Edge-Preserving Filtering of Images with Low Photon Counts

John A. Lee, Xavier Geets, Vincent Grégoire, and Anne Bol

Abstract—Edge-preserving filters such as local M-smoothers or bilateral filtering are usually designed for Gaussian noise. This paper investigates how these filters can be adapted in order to efficiently deal with Poissonian noise. In addition, the issue of photometry invariance is addressed by changing the way filter coefficients are normalized. The proposed normalization is additive, instead of being multiplicative, and leads to a strong connection with anisotropic diffusion. Experiments show that ensuring the photometry invariance leads to comparable denoising performances in terms of the root mean square error computed on the signal.

Index Terms—Edge-preserving filter, local M-smoothers, bilateral filtering, anisotropic diffusion, photometry invariance, Poissonian noise.

1 Introduction

Many fields of imaging face the difficulty of providing meaningful information in poorly illuminated environments. Microscopy and optical astronomy, with the use of telescopes, enter into well-known domains where low photon counts are an issue. In medical imaging, this problem also occurs in modalities that involve high-energy photons such as positron emission tomography [1] (PET) and single photon emission computed tomography [2] (SPECT). In this case, the difficulty stems from the poor efficiency of the photon detectors. Therefore, the resulting images combine two weaknesses: statistical noise is consequential, and they have a low resolution. Edge-preserving filters [3] address this problem in an appealing way: They attenuate noise without smoothing the images, that is, without noticeably decreasing the resolution. This property is useful in numerous applications of PET and SPECT, where quantitative measures are performed on the images, in some region of interest, such as a tumor or a patient organ. In these applications, a filter that blurs edges and transitions eventually increases the spillover between contiguous regions of interest and jeopardizes their accurate delineation.

However, the use of edge-preserving filters suffer from two limitations. First, the majority of these filters are designed for i.i.d. Gaussian noise. This prevents us from applying them as such to images with low photon counts, because the pixel intensities actually follow a Poissonian distribution and so, the noise variance is not constant over the image. Second, the filters should also give some guarantees about the invariance of the total photon count (the sum of all pixel counts), either in the delineated region or in the whole image. As edges are never perfectly sharp in practice, the first condition is difficult to fulfill, and only the second milder constraint will be considered in this paper.

The literature indicates that Gaussian filtering [4] is used in PET [1], [5] and SPECT [6], [7], [8] imaging, because it is simple and preserves the total photometry; on the other hand, it lacks the property of edge preservation and degrades the resolution. Anisotropic diffusion [9] also preserves the photometry, which is edge preserving but relies on an additive noise model. If one preprocesses the image with a variance-stabilizing transform (VST) such as Anscombe's [10], then anisotropic diffusion can deal with Poissonian noise [11] but looses the property of photometry invariance. Bilateral filtering [12], [13], [14] and local M-smoothers [15], [16] also preserve edges, but neither guarantee the photometry invariance nor work optimally with Poissonian noise. In this context, our aim is twofold. First, we derive edge-preserving filters based on local M-smoothers that ensure the photometry invariance. Next, we adapt these filters in order to deal with Poissonian noise.

The remainder of this paper is organized as follows: In Section 2, we define the image model and present basic denoising schemes like Gaussian filtering. Section 3 deals with the invariance of the total photometry in an image. Edge-preserving denoising techniques such as local M-smoothers and bilateral filtering are briefly reviewed in Section 4. Additive normalization for these filters is introduced in Section 5, along with the corresponding objective functions and some examples. Section 6 relates additive normalization with anisotropic diffusion. Section 7 explains how we can adapt edge-preserving filters and additive normalization to Poissonian data and how this allows us to ensure the photometry invariance. Experiments and results with sharp and blurred images are detailed in Section 8. Finally, Section 9 gathers the conclusions.

2 Image Model and Local Smoothing

We consider the case of imaging devices with low photon counts. The resulting images are made of single-valued pixels (or voxels in 3D) whose intensity value is representative of the number of photons emitted in the corresponding region of the space. When photon counts are low, noise is Poissonian, and

• *The authors are with the Molecular Imaging and Experimental Radiotherapy Unit (IMRE), Université Catholique de Louvain, Avenue Hippocrate 54/69, B-1200 Brussels, Belgium.*
E-mail: {John.Lee, Anne.Bol, Vincent.Gregoire}@uclouvain.be, Xavier.Geets@imre.ucl.ac.be.

Manuscript received 18 Apr. 2007; revised 20 Sept. 2007; accepted 10 Dec. 2007; published online 3 Jan. 2008.
Recommended for acceptance by W. Föstner.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-04-0221.
Digital Object Identifier no. 10.1109/TPAMI.2008.16.

the assumption of an image-independent noise model does not hold. For the sake of simplicity, we adopt a one-dimensional notation although real images are often at least two-dimensional. The generalization to two or three dimensions is shown to be straightforward further ahead.

In statistical estimation, the mean value of an unknown constant signal μ is estimated using a finite set of noisy observations f_i. If the noise is additive, i.i.d., and Gaussian, we can write $f_i \sim G(\mu, \nu)$, where ν is the noise standard deviation. In the case of Poissonian noise, we have $f_i \sim P(\mu)$, and the noise standard deviation is equal to $\sqrt{\mu}$. In both cases, the sample mean $\frac{1}{N}\sum_{i=1}^{N} f_j$ is the maximum likelihood estimate of the expectation μ. The sample mean minimizes the ℓ_2 error $E(u) = \frac{1}{2}\sum_{j=1}^{N}(u - f_j)^2$.

In image analysis, the data f_i (gray values or photon counts) are measured at different positions (pixels) x_i, and we want to find a solution vector $\mathbf{u} = [u_i]_{i=1,\ldots,N}$ whose entry u_i estimates the unknown signal value μ_i at position x_i. For the sake of simplicity, we assume that the spacing between successive positions is constant, that is, $(x_{i+1} - x_i) = (x_{j+1} - x_j) = \Delta x$, for all $0 < i, j < N$. As only one value is available at each position x_i, the only way to estimate μ_i is to involve values of neighboring pixels. Given a distance function $\delta(x_i, x_j)$, the reflexive neighborhood around x_i is defined to be the set $\mathcal{N}_i = \{j \mid \delta(x_i, x_j) < \epsilon\}$, for some constant neighborhood radius ϵ. In order to have $|\mathcal{N}_i| = |\mathcal{N}_j|$ for all $0 < i, j < N$, we define the distance function

$$\delta(x_i, x_j) = \min\{|x_i - x_j|, N\Delta x - |x_i - x_j|\}, \tag{1}$$

such that the vector $\mathbf{x} = [x_i]_{i=1,\ldots,N}$ can be considered to be "circular" (for $\epsilon \geq \Delta x$, $x_1 \in \mathcal{N}_N$ and $x_N \in \mathcal{N}_1$).

Within this framework, local filtering is achieved by minimizing the objective function

$$E(\mathbf{u}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} w_{ij}(u_i - f_j)^2. \tag{2}$$

The solution is obtained by differentiating E with respect to u_i and equating the partial derivative to 0

$$\frac{\partial E(\mathbf{u})}{\partial u_i} = \sum_{j\in\mathcal{N}_i} w_{ij}(u_i - f_j) = 0. \tag{3}$$

After rearranging the terms, it follows that

$$\hat{u}_i = \frac{\sum_{j\in\mathcal{N}_i} w_{ij} f_j}{\sum_{j\in\mathcal{N}_i} w_{ij}}, \tag{4}$$

where the right-hand term is a weighted estimate (or W-estimate, see [17]), with constant weights in this case. This estimator relies on the assumption that nearby pixels are i.i.d. with a high probability, as they likely lie in the same uniform region of the image. Accordingly, weights are assumed to vanish while moving away from the central pixel x_i, and this justifies the restriction of the sum to the neighborhood $\mathcal{N}_i$, which simplifies the computations. The weights w_{ij} can be defined using either a hard window or a soft one. Most often, the second option is chosen, for instance, by using a Gaussian function

$$w_{ij} = \exp\left(-\frac{\delta(x_i, x_j)^2}{2\rho^2}\right), \tag{5}$$

where the adjustable "width" of the function is denoted ρ. This is the principle of Gaussian smoothing.

The generalization to two or three dimensions merely requires the adaptation of the distance function. In the 2D case, if $[x_i, y_i]$ denote the coordinates of the ith pixel in a regular grid, then

$$\delta([x_i, y_i], [x_j, y_j]) = \sqrt{\delta^2(x_i, x_j) + \delta^2(y_i, y_j)} \tag{6}$$

corresponds to the euclidean distance. The property $|\mathcal{N}_i| = |\mathcal{N}_j|$ for all $0 < i, j < N$ remains valid and is important in photometry invariance.

3 Total Photometry Invariance

Thanks to the normalizing denominator in (4), local filtering ensures the invariance of the total photometry (or average gray level in [18])

$$\sum_{i=1}^{N}\hat{u}_i = \sum_{i=1}^{N} f_i \approx \sum_{i=1}^{N} u_i. \tag{7}$$

To demonstrate the left equality, let us define the effective weights of neighbors to be

$$W_{ij} = \begin{cases} \frac{w_{ij}}{\sum_{j\in\mathcal{N}_i} w_{ij}} & \text{if } i \in \mathcal{N}_i \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

and rewrite the estimate in (4) as $\hat{u}_i = \sum_{j=1}^{N} W_{ij} f_j$. Equality (7) holds because the effective weights in (8) are such that

- there are no border effects ($\mathbf{x}$ is "circular" according to the distance function δ used to define the neighborhoods),
- neighborhoods and effective weights are symmetric, that is, $j \in \mathcal{N}_i \Leftrightarrow i \in \mathcal{N}_j$ and $W_{ij} = W_{ji}$, and
- the effective weights are normalized, that is, $\sum_{j=1}^{N} W_{ij} = 1$.

See [18] for a thorough discussion about the above prerequisites, which eventually allow us to write

$$\sum_{i=1}^{N}\hat{u}_i = \sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} f_j \tag{9}$$

$$= \sum_{j=1}^{N} f_j \sum_{i=1}^{N} W_{ji} = \sum_{j=1}^{N} f_j. \tag{10}$$

4 Edge-Preserving Filtering

The use of a vanishing function for the spatial weights w_{ij} in (4) may prove to be insufficient near an edge, that is, a jump or discontinuity of the signal u. In that case, the edge will be more or less smoothed, depending on parameter ρ. In order to overcome this limitation, the estimator of the sample mean can be replaced with a robust M-estimator [19]. The objective function is then

$$E(\mathbf{u}) = \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} w_{ij}\Psi_{ij}(\mathbf{u}). \tag{11}$$

Choosing $\Psi_{ij}(\mathbf{u}) = \frac{1}{2}(u_i - f_j)^2$ trivially leads to the sample mean estimator. If we assume however that in

neighborhood $\mathcal{N}_i$, two or more constant signals are mixed, a mode estimator is more appropriate. A usual choice consists in taking

$$\Psi_{ij}(\mathbf{u}) \doteq \sigma^2\big(1-\phi\big(u_i-f_j\big)\big), \tag{12}$$

where

$$\phi(s) \doteq \exp\left(-\frac{s^2}{2\sigma^2}\right). \tag{13}$$

In this case, the edge preservation results from the fact that $\Psi_{ij}(\mathbf{u})$ saturates for $|u_i - f_j| \gg \sigma \approx \nu$. If x_i and x_j are not on the same side of an edge, then we can assume that $|\mu_i - \mu_j| \gg \sigma$ and, so, we have

$$\frac{\partial \Psi_{ij}(\mathbf{u})}{\partial u_i} \approx 0 \tag{14}$$

for $u_i \approx \mu_i$ and $f_j \approx \mu_j$. This means that f_j has little influence on the filtered value at x_i. Notice also that

$$\lim_{\sigma \to 0} \Psi_{ij}(\mathbf{u}) = 0, \tag{15}$$

and

$$\lim_{\sigma \to \infty} \Psi_{ij}(\mathbf{u}) = \frac{(u_i - f_j)^2}{2}, \tag{16}$$

which shows that for large values of σ, the same behavior as local filtering will be observed.

