



11 July 2003

EDGE-PRESERVING ENHANCEMENT OF SEISMIC IMAGES BY NONLINEAR ANISOTROPIC
DIFFUSION

The present invention relates to a method of
processing an image, for example a seismic image,
applying a diffusion process. Such methods are known, see
for example the thesis 'Anisotropic diffusion in image
processing' by J Weickert, January 1996, pages 41-44.

Known is a diffusion method that comprises the steps
of

(a) obtaining an n-dimensional initial image data set,
wherein each element $u(m=0)$ of the data set is the
initial image intensity of a point of the image, and
wherein $n=2$ or $n=3$;

(b) calculating for each point the partial derivatives of
 $u(m)$ in n directions to obtain n derivative data sets of
the partial derivatives u_{xi} ;

(c) calculating for each point a symmetric $n \times n$ structural
matrix S , wherein the elements s_{pq} equal $u_{xp} \cdot u_{xq}$;

(d) calculating for each point an $n \times n$ diffusion matrix D ,
wherein the elements d_{ij} are a function of the
elements s_{pq} ;

(e) calculating for each point $u(m+1)$ from $u(m)$ using the
following equation $u(m+1)=u(m)+\text{div}(D\text{grad}(u(m)))$; and

(f) repeating steps (b) through (e) M times to obtain the
processed image.

Unless otherwise specified, the following holds
throughout the specification and the claims:

$$i, j, p, q = 1, \dots, n;$$

$$u_{xi} = \partial u / \partial x_i.$$

An element of the image data set is also referred to
as a point of the image.

In the known method an initial image is processed to
obtain successive images, wherein each image is obtained



from the previous one by a diffusion process. The diffusion method can as well be described by the diffusion equation $\frac{\partial u}{\partial t} = \text{div}(D \cdot \nabla u)$.

5 There are four types of diffusion methods. At first there is the isotropic diffusion in which the diffusivity D is a scalar and the anisotropic diffusion in which D is a matrix. Then for each of these two types, the diffusion can be linear, in which D is not a function of u or the diffusion can be non linear in which D is a
10 function of u .

This method can as well be applied to processing a seismic image, in that case the image intensity is the amplitude of the seismic signal.

15 An advantage of the diffusion method is that noise is suppressed. A further advantage, which is particularly relevant to processing a seismic image is that after a few cycles the geological structure is highlighted more clearly. However, a problem with this method is preserving edges.

20 Edge preservation is of great interest in the interpretation of seismic images, which play an important role in the study of underground formations, and in particular in the study of underground formations that contain hydrocarbons. Moreover there is a great interest
25 in edge preserving techniques, which allow highlighting of faults while removing noise from the seismic data.

For edge preservation in combination with isotropic diffusion processing is proposed a method based on the Perona-Malik model (see the thesis of Weickert, page 10)
30 of which the diffusion equation is $\frac{\partial u}{\partial t} = \text{div}(g(|\nabla u|^2) \nabla u)$, wherein the non-linear diffusivity is

$g(|\nabla u|^2) = \left(1 + |\nabla u|^2 / \lambda^2\right)^{-1}$, the scalar λ being a pre-determined edge preservation parameter.

Applicant proposes a simple addition to the known method that has proved to be an effective way of preserving edges while capable of removing noise. Moreover Applicant proposes such a method that is applicable to anisotropic diffusion.

To this end the method of processing an image according to the present invention comprises the steps of (a) obtaining an n-dimensional initial image data set, wherein each element $u(m=0)$ of the data set is the initial image intensity of a point of the image, and wherein $n=2$ or $n=3$;

(b) calculating for each point the partial derivatives of $u(m)$ in n directions to obtain n derivative data sets of the partial derivatives u_{xi} ;

(c) calculating for each point a symmetric $n \times n$ structural matrix S , wherein the elements s_{pq} equal $u_{xp} \cdot u_{xq}$;

(d) calculating for each point an $n \times n$ diffusion matrix D , wherein the elements d_{ij} are a function of the elements s_{pq} ;

(e) calculating for each point $u(m+1)$ from $u(m)$ using the following equation $u(m+1) = u(m) + c \cdot \text{div}(\epsilon \cdot D \cdot \text{grad}(u(m)))$, wherein c is a predetermined constant, $0 \leq c \leq 1$, and wherein ϵ is a scalar, $0 \leq \epsilon \leq 1$, wherein ϵ is close to zero when near an edge and close to 1 when far away from an edge; and

(f) repeating steps (b) through (e) M times to obtain the processed image.

