Topological Derivative: A Tool for Image Processing

I.Larrabide\textsuperscript{a}, R.A.Feij\~{o}o\textsuperscript{a}, A.A.Novotny\textsuperscript{a} and E.A.Taroco\textsuperscript{a}

\textsuperscript{a}LNCC - Laboratório Nacional de Computação Científica
Rua Getúlio Vargas 333 - Petrópolis - RJ - Brazil
email: \{nacho,feij,novotny,etam\}@lncc.br

Abstract

The increasing complexity in several fields of science and technology has motivated the use of techniques originally conceived in other areas of applications. An illustrative example of this is given by the Topological Derivative which quantifies the sensitivity of a problem when the domain under consideration is perturbed by changing its topology. This concept, initially conceived to deal with topology optimization problems, has also been successfully applied to inverse problems and material properties characterization. Our aim in this paper is to present another field of application for the topological derivative: image processing. An appropriate functional and a variational problem are associated to the cost endowed to a specific image processing application. Thus, the corresponding topological derivative can be used as an indicator function that leads, through a minimization process, to the processed image. We focus our attention on two image processing applications. In the first, the topological derivative is used in image restoration i.e. to restore an image that was somehow degraded (acquisition process, transmission, storage, etc.). Moreover, a novel fully discrete algorithm based on the topological derivative concept is presented. In the second application, we use the topological derivative to derive a "continuous" and a fully discrete novel image segmentation algorithm, i.e. for objects identification in an image. Finally and in order to show the performance of these algorithms, several numerical examples are also presented in this work.

Key words: Topological derivative, image restoration, image segmentation.

1 Introduction

As a consequence of the technological advance a variety of instruments and tools have been introduced in medicine. For instance, we can refer to medical imaging devices. More specifically, techniques like Computed Tomography
(CT), Magnetic Resonance Imaging (MRI), Single Photon Emission Tomography (SPECT), Positron Emission Tomography (PET) and Ultrasound (US) among others, have provided useful information (anatomical and functional) to specialists, no matter which is the area of interest (practical medicine, research, etc.). Therefore, the demand for tools to manipulate medical images has grown considerably since the appearance of these technologies. Also different issues have appeared in this field, and to recall some of them we can mention volume data visualization, image restoration, image segmentation, image registration, pattern recognition, etc.

The inherent complexity of this area has motivated interdisciplinary research and the use of techniques actually born in other areas into medical imaging, as is the case of image processing. In this work, we focus our attention in the Topological Derivative as a tool for image processing, specifically in image restoration and segmentation. As it is known, the topological derivative allows us to quantify the sensitivity of a problem when the domain under consideration is perturbed by changing its topology, for example by the introduction of an arbitrary shaped hole, an inclusion or a source term. Early work on this subject can be found in papers by Masmoudi, Sokolowsky and their co-workers (Masmoudi, 1987; Sokolowski and Żochowski, 1999). This derivative has been originally conceived as a tool to solve topology optimization problems. Nevertheless, this concept is wider and has shown interesting results when applied in inverse problems. See also Novotny (2003); Amstutz (2003); Céa et al. (2000); Eschenauer et al. (1994); Novotny et al. (2003); Masmoudi (2002) for applications of the topological derivative to the above equations topology optimization and inverse problems considering Navier, Laplace, Poisson, Helmholtz, Stokes and Navier-Stokes equations among others.

In this paper another field of application for the topological derivative namely image restoration and image segmentation is studied. In particular, the topological derivative has proven to be a powerful tool in this field of research (Auroux et al., 2006; Belaid et al., 2007; Hintermüller, 2005). We can also mention the work of He and Osher (2006), where a connection is made between level set and the topological derivative. More specifically, appropriated functionals and variational formulations associated to the cost endowed to image restoration and segmentation are proposed. The corresponding topological derivatives can be used as an indicator function that allow us to obtain, through a minimization process, the processed image. Following this approach, novel algorithms for image restoration and segmentation are proposed.

This article is organized as follows. In Section 2 the concept of topological derivative is introduced. The context of image restoration using the topological derivative is analyzed and a fully discrete algorithm that removes noise preserving details (edges) is proposed in Section 3. Furthermore several numerical applications are also presented in this section showing the effectiveness
of this new algorithm for image restoration. In Section 4, the image segmentation problem is tackled defining an appropriate functional and calculating its topological derivative. As shown in this section, “continuous” and fully discrete novel algorithms are also presented and used in segmenting several images with excellent performance.

2 The Topological Derivative

As mentioned above, the topological derivative \((D_T)\) allows us to quantify the sensitivity of a problem when the domain under consideration is perturbed, for example by the introduction of an arbitrary shaped hole, an inclusion or a source term at an arbitrary point.

Then, let \(\Omega\) be a bounded open set in \(\mathbb{R}^N (N = 2, 3)\) and \(\omega\) be a fixed bounded domain containing the origin. Let also \(\mathcal{J}(u) = \mathcal{J}(u(\Omega))\) be the cost functional to be minimized and \(u(\Omega)\) the solution of a variational problem defined over the domain \(\Omega\). For a small parameter \(\epsilon \geq 0\), let \(\Omega_\epsilon = \Omega - \omega_\epsilon\) be the perturbed domain defined by inserting a small hole at point \(\hat{x} \in \Omega\) given by \(\omega_\epsilon = \hat{x} + \epsilon \omega\). Let also \(u_\epsilon\) be the corresponding solution of the same variational problem but now defined over the perturbed domain \(\Omega_\epsilon\).

For small values of \(\epsilon\) the topological asymptotic expansion for \(\mathcal{J}_\epsilon(u_\epsilon)\) reads

\[
\mathcal{J}_\epsilon(u_\epsilon) = \mathcal{J}(u) + f(\epsilon)D_T(\hat{x}) + o(f(\epsilon)),
\]

where \(f(\epsilon)\) is a known positive function going to zero with \(\epsilon\) and \(D_T(\hat{x})\) is the topological derivative at point \(\hat{x}\). Since \(f\) is positive, by introducing a perturbation at any point \(\hat{x}\) where \(D_T\) is negative the cost function \(\mathcal{J}\) will be decreased. Then, \(D_T\) can be taken as an indicator function defining the best places where the perturbations could be introduced in order to reduce the value of the cost function. As will be shown in the next sections, this information can be used to develop fast algorithms for image restoration and segmentation.

Furthermore, dividing Eq. (1) by \(f(\epsilon)\) and after taking the limit \(\epsilon \to 0\) we obtain

\[
D_T(\hat{x}) = \lim_{\epsilon \to 0} \frac{\mathcal{J}_\epsilon(u_\epsilon) - \mathcal{J}(u)}{f(\epsilon)}.
\]

The numerator of the right hand side of the above expression is associated with the total variation of the cost function. This information will be used to derive a fully discrete algorithm for image restoration and segmentation.
3 Image Restoration via Topological Derivative

For visual analysis, clarity of details and object visibility are important features in medical images, but when the interest is to use more advanced image processing procedures like segmentation, a high signal-to-noise (SNR) is mandatory to reduce errors as much as possible.