The objective function $E(\mathbf{u})$ can be minimized by a gradient descent, starting from the initial guess $\hat{u}_i^0 = f_i$ and converging toward a local minimum if Ψ_{ij} is nonconvex

$$\hat{u}_i^{k+1} = \hat{u}_i^k - \alpha \frac{\partial E(\mathbf{u})}{\partial u_i}\bigg|_{u_i = \hat{u}_i^k}, \tag{17}$$

$$= \hat{u}_i^k - \alpha \sum_{j \in \mathcal{N}_i} w_{ij} \phi\big(\hat{u}_i^k - f_j\big)\big(\hat{u}_i^k - f_j\big), \tag{18}$$

$$= \left(1 - \alpha \sum_{j \in \mathcal{N}_i} w_{ij} \phi\big(\hat{u}_i^k - f_j\big)\right) \hat{u}_i^k + \alpha \sum_{j \in \mathcal{N}_i} w_{ij} \phi\big(\hat{u}_i^k - f_j\big) f_j. \tag{19}$$

Setting the step size

$$\alpha = \frac{1}{\sum_{j \in \mathcal{N}_i} w_{ij} \phi\big(\hat{u}_i^k - f_j\big)} \tag{20}$$

leads to the iterative update rule

$$\hat{u}_i^{k+1} = \frac{\sum_{j \in \mathcal{N}_i} w_{ij} \phi\big(\hat{u}_i^k - f_j\big) f_j}{\sum_{j \in \mathcal{N}_i} w_{ij} \phi\big(\hat{u}_i^k - f_j\big)}, \tag{21}$$

which turns out to be a W-estimate with adaptive weights and corresponds to the formula of the local M-smoothers [15], [16].

The last equation is also closely related to bilateral filtering [13]. This filter results from an intuitive generalization of local filtering [14], wherein tonal (or photometric) similarity is involved in addition to spatial proximity. Bilateral filtering is primarily intended to be noniterative [13], like local filtering; however, its iterative application (proposed in [14]) can be written as

$$\hat{u}_i^{k+1} = \frac{\sum_{j \in \mathcal{N}_i} w_{ij} \phi\Big(\hat{u}_i^k - \hat{u}_j^k\Big) \hat{u}_j}{\sum_{j \in \mathcal{N}_i} w_{ij} \phi\Big(\hat{u}_i^k - \hat{u}_j^k\Big)}, \tag{22}$$

where ϕ denotes the tonal kernel and measures the photometric similarity. The last update rule is equivalent to (21) only for the first iteration ($\hat{u}_j^0 = f_j$); in subsequent iterations, the reference signal f_j is replaced with the estimate $\hat{u}_j^k$. This induces a coupling and makes bilateral filtering nonlocal, in contrast to local M-smoothers. Therefore, bilateral filtering can converge to a solution $\hat{\mathbf{u}}$ that is uniform. This does not happen with local M-smoothers [16]. In the case of bilateral filtering, let us define

$$W_{ij} = \begin{cases} \frac{w_{ij} \phi\left(\hat{u}_i^k - \hat{u}_j^k\right)}{\sum_{j \in \mathcal{N}_i} w_{ij} \phi\left(\hat{u}_i^k - \hat{u}_j^k\right)} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

This allows us to rewrite the update rule in (22) as $\hat{u}_i^{k+1} = \sum_{j=1}^N W_{ij} \hat{u}_j^k$. Although the effective weights are normalized ($\sum_{j=1}^N W_{ij} = 1$), it is however straightforward to observe that the total photometry is not guaranteed to be preserved. Indeed, equality $W_{ij} = W_{ji}$ no longer holds, because the respective denominators of W_{ij} and W_{ji} can be different (they are computed over two different neighborhoods and involve the corresponding estimates of the signal).

The case of local M-smoothers proves to be even worse; using the update rule $\hat{u}_i^{k+1} = \sum_{j=1}^N W_{ij} f_j$, we can write the effective weights as

$$W_{ij} = \begin{cases} \frac{w_{ij} \phi\left(\hat{u}_i^k - f_j\right)}{\sum_{j \in \mathcal{N}_i} w_{ij} \phi\left(\hat{u}_i^k - f_j\right)} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

In this case, even the numerator in (24) is problematic, as equality $|\hat{u}_i^k - f_j| = |\hat{u}_j^k - f_i|$ is not guaranteed to hold.

It is noteworthy that the Hessian of (11) is diagonal and easy to estimate. However, the choice of the step size in (19) does not take advantage of that fact, although it would enable the gradient descent to reach quadratic convergence. Instead, the value of the step size is constrained by the necessity to obtain the weight normalization, that is, $\sum_{i=1}^N W_{ij} = 1$, for each pixel and at each update. This means that convergence may be slow and that its speed may be different for each pixel.

5 Additive Weight Normalization

The asymmetry in (24) can be addressed by changing the value of the step size in the gradient descent (19) to $\alpha = 1/\sum_{i \in \mathcal{N}_i} w_{ij}$. Thereby, we obtain symmetry for the first iteration ($\hat{u}_j^0 = f_j$), and using nonreflexive neighborhoods

$$\mathcal{N}_i^0 \doteq \{j \mid j \neq i \text{ and } \delta(x_i, x_j) < \epsilon\} = \mathcal{N}_i \setminus \{i\}, \tag{25}$$

we can write $\hat{u}_i^1 = \sum_{j=1}^N W_{ij} f_j$ with the effective weights

$$W_{ij} = \begin{cases} \frac{w_{ij} \phi\left(\hat{u}_i - f_j\right)}{\sum_{j \in \mathcal{N}_i^0} w_{ij}} & \text{if } j \in \mathcal{N}_i^0 \\ 1 - \sum_{j \in \mathcal{N}_i^0} W_{ij} & \text{if } j = i \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

It is noteworthy that the change of the step size leads to an "additive" normalization in the last equation, instead of a multiplicative one, as in (24). The normalization constraint $\sum_{j=1}^{N} W_{ij} = 1$ is enforced by adjusting only the central weight W_{ii}, rather than by scaling all weights W_{ij} in the neighborhood $\mathcal{N}_i$. Unfortunately, using (26) for the second and subsequent iterations breaks the symmetry and normalization properties.

In order to address this issue, we first define a symmetrization function

$$S(D_{ij}, D_{ji}) = \sqrt{\theta_{ij}^2 D_{ij}^2 + (1-\theta_{ij})^2 D_{ji}^2}, \tag{27}$$

where $D_{ij} = u_i - f_j$ and $0 \le \theta_{ij} \le 1$. If S is chosen to be the minimum of its squared arguments, then we fix θ_{ij} as follows:

$$\theta_{ij} = \begin{cases} 1 & \text{if } D_{ij}^2 \le D_{ji}^2 \\ 0 & \text{if } D_{ij}^2 > D_{ji}^2. \end{cases} \tag{28}$$

A similar equation can be obtained for the maximum. In all cases, we have $\theta_{ji} = 1 - \theta_{ij}$ and, therefore, $S(D_{ij}, D_{ji}) = S(D_{ji}, D_{ij})$.

Next, we define the objective function

$$E(\mathbf{u}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i^0} w_{ij} \Xi_{ij}(\mathbf{u}), \tag{29}$$

where

$$\Xi_{ij}(\mathbf{u}) \doteq \frac{D_{ii}^2}{2} + \frac{(f_i - f_j)}{\theta_{ij}} \Phi\big(\theta_{ij} D_{ij}\big) \phi\big((1-\theta_{ij}) D_{ji}\big). \tag{30}$$

In the last equation, Φ is proportional to the Gauss error function

$$\Phi(s) \doteq \frac{\sigma\sqrt{\pi}}{\sqrt{2}} \operatorname{erf}\left(\frac{s}{\sigma\sqrt{2}}\right) \tag{31}$$

such that $\Phi'(s) = \phi(s)$. Notice that

$$\lim_{\sigma \to \infty} \Xi_{ij}(\mathbf{u}) = \frac{(u_i - f_i)^2}{2} + (f_i - f_j)(u_i - f_j) \tag{32}$$

$$= \frac{(u_i - f_j)^2}{2} + \frac{(f_j - f_i)^2}{2} \tag{33}$$

$$= \frac{(f_j - f_i)^2}{2} + \lim_{\sigma \to \infty} \Psi_{ij}(\mathbf{u}), \tag{34}$$

whereas

$$\lim_{\sigma \to 0} \Xi_{ij}(\mathbf{u}) = \frac{(u_i - f_i)^2}{2}, \tag{35}$$

which constitutes an essential difference with Ψ_{ij}. Hence, we see that, in this case, edge preservation is ensured by shifting the global minimum of Ξ_{ij}. If x_i and x_j are not on the same side of an edge, then we can assume that $|\mu_i - \mu_j| \gg \sigma$, and so, for $u_i \approx \mu_i$ and $f_j \approx \mu_j$, we have $\Xi_{ij}(\mathbf{u}) \approx (u_i - f_i)^2/2$. On the other hand, if $u_i \approx \mu_i$ and $f_j \approx \mu_i$, then $\Xi_{ij}(\mathbf{u}) \approx (u_i - f_j)^2/2$. This is illustrated in Fig. 1, for the case $\theta_{ij} = 1$. For $\theta_{ij} = 0$, l'Hôpital's rule allows us to compute

$$\lim_{\theta_{ij} \to 0} \Xi_{ij}(\mathbf{u}) = \frac{D_{ii}^2}{2} + D_{ij} \phi\big((1-\theta_{ij}) D_{ji}\big). \tag{36}$$

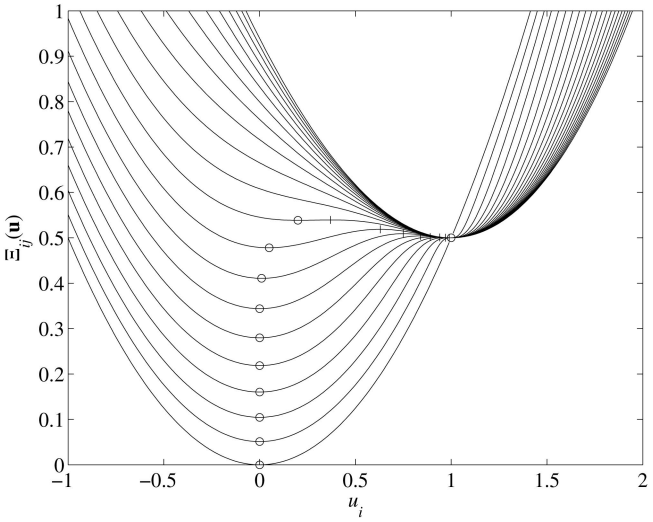

Fig. 1. Function $\Xi_{ij}(\mathbf{u})$, for $f_i = 0$, $f_j = 1$, θ_{ij}, and σ varying between 0 and 1.6. Depending on σ, $\Xi_{ij}(\mathbf{u})$ can have one or two minima (circles) that are separated by a local maximum (crosses). For $\sigma = 0$, $\Xi_{ij}(\mathbf{u}) = (u_i - f_i)^2/2$, whereas for $\sigma \to \infty$, $\Xi_{ij}(\mathbf{u}) \to (u_i - f_i)^2/2$.

Equating the partial derivative

$$\frac{\partial \Xi_{ij}(\mathbf{u})}{\partial u_i} = D_{ii} + (f_i - f_j) \phi\big(\theta_{ij} D_{ij}\big) \phi\big((1-\theta_{ij}) D_{ji}\big) \tag{37}$$

$$= D_{ii} + (f_i - f_j) \phi\big(S(D_{ij}, D_{ji})\big) \tag{38}$$

to zero shows that if $\sigma > 0$, then Ξ_{ij} reaches a minimum for $u_i = f_j$ and $u_j = f_i$. In particular, if S returns the minimum of its squared arguments, then the conjunction degenerates into a disjunction: The minimum is reached for $u_i = f_j$ or $u_j = f_i$. The presence of a minimum of Ξ_{ij} for $u_i = f_j$ is a common point with Ψ_{ij}. Depending on the value of σ, a second minimum can appear in the interval $[f_i, f_i + 0.285(f_j - f_i)]$, for $\sigma < 0.455|f_j - f_i|$. In this case, the two minima are separated by a local maximum that is located in the interval $[f_i + 0.285(f_j - f_i), f_j]$. The minimum for $u_i = f_j$ is global for $\sigma > 0.41|f_j - f_i|$.

