CWP-695

Structure-oriented bilateral filtering

Dave Hale

Center for Wave Phenomena, Colorado School of Mines, Golden CO 80401, USA

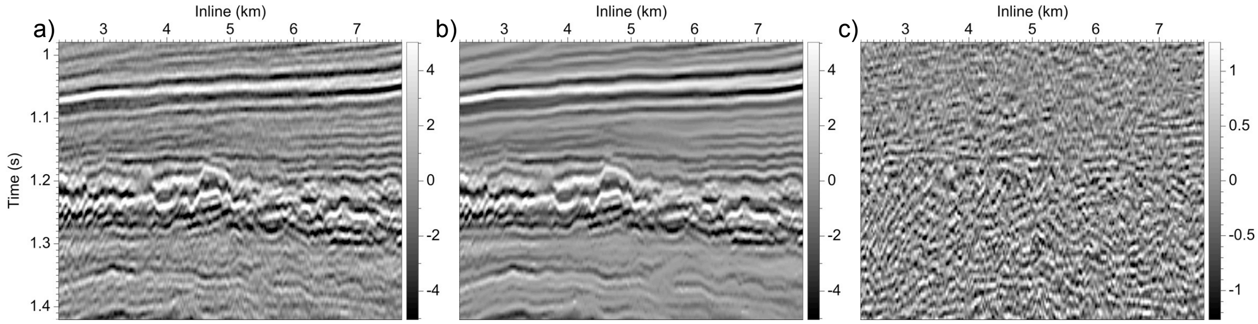

Figure 1. An input seismic image (a), the output of structure-oriented bilateral filtering (b), and the difference (c) between input and output images. For visibility, the input-output difference is displayed for a smaller gray-scale range of amplitudes.

ABSTRACT

Bilateral filtering is widely used to enhance photographic images, but in most implementations is poorly suited to seismic images. A bilateral filter consists of two filter kernels. By replacing one of those kernels with a smoothing filter that conforms to image structures, we obtain a bilateral filter suitable for seismic image processing. Examples and comparison with conventional edge-preserving smoothing illustrate both advantages and disadvantages of structure-oriented bilateral filtering.

Key words: tensor interpolation parametric surface

1 INTRODUCTION

Bilateral filters (Tomasi and Manduchi, 1998) are today widely used to smooth photographic images while preserving significant edges. Paris et al. (2008) provide a thorough review of bilateral filters and their applications, which include denoising, image abstraction, and texture and tone adjustment. Despite such widespread application to photographic images (and to medical CT and MRI scans), bilateral filters are seldom used to enhance seismic images. Why not?

One reason is that edges in seismic images differ significantly from those in photographic images (and in medical CT or MRI scans). Consider for example the seismic image displayed in Figure 2. The most obvious edges in this image are the alternating black and white features that correspond to seismic horizons. However, these sinusoidal features are unlike the edges apparent in most photographs. Rather, these features correspond to

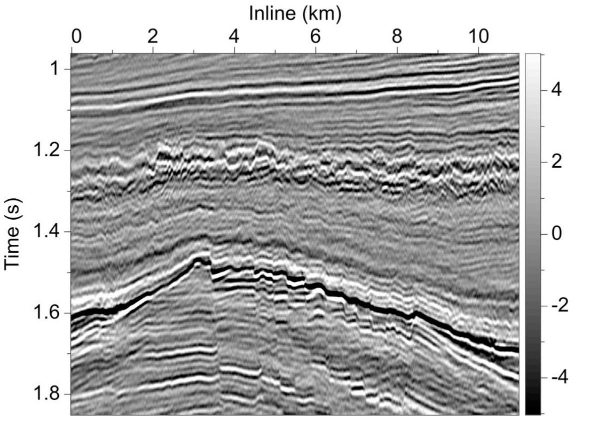

Figure 2. A seismic image used as the input p to various filters described in this paper.

reflections of seismic waves caused by changes in seismic impedance. Edges in images of seismic impedance, when such images are available, more closely resemble edges in photographs.

Also important are edges corresponding to lateral discontinuities in seismic reflections, the chaotic structures at about 1.2 s and the geologic faults below 1.5 s in the image of Figure 2. In processing seismic images, we seek to denoise (enhance the continuity of) coherent reflections, while preserving these lateral discontinuities.

The anisotropic diffusion filter (Weickert, 1999; Fehmers and Höcker, 2003) is one example of a filter that does this for seismic images, as well as for photographic images. However, although the anisotropic diffusion filter has been compared with the bilateral filter in applications to other types of images (e.g., Barash, 2002), I have found no such comparison in their application to seismic images.

In this paper I provide this comparison, for a modified bilateral filter that accounts for the different types of edges apparent in typical seismic images. I used this modified bilateral filter in the example of Figure 1, for a small subset of the image in Figure 2. I show that bilateral filtering of seismic images, like that of photographic images, preserves edges without any prior detection of those edges using coherence or semblance attributes.

Like any denoising filter, the modified bilateral filter proposed in this paper attempts to remove noise while preserving signal. Here, I define noise to be anything removed by a filter; signal is what remains. In other words, if q (Figure 1b) denotes the output image obtained by filtering an input image p (Figure 1a), then noise is the difference $p - q$ (Figure 1c) between input and output. For coherency-enhancing filters, spatial correlation in this difference should be insignificant, when compared with that in the input or output images.

2 SMOOTHING FILTERS

Let $p[\mathbf{i}]$ and $q[\mathbf{i}]$ denote input and output images, respectively, where $\mathbf{i} = (i_1, i_2, \ldots, i_n)$ is an n-dimensional sample index with n integer components. Convolutional smoothing filters have the form

$$q[\mathbf{i}] = \sum_j p[\mathbf{j}]\, s(\mathbf{i} - \mathbf{j}), \tag{1}$$

where $s(\mathbf{k})$ denotes the filter's impulse response. (For simplicity I omit summation limits in this paper; unless stated otherwise, sums include all indices for which the summand is non-zero.) This filter is shift-invariant (convolutional) because the filter coefficients $s(\mathbf{k})$ depend only on the difference $\mathbf{k} = \mathbf{i} - \mathbf{j}$ between output and input sample indices.

2.1 Gaussian smoothing

In a smoothing filter the coefficients $s(\mathbf{k})$ are chosen so that each output sample $q[\mathbf{i}]$ is some sort of weighted average of input samples $p[\mathbf{j}]$. In effect, a smoothing filter attempts to predict signal at sample index $\mathbf{i}$ from nearby noisy input samples $p[\mathbf{j}]$.

