Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 25 (2004) 603–618
Pattern Recognition Letters
www.elsevier.com/locate/patrec

A fast multi-scale edge detection algorithm

Benoit Tremblais *, Bertrand Augereau

IRCOM-SIC, UMR-CNRS 6615, Bvd Marie et Pierre Curie, BP 30179, F-86962 Futuroscope Chasseneuil Cedex, France

Received 8 July 2003; received in revised form 18 November 2003

Abstract

In this paper we present a new explicit numerical scheme to approximate the solution of the linear diffusion filtering. This scheme is fast, stable, easy to program, applicable to arbitrary dimensions, and preserves the discontinuities of the objects. Experimental results support the efficiency of the proposed approach for the multi-scale detection of edges in greyscale, and color images.

Keywords: Computer vision and image understanding; Adaptative filtering; Diffusion; PDE; Edge detection; Multi-scale analysis

1. Introduction

One of the main tasks in vision systems is to analyze unknown scenes that are composed of real-world objects which appearance is related to the scale of the observation. That is why it is considered that a scene should be described at multiple scales. Witkin (1983) and Koenderink (1984) were the first to formalize this concept with the idea of scale-space filtering. The principle is to convolve the original image by a family of Gaussians of increasing variance related to the studied scale, and then to progressively eliminate the smallest structures in the image. This kind of filtering is linear, and consequently a wide class of efficient resolution techniques have been proposed in (Alvarez, 1996; Zhao and Li, 1996; Deriche, 1990; Nielsen et al., 1997; Lindeberg, 1990). However, this approach suffers from several drawbacks: the edges are blurred, the locations of edges at the coarse scale are shifted from their true locations, and edge junctions are destroyed. It is so necessary to track edges across the scale-space which is a complicated and expensive task (Clark, 1998; Lindeberg, 1999). Multiple nonlinear diffusion filterings have been suggested to overcome these drawbacks. The main contributions on this subject can be found in (Perona and Malik, 1990; Osher and Rudin, 1990; Catte et al., 1992; Alvarez and Mazorra, 1994; Alvarez and Morel, 1994; Romeny, 1994; Whitaker and Gerig, 1994; Malladi and Sethian, 1996; Weickert, 1998; Sochen et al., 1998; Aubert and Kornprobst, 2002; Tschumperle, 2002). These approaches give impressive results but generally suffer from a poor computational efficiency due to the two-level explicit finite difference scheme which is only stable for small time steps. More elaborated

* Corresponding author. Tel.: +33549496599; fax: +33549496570.
E-mail address: tremblai@sic.sp2mi.univ-poitiers.fr (B. Tremblais).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2003.12.016

approaches have been suggested to accelerate the resolution, among them semi-implicit approaches (Catte et al., 1992; Weic- kert et al., 1998), three-level, wavelets based and pseudospectral ones (Fröhlich and Weickert, 1994), finite element approaches with adaptative mesh coarsening (Baensch and Mikula, 1997), multigrid methods (Acton et al., 1994), fast level-set methods (Sethian, 1996), high-order ENO (Siddiqi et al., 1997), implicit schemes for mean curvature motion (Alvarez, 1996; Cohignac et al., 1993; Marquina and Osher, 1999). But most of these approaches need to compute matrix inversions which is time expensive and a source of computational errors.

The purpose of the present paper is to propose a fast and easy to program technique for the multi-scale detection of the edges. We suggest to apply a new explicit scheme to the linear diffusion filtering which preserves edges. We get then a fast filtering algorithm that we combine with a simple multi-scale edge detection algorithm.

This paper is organized as follows: Section 2 establishes that the resolution of the two conventional limits' problems linear diffusion and non-linear diffusion filtering, using an explicit scheme, are fixed point searches. In Section 3 we suggest a new edge preserving scheme and show its good properties concerning the rotational invariance, and its computational efficiency. In Section 4 we present a new multi-scale edge detection algorithm adapted from an algorithm presented in (Tremblais et al., 2003) for the detection of bottom-lines in 2D X-ray angiograms. In Section 5 some experimental results are presented, as well as some comparison with other diffusion filtering schemes. Finally, we conclude in Section 6.

2. The explicit PDE resolution scheme as a fixed point search

2.1. The continuous case

Generally speaking, a diffusion filtering can be looked upon as a process applied to objects where objects may be a one-dimensional signal, a grey-level image, a color image, a volume, a film, etc. Let $\Omega \subset \mathbb{R}^N$ be the object domain. The object is modelized as a bounding mapping from Ω to $\mathbb{R}^P$, that is a N-dimensional differential manifold embedded in an Euclean space isomorph to $\mathbb{R}^{N+P}$. We denote such an object by the mapping $\boldsymbol{f} : \Omega \to \mathbb{R}^P$ where the image of $\boldsymbol{x} = (x^i)_{i=1,\ldots,N}$ is denoted $\boldsymbol{f}\ (\boldsymbol{x}) = (f^j(\boldsymbol{x}))_{j=1,\ldots,P}$. The process computes a filtered object $\boldsymbol{I}(\boldsymbol{x},t)$ where t is a real positive variable usually called the *scale* or *time parameter*. Increasing t leads to simpler object representations. The whole embedding of the original object into such a one-parameter family of processed (or simplified) objects produces a $(N+P+1)$-dimensional object called *scale-space*. Consequently, the diffusion filtering is a mapping from $\Omega \times \mathbb{R}^+$ to $\mathbb{R}^P$ which associates $\boldsymbol{I}(\boldsymbol{x},t) = (I^j(\boldsymbol{x},t))_{j=1,\ldots,P}$ to $(\boldsymbol{x},t) = ((x^i)_{i=1,\ldots,N},t)$. Such a process can be described as the limits' problem:

$$\begin{cases} \dfrac{\partial \boldsymbol{I}(\boldsymbol{x},t)}{\partial t} = G(\boldsymbol{I},\boldsymbol{x},t) & \forall \boldsymbol{x} \in \Omega,\ \forall t \in \mathbb{R}^+ \\ \boldsymbol{I}(\boldsymbol{x},0) = \boldsymbol{f}\ (\boldsymbol{x}) & \forall \boldsymbol{x} \in \Omega \\ \dfrac{\partial \boldsymbol{I}(\boldsymbol{x},t)}{\partial \boldsymbol{n}} = 0 & \forall \boldsymbol{x} \in \partial\Omega \end{cases} \tag{1}$$

where $G(\boldsymbol{I},\boldsymbol{x},t)$ and $\boldsymbol{n}$ respectively denote a combination of $\boldsymbol{I}$ and its spatial partial differentials, and the normal to the object boundary $\partial\Omega$. In the following, we will focus on the two conventional diffusion filtering related to the Partial Differential Equations (PDEs):

$$\frac{\partial \boldsymbol{I}(\boldsymbol{x},t)}{\partial t} = \sum_{i=1}^{N} \frac{\partial^2 \boldsymbol{I}(\boldsymbol{x},t)}{\partial x^{i2}} \tag{2}$$

$$\frac{\partial \boldsymbol{I}(\boldsymbol{x},t)}{\partial t} = \sum_{i=1}^{N} \frac{\partial}{\partial x^i}\left(\Psi_i(\boldsymbol{x},t)\frac{\partial \boldsymbol{I}(\boldsymbol{x},t)}{\partial x^i}\right) \tag{3}$$

where the Ψ_i are positive modulation functions. The diffusion filtering related to Eq. (2) namely the *linear diffusion filtering* is equivalent to a convolution of the object by a Gaussian and consequently do not preserve the edges. In order to reduce smoothing at edges, the diffusion filtering associated to the PDE (3) called *the nonlinear diffusion filtering* (or *anisotropic diffusion filtering*) have been introduced by Perona and Malik (1990) with $\forall i, \Psi_i(\boldsymbol{x},t) = c(\|\nabla \boldsymbol{I}\|, K)$, where K plays the role of a contrast parameter that forces the process to prefer intraregional smoothing to interregional blurring. A spatial regularization of this filter have been firstly

suggested by Catte et al. (1992) using $\|\nabla \boldsymbol{I}_\sigma\|$ instead of $\|\nabla \boldsymbol{I}\|$. Here, $\nabla \boldsymbol{I}_\sigma$ is the gradient of a smoothed version of $\boldsymbol{I}$ that is obtained by convolving $\boldsymbol{I}$ with a Gaussian of standard deviation σ. This parameter makes the filter insensitive to noise at scale smaller than σ and can be assumed as a regularization parameter which ensures well-posedness of the scale-space generated with this process.