Over the years different techniques have been studied to improve the SNR of a degraded image. These methods can be classified as those affecting acquisition time or pixel size (time averaging over pixels for repeated acquisitions, scanning with larger image elements), and methods not affecting these parameters (for example, improvement of the acquisition hardware and postprocessing the image data after acquisition, that is image processing techniques). The former techniques have the advantage of not affecting the acquisition process (Jain, 1989; Gonzalez and Woods, 2001). To this end different approaches have been proposed over the years. In particular, we can mention techniques based on the framework of variational approaches and energy minimization (Aubert and Kornprobst, 2002). The following energy was considered by these authors

\[ J(u) = \frac{1}{2} \int_{\Omega} (v - u)^2 d\Omega + \lambda \int_{\Omega} \phi(|\nabla u|) d\Omega. \]  

(3)

In the above expression the first term measures the misfit between the (degraded) image data \( v \) and the restored image \( u \) and the second term is associated to the smoothing process. The parameter \( \lambda \) is used to control the weight given to each term. Moreover, the Euler equation for this functional is given by

\[ -\lambda \text{div} \left\{ \frac{\phi'(|\nabla u|)}{|\nabla u|} \nabla u \right\} + u = v \]  

(4)

together with homogeneous Newmann boundary condition on \( \partial \Omega \). The authors also impose several restrictions over \( \phi \) in order to ensure satisfactory noise reduction, as well as existence and uniqueness of the solution \( u \) which minimizes the functional in Eq. (3). Another approach for image restoration is based on nonlinear diffusion methods. In this case the evolution of the (restored) image \( u = u(x, t) \) is processed by a nonlinear anisotropic diffusion governed by a partial differential equation (PDE)

\[ \frac{\partial u}{\partial t} = \text{div} (k(|\nabla u|) \nabla u) \]  

(5)

with initial condition \( u_0 = v \) and where \( k(|\nabla u|) \) is a tensor field called diffusivity and is designed to steer the direction of the diffusion process. When \( k(|\nabla u|) \) is reduced to a constant \( k_0 \) the linear isotropic diffusion method is obtained. In order to improve the outcome of linear isotropic diffusion filtering, specially edge preservation, in Perona and Malik (1990) is introduced a
monotonically decreasing function $\phi(|\nabla u|)$ instead of the constant $k_0$ such that the diffusion flux is reduced over edges. Despite the fact that a number of different alternatives for function $\phi$ can be found in the literature, this approach is still limited due to the isotropic nature of the diffusion equation. In fact only the intensity of the diffusion flux is controlled preserving noise along edges. This highly undesirable property of the nonlinear isotropic diffusion method is partly reduced when a nonlinear anisotropic diffusivity tensor is introduced (see Frangakis and Hegerl (2001)). In that case, the diffusivity tensor $k(|\nabla u|)$ is constructed from the eigenvectors and corresponding eigenvalues of the tensor $J = (\nabla u \otimes \nabla u)$ and also satisfying that $k(x) \rightarrow 0$ as $x \rightarrow \infty$. These conditions provokes denoising along edges and reduction of the diffusion in the presence of large gradients, for example, across boundaries. However, and since the diffusion across edges is not completely avoided, heuristic stop criteria must be introduced during the evolution in time otherwise details of the image may not be preserved. On the other hand and as will be shown in this paper, the methods based on the topological derivative also introduce anisotropic diffusion tensors with components taking a value $k_0$ in the direction diffusion is allowed (homogeneous regions of the image) and null otherwise (i.e., in the orthogonal direction of boundaries inside the image).

### 3.1 $D_T$ image restoration - Continuum approach

An image restoration algorithm based on the topological derivative was proposed by Masmoudi and his collaborators in (Belaid et al., 2005, 2007; Auroux et al., 2006). The cost functional associated to the edge detection adopted by these authors is given by

$$J(u) = \int_{\Omega} \nabla u \cdot \nabla u d\Omega$$

(6)

where $u \in H^1(\Omega)$ is the solution of the following variational equation

$$\int_{\Omega} (k_0 \nabla u \cdot \nabla \eta + u \eta) d\Omega = \int_{\Omega} v \eta d\Omega \quad \forall \eta \in H^1(\Omega)$$

(7)

and $v$ is the noisy image data. At any point $x \in \Omega$, the topological derivative of the cost functional (Eq. (6)) when the domain perturbation is characterized by a straight crack with unit normal vector $n$ orthogonal to the crack direction is obtained by these authors and is given by

$$D_T(x, n) = M n \cdot n = -\pi (\nabla u \otimes_s \nabla p + \nabla u \otimes \nabla u) n \cdot n$$

(8)

where $p \in H^1(\Omega)$ is the solution of the adjoint equation

$$\int_{\Omega} (k_0 \nabla p \cdot \nabla \eta + p \eta) d\Omega = \int_{\Omega} \frac{\partial J(u)}{\partial u} \eta d\Omega \quad \forall \eta \in H^1(\Omega).$$

(9)
Moreover, the orientation $\mathbf{n}$ of the crack at each point $\mathbf{x} \in \Omega$ is taken such that produces the minimum value for $D_T(\mathbf{x}, \mathbf{n})$. This is attained when $\mathbf{n}$ coincides with the eigenvector associated to the lowest eigenvalue, $\lambda_{min}$, of the symmetric tensor $M$. Using this information the authors propose that the restored image $\overline{\pi} \in H^1(\Omega)$ is the solution of the variational problem (isotropic/anisotropic diffusion)

$$ \int_{\Omega} (k \nabla \overline{\pi} \cdot \nabla \eta + \overline{\pi} \eta) \, d\Omega = \int_{\Omega} v \eta \, d\Omega \quad \forall \eta \in H^1(\Omega)$$

(10)

where the diffusivity tensor $k(\mathbf{x})$ is adopted as

- Isotropic diffusion based on $D_T$:
  - $k(\mathbf{x}) = \varepsilon \mathbf{I}$ if $D_T \leq \alpha D_{T,MIN}$;
  - $k(\mathbf{x}) = k_0 \mathbf{I}$ otherwise.

- Anisotropic diffusion based on $D_T$:
  - $k(\mathbf{x}) = \varepsilon (\mathbf{n} \otimes \mathbf{n}) + k_0 (\mathbf{t} \otimes \mathbf{t})$ if $D_T \leq \alpha D_{T,MIN}$;
  - $k(\mathbf{x}) = k_0 \mathbf{I}$ otherwise.

Then, to find the restored image $\overline{\pi}$ we must compute the solution of 3 systems of equations corresponding to the scalar fields $u$, $p$ and $\pi$.

At this point one question arises: is it possible to use the information given by the topological derivative in order to derive a fully discrete algorithm for image restoration without loss of quality when compared to the continuous approach?. The answer is affirmative and will be presented in the next section.

### 3.2 A fully discrete $D_T$ based approach

Typically, medical images as CT or MRI, are three dimensional data sets. For the sake of simplicity we consider only one slice of the whole image volume\(^1\), extending this idea to 3D is straightforward. The 2D image is characterized by a matrix of pixels, with an intensity associated to each one of them. More specifically, let us consider a two dimensional image $v$ given by a set of $M \times N$ pixels $s$. At each pixel $s$ the intensity of image $v$ will be denoted $v^s$. Then, any medical image belongs to the space $\mathcal{U}^d$

$$ \mathcal{U}^d := \{ v; v^s = \text{constant in } \omega^s, s = 1 \cdots M \times N \},$$

(11)

with $\Omega = \bigcup_s \omega^s$, being $\omega^s$ the domain of pixel $s$. The set $n^s$ of neighbors of a pixel $s$ was defined as the 4 pixels\(^2\) (see Fig. 1(a)) sharing a side with $s$.

\(^1\) A 3D medical image is composed of a stack of 2D images.

\(^2\) That is west, east, north and south of pixel $s$. 

6
The functional adopted for this fully discrete approach is taken as
\[
\mathcal{J}^d(u_t^s) = \sum_s \sum_{p \in n_s} k_s^{s,p} \Delta u_t^{s,p} \cdot \Delta u_t^{s,p},
\]
which could be interpreted as a discrete approximation of the square of the energy norm of the field \( u \). In the above expression the term \( k_s^{s,p} \) is the diffusion coefficient of pixel \( s \) with neighbor \( p \), \( n_s = \{ w, e, n, s \} \) stands for the neighbors of pixel \( s \) and \( \Delta u_t^{s,p} \) is defined as
\[
\Delta u_t^{s,p} = u_p^t - u_s^t.
\]
In this case, \( u_t^s \) is not the solution of a variational problem, but is explicitly computed using the following expression
\[
u_t^s(k^s) = u_t^{s-1} + \lambda \sum_{p \in n_s} k_s^{s,p} \Delta u_t^{s,p}
\]
where subindex \( t \geq 1 \) represents the iteration number, being \( u_0^s = v^s \) the intensity of the original image at pixel \( s \), \( k^s = \{ k_s^w, k_s^e, k_s^n, k_s^s \} \) denotes the set of diffusion coefficients associated to pixel \( s \), \( \lambda = \Delta t \) is the artificial time step size and with the notation \( u_t^s(k^s) \) we want to emphasize that \( u_t^s \) is an explicit function of the diffusion coefficients associated to pixel \( s \). Finally, Eq. (14) can be seen as a finite difference approximation for Eq. (5).