One such smoothing filter, widely used in image processing, has a Gaussian impulse response

$$s(\mathbf{k}) = s_0\, e^{-\frac{\mathbf{k}\cdot\mathbf{k}}{2\sigma^2}}, \tag{2}$$

where σ denotes the filter half-width or radius.

The scale factor s_0 is chosen so that $\sum_k s(\mathbf{k}) = 1$. This condition ensures that the smoothing filter does nothing to an input image p that is constant, an image that is already as smooth as it can be.

An alternative to normalizing the filter coefficients in this way would be to scale each output sample $q[\mathbf{i}]$ by the sum of the weights:

$$q[\mathbf{i}] = \frac{\sum_j p[\mathbf{j}]\, s(\mathbf{i} - \mathbf{j})}{\sum_j s(\mathbf{i} - \mathbf{j})}. \tag{3}$$

However, for shift-invariant filters, the denominator (with a simple change of summation index) $\sum_j s(\mathbf{i}-\mathbf{j}) = \sum_k s(\mathbf{k})$ is a constant, independent of the output sample index $\mathbf{i}$. For such filters we must compute the scale factor $1/\sum_k s(\mathbf{k})$ only once.

Applying the Gaussian smoothing filter with radius $\sigma = 16$ samples (0.064 s vertically, 0.4 km horizontally) to the input seismic image p shown in Figure 2, I obtained the output image q shown in Figure 3a. This output image is everywhere nearly zero! The estimate of the noise $p - q$ displayed in Figure 3b (with a different gray scale) is nearly identical to the input image p displayed in Figure 2, and is anything but uncorrelated. This poor result is not surprising because, within any Gaussian window with radius $\sigma = 16$ samples, the input image p has nearly zero mean.

For consistency, all filters shown in this paper have a maximum smoothing half-width of $\sigma = 16$ samples. If we use a smaller filter radius, the isotropic Gaussian filter will perform less smoothing in all directions, and the output will not be so nearly zero.

However, for images p like that in Figure 2, simple shift-invariant filters, such as Gaussian filters or dip filters, will be ineffective, because the shapes and orientations of image features vary spatially. For such images, a more effective alternative is a filter with an anisotropic and spatially varying impulse response, one that conforms to image structures.

2.2 Structure-oriented smoothing

To smooth *along* structures apparent in images, without smoothing *across* those structures, I solve a discrete approximation to the following partial differential equa-

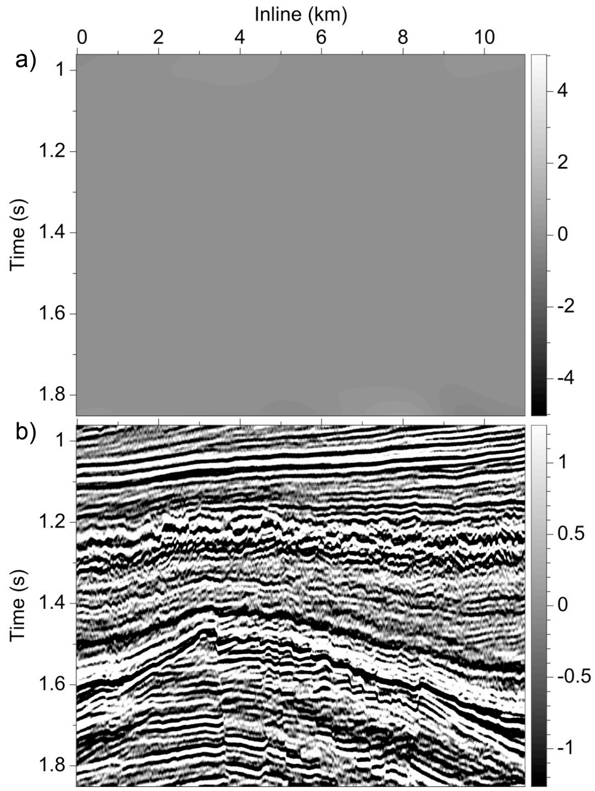

Figure 3. The output q of Gaussian smoothing (a) and the input-output difference $p-q$ (b). For the input p in Figure 2, the output q is everywhere nearly zero.

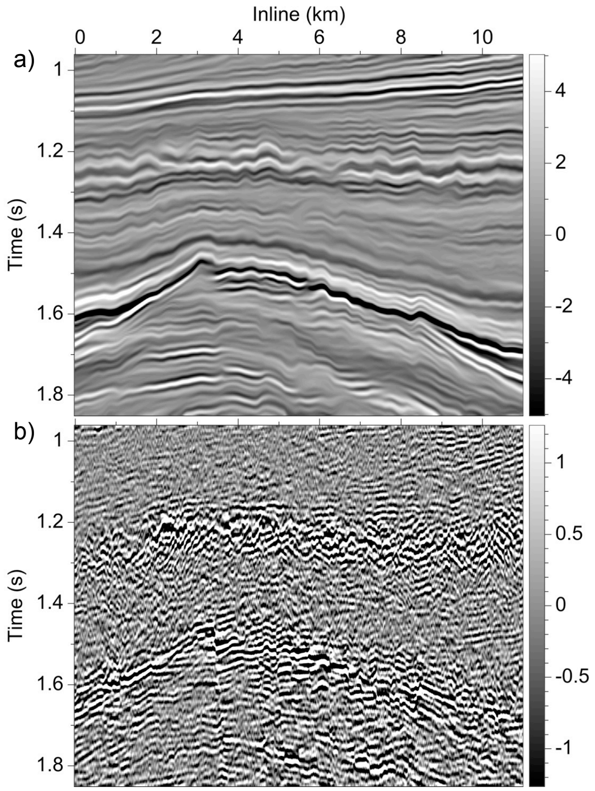

Figure 4. The output q of structure-oriented smoothing (a) and the input-output difference $p-q$ (b).

tion:

$$q(\mathbf{x}) - \frac{\sigma^2}{2}\boldsymbol{\nabla}\bullet\mathbf{D}(\mathbf{x})\bullet\boldsymbol{\nabla}q(\mathbf{x}) = p(\mathbf{x}) \tag{4}$$

for tensor-valued filter coefficients $\mathbf{D}(\mathbf{x})$. Here, $\mathbf{x}$ represents coordinates in space or space-time that when sampled become indices $\mathbf{i}$ and $\mathbf{j}$ in equations 1 and 3.