Starting from the initial guess $\hat{u}_i^0 = f_i$, we can write the gradient descent of $E(\mathbf{u})$ as

$$\hat{u}_i^{k+1} = \hat{u}_i^k - \alpha \left.\frac{\partial E(\mathbf{u})}{\partial u_i}\right|_{u_i = \hat{u}_i^k} \tag{39}$$

$$= \hat{u}_i^k - \alpha \sum_{j \in \mathcal{N}_i^0} w_{ij} \Big(D_{ii} + \phi\Big(S\Big(\hat{D}_{ij}^k, \hat{D}_{ji}^k \Big) \Big) (f_i - f_j) \Big), \tag{40}$$

where $\hat{D}_{ij}^k$ extend the previous notations to the estimate $\hat{u}_i^k$. Setting $\alpha = 1/\sum_{j \in \mathcal{N}_i^0} w_{ij}$, we find

$$\hat{u}_i^{k+1} = \left(1 - \frac{\sum_{j \in \mathcal{N}_i^0} w_{ij} \phi\Big(S\Big(\hat{D}_{ij}^k, \hat{D}_{ji}^k \Big) \Big)}{\sum_{j \in \mathcal{N}_i^0} w_{ij}} \right) f_i + \frac{\sum_{j \in \mathcal{N}_i^0} w_{ij} \phi\Big(S\Big(\hat{D}_{ij}^k, \hat{D}_{ji}^k \Big) \Big) f_j}{\sum_{j \in \mathcal{N}_i^0} w_{ij}}. \tag{41}$$

By comparison with (21), the last equation incorporates the symmetrization function S and the first term in the right-hand side comprises f_i instead of $\hat{u}_i^k$. Once translated into the generic form $\hat{u}_i^{k+1} = \sum_{j=1}^{N} W_{ij} f_j$, the effective weights are given by

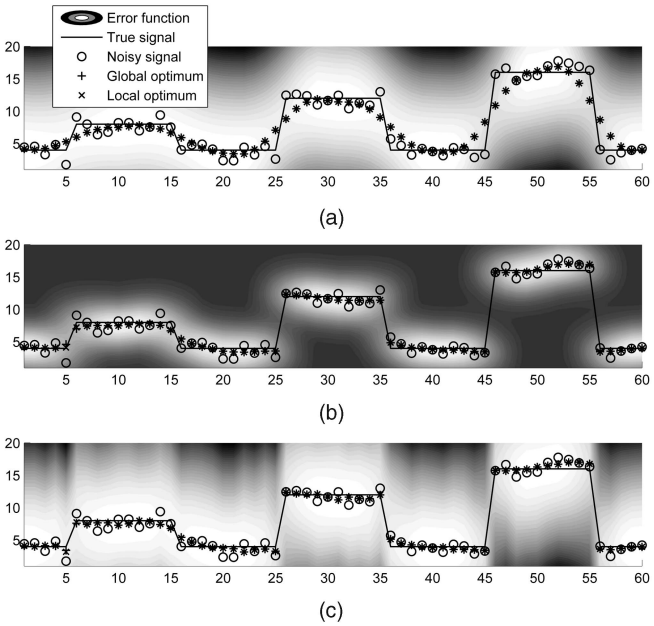

Fig. 2. Toy example with a piecewise constant signal and Gaussian noise: (a) results of Gaussian filtering, (b) LMSMN, and (c) LMSAN with nonreflexive neighborhoods for $\rho = 2\Delta x$ and $\sigma = 2.5$. Levels of gray in the background indicate the profile of the objective function. The genuine and noisy signals are plotted along with the global optima of the objective function and its local optima after initialization with the noisy signal and six iterations.

$$W_{ij} = \begin{cases} \frac{w_{ij}\phi\left(S(\hat{D}_{ji}^k, \hat{D}_{ji}^k)\right)}{\sum_{j\in\mathcal{N}_i^0} w_{ij}} & \text{if } j \in \mathcal{N}_i^0 \\ 1 - \sum_{j\in\mathcal{N}_i^0} W_{ij} & \text{if } j = i \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

and involve an additive normalization, as in (26). As equalities $w_{ij} = w_{ji}$ and $S(\hat{D}_{ij}^k, \hat{D}_{ji}^k) = S(\hat{D}_{ji}^k, \hat{D}_{ij}^k)$ hold, and since the denominator in (42) is constant, the symmetry $W_{ij} = W_{ji}$ is guaranteed. For bilateral filtering, the update rule $\hat{u}_i^{k+1} = \sum_{j=1}^N W_{ij}\hat{u}_j^k$ with effective weights

$$W_{ij} = \begin{cases} \frac{w_{ij}\phi\left(\hat{u}_i^k - \hat{u}_j^k\right)}{\sum_{j\in\mathcal{N}_i^0} w_{ij}} & \text{if } j \in \mathcal{N}_i^0 \\ 1 - \sum_{j\in\mathcal{N}_i^0} W_{ij} & \text{if } j = i \\ 0 & \text{otherwise,} \end{cases} \tag{43}$$

can be used, since $S(\hat{u}_i^k - \hat{u}_j^k, \hat{u}_j^k - \hat{u}_i^k) = \hat{u}_i^k - \hat{u}_j^k$.

Several differences between local M-smoothers with multiplicative normalization (LMSMN) and additive normalization (LMSAN) can be pointed out. First, it is noteworthy that in the minimization of (11), the step size α given in (20) differs for each pixel. In contrast, the gradient descent of (29) involves a constant step size $(\alpha = 1/\sum_{j\in\mathcal{N}_i^0} w_{ij})$. Second, the contribution of the noisy value of a pixel to the filtered value of another pixel depends on other surrounding pixels in the case of a multiplicative normalization. For additive normalization, this contribution is computed regardless of neighboring pixels.

Looking at the minima of the respective objective functions also reveals a difference. For instance, let us consider the toy example illustrated in Fig. 2, where Gaussian noise pollutes a 1D piecewise constant signal. The variance of the noise is equal to 1. The three panels represent Gaussian filtering, LMSMN, and LMSAN, respectively. For all three filters, $|\mathcal{N}_i| = 7$, and the weights w_{ij} are computed according to (5), with $\rho = 2\Delta x$. For both versions of local M-smoothers, the kernel width σ is set to 2.5. Six iterations of the gradient descent have been performed, after initialization with the noisy signal. In addition to the local minima that result from the gradient descent, global optima are also depicted. For the considered value of σ, the local and global minima of each filter are very close, if not identical. Gray levels in the background of each diagram indicate the profile of the underlying objective functions for each pixel index in abscissa. Due to the quadratic term in (29), the profiles of Gaussian filtering and LMSAN look similar, whereas those of LMSMN have narrower basins. However, a careful inspection reveals that LMSMN and LMSAN yield comparable results, whereas Gaussian filtering smooths the signal.

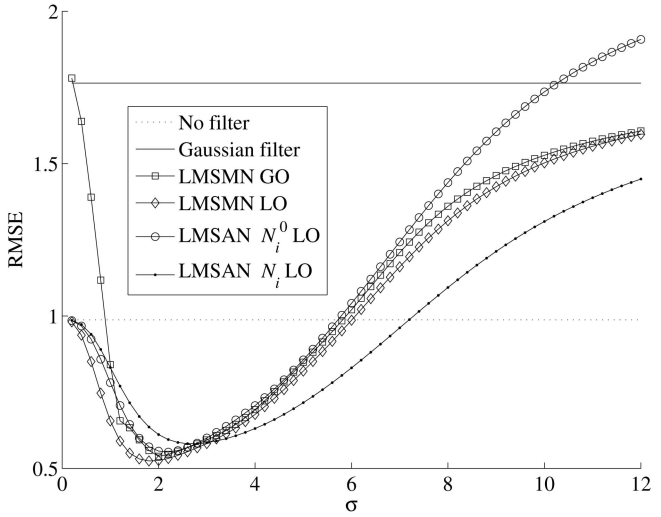

Fig. 3. Average RMSEs for 100 data sets based on the piecewise constant signal in Fig. 2. Noise is Gaussian with unit variance. Curves for Gaussian filtering, LMSMN, and LMSAN (without symmetrization function, that is, $S(a, b) = a$) are plotted as a function of σ. For LMSMN, the RMSEs corresponding to the global optima (GO) of the objective function are shown in addition to the local optima (LO) obtained with the gradient descent.

The recovery of the underlying signal can be assessed by looking at the root mean square error (RMSE) between the true signal and the filtered one

$$\text{RMSE} \doteq \sqrt{\frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{N}(\hat{u}_{im} - \mu_{im})^2}, \tag{44}$$

where M is the number of trials, and m is the trial index. Fig. 3 illustrates the RMSE (over 100 trials) of the considered filters for increasing values of σ. An additional curve gives the RMSE of the noisy signal, without any filtering.

For small values of σ, the gradient descent for both LMSMN and LMSAN leaves the noisy signal almost unchanged. For LMSMN, however, we observe a discrepancy between the global optimum of the objective function and the local optimum reached by the gradient descent, starting with the initial guess $\hat{u}_i^0 = f_i$. For each location x_i, the objective function of LMSMN is a sum of several functions Ψ_{ij} that are centered on the noisy values f_i. If σ is low, as illustrated in Fig. 4, the number of basins increases and their depth critically depends on the observed distribution of neighboring noisy values. In

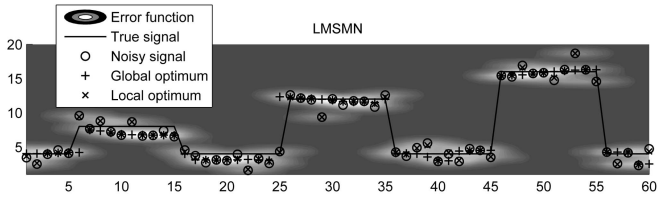

Fig. 4. Similar toy example as that in Fig. 2: result of LMSMN with $\rho = 2\Delta x$ and $\sigma = 0.5$. For a low value of σ, the global minima of the objective function and those found by the gradient descent starting from the noisy values as an initial guess do not always correspond.

some cases (see, for example, at $x_i = 25$), the global maximum can be considered to be spurious and explains the high RMSE value in Fig. 3. For LMSAN, such a discrepancy is not likely to happen: due to the quadratic term in (30), the solution found by the gradient descent is in good correspondence with the global minimum of the objective function. Those two different behaviors can also be explained by observing that the limits (15) and (35) are different.

For large values of σ, the limits (16) and (34) allow us to write

$$\lim_{\sigma\to\infty} E(\mathbf{u}) = \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} w_{ij} \lim_{\sigma\to\infty} \Psi_{ij}(\mathbf{u}) \tag{45}$$

$$= \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} w_{ij}\frac{(u_i - f_j)^2}{2} \tag{46}$$

for LMSMN, and

$$\lim_{\sigma\to\infty} E(\mathbf{u}) = \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i^0} w_{ij} \lim_{\sigma\to\infty} \Xi_{ij}(\mathbf{u}) \tag{47}$$

$$= \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i^0} w_{ij}\frac{(u_i - f_j)^2}{2} + \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i^0} w_{ij}\frac{(f_i - f_j)^2}{2} \tag{48}$$

for LMSAN. Both objective functions turn out to be closely related to that of local filtering. However, in contrast to LMSMN and local filtering, the inner sum for LMSAN runs over nonreflexive neighborhoods. Consequently, LMSAN does not perform as well as LMSMN and Gaussian filtering for large values of σ, as shown in Fig. 3. The objective function of LMSAN can also be defined with reflexive neighborhoods, that is, $\mathcal{N}_i$ instead of $\mathcal{N}_i^0$. This leads to the same effective weights, as in (42), except that $\mathcal{N}_i^0$ is replaced with $\mathcal{N}_i$ in the denominator of the first conditional assignment. In this case, the asymptotic behavior for large values of σ is the same as that of Gaussian filtering, as can be seen on the right part in Fig. 3.

Optimal values of σ are different for LMSMN ($\sigma \approx 1.7$), LMSAN with nonreflexive neighborhoods ($\sigma \approx 2.0$), and LMSAN with reflexive neighborhoods ($\sigma \approx 2.6$). LMSMN outperforms LMSAN because it better smooths outliers: if f_i is an outlier, then its effective weight W_{ii} is lower with multiplicative normalization than that produced by additive normalization. The difference between reflexive and nonreflexive neighborhoods in LMSAN can be explained in a similar way: For the definition with $\mathcal{N}_i^0$, we have $0 \le W_{ii} \le 1$, whereas we have

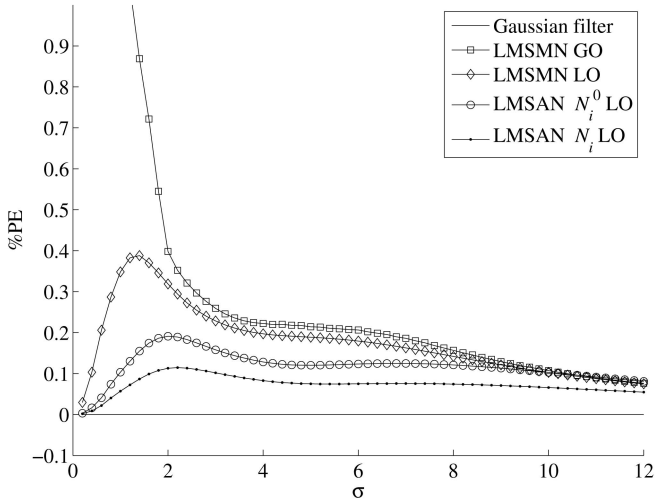

Fig. 5. Average photometric errors for 100 data sets based on the piecewise constant signal of Fig. 2. Noise is Gaussian with unit variance. Curves for Gaussian filtering, LMSMN, and LMSAN (without symmetrization function, that is, $S(a,b) = a$) are plotted as a function of σ. For LMSMN, the RMSEs corresponding to the global optima (GO) of the objective function are shown in addition to the local optima (LO) obtained with the gradient descent.

$$\frac{w_{ii}}{\sum_{j\in\mathcal{N}_i} w_{ij}} \le W_{ii} \le 1 \tag{49}$$

for the alternative definition with $\mathcal{N}_i$. Therefore, in the case of an outlier, the central weight W_{ii} is slightly lower with nonreflexive neighborhoods than with reflexive ones, and this leads to better filtering performance.