### 3.3 Topological Derivative Computation

In the continuum approach, the introduction of a null (or very small) diffusion coefficient was interpreted as the creation of a hole or a crack. This will also be the case for the discrete approach. The fact that \( k_s^{s,p} = 0 \) for pixel \( s \) with neighbor \( p \) is interpreted as introducing a crack along the edge they share (no diffusion occurs across this edge). Then, we want to find the best configuration for \( k^s \) that most preserve image details and remove noise when we compute \( u_t^s \). As in this case \( u_t^s \) is an explicit function of the set \( k^s \), we are able to compute the exact total variation of the cost functional for a given perturbation in \( k_s^{s,p} \). Moreover, let us denote by \( k_t^s \) the perturbed configuration on the diffusivity coefficients considered at pixel \( s \) then, the value of the cost functional when this perturbation is introduced is given by
\[
\mathcal{J}^d(u_t^s(k^s_t)) = \mathcal{J}^d(u_t^s(k^s)) + D_T(s, k_t^s)
\]
where \( D_T(s, k_t^s) \) represents the total variation of the functional due to the perturbation on the diffusivity coefficients at pixel \( s \) characterized by the set \( k_t^s \). As in the continuum approach, by introducing a perturbation at pixel \( s \) where \( D_T \) is negative the cost functional \( \mathcal{J}^d \) will be decreased. Then, using this
information we are able to select the best pixels for which this perturbation could be introduced.

![Pixel neighborhood and diffusion coefficient](image1)

![Neighborhood cases](image2)

(a) Pixel $s$ neighborhood and diffusion coefficient $k^s$.

(b) Neighborhood cases.

Fig. 1. Pixel neighborhood considered.

To this end, we assume $k^{s,p} \in \{0, k_0\}$. Then, the set of all possible configurations for $k^s$ can be defined as

$$C(s) := \{k^s = (k^{s,w}, k^{s,e}, k^{s,n}, k^{s,s}); k^{s,p} \in \{0, k_0\}, p = \{w, e, n, s\}\}$$

(16)

It is easy to notice that 16 different combinations are possible for $k^s$ (the possibilities are $k^{s,p} = 0$ or $k^{s,p} = k_0$, for each neighbor, so $2^4 = 16$ cases are possible). The case $k^{s,w} = k^{s,e} = k^{s,n} = k^{s,s} = 0$ is discarded because, for this case, the cost function assumes the value 0 (and the original image is not changed). The remaining 15 combinations are

- do not apply diffusion with one neighbor,
- do not apply diffusion with two neighbors that share a vertex,
- do not apply diffusion with three neighbors,
- apply diffusion in x-direction,
- apply diffusion in y-direction,
- apply diffusion in all the directions (isotropic diffusion).

The last combination corresponds to isotropic diffusion and is defined by $k^s_{iso} = \{k_0, k_0, k_0, k_0\}$. Since, the non perturbed configuration corresponds to the isotropic case we will denote by $C_\sigma$ the subset of $C$ containing the remaining $N_c = 14$ cases (Fig. 1(b)).

In order to compute the value of the $D_T$ for a specific pixel, we need to introduce a perturbation. This is done by changing at a particular pixel $s$ the set $k^s_{iso}$ by $k^s_{t} \in C_\sigma$. Hence, the cost function $J^d(\epsilon, u^t)$ takes the form

$$J^d(\epsilon, u^t) = J^d(\epsilon, u^t) - \sum_{p \in n^s} k^{s,p} \Delta u^t_{s,p} \cdot \Delta u^t_{s,p} + \sum_{p \in n^s} k^s_{t,p} \Delta \epsilon u^t_{s,p} \cdot \Delta \epsilon u^t_{s,p},$$

(17)
for \( u^*_t = u^*_t(k^*_s) \) and \( u^*_t = u^*_t(k^*_{iso}) \) calculated using Eq. (14) and \( \Delta_{\epsilon}u^*_t \).

Then, from Eqs. (15) and (17) the total variation of the cost function \( J^d \) due to the perturbation \( k^*_s \) can be written as

\[
D_T(s, k^*_s) = \sum_{p \in n^s} k^*_s \Delta_{\epsilon}u^*_t \cdot \Delta_{\epsilon}u^*_t - \sum_{p \in n^s} k^*_s \Delta u^*_t \cdot \Delta u^*_t. \tag{18}
\]

As in the continuum approach, the perturbation \( k^*_s \) (orientation of the crack at pixel \( s \)) is taken such that minimizes the value of \( D_T(s, k^*_s) \). Using this information we propose the following fully discrete image restoration algorithm based on the topological derivative (Algorithm 1).

**Algorithm 1** Image restoration based on fully discrete version of the topological derivative

**Require:** A 2D image \( v \in \mathcal{U} \) and the parameter \( \alpha \)

**Ensure:** The restored image \( \overline{v}^s \in \mathcal{U} \)

set \( u^*_0 = v^s, t = 1, \text{Stop} = \text{FALSE} \)

\( k^s = k^*_{iso}, s = 1..M \times N \)

while \( \text{Stop} = \text{FALSE} \) do

  for every pixel \( s \) do

    for every \( k^*_s \in C^s \) do

      compute \( D_T(s, k^*_s) \) following Eq. (18)

    end for

  end for

set \( D_T(s) = \min_{k^*_s \in C^s} \{ D_T(s, k^*_s), k^*_s \in C^s \} \)

for every pixel \( s \in \mathcal{M}_\alpha \) do

  set \( k^*_s = k^*_s \) the diffusivity coefficients associated to \( D_T(s) \)

end for

compute \( u^*_t(k^*_s) \) using Eq. (14).

if \( |J^d_\epsilon(u^*_t) - J^d_\epsilon(u^*_t-1)| > tol \) then

  \( t = t + 1 \)

else

  \( \overline{v}^s = u^*_t, s = 1, \ldots , M \times N, \text{Stop} = \text{TRUE} \)

end if

end while

The set \( \mathcal{M}_\alpha \) is defined as

\[
\mathcal{M}_\alpha := \{ D_T(s) : D_T(s) < 0 \}
\]

and \( D_T(s) \) is in the \( \alpha\% \) most negative values of the \( D_T \). \( \tag{19} \)

As in the continuum approach, in the above algorithm the \( \alpha \) parameter allows
us to control the values of the topological derivative that will produce changes in $k^i$. In the next section we present some results obtained using this technique.

### 3.4 Results

In order to show the performance of the proposed technique, in Figure 2 is presented the Lena image (size $256 \times 256$ pixels, polluted with $\sigma = 25$ White Gaussian Noise) processed with the nonlinear isotropic diffusion technique (Perona and Malik, 1990), the continuous $D_T$ image restoration technique and the fully discrete version proposed in this work. In this case the following two alternatives for the function $k(|\nabla u|)$ (nonlinear isotropic diffusion coefficient) where adopted

- Perona and Malik function (1990):
  \[
  k(x) = \frac{1}{1 + \frac{x^2}{\sigma^2}}
  \]

- Tukey error norm function (Black et. al. (1998)):
  \[
  k(x) = \begin{cases} 
  \frac{1}{2} \left(1 - \frac{x^2}{\sigma^2}\right)^2 & |x| < \sigma \\
  0 & \text{otherwise.}
  \end{cases}
  \]

As can be seen, the results for the proposed method ($D_T$ Discrete) are similar to the results for the other methods. Moreover, using the proposed fully discrete algorithm two different images were also processed. The first of these images (which will be referred as Case 1) is a phantom image with $200 \times 204$ pixels and composed of a series of ellipses of sizes and shapes similar to the structures that could be found in a real medical image (see Kak and Roberts (1986) for more details about this image). Five different regions are present in this image: bg, intensity = 0; $r_1$, intensity = 70; $r_2$, intensity = 120; $r_3$, intensity = 170; $r_4$, intensity = 210. The non polluted image is presented in Fig. 2(b). This image was polluted using additive White Gaussian Noise (WGN) with different standard deviations $\sigma$ ($\sigma = 18$ for Fig.3(c) and $\sigma = 25$ for Fig.3(d)).