As shown by Hale (2009), solution of this equation approximates Gaussian smoothing with half-width σ in the directions of eigenvectors of $\mathbf{D}(\mathbf{x})$ for which corresponding eigenvalues equal one. By choosing those directions to to be tangent to structures apparent in an input image p, and by choosing eigenvalues for orthogonal directions to be much less than one, smoothing is oriented along image structures.

Solving equation 4 is similar to filtering with coherence-enhancing anisotropic diffusion (Weickert, 1999; Fehmers and Höcker, 2003). As for that process, I derive the tensor-valued coefficients $\mathbf{D}(\mathbf{x})$ from structure tensors computed for the input image. An efficient method for solving equation 4 is described by Hale (2009).

Figure 4 shows an example of structure-oriented smoothing by solving equation 4 for $\sigma = 16$ samples. Although smoothing has been performed primarily along structures apparent in the input image p shown in Figure 2, significant details have not been preserved in the output image q shown in Figure 4a. In particular, faults in the lower part of the image are poorly resolved, as are sharp discontinuities in chaotic structures just below 1.2 s. Moreover, significant spatial correlation is apparent in the input-output difference displayed in Figure 4b.

2.3 Edge-preserving smoothing

To preserve edges while smoothing, we may scale the tensors $\mathbf{D}(\mathbf{x})$ in equation 4:

$$q(\mathbf{x}) - \frac{\sigma^2}{2}\boldsymbol{\nabla}\bullet c^2(\mathbf{x})\mathbf{D}(\mathbf{x})\bullet\boldsymbol{\nabla}q(\mathbf{x}) = p(\mathbf{x}), \tag{5}$$

where $c(\mathbf{x})$ denotes some normalized measure of coherence that is nearly zero near discontinuities and nearly one where image features are coherent.

In effect, coherence $c(\mathbf{x})$ in equation 5 reduces the maximum half-width σ of the structure-oriented smoothing filter by a factor that varies spatially. Fehmers and Höcker (2003) describe one way to compute the scale factors $c(\mathbf{x})$, using a measure of coherence that requires the computation of two structure tensor fields. Here I compute coherence $c(\mathbf{x})$ as structure-oriented semblance (Hale, 2009) raised to the power 8.

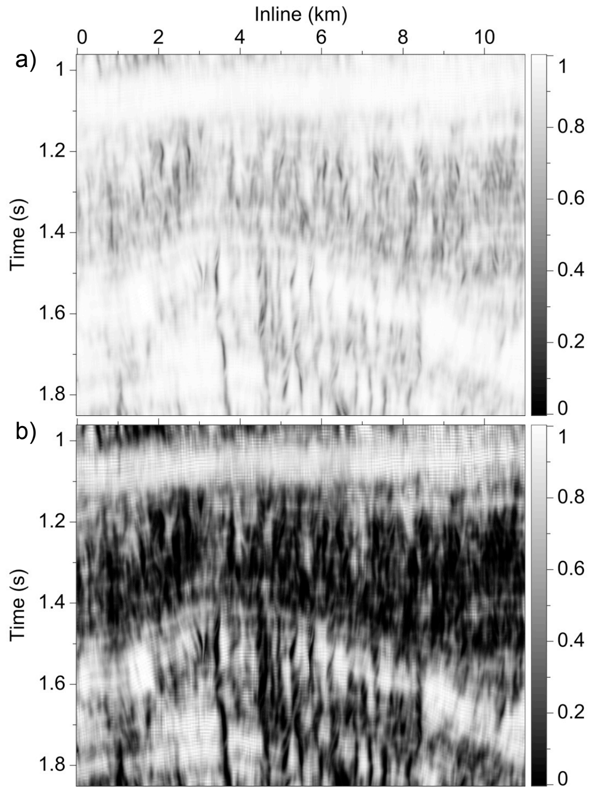

Figure 5. Structure-oriented semblance (a) and coherence (b). Coherence (here, semblance to the power 8) scales tensors used in edge-preserving smoothing.

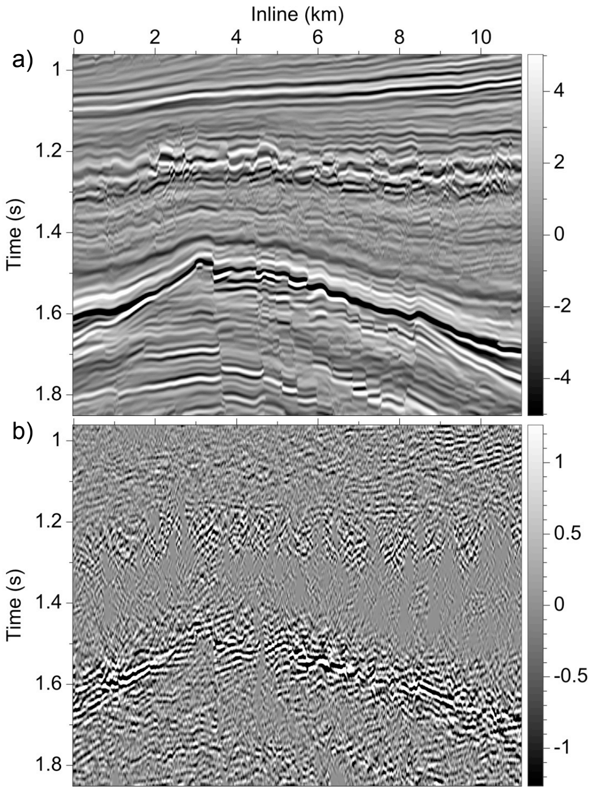

Figure 6. The output q of structure-oriented edge-preserving smoothing (a) and the input-output difference $p - q$ (b).

(This simply formula makes small values of semblance even smaller; only those image features with semblance near one have high coherence. Experimentally, I found that smaller powers do not sufficiently inhibit smoothing across faults.) Both semblance and coherence are displayed in Figure 5.

Figure 6 shows the effect of edge-preserving structure-oriented smoothing with equation 5, using coherence $c(\mathbf{x})$ displayed in Figure 5b. Near discontinuities, such as faults, coherence is low and little smoothing is performed. This smoothing near but not across discontinuities appears to enhance the definition of faults in the output image q of Figure 6a.