2.2. The explicit scheme

The influence of the dimension P of the image space is limited to the existence of a set of P one-dimensional replies of (1) applied to the $I^j(\boldsymbol{x},t)\ \forall j \in \{1,\ldots,P\}$. Thus in the following we will restrict the study to the case $P=1$. The resolution of the process comes down to the resolution of the problem (1) with the PDEs (2) or (3). However we only have the sampled data of I on a rectangular spatial lattice. To solve the problem, it is necessary to use a numerical scheme implying the discretization of the initial continuous problem and the use of partial derivatives approximations. Let $\boldsymbol{X} = \{x^p\}_{p=1,\ldots,N}$ and n_i respectively be a $\mathbb{R}^N$ discrete grid and the number of components of the ith dimension discretization. The total point number of a grid is therefore $\Pi_N = \prod_{i=1}^N n_i$ or $\Pi_N = n_N \Pi_{N-1}$ with $\Pi_0 = 1$ as a convention. To identify the different points, we choose a linear numbering system. Each grid point x^p, with $p \in \{1,\ldots,\Pi_N\}$, is so related to the $p = p_1 + \sum_{i=2}^N [(p_i-1)\prod_{j=1}^{i-1} n_j]$ number with $\forall i,\ p_i \in \{1,\ldots,n_i\}$. The object is consequently regarded as a vector $f \in \Omega$, whose components $f^i, i \in \{1,\ldots,N\}$ are the samples. Let $\mathbf{1}_i$ be the canonical vector of the ith component of $\mathbb{R}^N$. We respectively denote δx_i and $\boldsymbol{\Delta x_i} = \delta x_i \mathbf{1}_i$ the spatial sampling and the spatial sampling vector according to the ith component of $\mathbb{R}^N$. The Taylors series give the following first and second spatial derivatives approximations:

$$\frac{\partial I(\boldsymbol{x},t)}{\partial x^i} = \frac{1}{\delta x_i}\left[I\left(\boldsymbol{x}+\frac{\boldsymbol{\Delta x_i}}{2},t\right) - I\left(\boldsymbol{x}-\frac{\boldsymbol{\Delta x_i}}{2},t\right)\right]$$
$$\frac{\partial^2 I(\boldsymbol{x},t)}{\partial x^{i2}} = \frac{1}{\delta x_i^2}[I(\boldsymbol{x}+\boldsymbol{\Delta x_i},t) + I(\boldsymbol{x}-\boldsymbol{\Delta x_i},t) - 2I(\boldsymbol{x},t)] \tag{4}$$

In the same way, we use for the left-hand side member the following approximation:

$$\frac{\partial I(\boldsymbol{x},t)}{\partial t} = \frac{1}{\Delta t}[I(\boldsymbol{x},t+\Delta t) - I(\boldsymbol{x},t)] \tag{5}$$

where Δt refers to the "temporal" sampling on the t variable. From the choice of these approximations comes the numerical resolution method, called the explicit or Euler forward scheme. The PDEs (2) and (3) are then respectively approximated as follows:

$$I(\boldsymbol{x},t+\Delta t) = I(\boldsymbol{x},t) + \Delta t \sum_{i=1}^N \frac{I(\boldsymbol{x}+\boldsymbol{\Delta x_i},t) + I(\boldsymbol{x}-\boldsymbol{\Delta x_i},t) - 2I(\boldsymbol{x},t)}{\delta x_i^2}$$

$$I(\boldsymbol{x},t+\Delta t) = I(\boldsymbol{x},t) + \sum_{i=1}^N \frac{\Delta t}{\delta x_i}\Psi_i\left(\boldsymbol{x}+\frac{\boldsymbol{\Delta x_i}}{2},t\right)\frac{I(\boldsymbol{x}+\boldsymbol{\Delta x_i},t) - I(\boldsymbol{x},t)}{\delta x_i} - \sum_{i=1}^N \frac{\Delta t}{\delta x_i}\Psi_i\left(\boldsymbol{x}-\frac{\boldsymbol{\Delta x_i}}{2},t\right)\frac{I(\boldsymbol{x},t) - I(\boldsymbol{x}-\boldsymbol{\Delta x_i},t)}{\delta x_i}$$

We first assimilate the t increase to the evolution of the process from a step n to the step $n+1$, without specifying this Δt value. Then, rather than considering the $I(\boldsymbol{x} \pm \boldsymbol{\Delta x_i},t)$, respectively $\Psi_i(\boldsymbol{x} \pm \boldsymbol{\Delta x_i},t)$, by interpolation given any $\boldsymbol{\Delta x_i}$ we prefer to consider that δx_i is unknown but corresponds de facto to the $I(\boldsymbol{x},t)$, respectively $\Psi_i(\boldsymbol{x},t)$, shifting towards one of the neighbours $I(\boldsymbol{x} \pm \mathbf{1}_i,t)$, respectively $\Psi_i(\boldsymbol{x} \pm \mathbf{1}_i,t)$. If moreover we gather the unknown Δt and δx_i increases under a same parameter $\lambda_i^n(\boldsymbol{x})$ which might as well depends on $\boldsymbol{x}$ as n, we get the iterative expressions of the processes:

$$I^{n+1}(\boldsymbol{x}) = I^n(\boldsymbol{x}) + \sum_{i=1}^N \lambda_i^n(\boldsymbol{x})[I^n(\boldsymbol{x}+\mathbf{1}_i) + I^n(\boldsymbol{x}-\mathbf{1}_i) - 2I^n(\boldsymbol{x})] \tag{6}$$

$$I^{n+1}(\boldsymbol{x}) = I^n(\boldsymbol{x}) + \sum_{i=1}^N \frac{\lambda_i^n(\boldsymbol{x})}{2}\left[\Psi_i^n(\boldsymbol{x}) + \Psi_i^n(\boldsymbol{x}+\mathbf{1}_i)\right] \times \left[I_i^n(\boldsymbol{x}+\mathbf{1}_i) - I_i^n(\boldsymbol{x})\right] - \sum_{i=1}^N \frac{\lambda_i^n(\boldsymbol{x})}{2}\left[\Psi_i^n(\boldsymbol{x}) + \Psi_i^n(\boldsymbol{x}-\mathbf{1}_i)\right] \times \left[I_i^n(\boldsymbol{x}) - I_i^n(\boldsymbol{x}-\mathbf{1}_i)\right] \tag{7}$$

Then, by using the following interpolations:

$$\Psi_i^n\left(\boldsymbol{x}+\frac{\mathbf{1_i}}{2}\right)=\frac{\Psi_i^n(\boldsymbol{x})+\Psi_i^n(\boldsymbol{x}+\mathbf{1_i})}{2}$$

$$\Psi_i^n\left(\boldsymbol{x}-\frac{\mathbf{1_i}}{2}\right)=\frac{\Psi_i^n(\boldsymbol{x})+\Psi_i^n(\boldsymbol{x}-\mathbf{1_i})}{2}$$

and organizing Eq. (7), we obtain

$$\begin{aligned}I^{n+1}(\boldsymbol{x})=I^n(\boldsymbol{x})&-\sum_{i=1}^{N}\frac{\lambda_i^n(\boldsymbol{x})}{2}\\&\times[2\Psi_i^n(\boldsymbol{x})+\Psi_i^n(\boldsymbol{x}+\mathbf{1_i})+\Psi_i^n(\boldsymbol{x}-\mathbf{1_i})]I^n(\boldsymbol{x})\\&+\sum_{i=1}^{N}\frac{\lambda_i^n(\boldsymbol{x})}{2}[\Psi_i^n(\boldsymbol{x})+\Psi_i^n(\boldsymbol{x}+\mathbf{1_i})]I^n(\boldsymbol{x}+\mathbf{1_i})\\&+\sum_{i=1}^{N}\frac{\lambda_i^n(\boldsymbol{x})}{2}[\Psi_i^n(\boldsymbol{x})+\Psi_i^n(\boldsymbol{x}-\mathbf{1_i})]I^n(\boldsymbol{x}-\mathbf{1_i})\end{aligned}\tag{8}$$

Eqs. (6) and (8) can therefore be considered as particular cases of the generalized diffusion process:

$$\begin{cases}I^{n+1}(\boldsymbol{x})=I^n(\boldsymbol{x})+\sum_{i=1}^{N}\lambda_i^n(\boldsymbol{x})[a_i^n(\boldsymbol{x})I^n(\boldsymbol{x}+\mathbf{1_i})\\\qquad\qquad+b_i^n(\boldsymbol{x})I^n(\boldsymbol{x}-\mathbf{1_i})\\\qquad\qquad-(a_i^n(\boldsymbol{x})+b_i^n(\boldsymbol{x}))I^n(\boldsymbol{x})]\\I^0(\boldsymbol{x})=f(\boldsymbol{x})\end{cases}\tag{9}$$

- with $a_i^n(\boldsymbol{x})=b_i^n(\boldsymbol{x})=1$ in the case of Eq. (6);
- with $a_i^n(\boldsymbol{x})=\frac{1}{2}\left(\Psi_i^n(\boldsymbol{x})+\Psi_i^n(\boldsymbol{x}+\mathbf{1_i})\right)$ and $b_i^n(\boldsymbol{x})=\frac{1}{2}(\Psi_i^n(\boldsymbol{x})+\Psi_i^n(\boldsymbol{x}-\mathbf{1_i}))$ in the case of Eq. (8).