Concerning the preservation of the total photometry, Fig. 5 shows results that correspond to those in Fig. 3. The photometric error is expressed as a percentage and is computed as

$$\%\mathrm{PE} \doteq \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(\frac{100}{N}\sum_{i=1}^{N}\left(1 - \frac{\hat{u}_{im}}{f_{im}}\right)\right)^2}. \tag{50}$$

Notice that in this preliminary example, LMSAN is used without any symmetrization function (that is, $S(a,b) = a$). This shows that even without symmetrization function, the additive normalization reduces the photometric error. We observe that LMSAN with reflexive neighborhoods achieves a lower photometric error than with nonreflexive ones at the expense of a higher RMSE. With any symmetrization function, the photometric error of LMSAN is trivially null; see also Section 8 for additional experiments with a nondegenerate symmetrization function. The impact of the symmetrization function on the RMSE is illustrated in Fig. 6 in the different cases (the minimum, the maximum, and the average, that is, $\theta_{ij} = 0.5$). The minimum delivers the lowest RMSE in this example.

6 Connection with Anisotropic Diffusion

The update rules with additive normalization described in the previous section for local M-smoothers and bilateral filtering can be related to anisotropic diffusion [9]. For a continuous signal $\hat{u}(x,t)$ that evolves in time, with initial conditions $\hat{u}(x,0) = f(x)$, the partial differential equation representing the diffusion-reaction process is

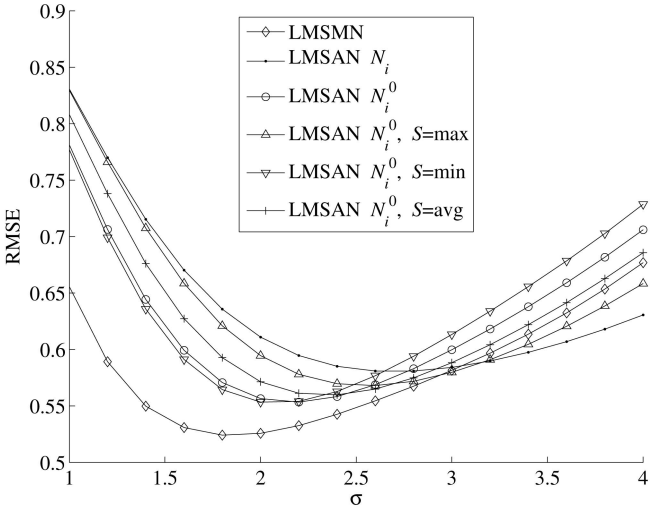

Fig. 6. Closer look at the curves in Fig. 3. The results of LMSMN and LMSAN are given, as a function of σ; the average RMSEs over 100 trials are shown. For LMSAN, additional curves show the effect of different symmetrization functions S.

$$\frac{\partial \hat{u}(x,t)}{\partial t} = \operatorname{div}\left(g\left(\frac{|\nabla_x \hat{u}(x,t)|^2}{2}\right)\nabla_x \hat{u}(x,t)\right). \quad (51)$$

In [9], g is a gradient-stopping function (or diffusivity function) such that $g(\frac{1}{2}s^2) \to 0$ when $s \to \infty$. The last condition ensures that the diffusion slows down across edges [20].

In order to obtain a discretized formulation of (51), let us assume that for $t = k$ the approximation

$$\nabla_x \hat{u}(x,t)\big|_{x \to x_i} \approx \begin{cases} \hat{u}_{i+1}^k - \hat{u}_i^k & \text{if } x > x_i \\ \hat{u}_{i-1}^k - \hat{u}_i^k & \text{if } x < x_i \end{cases} \quad (52)$$

holds [20], and let us redefine $\mathcal{N}_i^0 \doteq \{j | \delta(x_i, x_j) = \Delta x\}$, so that only direct neighbors are included. This leads to a straightforward discretization [18] of (51), which is

$$\frac{\hat{u}_i^{k+1} - \hat{u}_i^k}{\tau} = \frac{1}{|\mathcal{N}_i^0|}\sum_{j \in \mathcal{N}_i^0} g\left(\frac{1}{2}(\hat{u}_j^k - \hat{u}_i^k)^2\right)(\hat{u}_j^k - \hat{u}_i^k). \quad (53)$$

The corresponding update rule [20], [18] is written as

$$\hat{u}_i^{k+1} = \hat{u}_i^k + \frac{\tau}{|\mathcal{N}_i^0|}\sum_{j \in \mathcal{N}_i^0} g\left(\frac{1}{2}(\hat{u}_j^k - \hat{u}_i^k)^2\right)(\hat{u}_j^k - \hat{u}_i^k). \quad (54)$$

According to [20], the diffusion process can also be viewed as a statistical estimation problem. Within that framework, $g(\frac{1}{2}s^2) = \phi(s)$. Setting the step size $\tau = 1$ allows us to rewrite the above update rule as $\hat{u}_i^{k+1} = \sum_{j=1}^N W_{ij}\hat{u}_j^k$ with effective weights

$$W_{ij} = \begin{cases} \frac{\phi(\hat{u}_i - \hat{u}_j)}{|\mathcal{N}_i^0|} & \text{if } j \in \mathcal{N}_i^0 \\ 1 - \sum_{j \in \mathcal{N}_i^0} W_{ij} & \text{if } j = i \\ 0 & \text{otherwise.} \end{cases} \quad (55)$$

This corresponds to bilateral filtering with additive normalization, as in (43), but with restricted neighborhoods, and $w_{ij} = 1$. Hence, this discrete definition of anisotropic diffusion preserves the total photometry of the image. A different approach for relating anisotropic diffusion to bilatering filtering has been developed in [21] and [22], in the case of multiplicative normalization.

7 Poissonian Noise

The edge-preserving filters that have been described in the previous section rely on the assumption of i.i.d. Gaussian noise for all pixels. In that context, the Gaussian function $\phi(s)$ involved in (42) and (43) can be intuitively interpreted as a similarity measure. More precisely, the filtered value of the ith pixel is a weighted average in which heavier weights are given to neighboring pixels having a similar count as that of the ith pixel. The similarity depends on the width parameter σ, whose value is usually chosen to be proportional to the noise standard deviation ν. A good trade-off yields a good denoising efficiency in constant regions of the images, along with the preservation of the intensity jumps, provided they are much larger than σ.

If noise is Poissonian, σ should be adjusted independently for each pixel, in order to keep the same trade-off between denoising efficiency and edge preservation everywhere in the image. Otherwise, if σ remains constant, edges in regions with low counts are smoothed whereas denoising in regions with high counts is insufficient. Since Poissonian noise is considered to be multiplicative, a simple trick to overcome this limitation could be to replace σ with an estimate of the product $\sigma\sqrt{\mu_i}$, e.g., $\sigma\sqrt{f_i}$ for local M-smoothers or $\sigma\sqrt{\hat{u}_i}$ for bilateral filtering. Nevertheless, this leaves three unresolved issues. First, f_i (and $\hat{u}_i$ to a lesser extent) is a noisy estimate of μ_i. Second, Poissonian distributions are asymmetric, and this fact is not taken into account in the similarity measure $\phi(s)$. Third, factors for neighboring voxels i and j are different, what breaks the weight symmetry $W_{ij} = W_{ji}$ and jeopardizes the total photometry invariance.

A more elegant solution to the problem is brought by using a VST, like Anscombe's [10]. If $r \sim P(\lambda)$ and $\lambda \to \infty$, then $t = \mathcal{A}(r) \doteq 2\sqrt{r + 3/8}$ is such that $t \sim G(2\sqrt{\lambda}, 1)$. In practice, t is approximately Gaussian with unit variance for $\lambda \geq 10$ [23]. This suggests to follow the strategy described in [11]:

1. Apply $\mathcal{A}$ to the image.
2. Use a classical filter designed for Gaussian noise.
3. Apply $\mathcal{A}^{-1}$ to the denoised image.

Unfortunately, this strategy does not ensure the photometry invariance. Even if the filter preserves the photometry of the transformed image, this does not necessarily remain true for the image obtained after the third step, since Anscombe's VST and its inverse are nonlinear.

Instead, we propose to embed a VST in the objective function of the aforementioned filters and to compute the corresponding gradient descent update rule. This allows us to avoid any pre- and posttransformation of the image and to obtain the filtered pixel counts directly as a weighted average of the noisy ones. As the usual functions for robust mode-estimation involve a squared difference of two pixel counts, we use Fisz's transform [24] instead of Anscombe's. Fisz's transform is a special case of Fisz's theorem [25] and has already been used in wavelet-based denoising [24], [23]. According to Fisz's theorem, if $r, t \sim P(\lambda)$ are two independent Poissonian variables and $\lambda \to \infty$, then

$$s = \mathcal{F}(r,t) \doteq \frac{r - t}{\sqrt{r + t}} \sim G(0,1). \quad (56)$$

In practice, Fisz's transform provides us with a better "Gaussianization" than Anscombe's, and s is approximately Gaussian with unit variance for $\lambda \geq 1$ [23]. By comparison, for two independent Gaussian variables $r, t \sim G(\mu, 1)$, we have $s = r - t \sim G(0, \sqrt{2})$.

This allows us to rewrite the objective function of local filtering in (2) as

$$E(\mathbf{u}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} w_{ij} \mathcal{F}^2(u_i, f_j). \tag{57}$$

The derivative of $E(\mathbf{u})$ involves

$$\frac{\partial \mathcal{F}^2(r,t)}{\partial r} = (r-t)\,\mathcal{H}(r,t), \tag{58}$$

where

$$\mathcal{H}(r,t) = \frac{r+3t}{(r+t)^2}. \tag{59}$$

As the optimum cannot be expressed in closed form, the gradient descent leads to the update rule $\hat{u}_i^{k+1} = \sum_{j=1}^{N} W_{ij} f_j$, where the effective weights are

$$W_{ij} = \begin{cases} \frac{w_{ij}\mathcal{H}(\hat{u}_i^k, f_j)}{\sum_{j \in \mathcal{N}_i} w_{ij}\mathcal{H}(\hat{u}_i^k, f_j)} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise.} \end{cases} \tag{60}$$

This update rule does not preserve the total photometry.

For LMSMN, we can define

$$\Pi_{ij}(\mathbf{u}) = \sigma^2 \Big(1 - \phi\Big(\sqrt{2}\,\mathcal{F}(u_i, f_j)\Big)\Big), \tag{61}$$

which allows us to rewrite the objective function in (11) as

$$E(\mathbf{u}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} w_{ij} \Pi_{ij}(\mathbf{u}). \tag{62}$$

Function Π_{ij} resembles an "upside-down bell" like Ψ_{ij}. The only difference is that the bell width depends on the product of σ and $\sqrt{u_i + f_j}$ in this case. The derivative of $E(\mathbf{u})$ involves

$$\frac{\partial \Pi_{ij}(\mathbf{u})}{\partial u_i} = (u_i - f_j)\,\Phi\Big(\sqrt{2}\,\mathcal{F}(u_i, f_j)\Big)\,\mathcal{H}(u_i, f_j), \tag{63}$$

and the effective weights can be written as

$$W_{ij} = \begin{cases} \frac{w_{ij}\phi\left(\sqrt{2}\,\mathcal{F}(\hat{u}_i^k, f_j)\right)\mathcal{H}(\hat{u}_i^k, f_j)}{\sum_{j \in \mathcal{N}_i} w_{ij}\phi\left(\sqrt{2}\,\mathcal{F}(\hat{u}_i^k, f_j)\right)\mathcal{H}(\hat{u}_i^k, f_j)} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise.} \end{cases} \tag{64}$$

The use of Fisz's transform makes the derivation of an additive normalization more difficult than in the Gaussian case. Indeed, if we embed Fisz's transform into (30), the derivative of $\mathcal{F}$ appears in the gradient descent and prevents us from obtaining an update rule in the form of $\hat{u}_i^{k+1} = \sum_{j=1}^{N} W_{ij} f_j$. However, if we write the effective weights as

$$W_{ij} = \begin{cases} \frac{w_{ij}\phi\left(\sqrt{2}\,S(\mathcal{F}(\hat{u}_i^k, f_j), \mathcal{F}(\hat{u}_j^k, f_i))\right)}{\sum_{j \in \mathcal{N}_i} w_{ij}} & \text{if } j \in \mathcal{N}_i^0 \\ 1 - \sum_{j \in \mathcal{N}_i^0} W_{ij} & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \tag{65}$$

then we can test this ad hoc definition with a similar example to that in Fig. 2. The piecewise constant signal that is shown in Fig. 7 occupies four levels (50 for the baseline; 100, 150, and 200 for the bumps). Poissonian noise is generated according to these values using Ahrens' and Dieter's method [26]. As the RMSE is an optimal error measure for Gaussian noise only, we include Fisz's VST in the computation of the RMSE, that is,