The input-output difference in Figure 6b shows that little smoothing is performed near the middle of the image, at times near 1.4 s, where features in the input image p are least coherent. Note, however, that the largest differences in Figure 6b exhibit significant spatial correlation, and that these differences coincide with high-amplitude features in the input image.

2.4 Generalized smoothing filters

As for the Gaussian smoothing filter described earlier, solution of equation 4 when the input image p is constant yields an identical output image $q = p$, because the term in that equation with gradient operators is zero. In other words, like Gaussian smoothing, structure-oriented smoothing is normalized, and can be expressed as follows:

$$q[\mathbf{i}] = \frac{\sum_j p[\mathbf{j}]\, s(\mathbf{i},\mathbf{j})}{\sum_j s(\mathbf{i},\mathbf{j})}. \qquad (6)$$

Note that structure-oriented smoothing cannot be represented by the shift-invariant filter of equation 1. The filter coefficients $s(\mathbf{i},\mathbf{j})$ vary spatially; they depend on both the output index $\mathbf{i}$ and input index $\mathbf{j}$, not merely on their difference. In practice we will never explicitly compute these filter coefficients, because it is more efficient computationally to solve partial differential equations 4 or 5. However, as we next consider bilateral filtering, let us remember that solving equations 4 or 5 is equivalent to filtering with equation 6, for some unspecified filter coefficients $s(\mathbf{i},\mathbf{j})$.

3 BILATERAL FILTERS

The name *bilateral filter* (Tomasi and Manduchi, 1998) was chosen to imply that the kernel of this filter is a combination of two filter kernels, one a function of the input image's spatial *domain* and the other a function of its *range*. The basic idea is simple. We modify the shift-invariant filter equation 3 to scale the coefficients $s(\mathbf{i}-\mathbf{j})$ by a range function $r(p[\mathbf{i}]-p[\mathbf{j}])$ of the difference between two input sample values:

$$q[\mathbf{i}] = \frac{\sum_j p[\mathbf{j}]\, r(p[\mathbf{i}] - p[\mathbf{j}])\, s(\mathbf{i} - \mathbf{j})}{\sum_j r(p[\mathbf{i}] - p[\mathbf{j}])\, s(\mathbf{i} - \mathbf{j})}. \tag{7}$$

The range function $r(p)$ should be chosen to decrease monotonicially with increasing $|p|$. In practice (Durand and Dorsey, 2002), a simple and effective choice is Tukey's biweight function, defined by

$$r(p) \equiv \begin{cases} [1 - (p/\sigma_p)^2]^2 & \text{if } |p| < \sigma_p, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The half-width σ_p of the range function r controls the scaling of the spatial filter coefficients s in equation 7.

The definition of the range function in equation 8 implies that the choice of σ_p should depend on the range of values in the input image p. In practice, I find that a good choice is

$$\sigma_p = \frac{\sqrt{5}}{2}(p_{75} - p_{25}) \tag{9}$$

where p_{25} and p_{75} denote the 25'th and 75'th percentiles (1st and 3rd quartiles), respectively, of the sample values in the input image p. The factor $\sqrt{5}/2 \approx 1.1$ is an artifact of my tests for different types of range functions (as in Durand and Dorsey, 2002). Specifically, Tukey's biweight function with half-width $\sigma_p = \sqrt{5}$ is comparable to a Gaussian function with half-width $\sigma_p = 1$.

Figure 7 illustrates for a synthetic example the effect that the half-width σ_p has on bilateral smoothing. For small values of σ_p, little smoothing is performed, because then only values $p[\mathbf{j}] \approx p[\mathbf{i}]$ are averaged by equation 7 when computing the output value $q[\mathbf{i}]$. For large values of σ_p, scaling by the range function has little effect, and the bilateral filter is merely a spatial smoothing filter, one that does not preserve edges. For a range of intermediate values $2 < \sigma_p < 6$, the bilateral filter attenuates noise while more or less preserving edges in the signal. This synthetic example highlights the effectiveness of the bilateral filter when applied to photographs or medical images with similar step edges.

Recall, however, that edges most apparent in seismic images are reflections with sinusoidal waveforms, which are not step functions. Figure 8 shows the result of using a shift-invariant isotropic Gaussian (with half-width $\sigma = 16$ samples) for the spatial smoothing filter kernel s in equation 7, in bilateral filtering of the seismic image of Figure 2. The output image q shown in Figure 8a is no longer nearly zero (as in Figure 3a), but significant spatial correlation remains apparent in the input-output difference shown in Figure 8b. The bilateral filter preserves faults and other discontinuities, but it removes coherent signal from the seismic image as it attenuates incoherent noise.

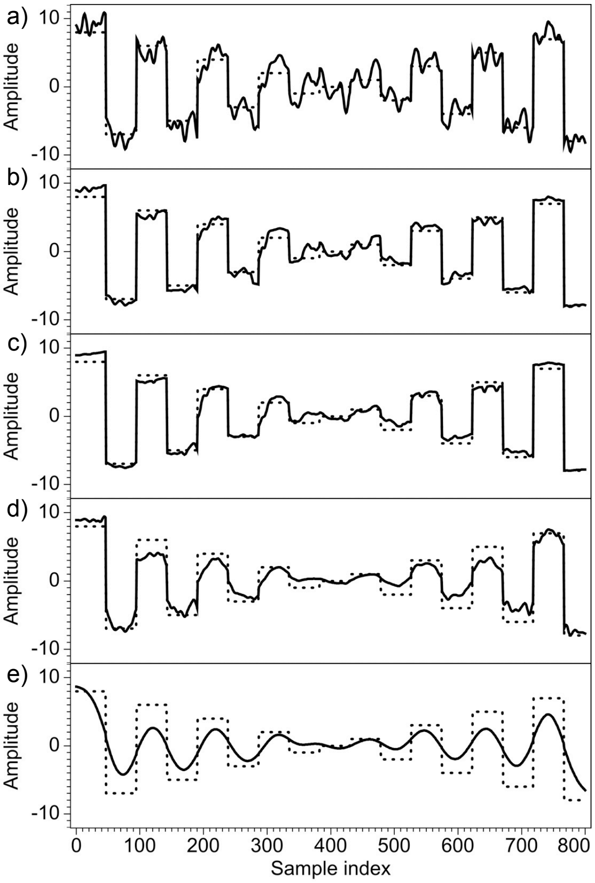

Figure 7. Bilateral filtering of a blocky signal (dashed lines) contaminated with additive random noise. The smoothing filter kernel is Gaussian with half-width $\sigma = 20$ samples. Half-widths σ_p of the Tukey range filter kernel are (a) 1/100, (b) 1/2, (c) 1, (d) 3/2, and (e) 10 times the value $\sigma_p \approx 4$ given by equation 9.