During the application of an iterative diffusion process, the λ parameter has a very particular importance. Its behavior, its convergence, and some characteristics of the generated scale spaces directly depend on the λ value. In the following we establish a global convergence requirement of such explicit schemes.

2.3. Convergence requirement

The iterative form of Eq. (9) globally expresses itself as

$$\begin{cases}I^{n+1}=\Phi(I^n)\\I^0=f\end{cases}\tag{10}$$

with a vectorial function Φ composed of the $\varphi_{\boldsymbol{x}}(I^n)$ functions such as

$$\begin{aligned}\varphi_{\boldsymbol{x}}(I^n)=I^n(\boldsymbol{x})+\sum_{i=1}^{N}\lambda_i^n(\boldsymbol{x})\big[&a_i^n(\boldsymbol{x})I^n(\boldsymbol{x}+\mathbf{1_i})\\&+b_i^n(\boldsymbol{x})I^n(\boldsymbol{x}-\mathbf{1_i})-(a_i^n(\boldsymbol{x})+b_i^n(\boldsymbol{x}))I^n(\boldsymbol{x})\big]\end{aligned}\tag{11}$$

and therefore such as $I^{n+1}(\boldsymbol{x})=\varphi_{\boldsymbol{x}}(I^n)$.

This expression shows that the iterative resolution process is a fixed point research method since if the process converges, it converges towards I^∞ such as $I^\infty=\Phi(I^\infty)$, i.e. such as for all $\boldsymbol{x}$ we shall have $I^\infty(\boldsymbol{x})=\varphi_{\boldsymbol{x}}(I^\infty)$. These fixed point research methods, called Newton methods, have been widely studied and are known to converge if all the eigenvalues of the Φ Jacobian matrix are inferior to 1 in absolute value (Ciarlet, 1998). Thus to study the λ parameter influence on the convergence of the iterative diffusion process, we bound the spectral radius of the Φ Jacobian matrix.

The vectorial function Φ has Π_N components the φ_{x^p}, also denoted φ^p. The Φ Jacobian matrix is so a $\Pi_N\times\Pi_N$ square matrix which pth row is composed of the derivations of $\varphi^p(I^n)$ with respect to $I^n(x^q)$:

$$J(\Phi)=\begin{pmatrix}\dfrac{\partial\varphi^1(I^n)}{\partial I^n(x^1)} & \cdots & \dfrac{\partial\varphi^1(I^n)}{\partial I^n(x^{\Pi_N})}\\ \vdots & \dfrac{\partial\varphi^p(I^n)}{\partial I^n(x^q)} & \vdots\\ \dfrac{\partial\varphi^{\Pi_N}(I^n)}{\partial I^n(x^1)} & \cdots & \dfrac{\partial\varphi^{\Pi_N}(I^n)}{\partial I^n(x^{\Pi_N})}\end{pmatrix}$$

By bounding the spectral radius, we can prove (Tremblais, 2002) that if we consider the generalized diffusion process and if we suppose that $\forall n,\forall i,\forall\boldsymbol{x}$ we have $a_i^n(\boldsymbol{x})\geqslant 0$ and $b_i^n(\boldsymbol{x})\geqslant 0$, which is obvious here, then the numerical processes will converge if $\forall n,\forall\boldsymbol{x}$, the parameter λ verifies

$$0\leqslant\sum_{i=1}^{N}\lambda_i^n(\boldsymbol{x})\left[a_i^n(\boldsymbol{x})+b_i^n(\boldsymbol{x})\right]<1\tag{12}$$

Note that unlike many authors we avoid the case $\sum_{i=1}^{N}\lambda_i^n(\boldsymbol{x})[a_i^n(\boldsymbol{x})+b_i^n(\boldsymbol{x})]=1$, since though stable, the process may then produce undesirable oscillations.

3. A new edge-preserving diffusion scheme

3.1. General principle: λ-resolution

As previously seen, the use of the explicit scheme does not ensure an unconditional convergence of the diffusion process. In this way, if $\forall i$, $\lambda_i^n(\boldsymbol{x}) = \lambda^n(\boldsymbol{x})$, then $\lambda^n(\boldsymbol{x})$ must belong to a convergence range $[0, \lambda_{\max}[$, depending on the diffusion process. The most widespread strategy is to choose the same evolution parameter for all of the local evolution process, i.e. $\forall \boldsymbol{x}$, $\forall n$, $\lambda^n(\boldsymbol{x}) = \lambda$ with $\lambda \ll \lambda_{\max}$ to ensure the global convergence of the scheme. But by making this choice, the whole filtering procedure becomes rather time expensive.

Here we choose a quite different approach, called λ-resolution. We choose a space and time localized $\lambda_i^n(\boldsymbol{x})$ parameters such that the condition (12) is verified. Thus we ensure the convergence for each local process. In this way, we can adapt the diffusion process' evolution speed according to the local features of the processed object. Consequently, we need to choose the expression of $\lambda_i^n(\boldsymbol{x})$ at best, knowing that on the one hand, the more the value of $\sum_{i=1}^{N} \lambda_i^n(\boldsymbol{x})[a_i^n(\boldsymbol{x}) + b_i^n(\boldsymbol{x})]$ is inferiorly close to 1, the faster the process converges or smoothes, and on the other hand, the more this value is close to 0, the more the evolution process is slowed, or even stopped. So, if we wish to smooth a signal while preserving some relevant structures, as edges for example, we have to write the $\lambda_i^n(\boldsymbol{x})$ parameters according to a relevant structure indicator and to define this indicator in such a way that $\sum_{i=1}^{N} \lambda_i^n(\boldsymbol{x})[a_i^n(\boldsymbol{x}) + b_i^n(\boldsymbol{x})]$ has to be low (or zero) when one wishes to preserve the structure and to be closed to 1 elsewhere. To do that, we can use some local statistic informations, as those of the data mean and variance, or some local geometric informations, as the norm of the gradient, local curvatures, etc. The present idea is to suggest a localized evolution parameter to control the diffusion process speed according to the local geometry or the local statistics object features. Such a behavior allows then to automatically preserve the relevant structures in the object even using the linear PDE (2). The nonlinearity is here produced by the adaptative resolution of the limits' problem (1).

3.2. Edge-preserving resolution parameter

From now on, we will focus on the resolution of Eq. (2), that is we consider the particular case $a_i^n(\boldsymbol{x}) = b_i^n(\boldsymbol{x}) = 1$. Then if for all i, $\lambda_i^n(\boldsymbol{x}) = \lambda^n(\boldsymbol{x})$ the condition of convergence (12) becomes: $0 \leqslant \lambda^n(\boldsymbol{x}) < \frac{1}{2N}\,\forall n,\ \forall \boldsymbol{x}$. Consequently, if we want to preserve edges, some possible choices for the resolution parameter are:

$$\lambda^n(\boldsymbol{x}) = \frac{1}{2N + \epsilon + \left(\exp\left(\dfrac{[E^r(\boldsymbol{x})]^2}{K^2}\right) - 1\right)} \tag{13}$$

$$\lambda^n(\boldsymbol{x}) = \frac{1}{2N + \epsilon + \left(\dfrac{E^r(\boldsymbol{x})}{K}\right)^2} \tag{14}$$

where $E^r(\boldsymbol{x})$ is a positive real local edge response at step n, for example a gradient of I norm, Euclidean or not, and K is a threshold that allows to distinguish the high edge response values points due to noise from those due to the edges, and ϵ is a small strictly positive real avoiding oscillations state. As an illustration, we present on Fig. 1 a plot of the two suggested λ functions for the edge response, with $N = 2$ and $E^r(\boldsymbol{x})$ a gradient norm. As mentioned by Perona and Malik (1990), the constant K can either be fixed by hand, or using the noise estimator described by Canny (1986).

From this point of view, one can interpret the λ-resolution as a nonlinear, or adaptative resolution of the linear diffusion equation which is conceptually different from the conventional nonlinear diffusion filtering where the nonlinearity is spatial.