These images were processed using different values for parameter $\alpha$ and the results are presented in Fig. 4. The first and second row correspond to the WGN $\sigma = 18$ and $\sigma = 25$ respectively. The columns correspond to the different values of parameter $\alpha$ (respectively and from left to right: 0.05, 0.10 and 0.15).

As we dispose of the original image, we are able to compute the error after the restoration process. To measure the quality of the results we use different
error measures. Let $v_0$ be the original unpolluted image and $\overline{u}$ be the restored image, then

- **Mean Square Error (MSE):**
  \[
  \text{MSE}(v_0, \overline{u}) = \frac{\sum_s (v_0^s - \overline{u}^s)^2}{N \times M} \tag{22}
  \]

- **Peak Signal-to-Noise Ratio (PSNR):**
  \[
  \text{PSNR}(v_0, \overline{u}) = 10 \times \log_{10} \left( \frac{\max(\overline{u}^s)^2}{\text{MSE}(v_0, \overline{u})} \right) \tag{23}
  \]

- **Signal-to-Noise Ratio (SNR):**
  \[
  \text{SNR}(v_0, \overline{u}) = 10 \times \log_{10} \left( \frac{\frac{1}{N \times M} \sum_s (\overline{u}^s)^2}{\text{MSE}(v_0, \overline{u})} \right) \tag{24}
  \]

Also, we are able to compute the mean $\mu(e)$ and standard deviation $\sigma(e)$ of the error $e(s)$, being:

\[
\begin{align*}
e(s) &= v_0^s - \overline{u}^s, \quad \text{\(s\in\mathbb{Z}^2\)} \tag{25} \\
\mu(e) &= \frac{\sum_s e(s)}{(N \times M)}, \quad \text{\(s\in\mathbb{Z}^2\)} \tag{26}
\end{align*}
\]
Fig. 3. Synthetic image - Case 1.

<table>
<thead>
<tr>
<th></th>
<th>PSNR</th>
<th>SNR</th>
<th>$\mu(e)$</th>
<th>$\sigma(e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>WGN with $\sigma = 18$</td>
<td>23.156610</td>
<td>61.855321</td>
<td>-0.250686</td>
<td>17.728266</td>
</tr>
<tr>
<td>Result for $\alpha = 0.05$</td>
<td>34.160666</td>
<td>72.859376</td>
<td>0.227721</td>
<td>4.989477</td>
</tr>
<tr>
<td>Result for $\alpha = 0.10$</td>
<td>34.030855</td>
<td>72.729566</td>
<td>0.257525</td>
<td>5.063332</td>
</tr>
<tr>
<td>Result for $\alpha = 0.15$</td>
<td>33.016007</td>
<td>71.714718</td>
<td>0.201765</td>
<td>5.694655</td>
</tr>
<tr>
<td>WGN with $\sigma = 25$</td>
<td>20.334903</td>
<td>59.033613</td>
<td>-0.908284</td>
<td>24.518690</td>
</tr>
<tr>
<td>Result for $\alpha = 0.05$</td>
<td>31.595765</td>
<td>70.294476</td>
<td>-0.378824</td>
<td>6.699753</td>
</tr>
<tr>
<td>Result for $\alpha = 0.10$</td>
<td>31.075751</td>
<td>69.774462</td>
<td>-0.401863</td>
<td>7.113128</td>
</tr>
<tr>
<td>Result for $\alpha = 0.15$</td>
<td>30.068495</td>
<td>68.767206</td>
<td>-0.407647</td>
<td>7.990077</td>
</tr>
</tbody>
</table>

Table I: Comparison table for Case 1.

\[
\sigma(e) = \sqrt{\frac{\sum_{s}[\mu(e) - e(s)]^2}{(N \times M)}}. \quad (27)
\]
From the results presented in Table 1, we can observe that this method actually removes noise from the polluted image. When we analyze the statistical indicators, we observe that the mean of the error was reduced and the standard deviation $\sigma$ was improved in $\approx 70\%$ in all cases.

As a second example we present another synthetic image with $256 \times 256$ pixels, in this case we also try to reproduce a medical image but with more complicated structures. Also the different structures present a smaller difference in their intensities. In this case 4 different regions are present: $bg$, intensity = 30; $r_1$, intensity = 50; $r_2$, intensity = 100; $r_3$, intensity = 150 (Fig. 5). As in the former case, this image was also polluted with different levels of WGN ($\sigma = 18$ for Fig.5(c) and $\sigma = 25$ for Fig.5(d)).

Following the same steps as in the first case, these images were processed using the proposed algorithm. Again, the same values for parameter $\alpha$ were tested. In Fig. 6 are presented the restored images for $\sigma = 18$ (first row) and $\sigma = 25$ (second row). In the columns are presented the images obtained for the different values of the parameter $\alpha$ (0.05, 0.10 and 0.15 from left to right respectively).

Again, the error was analyzed statistically and the results are presented in Table 2. In particular the statistical indicators point out an interesting improvement: the mean error remains close to 0 and the standard deviation $\sigma$
As mentioned above, the topological derivative is used in an optimization process where we are interested in finding the optimal distribution for the diffusivity coefficients such that, in the presence of noise, will preserve image...
Fig. 6. Results for synthetic image - Case 2.

(details eliminating noise. This process is stopped when the cost functional stabilizes and its variation in two consecutive iterations is smaller than a certain tolerance (denoted by tol). Fig. 7 presents the evolution of the cost functional as a function of iterations. As the number of iterations increase, the cost functional value stabilizes. This behavior also indicates that the restored image is stable with the number of iterations and no further degradation occurs as usually happens when nonlinear isotropic/anisotropic diffusion algorithms are used.

4 Image Segmentation via Topological Derivative

Medical imaging data provides information otherwise unavailable for clinical specialists to analyse. Quantitative information such as organ size and shape can be extracted from these images in order to support different activities, as surgical planning, disease diagnosis and monitoring among others. The first step in this process is to identify the different tissues and anatomical structures present in the image. This process, called segmentation, subdivides an image into its constituent regions or objects and must be accurate and repeatable in order to be clinically useful. On the other hand, the level to which the subdivision is done depends on the particular application.)
Classical image segmentation techniques are based on two basic pixel characteristics: discontinuities and similarities. Many of this classical techniques (e.g., multiple thresholds, region growing, morphologic filtering and others (Jain, 1989; Gonzalez and Woods, 2001)) have been applied to try to solve this problem with variable outcomes (Suri et al., 2001; Hohne et al., 1990). Such techniques tend to be unreliable when segmenting a structure that is surrounded by others with similar image intensity (e.g., low-contrast structures).

More sophisticated techniques, like Level Sets, use powerful numerical computations for tracking the evolution of moving surface fronts. These techniques are based on computing linear/nonlinear hyperbolic equation solutions for the appropriate equations of motion. An initial approximation of the solution (seed) evolves until it gets the limits of the region of interest. In this case user interaction is needed to introduce one or more seeds for the algorithm to evolve from (Sethian, 1999; Malladi and Sethian, 1997). Although this approach brings good results, the computational cost may become too high. A wide variety of works present the Active Contour (also called Snakes)
technique as the most robust for medical image segmentation (Xu et al., 2000, 1999; Li et al., 1995; Boscolo et al., 2002). With this technique good results are obtained, in particular for brain MRI segmentations. In this case input data must be pre-processed to extract spurious structures before the segmentation algorithm is started.

By means of Markov Random fields, in Held et al. (1997) and Zhang et al. (2001) fully automatic 3D-segmentation techniques especially designed for brain MRI images are described. These techniques capture three main spatial features of MRI images: non-parametric distribution of tissue intensities, neighborhood correlations and signal inhomogeneities. Once these fields are calculated (using suitable probabilistic models), an iterative optimization algorithm (Iterated Conditional Modes, Simulated Annealing, Expectation-Maximization, etc.) is used to recalculate them until convergence is achieved. Again, the limitation of this technique is its excessive computational cost.