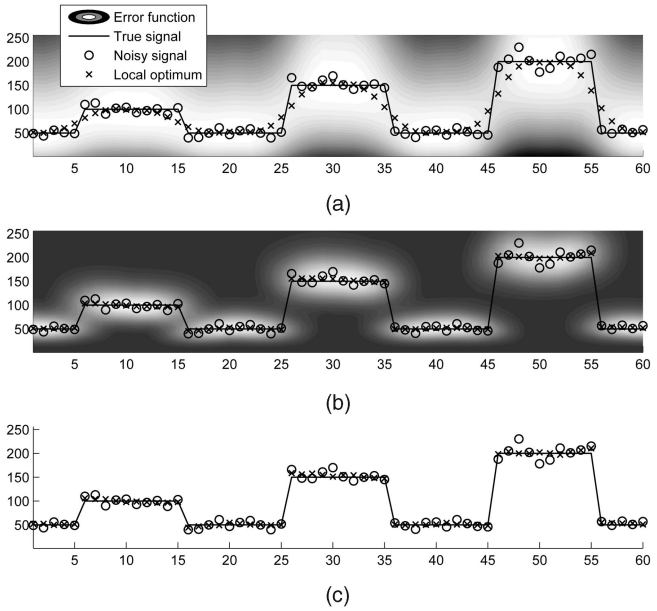

Fig. 7. Toy example with a piecewise constant signal and Poissonian noise: results of (a) Gaussian filtering, (b) LMSMN, and (c) LMSAN with nonreflexive neighborhoods and incorporated Fisz's transform for $\rho = 2\Delta x$ and $\sigma = 2.5$. Levels of gray in the background reveal the profile of the objective function for Gaussian filtering and LMSMN. The genuine and noisy signals are plotted along with the local optima (LO) of the objective functions after initialization with the noisy signal and six iterations of the gradient descent.

$$\mathrm{RMSE} \doteq \sqrt{\frac{2}{MN} \sum_{m=1}^{M} \sum_{i=1}^{N} \mathcal{F}^2(\hat{u}_{im}, \mu_{im})}. \tag{66}$$

The results over 100 trials are given in Fig. 8. Three lines are drawn for Gaussian filtering: The first one corresponds to Gaussian filtering applied to the raw, noisy signal; the second one preprocesses the signal with Anscombe's VST; the third curve corresponds to the version with embedded Fisz's transform, which minimizes (57) by the gradient descent. According to the definition of the RMSE for Poissonian noise in (66), the two implementations of Gaussian filtering that involve a VST are expected to outperform the version that is deprived of VST. However, more importantly, it is noteworthy that the gradient descent of (57) with Fisz's transform outperforms the solution that first applies Anscombe's VST and then compute the W-estimate (4) in closed form. This difference can also be pointed out for LMSMN but only for large values of σ. The ranking of edge-preserving filters remains the same as with Gaussian noise: LMSMN outperforms LMSAN with nonreflexive neighborhoods, which, in turn, achieves better results than LMSAN with reflexive neighborhoods.

The photometric error is plotted in Fig. 9. Gaussian filtering with a VST (either embedded or as a preprocessing) does not preserve the total photometry. Notice also that in this preliminary example, LMSAN is used without any symmetrization function (that is, $S(a, b) = a$). This shows that even

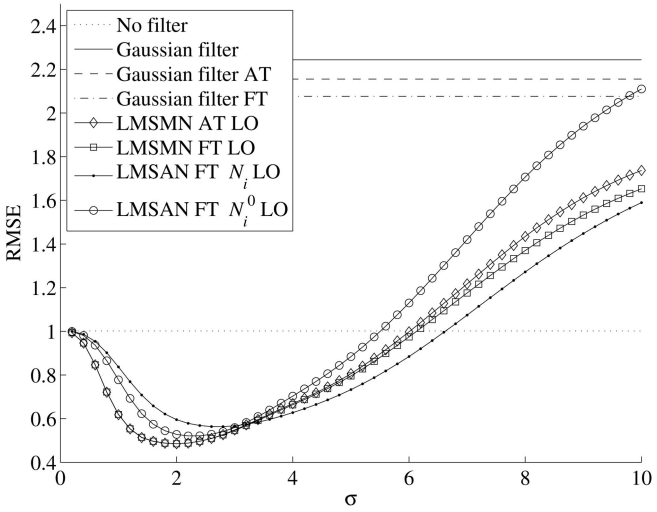

Fig. 8. Average RMSEs for 100 data sets based on the piecewise constant signal in Fig. 7. Noise is Poissonian. Curves for Gaussian filtering, LMSMN, and LMSAN (without symmetrization function, that is $S(a,b)=a$) are plotted as a function of σ. Gaussian filtering is applied to the noisy signal without preprocessing and using Anscombe's VST (AT), too. A third version involves Fisz's VST (FT) and minimizes (57). LMSMN is applied to preprocessed data using Anscombe's VST and (11) or directly using Fisz's VST embedded in (62). For LMSAN, results for reflexive and nonreflexive neighborhoods are given. The RMSE values correspond to local optima (LO) of the objective functions after the gradient descent.

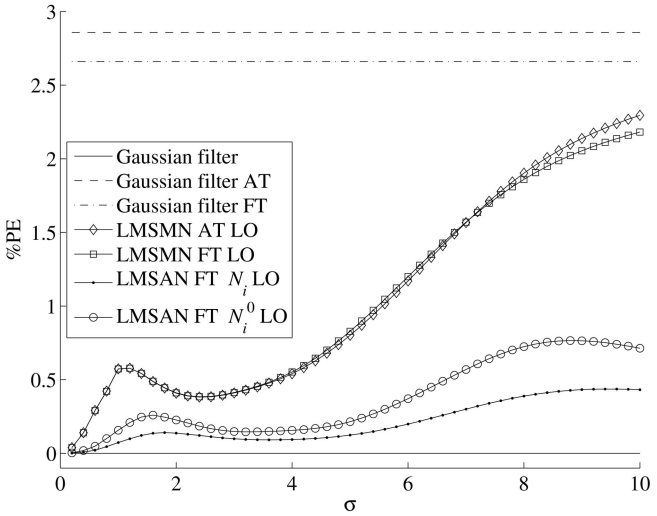

Fig. 9. Average photometric errors corresponding to the results in Fig. 8. LMSAN is used without symmetrization function, that is, $S(a,b)=a$.

without symmetrization function, the additive normalization reduces the photometric error. LMSAN achieves a lower photometric error with reflexive neighborhoods than with nonreflexive ones in this respect. The impact of the symmetrization on the RMSE is illustrated in Fig. 10 for three symmetrization functions S (minimum, maximum, and average). The minimum provides the lowest RMSE.

8 Experiments with Sharp and Blurred Images

For this series of experiments, we use gray-level images consisting of 64^2 pixels. Two kinds of images are used: sharp ones and blurred ones. Sharp images are based on the pattern

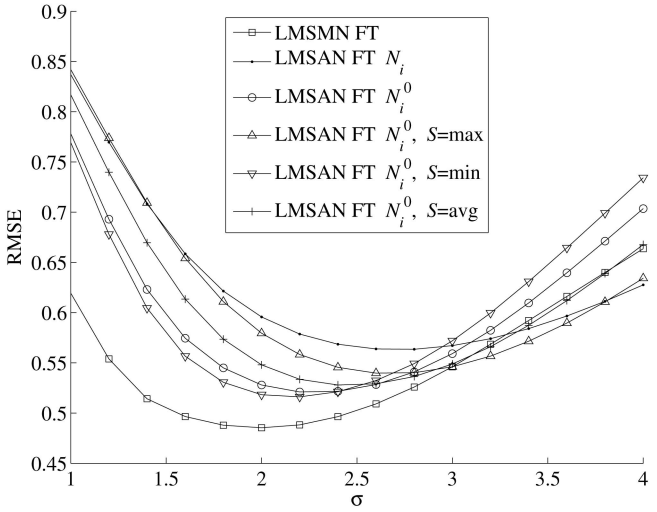

Fig. 10. Closer look at the curves in Fig. 8. The results of LMSMN and LMSAN are given as a function of σ; the average RMSEs over 100 trials are shown. For LMSAN, additional curves show the effect of different symmetrization functions S.

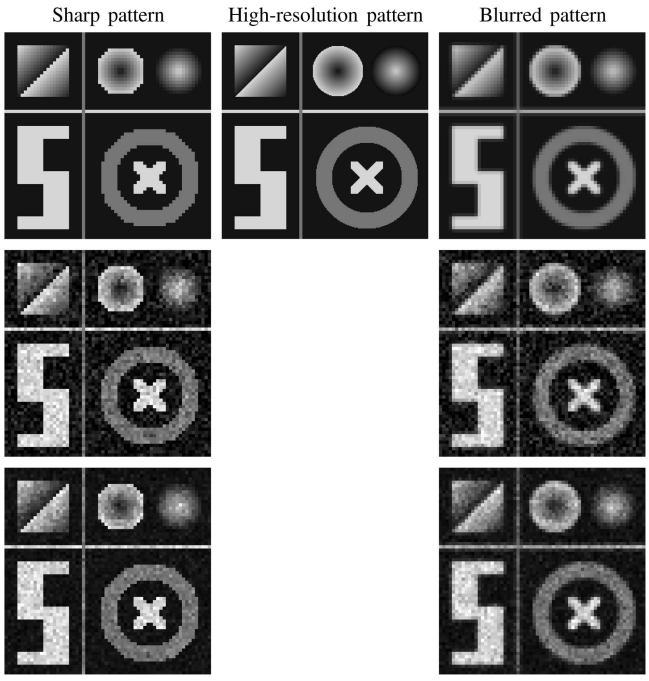

Fig. 11. Image patterns used in the experiments. In the left column, the image is sharp (each pixel fully belongs to a single region, and only three intensities are permitted). In the middle column, a similar pattern is shown with a four times finer grid of pixels. In the right column, this high-resolution pattern is blurred using bilinear interpolation; in this case, pixels near the boundary of two regions can have intermediate intensities. The last two rows show the corresponding images with Gaussian and Poissonian noise.

that can be seen in the left column in Fig. 11. The same pattern is also generated in a 256-by-256 grid, as shown in the middle column in Fig. 11. This second pattern is then resized to an image having 64^2 pixels using a bilinear interpolation in order to obtain the blurred pattern in the right column in Fig. 11. In the case of the sharp pattern, each pixel belongs to a single region, and all regions have strictly constant intensities. However, this model is not very realistic for most imaging devices. As the image grid is quite coarse, portions of two or

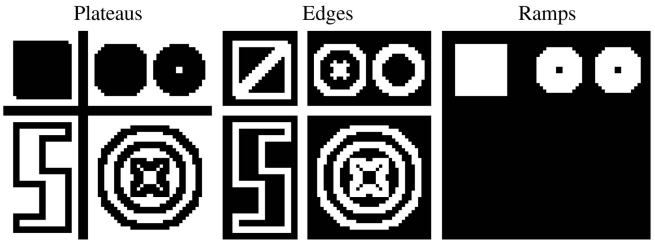

Fig. 12. Masks corresponding to plateaus, edges, and ramps in the computation of RMSE and %PE; black and white corresponds to values of ν_i equal to 0 and 1, respectively.

more objects can be included in the same pixel. In other words, near boundaries, several objects can contribute to the intensity of a single pixel. This phenomenon is also referred to as the "partial volume effect" [27]. As a consequence, transitions between uniform regions are smooth and approximately one-pixel wide. The second pattern simulates this minimal blur observed in most digital images. For very low-resolution devices, blur can be even more important, but then, the application of an edge-preserving filter becomes questionable, and other models should be used.

The experiments involve both Gaussian and Poissonian noise. In the former case (Fig. 11, penultimate row), noise is generated as follows: $f_i \sim G(\mu_i, 1)$ and $\mu_i \in \{1, 5.5, 10\}$. In the latter case (Fig. 11, last row), we have $f_i \sim P(\mu_i)$ and $\mu_i \in \{10, 55, 100\}$.