3.3. Characteristics of the edge-preserving resolution

3.3.1. Preserving edge localization

To illustrate the edge localization preserving property, we now apply to a one-dimensional test signal, the linear diffusion filtering process using our edge-preserving resolution scheme. Notice that the more the dimension is high the more the gradient norm indicator is suitable since the local information increase. The one-dimensional test signal is a block signal with values in the range $[-50, 50]$ corrupted with an additive white noise

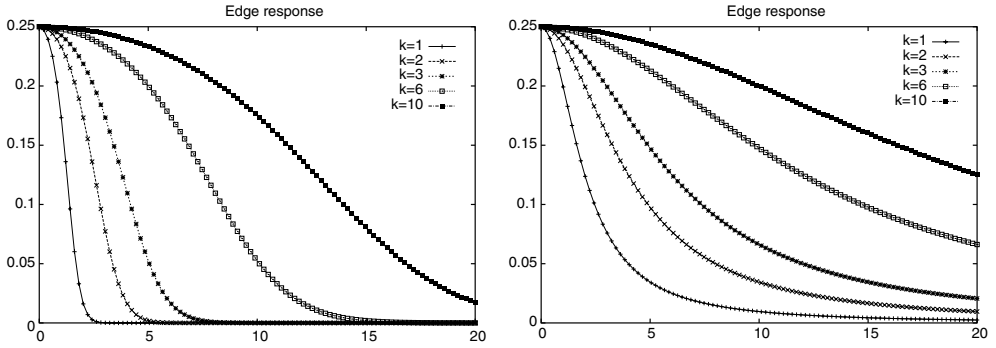

Fig. 1. Left: plot of $\left[4+\epsilon+\left(\exp\left(\frac{[E^r(\boldsymbol{x})]^2}{K^2}\right)-1\right)\right]^{-1}$ for some values of K, right: plot of $\left[4+\epsilon+\left(\frac{E^r(\boldsymbol{x})}{K}\right)^2\right]^{-1}$ for some values of K.

such that the signal noise ratio (SNR) is 15 dB. Fig. 2 depicts the evolution of this signal through our diffusion scheme using the resolution parameter

$$\lambda^n(x)=\left[2+\epsilon+\left(\exp\left(\frac{(I_x^n)^2}{5^2}\right)-1\right)\right]^{-1}$$

It is interesting to note that the high discontinuities are well preserved whereas small variations of the signal are smoothed.

3.3.2. Rotational invariance

In the continuous case, the linear diffusion filtering is known to be rotational invariant. In the discrete case, the rotational invariance depends on the spatial approximation of the second derivatives sum. Our scheme can be interpreted as a selective Gaussian smoothing where intra-regional areas are smoothed whereas inter-regional areas are not. Consequently, the scheme is likewise rotational invariant in intra-regional areas as the Gaussian smoothing and it is in inter-regional

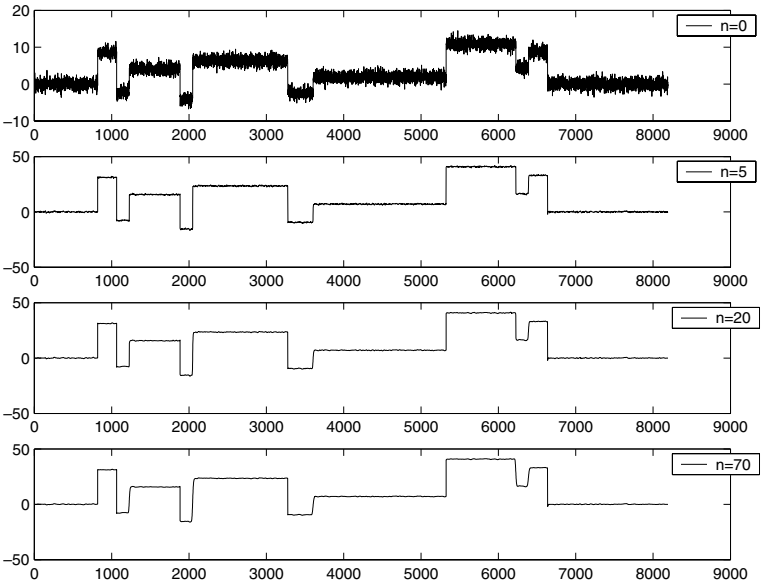

Fig. 2. A family of one-dimensional signals $I(x,t)$ obtained by applying the λ-resolution scheme with $\lambda^n(x)=\left[2+\epsilon+\left(\exp\left(\frac{|I_x^n|^2}{K^2}\right)-1\right)\right]^{-1}$, $K=5$ computed for all x and an increasing scale (or iteration n) from top to bottom.

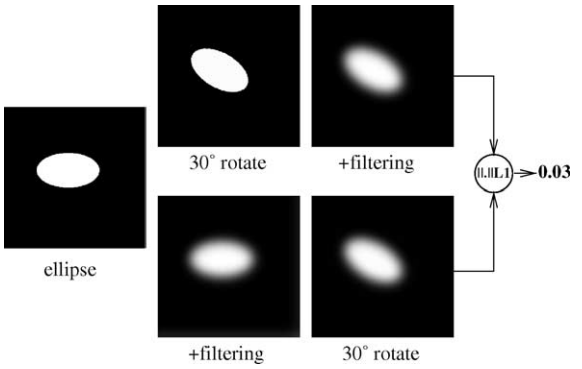

Fig. 3. Illustration of the rotational invariance.

areas too since they are preserved. Fig. 3 is used as a test for rotational invariance. On the one hand, we present the filtering of a rotated ellipse and on the other hand the rotation of a filtered image. We see that the resulting images are quite similar, and no preference of certain filtering directions appears.

3.3.3. Computational considerations

As the proposed scheme requires only a few programming lines more than the conventional linear diffusion filtering explicit one, i.e. the computation of $\lambda^n(\boldsymbol{x})$ for all $\boldsymbol{x}$ at each iteration n, we may expect a low computational cost. We summarize in Table 1 the computational requirements for one step of the λ-resolution using

$$\lambda^n(\boldsymbol{x}) = \left[2N + \epsilon + \left(\exp\left(\frac{\|\nabla I\|^2_{\mathscr{L}^2}}{K^2}\right) - 1\right)\right]^{-1}$$

one step of the Perona and Malik (1990) nonlinear diffusion filtering resolved using the conventional explicit scheme, and one step of the AOS scheme (Weickert et al., 1998) given here without the step of regularization. We provide as separate values the multiplications or divisions (M/D) efforts according to the number Π_N of points to treat and the dimension N, and the additions or subtractions (A/S) one. The computational cost of the λ-resolution is proportional to the Π_N and N numbers, and is only $(2N+4)\Pi_N + 1$ multiplications or divisions, and $(4N+1)\Pi_N + 1$ additions or subtractions that is less than those of the two others methods. Computational effort is thus linear in size and dimension of the object. The computational efficiency of our scheme increases with these two parameters. It is furthermore worth noticing that as for the conventional nonlinear diffusion filtering, our scheme can be accelerated using the $\mathscr{L}_1$ norm instead of the $\mathscr{L}_2$ norm and/or being hardware implemented (Perona and Malik, 1990; Gijbels et al., 1994).

Of course, it may seem improper to compare the one step complexity of an explicit scheme with an implicit one such as the AOS. If only one step is needed by implicit schemes whereas explicit ones require multiple steps, within the context of a multiscale analysis we need to compute at multiple scales for all schemes.

4. The multi-scale edge detection algorithm

In this section, in order to simplify both the explanations and the illustrations we present the two-dimensional case ($N = 2$). Extension to the third-dimensional case is straightforward. It is indeed enough to extend the definition of connected components to higher dimensions. The suggested

Table 1
Main operations for one step of the N-dimensional Perona and Malik, AOS, and adaptative resolution schemes (M/D: multiplications or divisions; A/S: additions or subtractions)

Scheme	Perona		AOS		λ-Resolution	
Task	M/D	A/S	M/D	A/S	M/D	A/S
Calculate $\|\nabla \boldsymbol{I}\|^2_{\mathscr{L}^2}$	$2N\Pi_N$	$(2N-1)\Pi_N$	$2N\Pi_N$	$(2N-1)\Pi_N$	$2\Pi_N N$	$(2N-1)\Pi_N$
$\boldsymbol{g}(\|\nabla \boldsymbol{I}\|^2_{\mathscr{L}^2})$	Π_N	–	–	–	Π_N	–
Diffusion step	$(2N+2)\Pi_N$	$6N\Pi_N$	$6N\Pi_N$	$3N\Pi_N$	$6\Pi_N$	$(2N+2)\Pi_N$
Others	1	–	–	–	1	1
Total	$(4N+2)\Pi_N + 1$	$(8N-1)\Pi_N$	$8N\Pi_N$	$(8N-1)\Pi_N$	$(2N+4)\Pi_N + 1$	$(4N+1)\Pi_N + 1$

algorithm is an adaptation of the one we have presented in (Tremblais et al., 2003) for the detection of bottom-lines in 2D X-rays angiograms.