In this way edge preservation is introduced in anisotropic, non-linear diffusion.

Suitably step (c) comprises calculating for each point a symmetric $n \times n$ structural precursor matrix S^0 , wherein the elements s^0_{pq} equal $u_{xp} \cdot u_{xq}$; creating

(1/2)n(n+1) data sets, wherein the elements of the first data set are s^0_{11} pertaining to each point, the elements of the second data set are s^0_{12} and so on; and filtering each of these data sets by convolution with a suitable kernel to obtain the elements s_{pq} of the nxn structural matrix S.

Alternatively step (c) comprises filtering for each point the partial derivatives u_{xi} by convolution with a suitable kernel to obtain regularized partial derivatives u^r_{xi} ; calculating for each point a symmetric nxn structural precursor matrix S^0 , wherein the elements s^0_{pq} equal $u^r_{xp} \cdot u^r_{xq}$; creating (1/2)n(n+1) data sets, wherein the elements of the first data set are s^0_{11} pertaining to each point, the elements of the second data set are s^0_{12} and so on; and filtering each of these data sets by convolution with a suitable kernel to obtain the elements s_{pq} of the nxn structural matrix S.

The above described alternatives for step (c), wherein a precursor matrix is calculated allows defining the edge preservation parameter as follows:
 $\epsilon = \text{trace}(S^0 S) / (\text{trace}(S^0) \cdot \text{trace}(S))$. In this way the edge preservation parameter is calculated for each point.

The present invention also relates to a method of processing an image in order to display the edge preservation parameter. This method comprises the steps of

- (a) obtaining an n-dimensional initial image data set, wherein each element $u(m=0)$ of the data set is the initial image intensity of a point of the image, and wherein $n=2$ or $n=3$;
- (b) calculating for each point the partial derivatives of $u(m)$ in n directions to obtain n derivative data sets of the partial derivatives u_{xi} ;
- (c) calculating for each point a symmetric nxn structural matrix S^0 , wherein the elements s^0_{pq} equal $u_{xp} \cdot u_{xq}$;

11 July 2003

creating $(1/2)n(n+1)$ data sets, wherein the elements of the first data set are s^0_{11} pertaining to each point, the elements of the second data set are s^0_{12} and so on; and filtering each of these data sets by convolution with a suitable kernel to obtain the elements s_{pq} of the $n \times n$ structural matrix S ; and

(d) calculating for each point $\varepsilon = \text{trace}(S^0 S) / (\text{trace}(S^0) \cdot \text{trace}(S))$ and attributing the calculated value of ε to each point to obtain an image in which faults are highlighted.

Alternatively, the method of processing an image comprises the steps of

(a) obtaining an n -dimensional initial image data set, wherein each element $u(m=0)$ of the data set is the initial image intensity of a point of the image, and wherein $n=2$ or $n=3$;

(b) calculating for each point the partial derivatives of $u(m)$ in n directions to obtain n derivative data sets of the partial derivatives u_{xi} ;

(c) filtering for each point the partial derivatives u_{xi} by convolution with a suitable kernel to obtain regularized partial derivatives u^r_{xi} ; calculating for each point a symmetric $n \times n$ structural precursor matrix S^0 , wherein the elements s^0_{pq} equal $u^r_{xp} \cdot u^r_{xq}$;

creating $(1/2)n(n+1)$ data sets, wherein the elements of the first data set are s^0_{11} pertaining to each point, the elements of the second data set are s^0_{12} and so on; and filtering each of these data sets by convolution with a suitable kernel to obtain the elements s_{pq} of the $n \times n$ structural matrix S ;

(d) calculating for each point $\varepsilon = \text{trace}(S^0 S) / (\text{trace}(S^0) \cdot \text{trace}(S))$ and attributing the calculated value of ε to each point to obtain an image in which faults are highlighted.

In the paper 'Coherence-Enhancing Diffusion Filtering' by J Weickert, International Journal of Computer Vision vol 31(2/3), pages 111-127, (1999), modifications of the method known from the above-mentioned thesis are discussed. The paper discloses a method for enhancing coherence in images with flow-like structures. This method is not a method for preserving edges, and no parameter corresponding to the parameter ϵ of the present invention is disclosed.

