\partial \boldsymbol{\theta}$. For these

reasons, we prefer the current formulation of the optimization procedure involving the numerical estimate of the gradient of the cost function E.

Overall, the results presented here demonstrate the practical use of the proposed framework in the case of BF. The results obtained with EABF outperform other adaptive filters obtained through empirical rules and intuition. Preliminary results (not reported here for reason of space) also showed that the same learning procedure can be applied to develop an adaptive Sliding Discrete Cosine Transform (DCT) filter,[20] where an adaptive threshold is used to eliminate DCT components locally associated to noise. The same framework was then applied to adaptively blending the output of EABF and Adaptive Sliding DCT filter, similar to the work described by Hammond and Simoncelli,[8] with significant image quality improvements. Based on the results described here, we are therefore going to apply our framework to the optimization of more complex filtering procedures, like BM3D[19] and Total Variation denoising.[18]

REFERENCES

1. C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, 1998.
2. M. Zhang and B. K. Gunturk, "Multiresolution bilateral filtering for image denoising," *IEEE TIP* **17**(12), 2008.
3. B. Zhang and J. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," in *ICIP*, 2007.
4. E. P. Bennett and L. McMillan, "Video enhancement using per-pixel virtual exposures," *ACM TOG* **24**(3), 2005.
5. F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM TOG* **21**(3), 2002.
6. M. Qi, Z. F. Zhou, J. Liu, J. Z. Cao, H. Wang, A. Q. Yan, D. S. Wu, H. Zhang, and L. N. Tang, "Image denoising algorithm via spatially adaptive bilateral filtering," *Advanced Materials Research* **760 - 762**, 2013.
7. A. Wong, "Adaptive bilateral filtering of image signals using local phase characteristics," *Signal Processing* **88**(6), 2008.
8. D. Hammond and E. Simoncelli, "A machine learning framework for adaptive combination of signal denoising method," in *ICIP*, 2007.
9. X. Zhu and P. Milanfar, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," *IEEE TIP* **19**(12), 2010.
10. J. Nocedal and S. J. Wright, *Numerical optimization*, Springer, New York, NY, 2. ed., 2006.
11. S. S. Mole and L. Ganesan, "Unsupervised texture classification of entropy based local descriptor using k-means clustering algorithm," *Int. J. of Computing* **10**(2), 2011.
12. M. A. El-Sayed, "Edges detection based on renyi entropy with split/merge," *Computer Engineering and Intelligent Systems* **3**(9), 2012.
13. A. Fathi and A. Naghsh-Nilchi, "Efficient image denoising method based on a new adaptive wavelet packet thresholding function," *IEEE TIP* **21**(9), 2012.
14. L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic Imaging* **20**(2), 2011.
15. "Kodak lossless true color image suite." http://r0k.us/graphics/kodak/.
16. A. Foi, "Clipped noisy images: Heteroskedastic modeling and practical denoising," *Signal Processing* **89**(12), 2009.
17. I. Frosio and N. A. Borghese, "Statistical based impulsive noise removal in digital radiography," *IEEE TMI* **28**(1), 2009.
18. I. Frosio, C. Olivieri, M. Lucchese, N. Borghese, and P. Boccacci, "Bayesian denoising in digital radiography: A comparison in the dental field," *Computerized Medical Imaging and Graphics* **37**(1), 2013.
19. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE TIP* **16**(8), 2007.
20. L. P. Yaroslavsky, K. O. Egiazarian, and J. T. Astola, "Transform domain image restoration methods: review, comparison, and interpretation," in *Photonics West 2001-Electronic Imaging*, 2001.

Table 2: PSNRs (in dB) for the test images, different levels of noise and different version of BF, ABF, and EABF. The method adopted by each filter to compute $\sigma_d(x,y)$ and $\sigma_r(x,y)$ is reported in the first two rows. For BF, the parameter values are reported within the brackets for each image and for each noise level. The bold character indicate the highest PSNR at each row of the table. The PSNR for EABF is typically 0.5dB higher than that of ABF,[6] independently from σ_n, and up to 2.8dB higher with respect to BF,[2] whose performance decreases faster for increasing σ_n values.

$\sigma_d(x,y)$		optimal			1.8	1.8	Eq. 2
$\sigma_r(x,y)$		optimal			$2\sigma_n$	$\sigma_n^2/(0.3\sigma_s)$	Eq. 2
image	σ_n	**BF**			**BF²**	**ABF⁶**	**EABF**
			σ_d	σ_r			
barbara (512 × 512)	5	36.13	(1.31,	10)	36.06	36.00	**36.27**
	10	**31.45**	(1.51,	19)	31.40	31.44	31.10
	20	27.11	(1.69,	40)	27.09	**27.36**	26.40
	30	25.07	(1.83,	67)	25.00	**25.32**	24.68
	40	23.94	(1.98,	101)	23.69	**24.11**	23.76
boat (512 × 512)	5	36.29	(0.87,	12)	36.04	35.92	**36.40**
	10	32.53	(0.99,	28)	32.17	32.03	**32.81**
	20	28.96	(1.20,	59)	28.48	28.75	**29.51**
	30	26.96	(1.34,	99)	26.31	26.98	**27.62**
	40	25.63	(1.47,	151)	24.72	25.68	**26.17**
goldhill (512 × 512)	5	36.50	(0.70,	15)	36.07	35.93	**36.54**
	10	32.64	(0.98,	28)	32.23	32.15	**32.78**
	20	29.32	(1.23,	64)	28.81	29.17	**29.65**
	30	27.71	(1.49,	109)	26.84	27.68	**28.01**
	40	26.66	(1.72,	172)	25.33	26.53	**26.82**
house (256 × 256)	5	37.69	(1.35,	10)	37.50	37.20	**37.81**
	10	34.00	(1.24,	28)	33.76	33.66	**34.37**
	20	30.31	(1.42,	62)	29.64	30.17	**31.11**
	30	28.20	(1.50,	108)	27.12	28.20	**29.24**
	40	26.86	(1.62,	163)	25.46	26.83	**27.78**
lena (512 × 512)	5	38.17	(0.97,	14)	37.86	37.64	**38.45**
	10	34.64	(1.18,	31)	34.09	34.18	**35.17**
	20	31.20	(1.45,	70)	30.02	31.05	**32.08**
	30	29.37	(1.67,	126)	27.78	29.24	**30.21**
	40	28.20	(1.89,	190)	25.97	27.83	**28.77**
peppers (512 × 512)	5	37.81	(1.23,	12)	37.74	37.37	**37.86**
	10	34.75	(1.37,	30)	34.31	34.22	**34.98**
	20	31.27	(1.62,	62)	30.40	31.22	**31.80**
	30	29.14	(1.76,	106)	27.85	29.30	**29.75**
	40	27.79	(1.92,	163)	25.95	27.71	**28.30**
average	5	37.10	(1.07,	12)	36.88	36.68	**37.22**
	10	33.33	(1.21,	27)	32.99	32.95	**33.53**
	20	29.69	(1.43,	59)	29.07	29.62	**30.09**
	30	27.74	(1.60,	102)	26.82	27.79	**28.25**
	40	26.51	(1.77,	157)	25.19	26.44	**26.93**
average	{5, 10, 20, 30, 40}	30.88	-	-	30.19	30.70	**31.21**