Let $\Sigma = \{\sigma_{i\in\{1,\dots,p\}}\}$ be a set of p scales. Let $\{E^r_{\sigma_i}\}_{i\in\{1,\dots,p\}}$, and $\{E_{\sigma_i}\}_{i\in\{1,\dots,p\}}$ respectively be the set of the edges response and the set of the edge map at scale σ_i which we will call the *edges response scale-space* and the *edge map scale-space*. Notice that the dimension of these spaces is $N+1$. There are two main strategies to perform a multi-scale analysis. The first one consists in fusionning the $E^r_{\sigma_i}$ and then make the decision step. The second strategy uses both the edges response, and the edge map scale-spaces to take a final decision. We adopt this last solution. In the multi-scale analysis framework the fusion step is generally a sensitive step. Anyway, due to the λ-resolution process, the edges remain well localized across the scale-space, and we will show that, contrary to Clark (1988) or Lindeberg (1999), no complex fusion strategy will be necessary. We have to establish the parameters of the scale-space the scales study range $[\sigma_{\min}, \sigma_{\max}]$ and the distribution of the chosen scales in this range. As the edges response generally depends on differential operators, the first scale is determinated so as to limit numerical approximation errors and justify the use of these operators. The choice of the maximal scale depends on the maximal size of the structures to detect, and so on the application. The lower it is, the lower computation times are. Though we do not use a constant resolution parameter we decide to use a logarithmic sampling of the scale parameter because of the asymptotic behaviour of the λ-resolution process (cf. Fig. 4).

As Monga et al. (1994), we construct then a multi-scale adjacency graph from those two scale-spaces. The nodes of the graph are the points P which have been detected at least once across the edges scale-space, that is to say the points such that there exists σ_i such that $E_{\sigma_i}(P) = 1$. Each node contains all the scales σ_i verifying $E_{\sigma_i}(P) = 1$ as well as the edge response, and possibly the direction of the gradient. Then, an edge connects two nodes if the points are adjacent, or neighbours, in the edge map scale-space. Let $\mathrm{NE}(P^{i-1})$, $\mathrm{NE}(P^i)$ and $\mathrm{NE}(P^{i+1})$ respectively be the set of the neighbours of P in the edge map at the scales σ_{i-1}, σ_i and σ_{i+1}. We define the set of the neighbours of P^i in the edge map scale-space as the set $N(P^i) = \{\mathrm{NE}(P^{i-1}) \cup P^{i-1}, \mathrm{NE}(P^i),\ \mathrm{NE}(P^{i+1}) \cup P^{i+1}\}$. Thus in the two-dimensional case nodes have 26 neighbours and in the three-dimensional case they have 80 neighbours.

As our diffusion filtering preserves edge localization we can use the stability hypothesis first planed by Witkin (1983): the structures that "survive over a broad of scales tends to leap out the eye…". So we decide to prune the graph keeping nodes that have been detected at least D times. It allows to eliminate a lot of not significant edges. Let $\#C$ be the number of connected components of the graph, we denote by $\{CC_k\}$, $k \in \{1,\dots,\#C\}$ the set of these connected components. Then to each connected components we attribute the following measure:

$$M(CC_k) = \frac{\sum_{P\in CC_k} E^r(P)}{\#CC_k} \tag{15}$$

where $\#CC_k$ and $E^r(P)$ respectively denote the number of nodes belonging to the connected components CC_k and the edge response corresponding to the P point, $M(CC_k)$ being the mean of measures corresponding to the connected component CC_k. Thereafter, we sort the connected components by descending order according to the $M(CC_k)$ values. Tuning the number S between 1 and $\#C$, we can keep the S "strongest" connected components. Projecting these components onto the image plane we obtain the "shadow" of the S strongest connected components. The choice of S may be let to the judgment of the user. We can see on Fig. 5 the impact of the choice of S on the results of the multi-scale detection. Then an homotopic thinning algorithm (Lantuéjoul, 1978) allows to obtain the edges as one pixel width 8-connected paths. [1] Finally, the remaining artifacts detections are eliminated deleting low length edges. Fig. 6 shows a more synthetic description of our fusion algorithm.

[1] As edges stay well localized this step could be avoided but it ensures one pixel width edges.

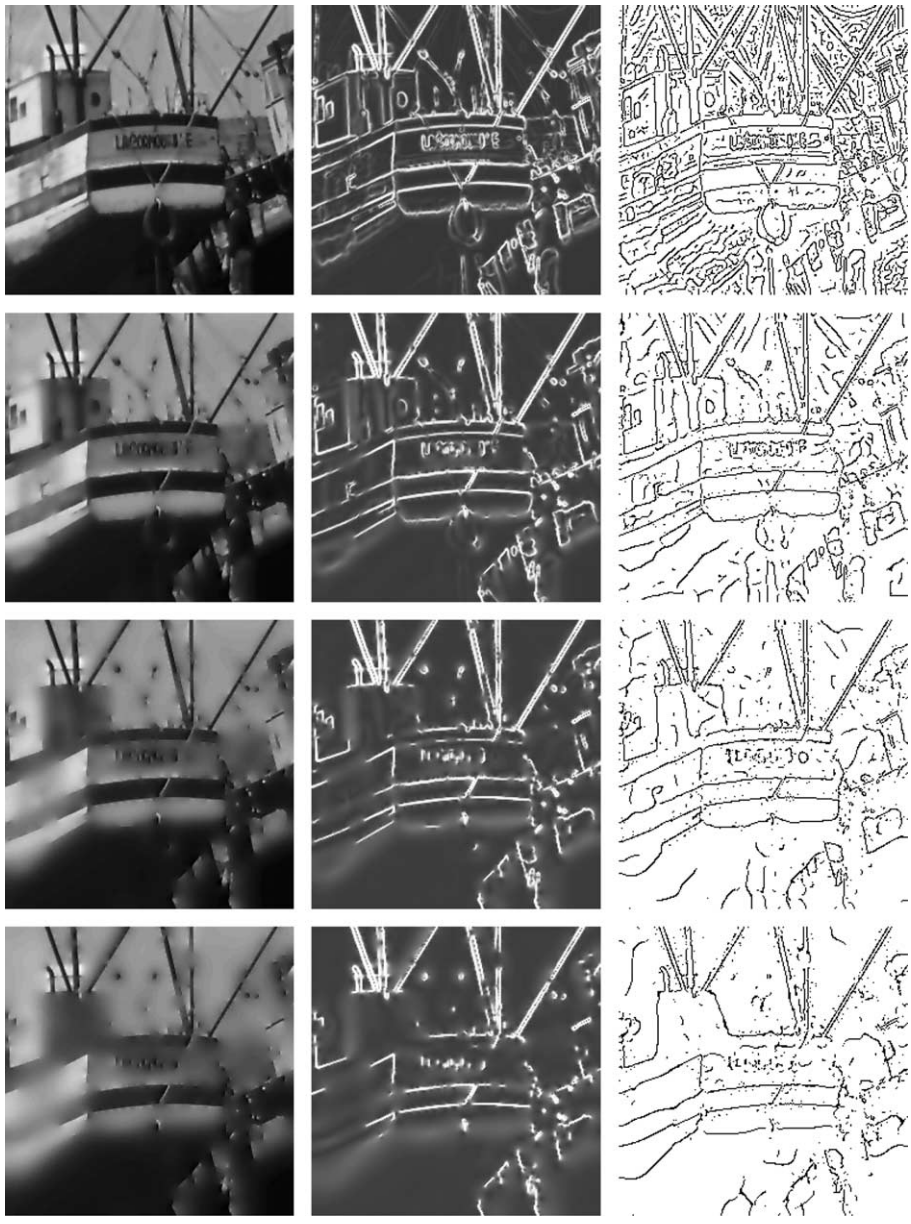

Fig. 4. Left: some scales of linear diffusion filtering resolved with Eq. (13) where $E^r(\boldsymbol{x}) = \|\nabla I(\boldsymbol{x})\|_{\mathscr{L}^2}$ and $K = 10$; middle: some scales of the edges response scale-space; right: some scales of the edge map scale-space, extracted here by the maxima of the gradient norm is the direction of the gradient.

5. Experimental results

Here we present some examples of detection on greyscale and color images using our diffusion filtering combined with the previous multi-scale algorithm.