3.1 Structure-oriented bilateral filtering

Although the coefficients $s[\mathbf{i}-\mathbf{j}]$ of the smoothing filter kernel in equation 7 are shift-invariant, the composite bilateral filter is not; the coefficients $r(p[\mathbf{i}]-p[\mathbf{j}])\, s[\mathbf{i}-\mathbf{j}]$ of the composite filter depend on both indices $\mathbf{i}$ and $\mathbf{j}$, not merely on their difference. Therefore, we may as well generalize equation 7 by writing:

$$q[\mathbf{i}] = \frac{\sum_j p[\mathbf{j}]\, r(p[\mathbf{i}], p[\mathbf{j}])\, s(\mathbf{i}, \mathbf{j})}{\sum_j r(p[\mathbf{i}], p[\mathbf{j}])\, s(\mathbf{i}, \mathbf{j})}. \tag{10}$$

This bilateral filter resembles equation 6, with the coefficients $s(\mathbf{i}, \mathbf{j})$ of the smoothing filter kernel again scaled by coefficients $r(p[\mathbf{i}], p[\mathbf{j}])$ of the range filter kernel. This observation leads to the key idea of this paper.

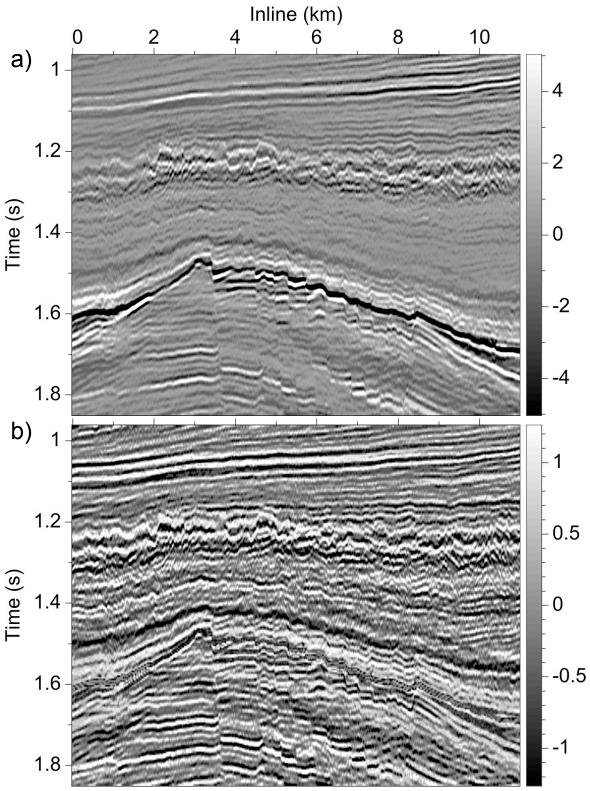

Figure 8. The output q of bilateral filtering (a) and the input-output difference $p - q$ (b).

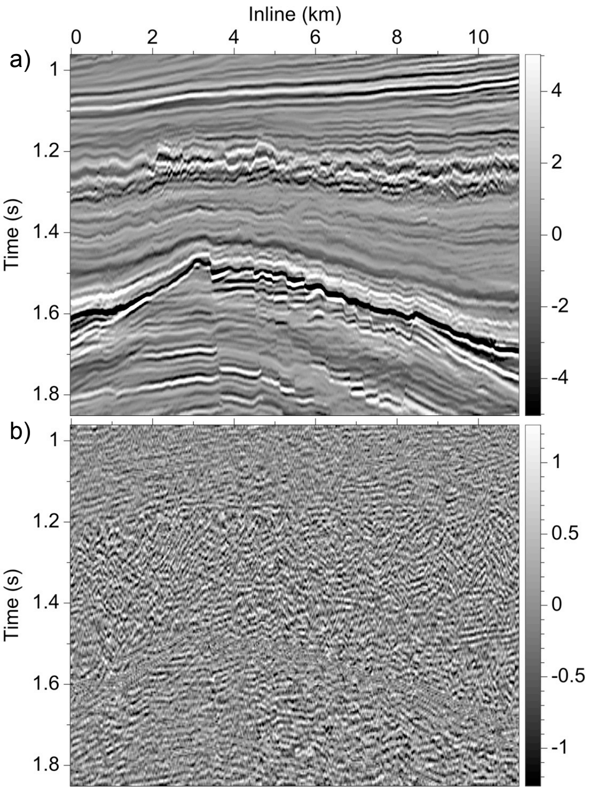

Figure 9. The output q of structure-oriented bilateral filtering (a) and the input-output difference $p - q$ (b).

For structure-oriented bilateral filtering, simply replace the smoothing filter kernel with structure-oriented smoothing.

Remember that the spatial filter coefficients $s(\mathbf{i},\mathbf{j})$ are unknown. We apply the spatial smoothing filter by solving the partial differential equation 4. Because the range filter kernel r is a function of both output and input indices $\mathbf{i}$ and $\mathbf{j}$, an efficient implementation of equation 10 may not be obvious.

My implementation is similar to that of Durand and Dorsey (2002). First, for any value of $p[\mathbf{j}]$, and for $p_1 \leq p[\mathbf{i}] \leq p_2$, let us use simple linear interpolation to approximate

$$r(p[\mathbf{i}],p[\mathbf{j}]) \approx \frac{(p[\mathbf{i}]-p_1)r(p_2,p[\mathbf{j}]) + (p_2-p[\mathbf{i}])r(p_1,p[\mathbf{j}])}{p_2-p_1}. \tag{11}$$