The quality of the filtering process is measured with both the RMSE and the photometric error. However, we generalize (44) and (50) in order to focus on particular regions in the images. This gives

$$\mathrm{RMSE} \doteq \sqrt{\frac{1}{M\sum_{i=1}^{N}\nu_i}\sum_{m=1}^{M}\sum_{i=1}^{N}\nu_i(\hat{u}_{im}-\mu_{im})^2}, \qquad (67)$$

and

$$\%\mathrm{PE} \doteq \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(\frac{100}{\sum_{i=1}^{N}\nu_i}\sum_{i=1}^{N}\nu_i\left(1-\frac{\hat{u}_{im}}{f_{im}}\right)\right)^2}, \qquad (68)$$

where the weights ν_i are chosen in {0, 1}. Obviously, %PE vanishes for filters that have the property of photometry invariance only if all pixels share the same weight. In the case of Poissonian noise, the RMSE with embedded Fisz's transform defined in (66) is generalized in the same way. Four masks are constituted. The first mask is uniform and assesses the recovery of the whole image (100 percent of the image). The second mask includes plateaus, that is, regions with constant intensity (approximately 54 percent of the image). The third one focus on pixels near edges (approximately 39 percent of the image). The last mask gathers all intensity ramps (approximately 14 percent of the image). The last three masks are shown in Fig. 12. There is some overlapping between edges and ramps.

The results of Gaussian smoothing are given in Tables 1 and 2 for $M = 100$. Four cases are detailed: Gaussian noise with classical Gaussian smoothing (GN, GS), Poissonian noise with no VST (PN, GS), with Anscombe's transform (PN, GSAT), and with Fisz's transform (PN, GSFT). In the last case, six iterations were performed. Each row in Tables 1 and 2

TABLE 1
Results with the Sharp Image Pattern in Fig. 11

	ρ σ	image	plateaus	edges	ramps
Gaussian noise, no VST					
GS					
image	0.46	*0.00* 0.83	*0.08* 0.72	*0.08* 0.97	*1.88* 0.83
plateaus	0.88	*0.00* 1.32	*1.58* 0.36	*1.04* 2.06	*6.91* 1.26
edges	0.40	*0.00* 0.87	*0.04* 0.85	*0.04* 0.91	*0.97* 0.88
ramps	0.46	*0.00* 0.83	*0.08* 0.72	*0.08* 0.97	*1.88* 0.83
LMSMN					
image	1.45 0.50	*0.51* **0.40**	*0.55* **0.26**	*0.97* **0.52**	*1.81* 0.59
plateaus	2.77 0.64	*2.06* 0.58	*0.68* **0.21**	*4.17* 0.85	*4.18* 0.72
edges	1.37 0.48	*0.41* **0.41**	*0.55* **0.27**	*0.74* **0.52**	*1.56* 0.59
ramps	0.97 0.94	*0.83* 0.51	*0.33 0.31*	*1.83* 0.71	*1.40* **0.45**
LMSAN					
image	1.02 0.90	*0.00* 0.47	*0.32* 0.30	*0.23* 0.64	*0.96* **0.50**
plateaus	1.26 1.09	*0.00* **0.51**	*0.79* 0.27	*0.42* **0.73**	*1.70* **0.50**
edges	0.70 0.83	*0.00* 0.49	*0.25* 0.37	*0.22* 0.62	*0.63* **0.50**
ramps	0.85 1.07	*0.00* **0.50**	*0.31* 0.32	*0.25* **0.69**	*1.14* 0.46
Poissonian noise, no VST					
GS					
image	0.38	*0.00* 0.93	*0.02* 0.89	*0.02* 0.99	*0.71* 0.92
plateaus	0.74	*0.00* 1.90	*0.63* 0.41	*0.43* 2.99	*5.52* 1.41
edges	0.34	*0.00* 0.96	*0.01* 0.96	*0.01* 0.96	*0.31* 0.96
ramps	0.42	*0.00* 0.96	*0.03* 0.81	*0.03* 1.15	*1.26* 0.91
LMSMN					
image	1.23 4.96	*0.64* **0.40**	*0.28* **0.26**	*0.86* **0.51**	*2.34* 0.68
plateaus	3.85 3.48	*1.20* 0.51	*0.42* **0.20**	*1.81* 0.68	*4.23* 1.17
edges	1.12 4.90	*0.55* **0.40**	*0.27* **0.28**	*0.72* **0.50**	*1.99* 0.66
ramps	0.83 9.09	*0.70* 0.57	*0.14* **0.35**	*1.37* 0.79	*1.27* **0.52**
LMSAN					
image	1.02 6.95	*0.00* 0.46	*0.18* 0.30	*0.13* 0.61	*0.77* **0.60**
plateaus	1.34 7.60	*0.00* **0.48**	*0.33* 0.27	*0.14* **0.66**	*1.09* **0.65**
edges	0.72 6.43	*0.00* 0.48	*0.13* 0.37	*0.14* 0.60	*0.53* **0.58**
ramps	0.73 9.19	*0.00* **0.55**	*0.16* 0.36	*0.15* **0.75**	*0.80* **0.52**
Poissonian noise, preprocessing with Anscombe's transform					
GS					
image	0.40	*1.65* 0.92	*0.29* 0.86	*3.30* 1.00	*1.83* 0.92
plateaus	0.80	*7.39* 1.89	*1.04* 0.39	*14.72* 2.98	*9.85* 1.74
edges	0.36	*0.86* 0.94	*0.15* 0.93	*1.72* 0.96	*0.94* 0.94
ramps	0.40	*1.65* 0.92	*0.29* 0.86	*3.30* 1.00	*1.83* 0.92
LMSMN					
image	1.28 0.95	*1.02* **0.37**	*0.80* **0.26**	*1.23* **0.47**	*2.29* 0.63
plateaus	2.90 0.72	*1.14* 0.46	*0.77* **0.21**	*1.68* 0.62	*3.05* 1.03
edges	1.14 0.92	*0.92* **0.37**	*0.79* **0.28**	*1.08* **0.46**	*1.95* 0.59
ramps	0.85 1.34	*1.07* **0.44**	*0.83* 0.36	*1.37* **0.54**	*1.67* 0.51
LMSAN					
image	1.11 1.22	*0.95* 0.42	*0.92* 0.28	*0.98* 0.57	*1.68* **0.55**
plateaus	1.45 1.37	*1.18* **0.44**	*0.89* 0.26	*1.33* **0.61**	*2.29* **0.61**
edges	0.80 1.17	*0.84* 0.44	*0.91* 0.35	*0.84* 0.56	*1.33* **0.51**
ramps	0.78 1.36	*0.96* 0.45	*0.90* **0.35**	*1.07* 0.57	*1.53* **0.50**
Poissonian noise, with embedded Fisz's transform					
GS					
image	0.44	*2.23* 0.88	*0.43* 0.79	*4.42* 1.02	*2.31* 0.89
plateaus	0.88	*7.59* 1.84	*1.23* 0.37	*14.87* 2.90	*9.65* 1.69
edges	0.38	*1.05* 0.92	*0.22* 0.91	*2.06* 0.96	*1.07* 0.92
ramps	0.44	*2.23* 0.88	*0.43* 0.79	*4.42* 1.02	*2.31* 0.89
LMSMN					
image	1.28 0.94	*1.01* **0.37**	*0.78* **0.26**	*1.24* **0.47**	*2.27* 0.63
plateaus	2.88 0.74	*1.16* 0.45	*0.75* **0.21**	*1.71* 0.62	*3.15* 1.02
edges	1.15 0.91	*0.91* **0.37**	*0.77* **0.28**	*1.08* **0.46**	*1.93* 0.59
ramps	0.85 1.32	*1.09* **0.44**	*0.81* 0.35	*1.44* **0.54**	*1.67* 0.52
LMSAN					
image	1.09 1.16	*0.00* 0.42	*0.17* 0.27	*0.16* 0.57	*0.55* **0.54**
plateaus	1.40 1.21	*0.00* **0.43**	*0.22* 0.26	*0.17* **0.59**	*0.73* **0.59**
edges	0.78 1.11	*0.00* 0.44	*0.15* 0.34	*0.15* 0.56	*0.39* **0.51**
ramps	0.78 1.35	*0.00* 0.45	*0.16* **0.34**	*0.15* 0.58	*0.57* **0.49**

The first column indicates the mask (whole image, plateaus, edges, and ramps; see Fig. 12) for which the RMSE is minimized. The second and third columns give the values of ρ and σ at the minimum. The last four columns show the photometric errors (%PE, in italics) and the RMSEs that are measured in all four masks. Bold figures indicate the best results.

shows the values of ρ that yields the lowest RMSE (the corresponding values of %PE precede in italics) for the mask mentioned in the first column. The last four columns give the RMSEs and photometric errors for all masks, which correspond to these parameter values. Figs. 13 and 14 show the residual noise over 100 trials, that is, the average over m of $\hat{u}_{im} - \mu_{im}$ and $\mathcal{F}(\hat{u}_{im}, \mu_{im})$ in the Gaussian and Poissonian cases, respectively. These figures correspond to the lowest RMSEs that are reached over the whole image. Fig. 15 shows graphically the evolution of both the RMSE and photometric error for values of ρ comprised in the interval [0, 1].

Looking at the results of Gaussian filtering in the case of Poissonian noise (see Fig. 15 and Tables 1 and 2), we can see that using a VST allows us to reach lower RMSE values. Moreover, the iterative Gaussian filter that incorporates Fisz's transform systematically outperforms the combination of the classical filter with data that have been preprocessed with Anscombe's transform. On the other hand, only the classical filter ensures the invariance of the total photometry. Filters relying on a VST lead to a photometric error of approximately 2 percent on the whole image.

The results of LMSMN and LMSAN are collected in Tables 1 and 2 ($M = 100$). LMSAN is run with nonreflexive neighborhoods ($\mathcal{N}_i^0$). As in the preliminary examples, the best results were achieved with the minimum as the symmetrization function. For each filter, four cases are