5.1. Greyscale images

We give here a comparison of different diffusion filtering techniques combined with the multi-scale edge detection algorithm previously presented

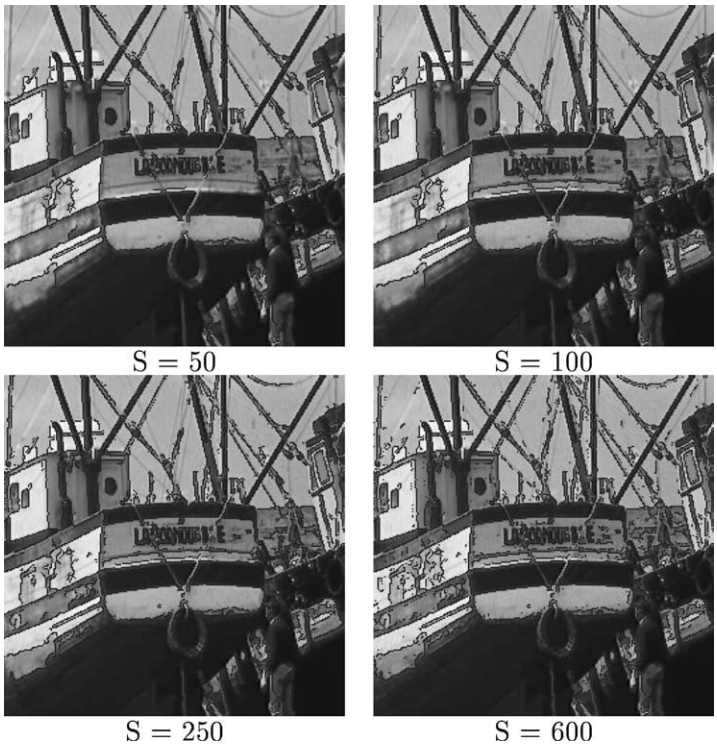

Fig. 5. Some edge detections for different values of S.

using 14 scales logarithmically sampled between $n = 4$, and $n = 103$, and using the parameters $D = 5$ and $S = 1800$ as well as the conventional Canny–Deriche edge detection (Deriche, 1990; Canny, 1986) techniques. The inner scale is set to $n = 4$ avoid aliasing problems. The outer scale is arbitrary chosen quite high to allow the detection of large structures. Without prior informations about the processed image it is difficult to choose this scale. The details of the different techniques are:

(1) Canny–Deriche technique with $\alpha = 0.7$;
(2) linear diffusion filtering resolved with $\lambda^n(\boldsymbol{x}) = 0.2$;
(3) nonlinear diffusion filtering resolved with $\lambda^n(\boldsymbol{x}) = 0.2$, $\Psi_i(\boldsymbol{x}, t) = \exp\left(-\frac{|I_x^2+I_y^2|^2}{6^2}\right)$;
(4) linear diffusion filtering resolved with $\lambda_g^n(\boldsymbol{x}) = \left[4 + \epsilon + \exp\left(\frac{I_x^2+I_y^2}{5^2}\right) - 1\right]^{-1}$, $\epsilon = 0.01$;
(5) linear diffusion filtering resolved with $\lambda_a^n(\boldsymbol{x}) = \left[4 + \epsilon + \frac{|I_x|+|I_y|}{0.5}\right]^{-1}$, $\epsilon = 0.01$.

All the results are presented in Fig. 7. Method (1) is linear and treats only one scale and so, as it appears in Fig. 7 all the objects of the image cannot be detected. Method (2) is still based on linear filtering but uses the information of several scales. As the edges are delocalized across the scale-space, the simple algorithm suggested in this paper is unadapted to follow the edges across scales, more complicated and time expensive algorithms should be used here to obtain better results. All other methods (3)–(5) give good results concerning the localization of the edges whether for small or large structures. Results of the methods (4) and (5) are almost similar while the method (3) gives more detections. This may be due to the enhancement properties of the conduction function used here.

Table 2 compares the measured computation times using AMD Athlon XP-1.4 GHz processor with 256 Mo RAM running on a linux OS for the different techniques and for different image sizes. The algorithms have been implemented using

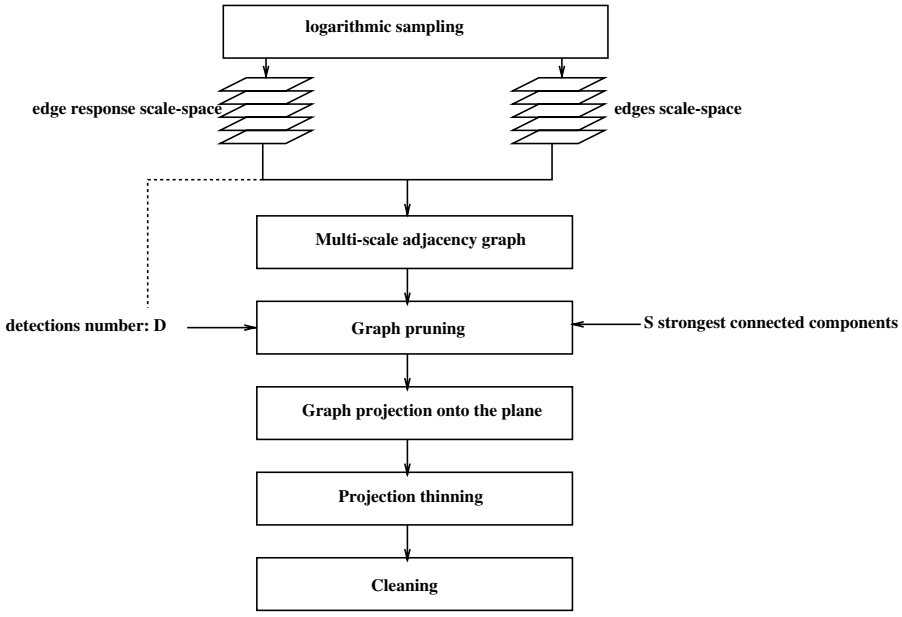

Fig. 6. Fusion algorithm.

the C/C++ programming language. We have omitted the Canny–Deriche technique computation times since it is a non multi-scale analysis and it is therefore nonsense to compare it here. As we may expect, the linear diffusion filtering resolved with a fixed resolution parameter is the fastest multi-scale technique. Among the edge-preserving techniques, the two techniques we have suggested in this paper are faster than the Perona and Malik (1990) one. As for these three techniques, the edge detection results are roughly similar our techniques offers an interesting alternative for a fast multi-scale edge detection.

5.2. Color images

There are several color spaces that can be divided in four main categories: the computer graphics color spaces: RGB, XYZ, etc.; TV/video, or luminance-chrominance ones: YUV, YCbCr; the human perception linked ones: HSL, HSI, etc., and the approximations of perceptually uniform space ones: Cie, 1976 $L^*a^*b^*$ Cie $L^*u^*v^*$, etc. (see Poyton, 1995, for detailed specifications of most of these color spaces). The choice of a color space generally depends on the device: using the luminance-chrominance color spaces is usually more adapted to the study of compressed images, or videos; the HSL or HSI choice allows to break shade effects; performing the calculations in Cie $L^*a^*b^*$ may produce perceptually better results. But on the other hand, converting degraded images between different spaces may in many instances degrade the information even more. To sum up, the choice of a color space will first depend on the degradations of the image, and then on the application.

Anyway in most of the color spaces, pixels may be seen as 3D vectors and color images are consequently modelized as the mapping $\boldsymbol{f} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$. The principles presented in Section 3.2 are quite general and independent of the color space. Let us not forget that the choice of the edge response $E^r(\boldsymbol{x})$ must be well-adapted to the color space metric. There are principally two ways to extend the previous diffusion filtering to color, or vector valued, images. The first, and the simpliest one is to apply the scalar diffusion filtering to each channel I^j. But as the marginal gradients ∇I^j are generally different when the noise is independent of