In the next section we propose a novel image segmentation method based on the topological derivative. To this end an appropriate functional associated to the cost of a specific segmentation is proposed and the corresponding topological derivative is calculated in order to use it in a segmentation algorithm called $D_T$ segmentation continuum approach. As was done in image restoration, in this section we also develop a discrete version of this method. Finally, several image segmentations with different levels of noise are presented and compared to classical methods in order to show the performance and capabilities of these two new methods (for a more exhaustive comparison see Larrabide et al. (2006)).

4.1 $D_T$ image segmentation - Continuum approach

In this work the following cost functional is adopted (Larrabide et al., 2006)

$$\mathcal{J}(\varphi) = \frac{1}{2} \int_{\Omega} k \nabla \varphi \cdot \nabla \varphi \, d\Omega + \frac{1}{2} \int_{\Omega} (\varphi - (v - u))^2 \, d\Omega, \quad (28)$$

where the diffusivity tensor $k$ could be adopted as $k = k_0 I$ (isotropic and homogeneous tensor) or, tensor $k$ could be taken equal to the one obtained using the restoration algorithms proposed in Section 3. Also in the above expression $v \in \mathcal{V}$ is the image data, $u \in \mathcal{U}$ is the segmented image and $\mathcal{V}$ and $\mathcal{U}$ are defined respectively by

\[
\mathcal{V} := \{ w \in L^2(\Omega) : w^s \text{ constant at s-pixel/voxel level} \}, \quad (29)
\]

\[
\mathcal{U} := \{ u \in \mathcal{V} : u(x) \in \mathcal{C}, \forall x \in \Omega \}. \quad (30)
\]
The set $C$ characterizes the number of classes ($N_c$) for which the original image $v$ will be segmented, and is defined as

$$C := \{ c_i \in \mathbb{R} : i = 1...N_c \}, \quad (31)$$

where $c_i$ is the intensity associated to the $i_{th}$ class. Moreover, $\varphi$ is the solution of the following variational problem: Find $\varphi \in H^1(\Omega)$ such that

$$\begin{cases}
a(\varphi, \eta) = l(\eta) \quad \forall \eta \in H^1(\Omega), \\
\varphi \in H^1(\Omega)
\end{cases}$$

with $a(\cdot, \cdot) : H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}$ and $l(\cdot) : H^1(\Omega) \to \mathbb{R}$

$$a(\varphi, \eta) := \int_{\Omega} k\nabla \varphi \cdot \nabla \eta \, d\Omega + \int_{\Omega} \varphi \eta \, d\Omega, \quad (32)$$

and

$$l(\eta) := \beta \int_{\Omega} (v - u) \eta \, d\Omega. \quad (33)$$

To identify a class different alternatives can be used. The values defined for the classes will depend on the specific application of the segmentation. Therefore, we will obtain different results according to the criteria adopted to define the set of classes $C$ (for instance, mean intensity inside a region). Other image information can be used to determine this values. In the case of CT the brighter intensities represent bone and darker areas represent soft tissues as inner organs or muscles. This a priori information can be used to determine classes values.

Associated to $\varphi$ is defined the function $\varphi_\epsilon$ that is the solution of the same variational formulation but now defined on the perturbed problem. The perturbation is introduced by changing the segmented image $u$ with a new one $u_T$ which is identical to $u$ at every point of the domain $\Omega$ except in the small region $B_\epsilon$ centered at point $\hat{x} \in \Omega$. In $B_\epsilon$, $u_T$ assumes one of the values $c_i \in C$.

Then, the perturbed cost functional becomes

$$J_\epsilon(\varphi_\epsilon) = \frac{1}{2} \int_{\Omega} (\varphi_\epsilon - (v - u_T))^2 \, d\Omega + \frac{1}{2} \int_{\Omega} k\nabla \varphi_\epsilon \cdot \nabla \varphi_\epsilon \, d\Omega, \quad (34)$$

where the field $\varphi_\epsilon$ is the solution of the perturbed variational problem: Find $\varphi_\epsilon \in H^1(\Omega)$ such that

$$\begin{cases}
a(\varphi_\epsilon, \eta) = l_\epsilon(\eta) \quad \forall \eta \in H^1(\Omega), \\
\varphi_\epsilon \in H^1(\Omega)
\end{cases}$$

with $a(\cdot, \cdot) : H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}$ defined as before and $l_\epsilon(\cdot) : H^1(\Omega) \to \mathbb{R}$
defined as
\[ l_\epsilon(\eta) = \beta \int_\Omega (v - u_T) \eta \, d\Omega. \] (35)

The associated adjoint problem is given by: Find \( p_\epsilon \in H^1(\Omega) \) such that
\[
\begin{cases}
  a(p_\epsilon, \eta) = -\langle \frac{\partial}{\partial \varphi}, J_\epsilon(\varphi_\epsilon) \rangle & \forall \eta \in H^1(\Omega), \\
  p_\epsilon \in H^1(\Omega)
\end{cases}
\]

It is easy to verify that
\[ p_\epsilon = \frac{(1 - \beta)}{\beta} \varphi_\epsilon. \] (36)

Using topological shape sensitivity analysis, the topological derivative for this problem is computed (Larrabide et al., 2006). In this case, the \( D_T(\hat{x}) \) is given by
\[
D_T(\hat{x}) = \frac{1}{2}(c_i - u) [(\varphi(\hat{x}) - (v - u)) + (\varphi(\hat{x}) - (v - c_i)) + \\
+ 2(1 - \beta) \varphi(\hat{x})] \quad \forall \hat{x} \in \Omega.
\] (37)

and the topological asymptotic expansion reads
\[
J(\varphi_\epsilon) = J(\varphi) + f(\epsilon)D_T(\hat{x}) + o(f(\epsilon)).
\] (38)

Then, introducing a perturbation at points where \( D_T \) is negative the cost functional will be decreased. Using this information, the Algorithm 2 (continuum approach) for image segmentation is proposed.

**Algorithm 2 Image segmentation based on the topological derivative \( D_T \)**

**Require:** An input image \( v \in L^2(\Omega) \), the set of classes \( C(\Omega) \), an initial guess \( u_0 \in U \) and the parameters \( \beta \) and \( k \)

**Ensure:** The segmented image \( u^* \in L^2(\Omega) \)

while \( \exists D_T < 0 \) do
  solve the variational problem (32) to obtain \( \varphi_n \),
  compute \( D_T \) at each point \( \hat{x} \in \Omega \)
  if \( \min\{D_T(\hat{x})\}, \ i = 1, \ldots, N_c \} < 0 \) then
    \( u_i(\hat{x}) = c_i \)
  else
    make \( n = n + 1 \) and \( u_{n+1} = u \),
  end if
end while
\( u^* = u_{n+1} \)
4.2 A fully discrete $D_T$ based approach

In this work is proposed an alternative segmentation algorithm based on a simplification of the former idea. As we will show, for the discrete case it is not necessary to compute the field $\varphi$ to obtain the topological derivative.