The present invention will now be described by way of example in more details. At first the equations for a two-dimensional image are discussed, then for a three dimensional image and thereafter alternative embodiments.

In two dimensions, the initial image data set comprises $k_1 \times k_2$ elements (or points) $u(m=0)$, wherein the value of each element u is the initial image intensity of a point of the image, and wherein k_1 and k_2 are the number of points in the two directions x_1 and x_2 of the two-dimensional image. The word pixel is used to refer to such a point in two dimensions, and in three dimensions, the point is called a voxel.

The next step comprises calculating the partial derivatives of all points $u(m)$ in n directions to obtain n derivative data sets of the partial derivatives u_{x_i} . The first data set comprises $k_1 \times k_2$ values of u_{x_1} ($=\partial u / \partial x_1$) and the second data set comprises $k_1 \times k_2$ values of u_{x_2} ($=\partial u / \partial x_2$).

Then for each point a symmetric $n \times n$ structural matrix S is calculated, wherein the elements s_{pq} equal $u_{xp} \cdot u_{xq}$. The structural matrix S is in two dimensions

$$\begin{pmatrix} u_{x1} \cdot u_{x1} & u_{x1} \cdot u_{x2} \\ u_{x2} \cdot u_{x1} & u_{x2} \cdot u_{x2} \end{pmatrix}.$$

Then for each point an $n \times n$ diffusion matrix D is calculated, wherein the elements d_{ij} are a function of

the elements s_{pq} , which elements s_{pq} are a function of the partial derivatives u_{xp} and u_{xq} .

Then the image comprising the elements $u(m=0)$ is processed in M steps to obtain a processed image comprising elements $u(m=M)$, wherein after each step the integer m is increased with 1. The $k_1 \times k_2$ values $u(m+1)$ of each image are obtained from the values $u(m)$ from a previous image by solving a diffusion type equation:

$u(m+1) = u(m) + c \cdot \text{div}(\varepsilon \cdot \text{Dgrad}(u(m)))$. The number of times the image is processed, M , is a predetermined value. In order to remove noise, M is suitably in the range of from 2 to 4, and when in addition thereto the image has to be simplified, M is suitably in the range of from 5 to 20.

In this equation, c is a predetermined constant, $0 \leq c \leq 1$, and ε is a scalar, $0 \leq \varepsilon \leq 1$, wherein ε is close to zero when near an edge and close to 1 when far away from an edge. The scalar ε can be understood to be a fault highlighter.

In two dimensions,

$$\text{div}(\varepsilon \cdot \text{Dgrad}(u)) = \sum_{i=1}^2 \frac{\partial}{\partial x_i} (\varepsilon (d_{i1} u_{x1} + d_{i2} u_{x2})).$$

In three dimensions, the image data set comprises $k_1 \times k_2 \times k_3$ image intensities $u(m)$. The first derivative data set comprises $k_1 \times k_2 \times k_3$ values of u_{x1} ($= \partial u / \partial x_1$), the second derivative data set comprises $k_1 \times k_2 \times k_3$ values of u_{x2} ($= \partial u / \partial x_2$), and the third derivative data set comprises $k_1 \times k_2 \times k_3$ values of u_{x3} ($= \partial u / \partial x_3$). The $n \times n$

$$\text{structural matrix } S \text{ is } \begin{pmatrix} u_{x1} \cdot u_{x1} & u_{x1} \cdot u_{x2} & u_{x1} \cdot u_{x3} \\ u_{x2} \cdot u_{x1} & u_{x2} \cdot u_{x2} & u_{x2} \cdot u_{x3} \\ u_{x3} \cdot u_{x1} & u_{x3} \cdot u_{x2} & u_{x3} \cdot u_{x3} \end{pmatrix}.$$

In three dimensions,

$$\text{div}(\varepsilon \cdot \text{Dgrad}(u)) = \sum_{i=1}^3 \frac{\partial}{\partial x_i} (\varepsilon (d_{i1} u_{x1} + d_{i2} u_{x2} + d_{i3} u_{x3})).$$

In the above, the diffusion matrix is not calculated directly from the image intensities. However, suitably the diffusion matrix is calculated from filtered data. Therefore the step of calculating the structural matrix
5 suitably comprises calculating for each point a symmetric nxn structural precursor matrix S^0 , wherein the elements s_{pq}^0 equal $u_{xp} \cdot u_{xq}$; creating $(1/2)n(n+1)$ data sets, wherein the elements of the first data set are s_{11}^0 pertaining to each point, the elements of the second data set are s_{12}^0 and so on; and filtering each of these data
10 sets by convolution with a suitable kernel to obtain the elements s_{pq} of the nxn structural matrix S .