Of course, for any value of $p[\mathbf{j}]$, $r(p[\mathbf{i}],p[\mathbf{j}])$ is not a linear function of $p[\mathbf{i}]$. Therefore, I use a piecewise-linear approximation for a finite number N_p of values $p_k = p_{min} + k\Delta p$, for $k = 0, 1, \ldots, N_p - 1$, where

$$N_p = 2 + \left\lfloor \frac{p_{max} - p_{min}}{\sigma_p} \right\rfloor \tag{12}$$

and

$$\Delta p = \frac{p_{max} - p_{min}}{N_p - 1}. \tag{13}$$

For this piecewise-linear approximation of the range function r, equation 10 becomes

$$q[\mathbf{i}] = \frac{\sum_k \Lambda(p[\mathbf{i}] - p_k) \sum_j p[\mathbf{j}]\, r(p_k, p[\mathbf{j}])\, s(\mathbf{i},\mathbf{j})}{\sum_k \Lambda(p[\mathbf{i}] - p_k) \sum_j r(p_k, p[\mathbf{j}])\, s(\mathbf{i},\mathbf{j})}, \tag{14}$$

where $\Lambda(p[\mathbf{i}]-p_k)$ is a shifted version of the hat function defined by

$$\Lambda(p) \equiv \begin{cases} 1 - \dfrac{|p|}{\Delta p} & \text{if } |p| < \Delta p, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Note that the $\sum_j$ terms in the numerator and denominator of equation 14 resemble those in equation 6. In equation 6 these terms represent structure-oriented smoothing of the images $p[\mathbf{j}]$ and 1 (a constant image). In equation 14 these terms imply exactly the same smoothing of images $p[\mathbf{j}]r(p_k, p[\mathbf{j}])$ and $r(p_k, p[\mathbf{j}])$. For all of these images, we perform spatial smoothing by solving the partial differential equation 4.

Figure 9 displays the result of structure-oriented bilateral filtering of the image in Figure 2. Faults and other discontinuities are well-preserved in the output image shown in Figure 9a; and the noise shown in Figure 9b exhibits the least spatial correlation, when compared with the noise for other filters demonstrated above.

Structure-oriented bilateral filtering preserves faults and other sharp discontinuities in Figure 9a, without using any prior estimate of semblance or coherence. The range function of the bilateral filter inhibits smoothing across a fault where values on each side of the fault differ significantly. As others have noted (e.g., Paris et al., 2008), this simplicity of the bilateral filter is one of its advantages.

For this example, I used $N_p = 19$, which implies that a total of $2N_p = 38$ structure-oriented smoothings were performed, 19 for the numerator and 19 for the denominator of equation 14. This rather large number $N_p = 19$ is necessary for the image of Figure 2 because for this image $p_{min} \ll p_{25}$ and $p_{75} \ll p_{max}$. Specifically, amplitudes of features at about 1.5 s are much higher than for other features in this image. As shown by examples in the following section, the number N_p is lower for images with more balanced amplitudes.

In any case, structure-oriented bilateral filtering generally requires significantly more solutions to equation 4 than the one solution of equation 5 required for edge-preserving structure-oriented smoothing. Relatively high computational cost is therefore a disadvantage of my implementation of structure-oriented bilateral smoothing.

It may be possible to decrease this cost. For example, an iterative solution of the structure-oriented smoothing equation 4 for index k in equation 14 may begin with the solution for index $k-1$. I have not yet tested this or other optimizations.

4 FILTER COMPARISON

To facilitate comparison of structure-oriented bilateral filtering with edge-preserving structure-oriented smoothing, this section displays side by side the outputs and input-output differences for these two filters. For each example, bilateral filtering required $2N_p$ solutions of equation 4, and edge-preserving filtering required one solution of equation 5. The tensors $\mathbf{D}(\mathbf{x})$ in these equations that guided the smoothing varied for different input images, but the same tensors were used for both types of filters. A smoothing half-width $\sigma = 16$ samples was used for both filters in all examples.

Figure 10 is a collection of images already displayed in several figures above. The input image p is displayed in Figure 10a. The output image q and the difference $p-q$ for bilateral filtering are displayed in Figures 10b and 10c, respectively. Figure 10d displays coherence c computed from the the input image p. I used this coherence to scale tensors in edge-preserving smoothing, for which the output q and difference $p-q$ are displayed in Figures 10e and 10f. In this example, $N_p = 19$ for the bilateral filter.

Figure 11 displays the same set of results for a different seismic image. As in the previous example, input-output differences for the bilateral filter exhibit less spatial correlation and are more uniform in amplitude than those for the edge-preserving filter. The bilateral filter appears to better preserve fine details in the signal, while attenuating uncorrelated noise. The output of the edge-preserving filter appears smoother; it has a more cartoonish appearance, which may or may not be an advantage, depending on how the output image is to be used. In this example, $N_p = 8$ for the bilateral filter.

Figure 12 displays the same set of results for a 2D horizontal slice from a 3D seismic image. The output images for both filters are similar. However, as for examples above, input-output differences for the bilateral filter are more uniform in amplitude, and less correlated with input amplitudes, than those for the edge-preserving filter. In this example, $N_p = 10$ for the bilateral filter.

5 CONCLUSION

As demonstrated by the examples above, a bilateral filter suitable for 2D or 3D seismic image processing may be obtained by (1) replacing the smoothing filter kernel in the usual bilateral filter with structure-oriented smoothing and (2) using a piecewise-linear approximation of the range filter kernel.

2D image examples show that bilateral filtering is comparable to edge-preserving smoothing for denoising of seismic images. Both filters attenuate noise while more or less preserving signal.

Edge-preserving smoothing is computationally more efficient, but requires an estimate of coherence (or some other attribute) that highlights edges to be preserved, such as faults. The effectiveness of edge-preserving smoothing depends on this prerequisite image of coherence.

Bilateral filtering requires only the input image, and the noise removed by the filter (the input-output difference), tends to be more uniformly distributed and to exhibit less spatial correlation than that removed by edge-preserving smoothing. Bilateral filtering preserves fine-scale details within coherent image features, as well as faults and other discontinuities.

The primary disadvantage of bilateral filtering is its computational cost, which is relatively high compared with that of edge-preserving smoothing.

ACKNOWLEDGMENTS

The seismic image in Figures 2 and 10a was provided by dGB Earth Sciences B.V. through OpendTect. The seismic image in Figure 11a was provided by the Rocky Mountain Oilfield Testing Center, sponsored by the U.S. Department of Energy. The seismic image in Figure 12a was provided courtesy of WesternGeco.