In fact, taking $\beta = 0$ in Eq.(32) we obtain the trivial solution $\varphi \equiv 0$ for any segmented image $u \in U$. In this case also the cost functional $\mathcal{J}(\varphi)$ reduces to a functional $\mathcal{J}(\cdot) : U \rightarrow \mathbb{R}$ and given by

$$\mathcal{J}(u) = \int_{\Omega} (v - u)^2 d\Omega$$  \hspace{1cm} (39)

With this in mind, let us consider a two-dimensional image\(^3\) characterized by a set of $M \times N$ pixels $s$ with $\Omega = \cup_s \omega_s$. For each pixel $s$ we denote by $v^s$ and by $u^s$ the intensity of the function $v \in V^d$ and of the segmented image $u^s \in U^d$ respectively and where the sets $V^d \subset V$ and $U^d \subset U$ are defined as

$$V^d := \{ v; v^s \text{ constant in } \omega_s; s = 1...M \times N \} ,$$  \hspace{1cm} (40)

$$U^d := \{ u; u^s \in C; s = 1...M \times N \} .$$  \hspace{1cm} (41)

Furthermore, the following cost functional is adopted

$$\mathcal{F}^d(u^s) = \theta \mathcal{J}^d(u^s) + (1 - \theta) \mathcal{B}^d(u^s) , \text{ with } \theta \in (0, 1] \subset \mathbb{R} ,$$  \hspace{1cm} (42)

where the first term of the cost function $\mathcal{F}^d(u^s)$, denoted by $\mathcal{J}^d(u^s)$, is associated to the distance between the input image pixels intensities $v^s$ and the segmented image pixels intensities $u^s$. This term can be interpreted as the discrete version of the functional given by Eq.(39). The second term $\mathcal{B}^d(u^s)$ can be seen as a regularization functional that measures (and penalizes) the interface (boundary) length between different regions. These terms respectively take the form

$$\mathcal{J}^d(u^s) = \sum_s (v^s - u^s)^2 \quad \text{and} \quad \mathcal{B}^d(u^s) = \frac{1}{4n} \sum_s \sum_{p \in \{w,e,n,s\}} \chi(u^s, u^p) ,$$  \hspace{1cm} (43)

where $n = 2$ ($n = 3$) for 2-dimensional (3-dimensional) images and $\chi(u^s, u^p)$ is a characteristic function of the boundary that the pixel $s$ shares with the neighbor pixel $p$ and is defined as taking the value 1 (one) when $u^s \neq u^p$ and 0 otherwise. The $\theta$ parameter controls the contribution of each term ($\mathcal{J}^d(u^s)$ and $\mathcal{B}^d(u^s)$) to the cost function $\mathcal{F}^d$. When we are interested in classifying pixels of the image in homogeneous regions, the size of the interfaces between

\(^3\) Extending this idea to 3D is straightforward.
both regions will produce a higher value of the cost function. This is done by setting $\theta < 1$.

### 4.3 Topological Derivative Computation

In order to compute the topological derivative, we need to perturb the cost function at a given pixel $s$. This is done by changing the class of pixel $s$ from $u^s$ to $c_i$. In this way we obtain the perturbed cost functional $F^d(u^s_T)$ given by

$$F^d(u^s_T) = \theta J^d(u^s_T) + (1 - \theta) B^d(u^s_T),$$

where $J^d(u^s_T)$ and $B^d(u^s_T)$ can be written as

$$J^d(u^s_T) = J^d(u^s) - (v^s - u^s)^2 + (v^s - c_i)^2,$$

$$B^d(u^s_T) = B^d(u^s) - \frac{1}{4n} \sum_{p=\{w,e,n,s\}} (\chi(u^s, u^p) - \chi(c_i, u^p))$$

for $i = 1 \cdots Nc$. (45)

Then, the total variation of the functional $F^d$ will be denoted by $D_T$ and is given by the difference $J^d(u^s_T) - J^d(u^s)$, that is

$$D_T(u^s_T) = \theta \left[ (v^s - u^s_T)^2 - (v^s - u^s)^2 \right] + (1 - \theta) \frac{1}{4n} \sum_{p=\{w,e,n,s\}} [\chi(u^s, u^p) - \chi(u^s, u^p)]$$

for $u^s_T = c_i, i = 1 \cdots Nc.$ (46)

Moreover, at each pixel $s$ the perturbation $u^s_T$ will be selected such that produces the minimum value for the total variation at that pixel.

### 4.4 An Image Segmentation Algorithm

Using the information given by the total variation, a fully discrete topological derivative segmentation algorithm for 2D images (Algorithm 3) is proposed in this work. The algorithm inputs are the 2D image $v \in \mathcal{V}^d$ to be segmented and a set of classes in which image pixels will be classified. The algorithm output is $\bar{u}^s \in \mathcal{U}^d$. As mentioned before, the topological derivative can be used as a descent criterion in an optimization process. The sufficient local minimum condition is given by

$$D_T(u^s_T) \geq 0 \quad \forall s = 1, \ldots, N \times M$$

(47)
**Algorithm 3** Image segmentation algorithm based on the Topological Derivative

**Require:** A 2D image \( v \in \mathcal{V}^d \), the set of classes \( \mathcal{C} \), \( \theta \in (0, 1] \) and an initial guess \( u_0 \in \mathcal{U}^d \)

**Ensure:** The segmented image \( \bar{u}^* \in \mathcal{U}^d \)

normalize the image and classes values to \([0, 1]\)

while \( \exists T \) and \( s \) such that \( D_T(u^*_s) < 0 \)

for every pixel \( s \) do

for every class \( c_i \in \mathcal{C} \) do

compute \( D_T(u^*_s) \) following (Eq. 46)

end for

if \( \min\{D_T(u^*_s), T = 1, \ldots, N_c\} < 0 \) then

\( u^*_s = c_i \)

end if

end for

end while

\( \bar{u}^{*s} = u^*, s = 1 \ldots N \times M \)

Depending on the case, the stop condition could be substituted by a criteria associated to the behavior of the cost functional, i.e., if in two consecutive iterations, the cost function decreased less than a certain given tolerance, the algorithm stops.

In some cases (e.g., strong noise), changing all the pixels whose topological derivative is negative, an oscillating behavior was observed. This problem was solved using the same technique shown in Algorithm 1. That is, only a percent of the pixels whose topological derivative is negative is taking in each step. This simple technique stabilizes the algorithm and produces a convergent result in all the cases.

### 4.5 Experimental Results

The quality of the corresponding segmented images was quantified using the following metrics (Alonso et al., 2004; Zijdenbos et al., 1994)

- **Tanimoto index**: This index is calculated as

\[
I(A_1, A_2) = \frac{n(A_1 \cap A_2)}{n(A_1 \cup A_2)}.
\]  

- **Overlap index**: Is defined as (Zijdenbos et al., 1994):

\[
O(A_1, A_2) = 2 \cdot \frac{n(A_1 \cap A_2)}{n(A_1) + n(A_2)}.
\]
• **Mass Center Deviation (MCD):** Is computed as the distance from the exact center of mass of a particular region and the corresponding region in the segmented image. This computation includes all the pixels in the segmented image that were classified in that region.

• **Distance between borders (DBB):** This index is given by

\[
D(C_1, C_2) = \frac{\sum_{i=1}^{np_1} d(x_i^1, C_2) + \sum_{i=1}^{np_2} d(x_i^2, C_1)}{(np_1 + np_2)}
\]

where \(d(x, C)\) means the distance (in pixels) of the point \(x\) to the curve \(C\). \(C_1\) and \(C_2\) are the boundaries of the original and segmented region respectively, \(x_i^1\) and \(x_i^2\) denote arbitrary points on the boundaries \(C_1\) and \(C_2\), finally \(np_1\) and \(np_2\) are the number of points which characterize the respective boundaries.

The \(D_T\) methods (Continuous and Discrete) where compared to other well established segmentation methods (Bootstrap, Fuzzy C-Means and K-means) usually used to segment medical images. In Figure 8 are presented the comparisons for the mentioned methods and indexes. As can be seen, only K-means is able to keep up with the results of both \(D_T\) methods.

Furthermore, using the Case 1 and Case 2 images shown in Section 3 the behavior of the DT discrete segmentation method was analyzed for different alternatives in the noise roving process. The considered alternatives are: no processing (NPP), 3 iterations (ISPP3), 6 iterations (ISPP6) of isotropic smoothing (convolution with a \(5 \times 5\) gaussian kernel), and finally the \(D_T\) Restoration method was used to remove noise (DTPP, with \(\alpha = 0.05, 0.10, 0.15\)). The values used to characterize the classes were exactly the intensities of the corresponding regions in the original synthetic image. For Case 1 the classes were \(bg\), intensity = 30; \(c_1\), intensity = 70; \(c_2\), intensity = 120; \(c_3\), intensity = 170 and \(c_4\), intensity = 210 (Fig. 3). The corresponding segmented images are presented in Fig. 9.