In an alternative embodiment, the nxn structural matrix S is calculated from filtered partial derivatives.
15 To this end, the partial derivatives u_{xi} are filtered for each point by convolution with a suitable kernel to obtain regularized partial derivatives u_{xi}^r . For each point a symmetric nxn structural precursor matrix S^0 is calculated, wherein the elements s_{pq}^0 equal $u_{xp}^r \cdot u_{xq}^r$.
20 Then the $(1/2)n(n+1)$ data sets are created, wherein the elements of the first data set are s_{11}^0 pertaining to each point, the elements of the second data set are s_{12}^0 and so on. And these data sets are filtered by convolution with a suitable kernel to obtain the
25 elements s_{pq} of the nxn structural matrix S .

Suitably the edge preservation parameter ϵ is derived from the image data set by calculations. Applicant had found that the structural precursor matrix S^0 can be used to estimate the scalar ϵ , suitably

30 $\epsilon = \text{trace}(S^0 S) / (\text{trace}(S^0) \cdot \text{trace}(S))$, wherein $\text{trace}(A)$ is the sum of the diagonal elements a_{kk} of the matrix A . In this way ϵ becomes a function of the partial derivatives of $u(m)$ in the two or three directions. And the scalar ϵ is a fault highlighter that is calculated for each point.

The scalar is about 1 in the absence of a fault and much smaller than 1 in the presence of a fault.

Suitably step (d) comprises determining the n eigenvalues λ_i and n eigenvectors v_i of each of the structural matrices S ; sorting the eigenvalues so that $\lambda_1 \geq \lambda_2 (\geq \lambda_3)$ and calculating for each point an $n \times n$ diffusion matrix D , wherein the elements d_{pq} equal $v_{2p} \cdot v_{2q}$ (wherein v_2 is the eigenvector pertaining to the smallest eigenvalue if $n=2$) or wherein the elements d_{pq} equal $v_{2p} \cdot v_{2q} + v_{3p} \cdot v_{3q}$ (wherein v_2 and v_3 are the eigenvectors pertaining to the smallest eigenvalues if $n=3$). The symbols v_{ip} and v_{iq} denote the p -th and q -th element of the eigenvector v_i . As a result the eigenvector pertaining to the largest eigenvalue does not contribute to the diffusion matrix, and the eigenvector(s) pertaining to the smaller eigenvalue(s) do contribute. This inhibits diffusion of the image luminance in the direction of the eigenvector pertaining to the largest eigenvalue, which latter eigenvector is directed perpendicular to the reflection.

The suitable kernel is a kernel for low pass filtering, such a kernel is symmetric and positive everywhere. A suitable kernel is the Gaussian kernel of width $\sigma > 0$, which is given by the following equation: The

Gaussian kernel is used as a $K_\sigma(x) = \frac{1}{(2\pi\sigma^2)^{n/2}} \cdot \exp\left(\frac{-\|x\|^2}{2\sigma^2}\right)$

convolution mask, wherein x is the position of the centre of the convolution mask and $\|\cdot\|$ is the Euclidean norm, and n is the dimension.

The invention will now be described by way of example with reference to the accompanying drawings, wherein

Figure 1 shows the original image;

Figure 2 shows the image treated according to the present invention with three anisotropic diffusion steps;

Figure 3 shows the image treated according to the present invention with ten anisotropic diffusion steps;

Figure 4 shows the image treated not according to the present invention with three anisotropic diffusion steps with no edge preservation, $\epsilon=1$;

Figure 5 shows the image treated not according to the present invention with ten anisotropic diffusion steps with no edge preservation, $\epsilon=1$;

Figure 6 shows the image treated not according to the invention with isotropic diffusion without edge preservation;

Figure 7 shows the image treated not according to the invention with isotropic diffusion with edge preservation according to Perona-Malik with $\lambda=10$; and

Figure 8 shows the image of Figure 1 treated so as to highlight the faults.