TABLE 2
Results with the Blurred Image Pattern in Fig. 11

	ρ σ	image	plateaus	edges	ramps
Gaussian noise, no VST					
GS					
image	0.56	*0.00* 0.68	*1.15* 0.56	*0.92* 0.84	*1.98* 0.64
plateaus	0.78	*0.00* 0.79	*2.57* 0.45	*2.00* 1.13	*3.67* 0.70
edges	0.48	*0.00* 0.73	*0.71* 0.68	*0.57* 0.79	*1.26* 0.71
ramps	0.60	*0.00* 0.69	*1.38* 0.52	*1.09* 0.89	*2.31* 0.63
LMSMN					
image	0.84 0.82	*0.54* 0.55	*1.57* 0.40	*2.31* 0.73	*1.61* 0.56
plateaus	1.87 0.52	*1.42* 0.64	*2.92* **0.32**	*4.91* 0.92	*3.43* 0.71
edges	0.70 1.00	*0.34* 0.57	*1.23* 0.46	*1.66* **0.71**	*1.38* 0.57
ramps	0.85 0.97	*0.70* 0.56	*1.70* 0.40	*2.76* 0.75	*1.96* 0.55
LMSAN					
image	0.97 0.93	*0.00* **0.53**	*2.06* **0.36**	*1.56* **0.72**	*1.68* **0.53**
plateaus	1.14 0.95	*0.00* **0.54**	*2.47* 0.36	*1.83* **0.73**	*1.91* **0.54**
edges	0.64 0.91	*0.00* **0.55**	*1.45* **0.42**	*1.15* **0.71**	*1.35* **0.55**
ramps	0.92 1.01	*0.00* **0.54**	*2.12* **0.37**	*1.61* **0.73**	*1.88* **0.52**
Poissonian noise, no VST					
GS					
image	0.48	*0.00* 0.80	*0.69* 0.70	*0.55* 0.94	*1.26* 0.75
plateaus	0.60	*0.00* 0.92	*1.34* 0.62	*1.06* 1.27	*2.30* **0.74**
edges	0.42	*0.00* 0.84	*0.39* 0.81	*0.31* 0.88	*0.71* 0.83
ramps	0.54	*0.00* 0.84	*1.02* 0.64	*0.80* 1.09	*1.80* 0.72
LMSMN					
image	0.69 6.59	*0.62* **0.65**	*0.89* **0.51**	*1.91* **0.83**	*1.16* **0.66**
plateaus	1.85 2.53	*2.06* 0.88	*0.78* **0.40**	*4.72* 1.27	*2.33* 1.01
edges	0.58 9.40	*0.31* 0.68	*0.78* 0.58	*1.22* **0.80**	*0.96* 0.69
ramps	0.71 9.51	*0.70* **0.67**	*1.11* **0.52**	*2.27* **0.87**	*1.67* 0.65
LMSAN					
image	0.82 5.84	*0.00* 0.66	*1.87* 0.51	*1.46* 0.84	*0.83* **0.66**
plateaus	1.13 5.08	*0.00* **0.68**	*2.11* 0.49	*1.67* **0.87**	*0.80* 0.77
edges	0.47 5.62	*0.00* **0.67**	*1.44* **0.56**	*1.13* 0.82	*0.64* **0.67**
ramps	0.82 7.90	*0.00* 0.71	*2.00* 0.54	*1.54* 0.92	*1.31* **0.61**
Poissonian noise, preprocessing with Anscombe's transform					
GS					
image	0.50	*2.18* 0.79	*0.14* 0.67	*4.39* 0.96	*2.41* 0.76
plateaus	0.66	*3.67* 0.96	*0.14* 0.57	*7.64* 1.37	*4.42* 0.83
edges	0.44	*1.42* 0.82	*0.15* 0.78	*2.82* 0.88	*1.53* 0.81
ramps	0.52	*2.40* 0.80	*0.13* 0.64	*4.87* 1.00	*2.69* 0.75
LMSMN					
image	0.75 0.89	*1.01* 0.64	*0.42* 0.47	*1.55* **0.84**	*1.48* 0.65
plateaus	1.85 0.50	*1.05* 0.80	*0.46* **0.36**	*1.67* 1.16	*2.05* 0.91
edges	0.59 1.51	*1.30* 0.68	*0.37* 0.57	*2.25* **0.81**	*1.66* 0.68
ramps	0.73 1.23	*1.48* 0.65	*0.36* 0.48	*2.57* 0.86	*1.98* 0.64
LMSAN					
image	0.95 1.01	*1.19* **0.62**	*0.45* **0.42**	*1.83* **0.84**	*1.93* **0.62**
plateaus	1.20 1.03	*1.24* **0.62**	*0.40* 0.40	*1.94* **0.85**	*2.07* **0.66**
edges	0.53 0.94	*1.01* **0.64**	*0.48* **0.50**	*1.51* 0.82	*1.54* **0.64**
ramps	0.83 1.16	*1.38* **0.63**	*0.36* **0.44**	*2.30* **0.84**	*2.25* **0.61**
Poissonian noise, with embedded Fisz's transform					
GS					
image	0.52	*2.29* 0.78	*0.21* 0.65	*4.54* 0.95	*2.51* 0.74
plateaus	0.68	*3.65* 0.93	*0.11* 0.55	*7.49* 1.33	*4.34* 0.82
edges	0.44	*1.35* 0.82	*0.21* 0.79	*2.60* 0.88	*1.42* 0.81
ramps	0.54	*2.50* 0.78	*0.20* 0.62	*4.99* 0.99	*2.77* 0.74
LMSMN					
image	0.75 0.94	*1.04* 0.64	*0.41* 0.47	*1.62* 0.84	*1.52* 0.65
plateaus	1.82 0.51	*1.05* 0.80	*0.43* **0.36**	*1.67* 1.17	*2.08* 0.90
edges	0.59 1.52	*1.35* 0.67	*0.39* 0.57	*2.33* 0.81	*1.71* 0.67
ramps	0.73 1.25	*1.52* 0.65	*0.36* 0.48	*2.65* 0.85	*2.02* 0.64
LMSAN					
image	0.95 0.99	*0.00* **0.61**	*0.80* **0.43**	*0.59* **0.82**	*0.92* **0.61**
plateaus	1.15 0.91	*0.00* **0.62**	*0.73* 0.42	*0.55* **0.83**	*0.80* **0.67**
edges	0.59 1.00	*0.00* **0.63**	*0.69* **0.49**	*0.52* **0.80**	*0.85* **0.61**
ramps	0.85 1.17	*0.00* **0.62**	*1.00* **0.47**	*0.72* **0.82**	*1.20* **0.59**

The first column indicates the mask (whole image, plateaus, edges, and ramps; see Fig. 12) for which the RMSE is minimized. The second and third columns give the values of ρ and σ at the minimum. The last four columns show the photometric errors ($\%\mathrm{PE}$, in italics) and the RMSEs that are measured in all four masks. Bold figures indicate the best results.

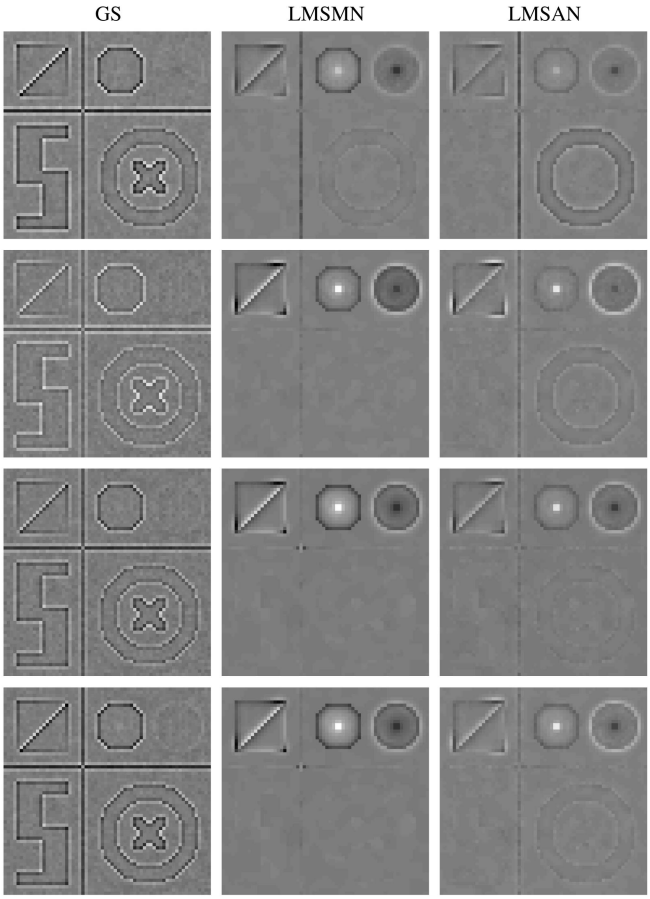

Fig. 13. Residual images for the sharp image pattern and for the different filters, averaged over 100 trials. The first row illustrates the case of Gaussian noise. The last three rows give the residues for Poissonian noise, without VST, with Anscombe's transform, and with Fisz's, respectively.

detailed: Gaussian noise, Poissonian noise without VST, Poissonian noise with Anscombe's transform, and Poissonian noise with embedded Fisz's transform. For all filters, six iterations were performed. Each row in Tables 1 and 2 shows the values of ρ and σ that yield the lowest RMSEs (the corresponding values of $\%\mathrm{PE}$ precede in italics) for the mask mentioned in the first column. The last four columns give the RMSEs and photometric errors for all masks, which correspond to these parameter values. The residual noise over 100 trials is shown in Figs. 13 and 14, when the RMSE is minimized over the whole image.

LMSMN and LMSAN always provide better results than Gaussian filtering. For the sharp image pattern (Table 1), LMSMN outperforms LMSAN in almost all cases, except for ramps corrupted by Poissonian noise. Applying LMSMN and LMSAN without VST to Poissonian data leads to good but suboptimal results; in that case, the photometric error remains low for LMSMN, except for ramps. Looking at the values of ρ and σ that correspond to the lowest RMSEs, we can see that optima for the various weighting masks are located far from each other ($0.83 \leq \rho \leq 3.85$ and $4.90 \leq \sigma \leq 9.09$ for LMSMN). The best trade-off for σ eventually depends on the image content in this case. If we use a VST, optima lie closer to each other ($0.85 \leq \rho \leq 2.88$ and $0.94 \leq \sigma \leq 1.32$ for LMSMN with Fisz's transform), in particular for LMSAN ($0.78 \leq \rho \leq 1.40$ and $1.11 \leq \sigma \leq 1.35$ with Fisz's transform). The type of the VST does not really influence the location of the RMSE optima, for all three filters. Notice that for plateaus, LMSMN reaches the minimal RMSE with $\rho \approx 2.8$ and $\sigma \approx 0.7$, whereas LMSAN minimizes the RMSE with $\rho \approx 1.4$ and $\sigma \approx 1.3$. This means that LMSMN increases the width of the spatial kernel while reducing that of the tonal kernel in order to recover plateaus. By comparison, in the case of LMSAN, the spatial width ρ is only slightly increased, and σ is kept high. To illustrate intuitively these two different behaviors, let us consider the $i\mathrm{th}$ pixel in some plateau, not so far away from an edge (the edge crosses $\mathcal{N}_i^0$). Thanks to the multiplicative normalization, the effective weights of LMSMN of faraway neighbors that lie inside the plateau can be scaled up; on the other hand, contributions of neighbors that lie outside the plateau remain negligible. This provides us with a good denoising performance in the plateaus. As for LMSAN, the effective weights do not involve any adaptive scaling, increasing ρ to a large extent tends to increase W_{ii}, in order to compensate for the low contributions of pixels that lie outside the plateau. Obviously, a high value of W_{ii} decreases the denoising efficiency of LMSAN.

For the blurred image pattern (Table 2), LMSMN and LMSAN achieve comparable results. LMSMN takes the advantage for plateaus, whereas LMSAN dominates in the ramps. With a VST, LMSAN globally outperforms LMSMN for all weightings, except for plateaus. In particular, LMSAN performs better with Fisz's transform than with Anscombe's. The photometric error follows a similar

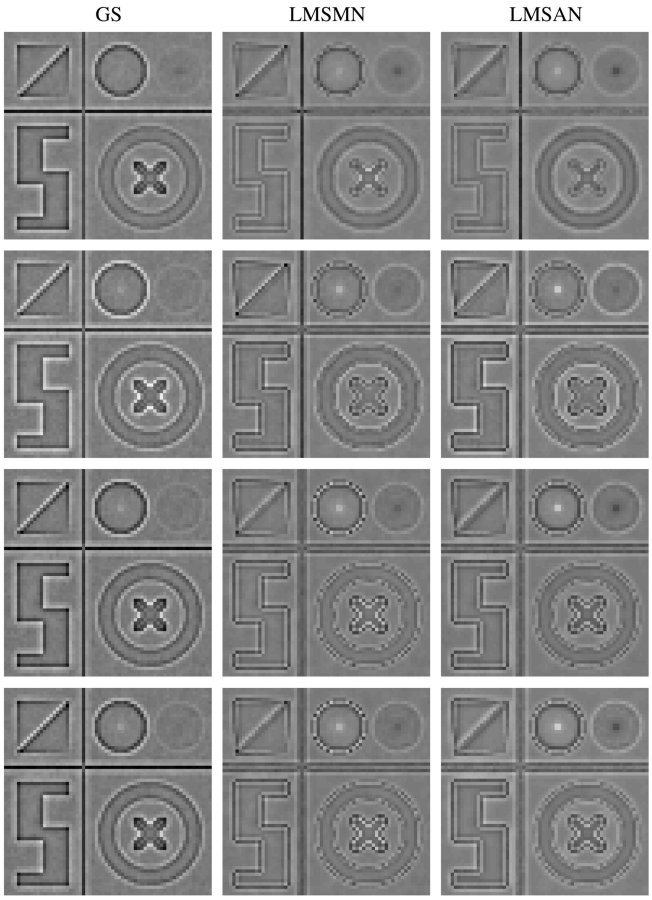

Fig. 14. Residual images for the blurred image pattern and for the different filters, averaged over 100 trials. The first row illustrates the case of Gaussian noise. The last three rows give the residues for Poissonian noise, without VST, with Anscombe's transform, and with Fisz's, respectively.