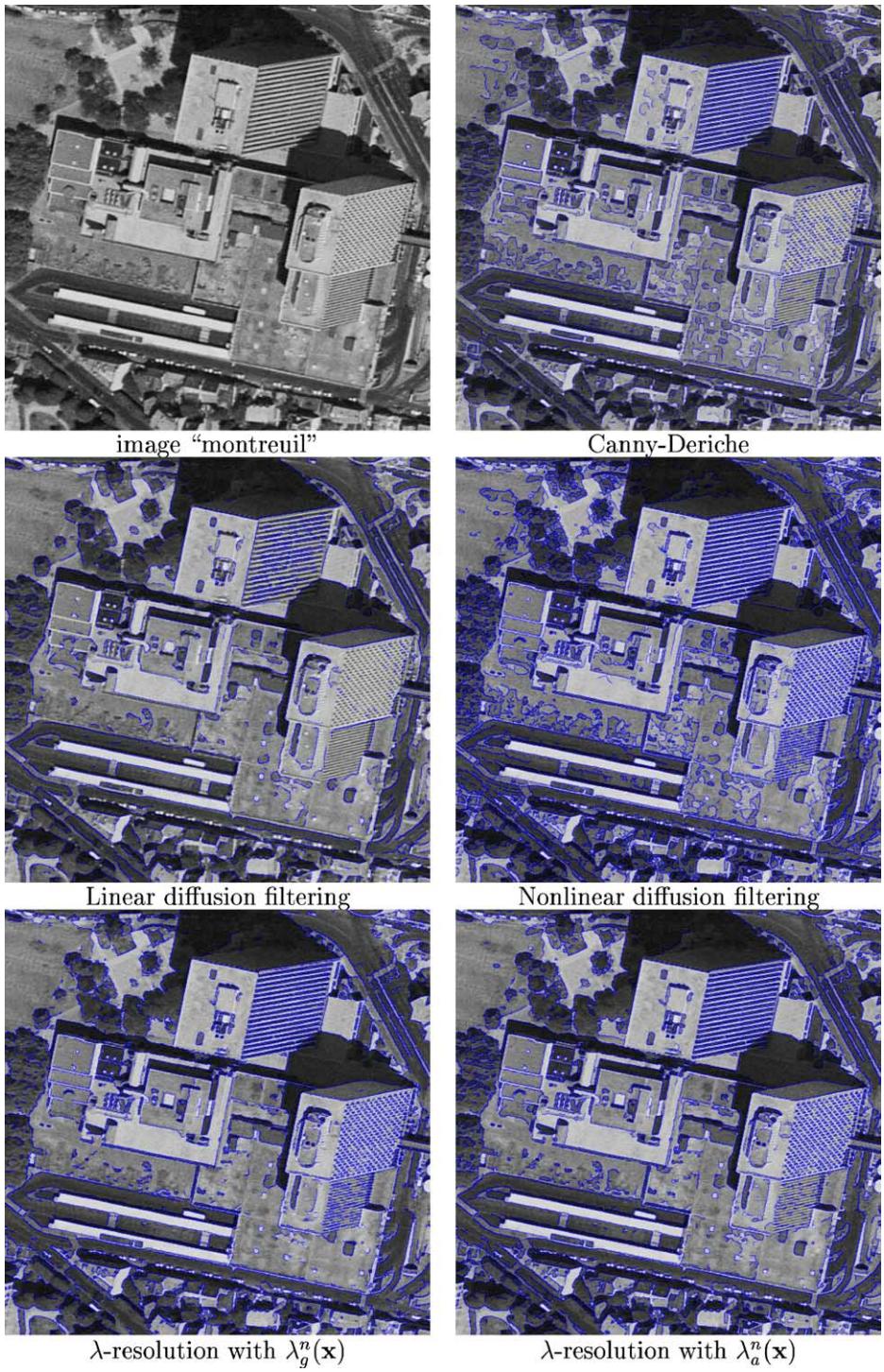

Fig. 7. Some edge detection results applied to the greyscale image "montreuil".

Table 2
Comparison of the CPU times for greyscale images of sizes 256×256 and 512×512

Technique	CPU time (s)	
	256×256	512×512
Linear diffusion filtering	2.45	9.18
Nonlinear diffusion filtering	6.66	26.61
λ-Resolution, $\lambda_g^n(\boldsymbol{x})$	4.93	18.71
λ-Resolution, $\lambda_a^n(\boldsymbol{x})$	2.78	11.15

each channel, this technique tends to blend colors and blur the edges. The second one is to define a vector-valued edge with a vector-valued edge response, and a vector-valued direction. Di-Zenzo (1986) was the first to suggest the Riemannian geometry framework for edge detection in multi-valued images. Lee and Cok (1991) show that using a vector valued edge response is more insensitive to noise than using a simple combination of the j scalar edge responses. The Di-Zenzo approach constitutes a mathematically consistent extension of scalar edges. The variations of the vector norm $\|\mathrm{d}\boldsymbol{I}\|^2$ are given by the first fundamental form:

$$\|\mathrm{d}\boldsymbol{I}\|^2 = \mathrm{d}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{G}\mathrm{d}\boldsymbol{x} \tag{16}$$

with

$$\boldsymbol{G} = (g_{kl}) = \begin{pmatrix} \sum_{j=1}^{3} (I_x^j)^2 & \sum_{j=1}^{3} I_x^j I_y^j \\ \sum_{j=1}^{3} I_x^j I_y^j & \sum_{j=1}^{3} (I_y^j)^2 \end{pmatrix} \tag{17}$$

This matrix has two positive eigenvalues λ^+, and λ^- which are the maximum, and the minimum of $\|\mathrm{d}\boldsymbol{I}\|^2$, and two eigenvectors $\boldsymbol{v}^+$, and $\boldsymbol{v}^-$ which are orthogonal and correspond to the orientations of the variations. Their formulae are given by

$$\lambda^{+/-} = \frac{g_{11} + g_{22} \pm \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2} \tag{18}$$

Fig. 8. Some edge detection results applied to the color image "miss" working on the RGB color space.

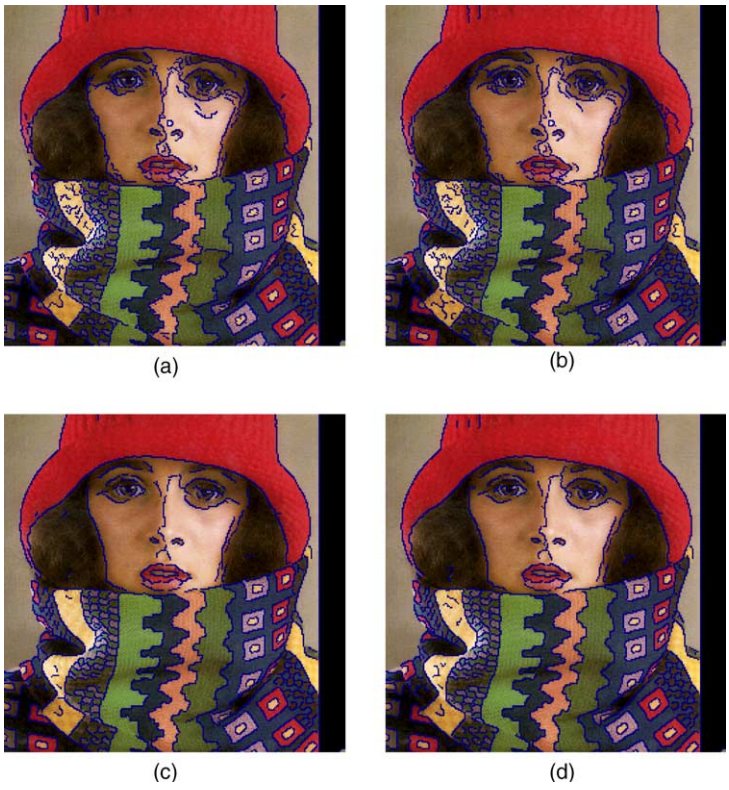

Fig. 9. A comparison of results obtained using different color spaces: (a) RGB color space and $E^r(\boldsymbol{x}) = \sqrt{\lambda^+}$; (b) YCbCr color space and $E^r(\boldsymbol{x}) = \sqrt{\lambda^+}$; (c) CieLab color space and $E^r(\boldsymbol{x}) = \sqrt{\lambda^+}$; (d) CieLab color space, $E^r(\boldsymbol{x}) = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$.

$$\boldsymbol{v}^{+/-} = \begin{pmatrix} 2g_{12} \\ g_{22} - g_{11} \pm \sqrt{(g_{22} - g_{11})^2 + 4g_{12}^2} \end{pmatrix} \quad (19)$$

Based on this work, several vector-valued edge responses have been suggested, including $E^r(\boldsymbol{x}) = \sqrt{\lambda^+}$ (Di-Zenzo, 1986; Blomgren, 1998; Tschumperle, 2002), $E^r(\boldsymbol{x}) = \sqrt{\lambda^+ - \lambda^-}$ (Sapiro and Ringach, 1996; Weickert, 1997), and $E^r(\boldsymbol{x}) = \sqrt{\lambda^+ + \lambda^-}$ [2] (Whitaker and Gerig, 1994; Blomgren, 1998). In the following we will simply use the edge response $E^r(\boldsymbol{x}) = \sqrt{\lambda^+}$ with its associated edge direction $\boldsymbol{v}^+$.