Reference is made to Figure 1, which shows the original image. The image is 128 pixels by 128 pixels having a value in the range from 0 to 255. The picture shows reflectors 1, 2, 3, 4 and 5 and faults 10, 11, 12, 14 and 15. For the sake of clarity, not all reflectors and faults have been referred to with a reference numeral.

Figures 2 and 3 show the original image treated according to the present invention. The structural tensor had been calculated as follows, at first for each point the partial derivatives u_{xi} were filtered by convolution with a Gaussian kernel to obtain regularized partial derivatives u^r_{xi} ; then for each point a symmetric $n \times n$ structural precursor matrix S^0 was calculated, wherein the elements s^0_{pq} equal $u^r_{xp} \cdot u^r_{xq}$; creating $(1/2)n(n+1)$ data sets, wherein the elements of the first data set are s^0_{11} pertaining to each point, the elements of the second data set are s^0_{12} and so on. To obtain the elements s_{pq}

of the $n \times n$ structural matrix S each of these data sets is filtered by convolution with a Gaussian kernel.

In order to get Figure 2, three anisotropic diffusion steps were applied. The edges are made clearer whilst noise is reduced.

Figure 3 shows a smoother picture, with less noise, obtained with the method according to the present invention after ten anisotropic diffusion steps.

In order to show the improvement obtained with the method according to the present invention, reference is made to Figures 4-7, which have been obtained with known methods. Figure 4 shows the image treated not according to the present invention with three anisotropic diffusion steps with no edge preservation, in which $\epsilon=1$.

Figure 5 shows the image treated not according to the present invention with ten anisotropic diffusion steps with no edge preservation, $\epsilon=1$.

Figure 6 shows the image treated not according to the invention with isotropic diffusion without edge preservation; and Figure 7 shows the image treated not according to the invention with isotropic diffusion with edge preservation according to Perona-Malik with $\lambda=10$.

Figure 8 shows an image obtained from Figure 1 processed to highlight the faults. The method of processing the image comprised calculating for each point of the original image (Figure 1) the partial derivatives of $u(m)$ in n directions to obtain n derivative data sets of the partial derivatives u_{xi} ; filtering for each point the partial derivatives u_{xi} by convolution with a suitable kernel to obtain regularized partial derivatives u^r_{xi} ; calculating for each point a symmetric $n \times n$ structural precursor matrix S^0 , wherein the elements s^0_{pq} equal $u^r_{xp} \cdot u^r_{xq}$; creating $(1/2)n(n+1)$ data sets, wherein the elements of the first data set are s^0_{11} pertaining to each point, the elements of the second data

11 July 2003

- 12 -

set are s_{012} and so on; and filtering each of these data sets by convolution with a suitable kernel to obtain the elements s_{pq} of the $n \times n$ structural matrix S . Then for each point $\epsilon = \text{trace}(S^0 S) / (\text{trace}(S^0) \cdot \text{trace}(S))$ was calculated, and the value of ϵ was attributed to each point to obtain Figure 8 in which faults are highlighted in black.

5

11 July 2003

C L A I M S

1. Method of processing an image comprising the steps of
(a) obtaining an n-dimensional initial image data set,
wherein each element $u(m=0)$ of the data set is the
initial image intensity of a point of the image, and
5 wherein $n=2$ or $n=3$;

(b) calculating for each point the partial derivatives of
 $u(m)$ in n directions to obtain n derivative data sets of
the partial derivatives u_{xi} ;

10 (c) calculating for each point a symmetric nxn structural
matrix S , wherein the elements s_{pq} equal $u_{xp} \cdot u_{xq}$;

(d) calculating for each point an nxn diffusion matrix D ,
wherein the elements d_{ij} are a function of the
elements s_{pq} ;

15 (e) calculating for each point $u(m+1)$ from $u(m)$ using the
following equation $u(m+1)=u(m)+c.\text{div}(\varepsilon.D\text{grad}(u(m)))$,
wherein c is a predetermined constant, $0 \leq c \leq 1$, and wherein
 ε is a scalar, $0 \leq \varepsilon \leq 1$, wherein ε is close to zero when
near an edge and close to 1 when far away from an edge;
and

20 (f) repeating steps (b) through (e) M times to obtain the
processed image.