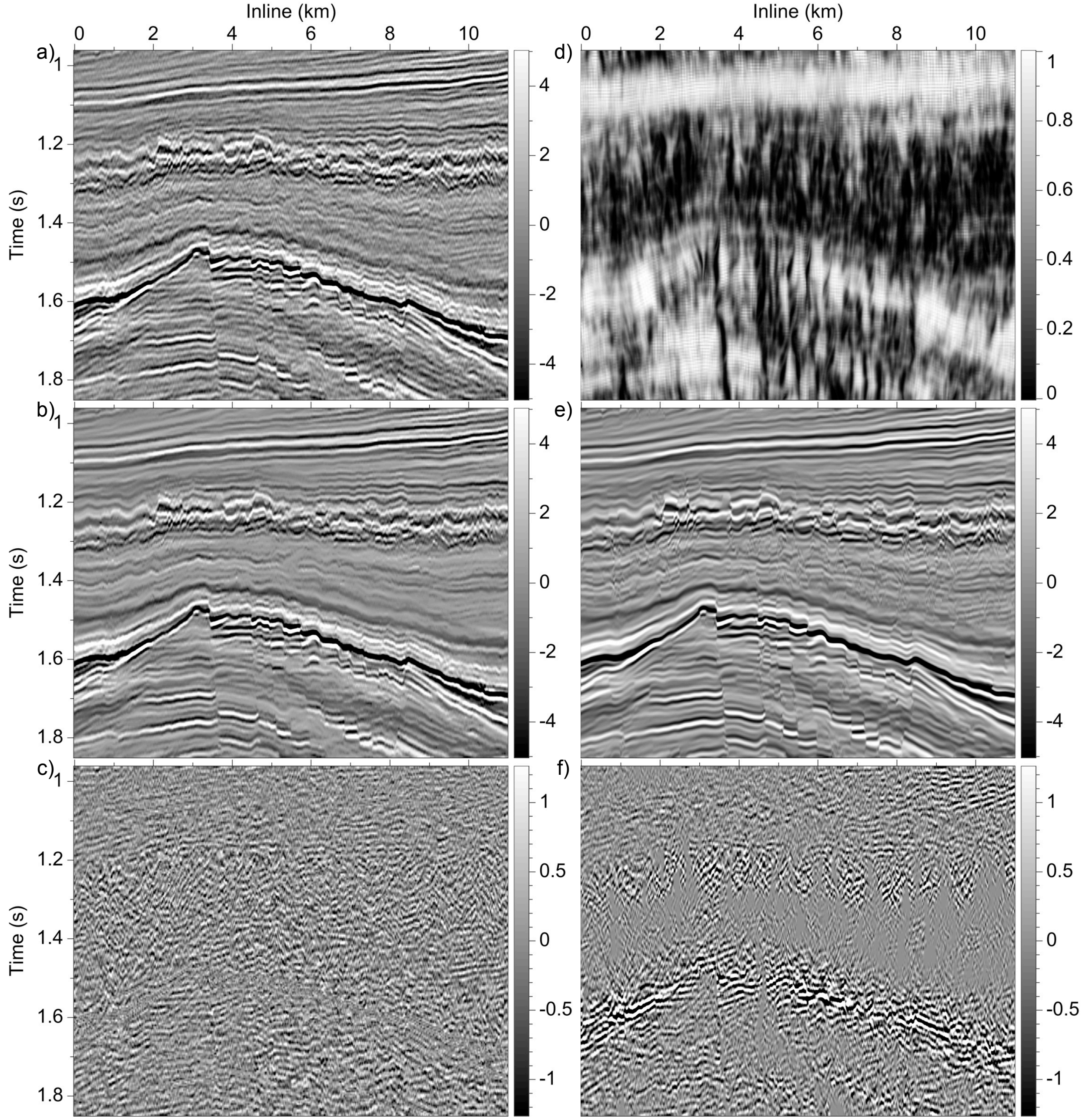

Figure 10. The input p (a), with output q (b) and input-output difference $p - q$ (c) for the bilateral filter. For comparison, coherence c (d) is used in the edge-preserving filter, with output (e) and input-output difference (f).

REFERENCES

Barash, D., 2002, A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation: IEEE Transactions on Pattern Analysis and Machine Intelligence, **24**, 1–5.

Durand, F., and J. Dorsey, 2002, Fast bilateral filtering for the display of high-dynamic-range images: ACM Transactions on Graphics, **21**, 257–266.

Fehmers, G., and C. Höcker, 2003, Fast structural interpretation with structure-oriented filtering: Geophysics, **68**, 1286–1293.

Hale, D., 2009, Structure-oriented smoothing and semblance: CWP Report 635.

Paris, S., P. Kornprobst, J. Tumblin, and F. Durand, 2008, A gentle introduction to bilateral filtering and its applications: ACM SIGGRAPH 2008 classes,