As we did previously, we give here a comparison of different diffusion filtering techniques combined with our multi-scale edge detection algorithm using 14 scales logarithmically sampled between $n = 4$ and $n = 103$, and the parameters $D = 5$ and $S = 300$. The color space used is the RGB one. The details of the different techniques are:

(1) linear diffusion filtering resolved with $\lambda^n(\boldsymbol{x}) = 0.2$;
(2) nonlinear diffusion filtering (Perona and Malik, 1990) resolved with $\lambda^n(\boldsymbol{x}) = 0.2$, $\Psi_i(\boldsymbol{x}, t) = \exp\left(-\frac{\lambda^+}{8^2}\right)$;
(3) linear diffusion filtering resolved with $\lambda^n(\boldsymbol{x}) = \left[4 + \epsilon + \left(\frac{\lambda^+}{1^2}\right)\right]^{-1}$, $\epsilon = 0.01$.

As for the greyscale image case, the linear diffusion filtering resolved with a constant resolution parameter give poor results (Fig. 8). The two others techniques give both well-localized edges whether for small and for large structures. Method (2) still gives more edges.

[2] This edge response is in fact $\|\nabla \boldsymbol{I}\|_{\mathscr{L}^2}$.

Table 3
Comparison of the CPU times for color images of sizes 256×256 and 512×512

Technique	CPU time (s)	
	256×256	512×512
Linear diffusion filtering	6.77	25.40
Nonlinear diffusion filtering	10.98	45.49
λ-Resolution	7.52	29.70

Fig. 9 compares the results obtained using the method (3) applied to the RGB (Fig. 9a), the YCbCr (Fig. 9b), the Cie $L^*a^*b^*$ (Fig. 9c) color spaces, to the results obtained using the third method taking the Cie $L^*a^*b^*$ color difference definition as the edges response, that is to say $E^r(\boldsymbol{x}) = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$.

The measured computation times for these three edge detection techniques are given in Table 3. As we may expect, λ-resolution is almost three times slower than those obtained for greyscale images.

6. Conclusion

In this paper, we first present a new explicit numerical scheme to approximate the solution to the linear diffusion filtering. This scheme is stable, fast and easy to program, extensible to higher dimensions, and preserves the discontinuities of the objects. Then we combine this diffusion scheme with an easy programming and fast multi-scale edge detection algorithm. Finally, we have shown the efficiency of the proposed approach for the detection of edges in greyscale and color images. Since the λ-resolution has an interesting complexity we aim at applying it on objects like volumes or image sequences.

Acknowledgements

This work was supported by the "Conseil Regional de Poitou-Charentes", France.

References

Acton, S., Bovik, A., Crawford, M., 1994. Anisotropic diffusion pyramids for image segmentation. IEEE ICIP 3 (November), 478–482.

Alvarez, L., 1996. Images and PDE's. In: ICAOS'96, 12th Internat. Conf. on Analysis and Optimization of Systems.

Alvarez, L., Mazorra, L., 1994. Signal and image restoration using shock filters and anisotropic diffusion. SIAM J. Numer. Anal. 31 (2), 590–605.

Alvarez, L., Morel, J.M., 1994. Formalization and computational aspects of image analysis. Acta Numer., 1–59.

Aubert, G., Kornprobst, P., 2002. Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations. In: Applied Mathematical Sciences, vol. 147. Springer-Verlag.

Baensch, E., Mikula, K., 1997. A coarsening finite element strategy in image selective smoothing. Comput. Visualizat. Sci. 1 (1), 53–61.

Blomgren, P., 1998. Total variation methods for restoration of vector valued images. Ph.D. thesis, University of California, Los Angeles.

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Machine Intell. 8 (6), 679–698.

Catte, F., Lions, P., Morel, J., Coll, T., 1992. Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. 29 (1), 182–193.

Ciarlet, P.G., 1998. Introductionà l'analyse numérique matricielle et à l'optimisation. Dunod, Paris.

Clark, J., 1988. Singularity theory and phantom edges in scale space. IEEE Trans. Pattern Recognition Machine Intell. 10 (5), 720–727.

Cohignac, T., Eve, F., Guichard, F., Lopez, C., Morel, J.M., 1993. Affine morphological scale-space: Numerical analysis of its fundamental equation. Technical Report, CEREMADE, Univ. Paris Dauphine.

Deriche, R., 1990. Fast algorithms for low-level vision. IEEE Trans. Pattern Anal. Machine Intell. 1 (12), 78–88.

Di-Zenzo, S., 1986. A note on the gradient of multi-image. Comput. Vision Graphics Image Process. 33, 116–125.

Fröhlich, J., Weickert, J., 1994. Image processing using a wavelet algorithm for non linear diffusion. Technical Report 104, Laboratory of Technomathematics, University of Kaiserslautern.

Gijbels, T., Six, P., Gool, L.V., Catthoor, F., Man, H.D., Oosterlinck, A., 1994. A VLSI-architecture for parallel nonlinear diffusion with applications in vision. In: IEEE Workshop on VLSI Signal Processing, vol. 7, pp. 398–707.

Koenderink, J., 1984. The structure of images. Biol. Cybernet. 50 (5), 363–370.

Lantuéjoul, C., 1978. Détection squelettisation et son application aux mesures topologiques des mosaiques polycristallines. Ph.D. thesis, École des Mines de Paris.

Lee, H.-C., Cok, D., 1991. Detecting boundaries in a vector field. IEEE Trans. Pattern Anal. Machine Intell. 39 (5), 1181–1194.

Lindeberg, T., 1990. Scale-space for discrete signals. IEEE Trans. Pattern Anal. Machine Intell. 12, 234–254.

Lindeberg, T., 1999. Automatic scale selection as a pre-processing stage for interpreting the visual world. In: Proc. Fundamental Structural Properties in Image and Pattern Analysis, vol. 130, Budapest, Hungary, pp. 9–23.

Malladi, R., Sethian, J., 1996. Image processing: flows under min/max curvature and mean curvature. Graphical Models Image Process. 58 (2), 127–141.

Marquina, A., Osher, S., 1999. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. Technical Report 99-5, Department of Mathematics, UCLA.

Monga, O., Lengagne, R., Deriche, R., 1994. A multi-scale approach. Technical Report 2338, INRIA.

Nielsen, M., Florack, L., Deriche, R., 1997. Regularization, scale-space, and edge detection filters. J. Math. Image Vision 66, 233–245.

Osher, S., Rudin, L., 1990. Feature-oriented image enhancement using shock filters. SIAM J. Numer. Anal. 27 (4), 919–940.

Perona, P., Malik, J., 1990. Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Machine Intell. 12 (7), 629–639.

Poyton, C., 1995. A guided tour of color space. In: SMPTE Advanced Television and Electronic Imaging Conference, San Fransisco, pp. 167–180.

Romeny, T.H. (Ed.), 1994. Geometry-Driven Diffusion in Computer Vision. Kluwer Academic Publishers, Dordrecht, the Netherlands.

Sapiro, G., Ringach, D., 1996. Anisotropic diffusion of multivalued images with applications to color filtering. IEEE Trans. Image Process. 5, 1582–1586.

Sethian, J., 1996. Level Set Methods. Cambridge University Press.

Siddiqi, K., Kimia, B.B., Shu, C.W., 1997. Geometric shock-capturing eno schemes for subpixel interpolation, computation, and curve evolution. Graphical Models Image Process. 59 (5), 278–301.

Sochen, N., Kimmel, R., Malladi, R., 1998. A general framework for low level vision. IEEE Trans. Image Process. 7, 310–318.

Tremblais, B., 2002. Décembre De la résolution numérique des EDP à l'extraction de caractéristiques linéiques dans les images: application à la détection multi-échelles d'un arbre vasculaire. Ph.D. thesis, Université de Poitiers.

Tremblais, B., Augereau, B., Leard, M., 2003. A multiscale approach for the extraction of vessels. In: Proceedings of SPIE Medical Imaging 2003, San Diego, CA, pp. 1331–1345.

Tschumperle, D., 2002. December PDE-based regularization of multivalued images and applications. Ph.D. thesis, University of Nice-ophia Antipolis.

Weickert, J., 1997. Coherence-enhancing diffusion of color images. VII NSPRIA 1, pp. 239–244.

Weickert, J., 1998. Anisotropic Diffusion in Image Processing. Teubner-Verlag, Stuttgart.

Weickert, J., ter Haar Romeny, B., Viergever, M., 1998. Efficient and reliable schemes for nonlinear diffusion filtering. IEEE Trans. Pattern Anal. Machine Intell. 7 (3), 398–410.

Whitaker, R., Gerig, G., 1994. Geometry-driven diffusion in computer vision. Kluwer. pp. 93–134 (Chapter 4).

Witkin, A.P., 1983. Scale-space filtering. In: AJCAI. Karlsruhe, pp. 1019–1022.

Zhao, D., Li, B., 1996. A new implementation of discrete multiscale filtering. In: ICIP96, pp. 383–386.