2. The method according to claim 1, wherein step (c)
comprises calculating for each point a symmetric nxn
structural precursor matrix S^0 , wherein the elements s^0_{pq}
25 equal $u_{xp} \cdot u_{xq}$; creating $(1/2)n(n+1)$ data sets, wherein
the elements of the first data set are s^0_{11} pertaining to
each point, the elements of the second data set are s^0_{12}
and so on; and filtering each of these data sets by
convolution with a suitable kernel to obtain the
30 elements s_{pq} of the nxn structural matrix S .

3. The method according to claim 1, wherein step (c) comprises filtering for each point the partial derivatives u_{xi} by convolution with a suitable kernel to obtain regularized partial derivatives u^r_{xi} ; calculating for each point a symmetric $n \times n$ structural precursor matrix S^0 , wherein the elements s^0_{pq} equal $u^r_{xp} \cdot u^r_{xq}$; creating $(1/2)n(n+1)$ data sets, wherein the elements of the first data set are s^0_{11} pertaining to each point, the elements of the second data set are s^0_{12} and so on; and filtering each of these data sets by convolution with a suitable kernel to obtain the elements s_{pq} of the $n \times n$ structural matrix S .

4. The method according to claim 2 or 3, wherein

$$\epsilon = \text{trace}(S^0 S) / (\text{trace}(S^0) \cdot \text{trace}(S)).$$

5. The method according any one of the claims 1-4, wherein step (d) comprises determining the n eigenvalues λ_i and n eigenvectors v_i of each of the $n \times n$ structural matrices S ; sorting the eigenvalues so that $\lambda_1 \geq \lambda_2 (\geq \lambda_3)$ and calculating for each point an $n \times n$ diffusion matrix D , wherein the elements d_{pq} equal $v_{2p} \cdot v_{2q}$ (if $n=2$) or wherein the elements d_{pq} equal $v_{2p} \cdot v_{2q} + v_{3p} \cdot v_{3q}$ (if $n=3$).

6. A method of processing an image comprising the steps of

(a) obtaining an n -dimensional initial image data set, wherein each element $u(m=0)$ of the data set is the initial image intensity of a point of the image, and wherein $n=2$ or $n=3$;

(b) calculating for each point the partial derivatives of $u(m)$ in n directions to obtain n derivative data sets of the partial derivatives u_{xi} ;

(c) calculating for each point a symmetric $n \times n$ structural matrix S^0 , wherein the elements s^0_{pq} equal $u_p \cdot u_q$; creating $(1/2)n(n+1)$ data sets, wherein the elements of the first data set are s^0_{11} pertaining to each point, the elements of the second data set are s^0_{12} and so on; and

filtering each of these data sets by convolution with a suitable kernel to obtain the elements s_{pq} of the $n \times n$ structural matrix S ; and

(d) calculating for each point $\varepsilon = \text{trace}(S^0 S) /$

($\text{trace}(S^0) \cdot \text{trace}(S)$) and attributing the calculated value of ε to each point to obtain an image in which faults are highlighted.

7. A method of processing an image comprising the steps of

(a) obtaining an n -dimensional initial image data set, wherein each element $u(m=0)$ of the data set is the initial image intensity of a point of the image, and wherein $n=2$ or $n=3$;

(b) calculating for each point the partial derivatives of $u(m)$ in n directions to obtain n derivative data sets of the partial derivatives u_{xi} ;

(c) filtering for each point the partial derivatives u_{xi} by convolution with a suitable kernel to obtain regularized partial derivatives u^r_{xi} ; calculating for each point a symmetric $n \times n$ structural precursor matrix S^0 , wherein the elements s^0_{pq} equal $u^r_{xp} \cdot u^r_{xq}$; creating $(1/2)n(n+1)$ data sets, wherein the elements of the first data set are s^0_{11} pertaining to each point, the elements of the second data set are s^0_{12} and so on; and filtering each of these data sets by convolution with a suitable kernel to obtain the elements s_{pq} of the $n \times n$ structural matrix S ; and

(d) calculating for each point $\varepsilon = \text{trace}(S^0 S) /$
 $(\text{trace}(S^0) \cdot \text{trace}(S))$ and attributing the calculated value of ε to each point to obtain an image in which faults are highlighted.