As before, the quality of the segmentation was analyzed using the indexes presented above. In particular, Table 3 presents the results of the \(D_T\) Discrete for Case 1 image with the same level of noise yet considered (\(\sigma = 18\) and \(\sigma = 25\)). Moreover, Fig. 9 presents the segmented images. In all cases our method together with the DTPP noise removal has shown good results.

For Case 2 the classes were \(bg\), intensity = 30; \(c_1\), intensity = 50; \(c_2\), intensity = 100 and \(c_3\), intensity = 150. As in the case presented before, 6 different alternatives where studied (NPP, ISPP3, ISPP6 and DTPP, with \(\alpha = 0.05, 0.10, 0.15\)) and in Fig. 10 the obtained segmented images are presented. The quality of the segmentations using the mentioned indexes is presented in Table 4 and the obtained segmentations are shown in Fig. 10. Again, the \(D_T\) Discrete method
<table>
<thead>
<tr>
<th>Class</th>
<th>MCD</th>
<th>Overlap</th>
<th>Tanimoto</th>
<th>DBB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPP, σ = 18</td>
<td>bg</td>
<td>0.08224</td>
<td>0.995661</td>
<td>0.994204</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>4.231997</td>
<td>0.880682</td>
<td>0.814241</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.044466</td>
<td>0.598734</td>
<td>0.998277</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>4.544830</td>
<td>0.916031</td>
<td>0.861244</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.233750</td>
<td>0.933105</td>
<td>0.993874</td>
</tr>
<tr>
<td>ISPP3, σ = 18</td>
<td>bg</td>
<td>0.165457</td>
<td>0.879011</td>
<td>0.844935</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>9.761895</td>
<td>0.332429</td>
<td>0.209984</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.066097</td>
<td>0.927756</td>
<td>0.892451</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>19.784352</td>
<td>0.106437</td>
<td>0.056640</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.647279</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>ISPP6, σ = 18</td>
<td>bg</td>
<td>0.196921</td>
<td>0.821802</td>
<td>0.777525</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>10.422243</td>
<td>0.200105</td>
<td>0.113623</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.055267</td>
<td>0.898263</td>
<td>0.851011</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>20.920997</td>
<td>0.056616</td>
<td>0.029343</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>7.943816</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>DTPP, α = 0.05, σ = 18</td>
<td>bg</td>
<td>0.039018</td>
<td>0.998967</td>
<td>0.998623</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>2.445294</td>
<td>0.840691</td>
<td>0.784946</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.028159</td>
<td>0.966661</td>
<td>0.994840</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>8.909897</td>
<td>0.701595</td>
<td>0.574627</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.088697</td>
<td>0.981293</td>
<td>0.975212</td>
</tr>
<tr>
<td>DTPP α = 0.10, σ = 18</td>
<td>bg</td>
<td>0.072224</td>
<td>0.998450</td>
<td>0.997934</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>4.348604</td>
<td>0.925532</td>
<td>0.867110</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.015530</td>
<td>0.966873</td>
<td>0.995693</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>5.870391</td>
<td>0.878487</td>
<td>0.798206</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.195792</td>
<td>0.993330</td>
<td>0.991051</td>
</tr>
<tr>
<td>DTPP α = 0.15, σ = 18</td>
<td>bg</td>
<td>0.125568</td>
<td>0.993891</td>
<td>0.988305</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>5.543596</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.067099</td>
<td>0.993030</td>
<td>0.991763</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>6.392879</td>
<td>0.794872</td>
<td>0.661922</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.226113</td>
<td>0.984548</td>
<td>0.979353</td>
</tr>
<tr>
<td>NPP, σ = 25</td>
<td>bg</td>
<td>0.137028</td>
<td>0.988615</td>
<td>0.974009</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>10.198286</td>
<td>0.512255</td>
<td>0.360967</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.051609</td>
<td>0.980622</td>
<td>0.973561</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>15.840950</td>
<td>0.690763</td>
<td>0.548811</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.367261</td>
<td>0.977055</td>
<td>0.968236</td>
</tr>
<tr>
<td>ISPP3, σ = 25</td>
<td>bg</td>
<td>0.190733</td>
<td>0.874346</td>
<td>0.839197</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>10.190951</td>
<td>0.327804</td>
<td>0.197442</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.085705</td>
<td>0.928545</td>
<td>0.894469</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>19.163325</td>
<td>0.109000</td>
<td>0.058037</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.954842</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>ISPP6, σ = 25</td>
<td>bg</td>
<td>0.211291</td>
<td>0.841928</td>
<td>0.767575</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>10.874060</td>
<td>0.229084</td>
<td>0.131117</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.125370</td>
<td>0.904393</td>
<td>0.858668</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>19.937577</td>
<td>0.605117</td>
<td>0.062572</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>8.164306</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>DTPP α = 0.05, σ = 25</td>
<td>bg</td>
<td>0.093383</td>
<td>0.995172</td>
<td>0.993516</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>8.695588</td>
<td>0.692029</td>
<td>0.589506</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.119534</td>
<td>0.987511</td>
<td>0.982133</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>17.063908</td>
<td>0.459215</td>
<td>0.308943</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.191939</td>
<td>0.926130</td>
<td>0.903322</td>
</tr>
<tr>
<td>DTPP α = 0.10, σ = 25</td>
<td>bg</td>
<td>0.088234</td>
<td>0.994540</td>
<td>0.992600</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>7.928047</td>
<td>0.726968</td>
<td>0.609551</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.051243</td>
<td>0.988749</td>
<td>0.984102</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>13.530635</td>
<td>0.636183</td>
<td>0.486322</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.130446</td>
<td>0.965860</td>
<td>0.953611</td>
</tr>
<tr>
<td>DTPP α = 0.15, σ = 25</td>
<td>bg</td>
<td>0.099417</td>
<td>0.992988</td>
<td>0.990256</td>
</tr>
<tr>
<td></td>
<td>c1</td>
<td>6.612474</td>
<td>0.646654</td>
<td>0.512222</td>
</tr>
<tr>
<td></td>
<td>c2</td>
<td>0.056550</td>
<td>0.984811</td>
<td>0.978095</td>
</tr>
<tr>
<td></td>
<td>c3</td>
<td>17.314316</td>
<td>0.615070</td>
<td>0.461538</td>
</tr>
<tr>
<td></td>
<td>c4</td>
<td>0.321158</td>
<td>0.969213</td>
<td>0.958562</td>
</tr>
</tbody>
</table>

Table 3
Comparison table for segmentation results - Case 1.
performs better when the TDPP preprocessing is adopted.

Finally and for Case 1, the behavior of the cost functional given by Eq. (42) as a function of iterations is presented in Fig. 11 (WGN with \( \sigma = 18 \) and \( \sigma = 25 \) in Fig. 11(a) and 11(b) respectively).

5 Conclusions

In this paper was presented a new area of application for the topological derivative: image processing, in particular to image restoration and segmentation. Four algorithms \( D_T \) continuum approach and \( D_T \) fully discrete for image restoration and image segmentation are proposed. These algorithms are based on computing the topological derivative for an appropriate cost functional and using this information to characterize the perturbation to be inserted. This
Fig. 9. Results for synthetic image - Case 1.