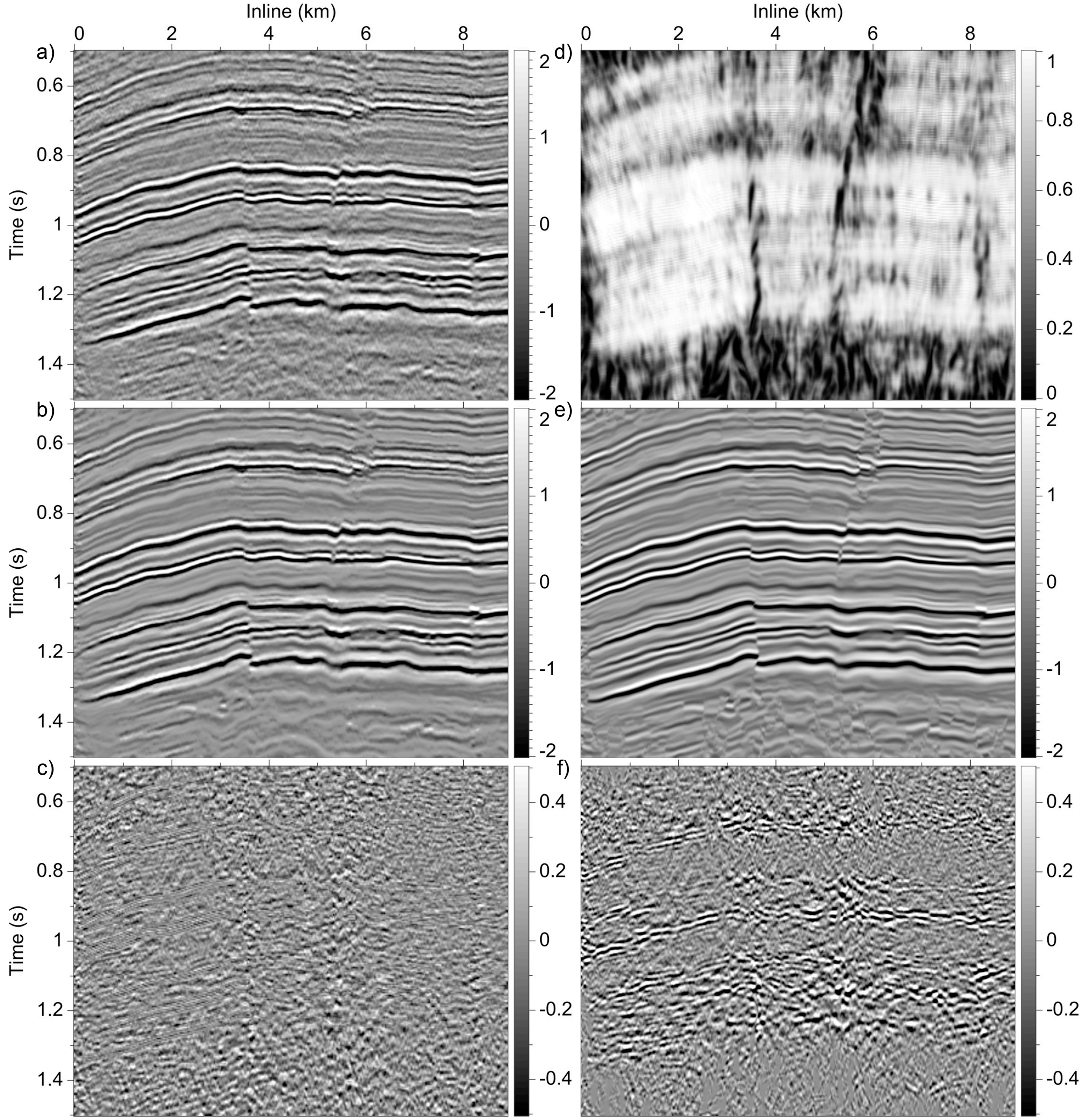

Figure 11. The input p (a), with output q (b) and input-output difference $p - q$ (c) for the bilateral filter. For comparison, coherence c (d) is used in the edge-preserving filter, with output (e) and input-output difference (f).

ACM, 1–50.

Tomasi, C., and R. Manduchi, 1998, Bilateral filtering for gray and color images: Proceedings of the Sixth International Conference on Computer Vision (ICCV 98), IEEE Computer Society, 839.

Weickert, J., 1999, Coherence-enhancing diffusion filtering: International Journal of Computer Vision, **31**, 111–127.

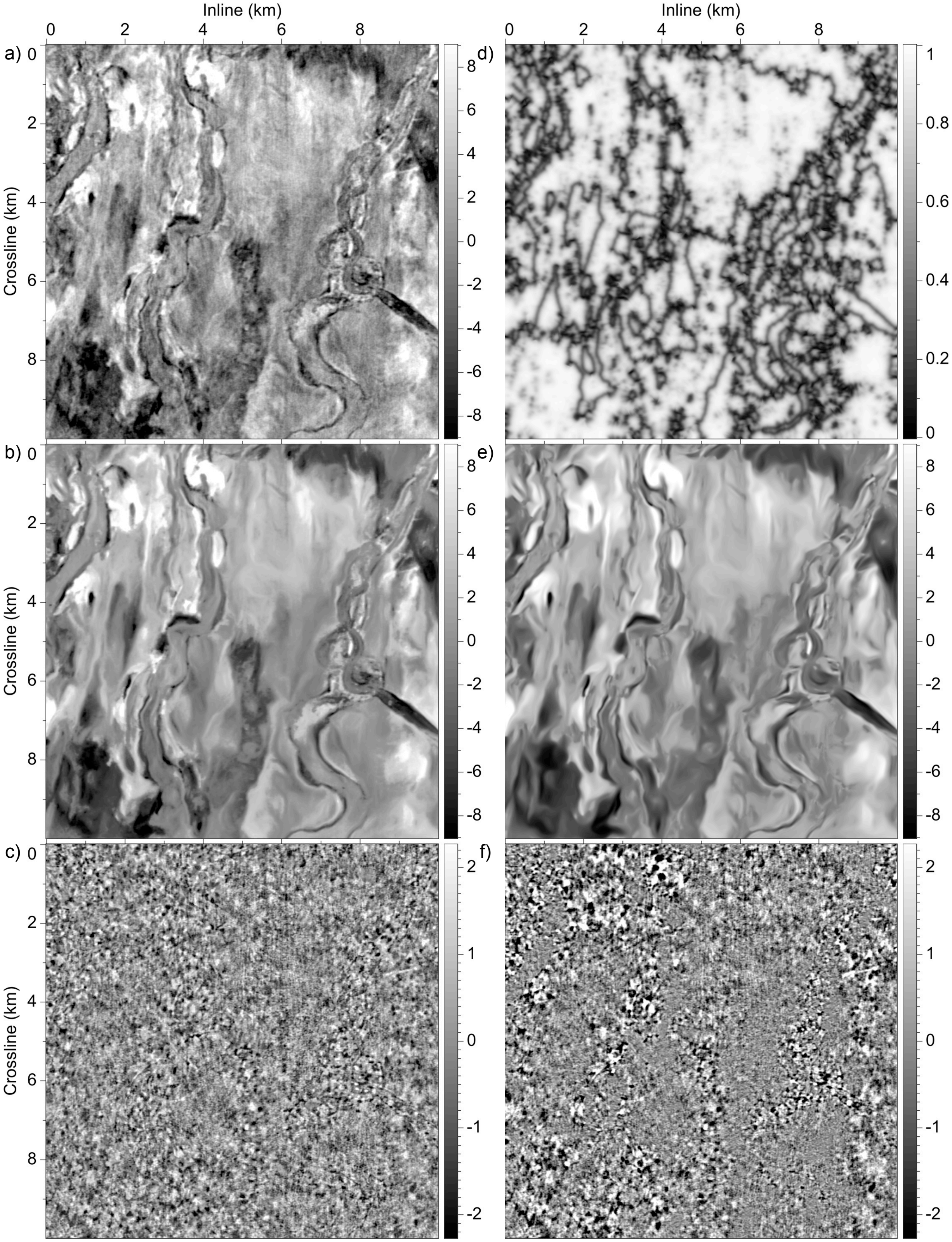

Figure 12. The input p (a), with output q (b) and input-output difference $p - q$ (c) for the bilateral filter. For comparison, coherence c (d) is used in the edge-preserving filter, with output (e) and input-output difference (f).