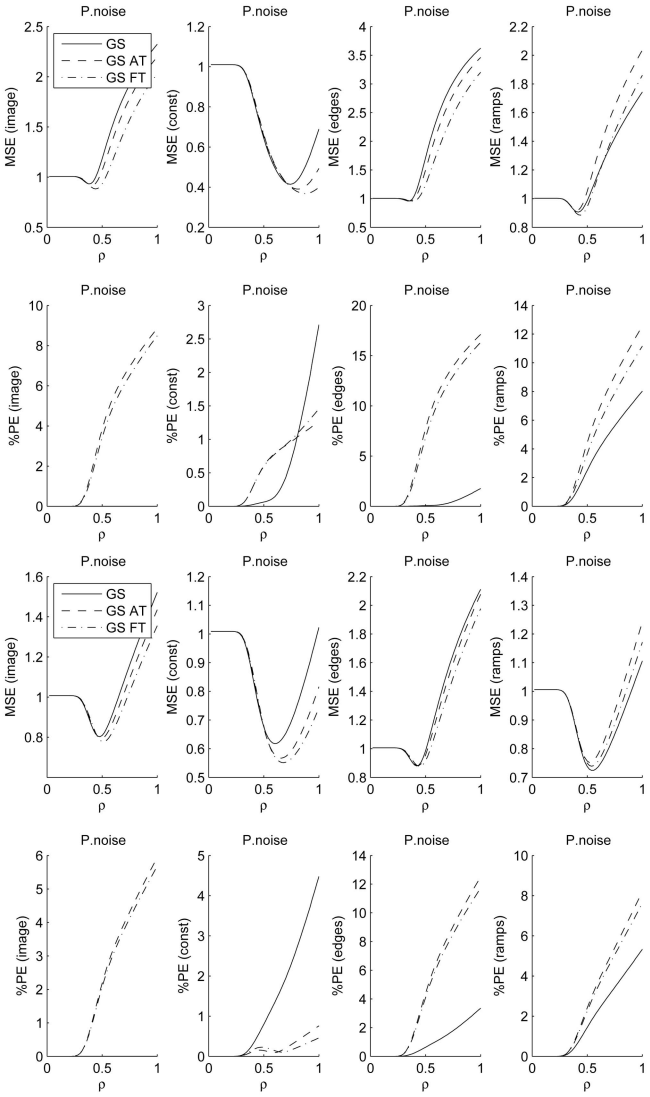

Fig. 15. Results of Gaussian filtering (RMSE and $\%\mathrm{PE}$) for the sharp and blurred image patterns (upper eight plots and lower eight plots, respectively). The columns give the results for the whole image, the plateaus, the edges, and the ramps in the image patterns. In each plot, the error value is shown as a function of ρ, the width of the spatial Gaussian kernel.

evolution and is lower with LMSAN than with LMSMN, except for plateaus. With the blurred pattern, LMSAN thus combines the lowest RMSEs with the invariance of the total photometry; in addition, low photometric errors are also observed in subsets of the image, except in plateaus. As was the case with the sharp pattern, the optima of LMSAN lie closer to each other than those of LMSMN. For LMSAN, we have $0.59 \leq \rho \leq 1.15$ and $0.91 \leq \sigma \leq 1.17$, compared to $0.59 \leq \rho \leq 1.82$ and $0.51 \leq \sigma \leq 1.52$.

By comparison with LMSMN, the overall better performance of LMSAN in difficult situations, that is, in intensity ramps and near blurred edges, can be explained as follows: In regions of the image that are not strictly piecewise constant, the reflexive weight W_{ii} computed by LMSAN is higher than that produced by LMSAN. Similarly, the weights of the neighbors ($W_{ij},\ j \in \mathcal{N}_i^0$) are smaller with LMSAN than LMSMN. In nonconstant regions, this strategy of giving more importance to the considered pixel and less to its neighbor leads to a good trade-off: denoising is lower, but the distortion of the underlying signal is weaker as well.

9 Conclusion

Classical edge-preserving filters such as local M-smoothers and bilateral filters can be modified in order to ensure the invariance of the total photometry. For this purpose, the usual normalization that scales all filter coefficients can be replaced with an additive normalization. In that case, the normalization adjusts only the coefficient of the pixel to be filtered, whereas the coefficient of neighboring pixels is left unchanged. In this way, all coefficients of the weighted average sum to one and the respective coefficients of two neighboring pixels can easily be made symmetric. Experiments show that preserving the total photometry with an additive normalization leads to comparable performances in terms of the RMSE.

In order to deal with Poissonian data, Anscombe's VST can be used. However, filtering the transformed data and applying the inverse transform do not guarantee the invariance of the total photometry, even if the filter involves an additive normalization. We show how to overcome this difficulty, namely, by incorporating another VST (Fisz's transform) directly in the filters. In this way, we obtain versions of for example, Gaussian filtering, local

M-smoothers, and bilateral filtering that can denoise Poissonian data without any preprocessing. Experiments show that Gaussian filtering with Fisz's transform outperforms classical Gaussian filtering, namely, without VST, applied on either the raw data or data preprocessed with Anscombe's transform. A similar improvement is observed for local M-smoothers, although the RMSE advantage of Fisz's transform over Anscombe's is smaller in this case. On the other hand, the combination of additive normalization with the embedded Fisz's transform allows the local M-smoothers to preserve the total photometry, whereas this is impossible with Anscombe's.

The model behind edge-preserving filters like local M-smoother makes them optimal for piecewise constant data. Provided this condition is met, experiments show that the usual multiplicative normalization allows local M-smoothers to yield better results than the proposed additive normalization. However, if the signal is slightly blurred, as is often the case in reality, then the additive normalization outperforms the multiplicative one. An intuitive explanation can be sketched as follows: In neighborhoods that encompass a blurred jump, the additive normalization leads to lower filter coefficients than the multiplicative one. Consequently, constant regions of the signal are denoised almost as well as with the multiplicative normalization, whereas the blurred jumps are filtered to a lesser extent. This proves to be a good trade-off in this more difficult case.

REFERENCES

[1] P.E. Valk, D.L. Bailey, D.W. Townsend, and M.N. Maisey, *Positron Emission Tomography: Basic Science and Clinical Practice.* Springer, 2003.

[2] R.J. Jaszczak and B.M.W. Tsui, "Single Photon Emission Computed Tomography," *Principles of Nuclear Medicine,* H.N.J. Wagner, ed., second ed., W.B. Saunders, 1995.

[3] G. Winkler, V. Aurich, K. Hahn, A. Martin, and K. Rodenacker, "Noise Reduction in Images: Some Recent Edge-Preserving Methods," *Pattern Recognition and Image Analysis,* vol. 9, no. 4, pp. 749-766, Apr. 1999.

[4] A.K. Jain, *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[5] C.D. Ramos, Y.E. Erdi, M. Gonen, E. Riedel, H.W.D. Yeung, H.A. Macapinlac, R. Chisin, and S.M. Larson, "FDG-PET Standardized Uptake Values in Normal Anatomical Structures Using Iterative Reconstruction Segmented Attenuation Correction and Filtered Back-Projection," *European J. Nuclear Medicine and Molecular Imaging,* vol. 28, no. 2, pp. 155-164, Feb. 2001.

[6] F.J. Beekman, K.C. Frey, and E.C. Frey, "Scatter Compensation Methods in 3D Iterative Spect Reconstruction: A Simulation Study," *Physics in Medicine and Biology,* vol. 42, pp. 1619-1632, 1997.

[7] F.J. Beekman, E.T.P. Slijpen, and W.J. Niessen, "Selection of Task-Dependent Diffusion Filters for the Post-Processing of SPECT Images," *Physics in Medicine and Biology,* vol. 43, pp. 1713-1730, 1998.

[8] E.T.P. Slijpen and F.J. Beekman, "Comparison of Post-Filtering and Filtering between Iterations for SPECT Reconstruction," *IEEE Trans. Nuclear Science,* vol. 46, no. 6, pp. 2233-2238, Dec. 1999.

[9] P. Perona and J. Malik, "Scale-Space and Edge-Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, pp. 629-639, 1990.

[10] F.J. Anscombe, "The Transformation of Poisson, Binomial and Negative-Binomial Data," *Biometrika,* vol. 35, nos. 3-4, pp. 246-254, 1948.

[11] D.L. Donoho, "Nonlinear Wavelet Methods for Recovery of Signals, Densities and Spectra from Indirect and Noisy Data," *Proc. Symp. Applied Math.,* vol. 47, 1993.

[12] V. Aurich and J. Weule, "Nonlinear Gaussian Filters Performing Edge Preserving Diffusion," *Proc. 17th Deutsche Arbeitsgemeinschaft für Mustererkennung Symp.,* pp. 538-545, 1995.

[13] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. Sixth Int'l Conf. Computer Vision,* pp. 839-846, 1998.

[14] M. Elad, "On the Origin of the Bilateral Filter and Ways to Improve It," *IEEE Trans. Image Processing,* vol. 11, no. 10, pp. 1141-1151, 2002.

[15] C.K. Chu, I. Glad, F. Godtliebsen, and J.S. Marron, "Edge-Preserving Smoothers for Image Processing," *J. Am. Statistical Assoc.,* vol. 93, no. 442, pp. 526-556, 1998.

[16] P. Mrazek, J. Weickert, and A. Bruhn, "On Robust Estimation and Smoothing with Spatial and Tonal Kernels," Technical Report Preprint no. 51, Univ. of Bremen, June 2004.

[17] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel, *Robust Statistics,* Wiley Series in Probability and Math. Statistics, John Wiley & Sons, 1986.

[18] J. Weickert, B.M. ter Haar Romeny, and M.A. Viergever, "Efficient and Reliable Schemes for Nonlinear Diffusion Filtering," *IEEE Trans. Image Processing,* vol. 7, no. 3, pp. 398-410, Mar. 1998.

[19] P.J. Huber, *Robust Statistics,* Wiley Series in Probability and Math. Statistics. John Wiley & Sons, 1981.

[20] M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger, "Robust Anisotropic Diffusion," *IEEE Trans. Image Processing,* vol. 7, no. 3, pp. 421-432, 1998.

[21] D. Barash, "A Fundamental Relationship between Bilateral Filtering, Adaptive Smoothing and the Nonlinear Diffusion Equation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 6, pp. 844-847, June 2002.

[22] D. Barash and D. Comaniciu, "A Common Framework for Nonlinear Diffusion, Adaptive Smoothing, Bilateral Filtering and Mean-Shift," *Image and Video Computing,* vol. 22, no. 1, pp. 73-81, 2004.

[23] B. Zhang, M.J. Fadili, and J.-L. Starck, "Multi-Scale Variance Stabilizing Transform for Multi-Dimensional Poisson Count Image Denoising," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing,* vol. 2, pp. II-81-II-84, May 2006.

[24] P. Frýlewicz and G.P. Nason, "A Haar-Fisz Algorithm for Poisson Intensity Estimation," *J. Computational and Graphical Statistics,* vol. 13, pp. 621-638, 2004.

[25] M. Fisz, "The Limiting Distribution Function of Two Independent Random Variables and Its Statistical Application," *Colloquium Math,* vol. 3, pp. 138-146, 1955.

[26] L. Devroye, *Non-Uniform Random Variate Generation.* Springer, 1986.

[27] M. Soret, S.L. Bacharach, and I. Buvat, "Partial-Volume Effect in PET Tumor Imaging," *J. Nuclear Medicine,* vol. 48, no. 6, pp. 932-945, 2007.

John A. Lee received the MSc degree in applied sciences (computer engineering) and the PhD degree in applied sciences (machine learning) from the Université Catholique de Louvain (UCL), Belgium, in 1999 and 2003, respectively. He is currently a postdoctoral researcher of the Belgian Fonds National de la Recherche Scientifique. His current work aims at developing specific image restoration techniques for positron emission tomography in the Center for Molecular Imaging and Experimental Radiotherapy, Saint-Luc University Hospital, Belgium. His research interests include machine learning techniques such as nonlinear dimensionality reduction, independent component analysis, cluster analysis, and vector quantization. He is a member of the UCL Machine Learning Group.

Xavier Geets received the MD degree from the Université Catholique de Louvain (UCL), Belgium, in 2000. He is currently working toward the PhD degree at the Center for Molecular Imaging and Experimental Radiotherapy, UCL. He was board certified in radiation oncology in Belgium in 2007. He is currently appointed at the Academic Hospital, UCL, Brussels. His research interests include multimodal imaging in radiation oncology. In particular, he is studying several aspects of accurate target volume definition with positron emission tomography on lung cancers and head and neck cancers.

Vincent Grégoire received the MD degree from the Université Catholique de Louvain (UCL), Belgium, in 1987. He was board certified in radiation oncology in Belgium in 1994 and received the PhD degree in radiation biology in 1996 after a fellowship at The Netherlands Cancer Institute in Amsterdam, Netherlands, and at the MD Anderson Cancer Center, Houston, Texas. Since his return from the United States, he was appointed at the Academic Hospital of the UCL, Brussels, where he is currently the director of the Center for Molecular Imaging and Experimental Radiotherapy, professor in radiation oncology, and head of the Clinic in the Department of Radiation Oncology. Besides his clinical activities, he is running a translational research program on tumor microenvironment and the integration of functional and molecular imaging for treatment planning. He is the director or codirector of seven PhD theses, 125 peer-reviewed publications, and six book chapters. He has delivered close to 200 lectures or teaching seminars worldwide. He is a member of the editorial board of *Radiotherapy & Oncology* and is a member of numerous scientific societies, including ASTRO and ESTRO, on which he serves on various committees. At ESTRO, he has been elected as the president-elect in 2005 and will become the president for two years during the ECCO meeting in 2007. He is the acting chairman of an ICRU Report Committee on "dose prescription, specification, and reporting in IMRT" and was recently appointed as a member of the UICC TNM advisory panel on Head and Neck Cancer.

Anne Bol received the PhD degree in physics from the Université Catholique de Louvain (UCL), Belgium, in 1983. Since then, she has been working as a research associate for the Belgian National Fund for Medical Scientific Research (FRSM), Positron Tomography Laboratory, which is part of the Center for Molecular Imaging and Experimental Radiotherapy of the UCL. She is a lecturer in medical imaging at the UCL. Her research interests include absolute quantization of PET data, tracer kinetic modeling, and image quality assessment.