The algorithms were tested with synthetic images degraded with different noise levels and compared with methods already used for these issues. In all cases the proposed algorithms are able to produce excellent results even for images degraded with strong noise where the others methods generally fails. Finally,
Fig. 10. Results for synthetic image - Case 2.

concerning the convergence issues for image processing algorithms based on topological derivative, we refer the reader to Belaid et al. (2005, 2007).
<table>
<thead>
<tr>
<th>Class</th>
<th>MCD</th>
<th>Overlap</th>
<th>Tanimoto</th>
<th>DBB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPP, $\sigma = 18$</td>
<td>bg</td>
<td>0.498795</td>
<td>0.925491</td>
<td>0.896859</td>
</tr>
<tr>
<td>c1</td>
<td>2.230574</td>
<td>0.782898</td>
<td>0.686635</td>
<td>11.357607</td>
</tr>
<tr>
<td>c2</td>
<td>0.070897</td>
<td>0.797276</td>
<td>0.994349</td>
<td>3.025453</td>
</tr>
<tr>
<td>c3</td>
<td>0.214670</td>
<td>0.972240</td>
<td>0.967284</td>
<td>0.079201</td>
</tr>
<tr>
<td>ISPP3, $\sigma = 18$</td>
<td>bg</td>
<td>0.447994</td>
<td>0.920598</td>
<td>0.896860</td>
</tr>
<tr>
<td>c1</td>
<td>1.717762</td>
<td>0.895116</td>
<td>0.817590</td>
<td>0.763943</td>
</tr>
<tr>
<td>c2</td>
<td>0.396328</td>
<td>0.941890</td>
<td>0.894251</td>
<td>0.519422</td>
</tr>
<tr>
<td>c3</td>
<td>0.635437</td>
<td>0.560402</td>
<td>0.487152</td>
<td>0.714340</td>
</tr>
<tr>
<td>ISPP6, $\sigma = 18$</td>
<td>bg</td>
<td>0.596886</td>
<td>0.891943</td>
<td>0.860933</td>
</tr>
<tr>
<td>c1</td>
<td>2.186333</td>
<td>0.856791</td>
<td>0.759035</td>
<td>0.987164</td>
</tr>
<tr>
<td>c2</td>
<td>0.698385</td>
<td>0.896412</td>
<td>0.815712</td>
<td>0.950743</td>
</tr>
<tr>
<td>c3</td>
<td>2.384734</td>
<td>0.000000</td>
<td>0.000000</td>
<td>3.709368</td>
</tr>
<tr>
<td>DTPP, $\alpha = 0.05, \sigma = 18$</td>
<td>bg</td>
<td>0.028543</td>
<td>0.990160</td>
<td>0.986867</td>
</tr>
<tr>
<td>c1</td>
<td>0.204254</td>
<td>0.961343</td>
<td>0.937861</td>
<td>5.176868</td>
</tr>
<tr>
<td>c2</td>
<td>0.377372</td>
<td>0.980026</td>
<td>0.969896</td>
<td>0.360938</td>
</tr>
<tr>
<td>c3</td>
<td>1.234926</td>
<td>0.943759</td>
<td>0.920871</td>
<td>0.219132</td>
</tr>
<tr>
<td>DTPP $\alpha = 0.10, \sigma = 18$</td>
<td>bg</td>
<td>0.041924</td>
<td>0.987798</td>
<td>0.982859</td>
</tr>
<tr>
<td>c1</td>
<td>0.148215</td>
<td>0.949456</td>
<td>0.921411</td>
<td>5.628081</td>
</tr>
<tr>
<td>c2</td>
<td>0.311331</td>
<td>0.976744</td>
<td>0.965784</td>
<td>0.526392</td>
</tr>
<tr>
<td>c3</td>
<td>0.851640</td>
<td>0.949262</td>
<td>0.924965</td>
<td>0.419459</td>
</tr>
<tr>
<td>DTPP $\alpha = 0.15, \sigma = 18$</td>
<td>bg</td>
<td>0.101573</td>
<td>0.983861</td>
<td>0.977126</td>
</tr>
<tr>
<td>c1</td>
<td>0.707214</td>
<td>0.927620</td>
<td>0.880621</td>
<td>6.842766</td>
</tr>
<tr>
<td>c2</td>
<td>0.098943</td>
<td>0.973103</td>
<td>0.961079</td>
<td>0.853142</td>
</tr>
<tr>
<td>c3</td>
<td>0.172514</td>
<td>0.953219</td>
<td>0.929760</td>
<td>0.651386</td>
</tr>
<tr>
<td>NPP, $\sigma = 25$</td>
<td>bg</td>
<td>1.178724</td>
<td>0.847396</td>
<td>0.796283</td>
</tr>
<tr>
<td>c1</td>
<td>4.960136</td>
<td>0.511747</td>
<td>0.395506</td>
<td>13.859100</td>
</tr>
<tr>
<td>c2</td>
<td>0.192516</td>
<td>0.946047</td>
<td>0.922430</td>
<td>2.105597</td>
</tr>
<tr>
<td>c3</td>
<td>0.530470</td>
<td>0.934613</td>
<td>0.898838</td>
<td>1.429791</td>
</tr>
<tr>
<td>ISPP3, $\sigma = 25$</td>
<td>bg</td>
<td>0.526817</td>
<td>0.914074</td>
<td>0.886571</td>
</tr>
<tr>
<td>c1</td>
<td>1.884506</td>
<td>0.877452</td>
<td>0.791894</td>
<td>1.310068</td>
</tr>
<tr>
<td>c2</td>
<td>0.452387</td>
<td>0.931905</td>
<td>0.881464</td>
<td>0.554374</td>
</tr>
<tr>
<td>c3</td>
<td>1.036228</td>
<td>0.553777</td>
<td>0.478422</td>
<td>0.769443</td>
</tr>
<tr>
<td>ISPP6, $\sigma = 25$</td>
<td>bg</td>
<td>0.660702</td>
<td>0.882783</td>
<td>0.849587</td>
</tr>
<tr>
<td>c1</td>
<td>2.187299</td>
<td>0.830688</td>
<td>0.725929</td>
<td>1.127844</td>
</tr>
<tr>
<td>c2</td>
<td>0.895636</td>
<td>0.891374</td>
<td>0.809281</td>
<td>0.997886</td>
</tr>
<tr>
<td>c3</td>
<td>4.484137</td>
<td>0.000000</td>
<td>0.000000</td>
<td>2.595335</td>
</tr>
<tr>
<td>DTPP $\alpha = 0.05, \sigma = 25$</td>
<td>bg</td>
<td>0.112122</td>
<td>0.986612</td>
<td>0.980894</td>
</tr>
<tr>
<td>c1</td>
<td>0.561304</td>
<td>0.942178</td>
<td>0.909935</td>
<td>7.348805</td>
</tr>
<tr>
<td>c2</td>
<td>0.431706</td>
<td>0.963539</td>
<td>0.945628</td>
<td>1.085454</td>
</tr>
<tr>
<td>c3</td>
<td>1.242424</td>
<td>0.921996</td>
<td>0.887155</td>
<td>0.403577</td>
</tr>
<tr>
<td>DTPP $\alpha = 0.10, \sigma = 25$</td>
<td>bg</td>
<td>0.084423</td>
<td>0.982120</td>
<td>0.975029</td>
</tr>
<tr>
<td>c1</td>
<td>0.675668</td>
<td>0.920735</td>
<td>0.879487</td>
<td>8.488288</td>
</tr>
<tr>
<td>c2</td>
<td>0.568348</td>
<td>0.949385</td>
<td>0.927024</td>
<td>1.770457</td>
</tr>
<tr>
<td>c3</td>
<td>1.148880</td>
<td>0.912646</td>
<td>0.872424</td>
<td>0.872424</td>
</tr>
<tr>
<td>DTPP $\alpha = 0.15, \sigma = 25$</td>
<td>bg</td>
<td>0.084728</td>
<td>0.976656</td>
<td>0.967037</td>
</tr>
<tr>
<td>c1</td>
<td>0.475479</td>
<td>0.856048</td>
<td>0.831336</td>
<td>9.957039</td>
</tr>
<tr>
<td>c2</td>
<td>0.509805</td>
<td>0.934349</td>
<td>0.905783</td>
<td>2.551882</td>
</tr>
<tr>
<td>c3</td>
<td>1.186147</td>
<td>0.901604</td>
<td>0.854825</td>
<td>1.333364</td>
</tr>
</tbody>
</table>

Table 4
Comparison table for segmentation results - Case 2.

6 ACKNOWLEDGMENTS

This research was partly supported by the brazilian agencies CNPq/FAPERJ-PRONEX, under Grant E-26/171.199/2003. Ignacio Larrabide was partly supported by the brazilian agency CNPq (141336/2003-0). The support of these
Fig. 11. Cost function for Case 1 with $\sigma = 18$ and 25, with $\alpha = 0.05$ (cost function value vs. iteration number).

agencies is greatly appreciated.

References


Perona, P. and Malik, J. (1990). Scale-space and edge detection using


