

# Numerical implementation of the Ambrosio-Tortorelli functional using discrete calculus and application to image restoration and inpainting <sup>\*</sup>

Marion Foare<sup>1</sup>, Jacques-Olivier Lachaud<sup>1</sup>, Hugues Talbot<sup>2</sup>

<sup>1</sup> Laboratoire de Mathématiques  
Université Savoie Mont Blanc, F-73376, France  
{marion.foare,jacques-olivier.lachaud}@univ-smb.fr

<sup>2</sup> Laboratoire d'Informatique Gaspard-Monge  
Université Paris-Est – ESIEE  
hugues.talbot@univ-paris-est.fr

**Abstract.** The Mumford-Shah (MS) functional is one of the most influential variational model in image segmentation, restoration, and cartooning. Difficult to solve, the Ambrosio-Tortorelli (AT) functional is of particular interest, because minimizers of AT can be shown to converge to a minimizer of MS. This paper takes an interest in a new method for numerically solving the AT model [11]. This method formulates the AT functional in a discrete calculus setting, and by this way is able to capture the set of discontinuities as a one-dimensional set. It is also shown that this model is competitive with total variation restoration methods. We present here the discrete AT models in details, and compare its merit with recent convex relaxations of AT and MS functionals. We also examine the potential of this model for inpainting, and describe its implementation in the DGtal library, an open-source project.

**Keywords:** image denoising and restoration, Mumford-Shah functional, variational model, optimization, inverse problems, discrete calculus, Ambrosio-Tortorelli functional, image inpainting.

## 1 Introduction

This paper takes an interest in a new numerical method to solve the Ambrosio-Tortorelli (AT) functional [1]. This functional is a well known non-convex relaxation of the famous Mumford-Shah (MS) functional [17], which has many applications in image segmentation [2, 6, 18], restoration [2, 4, 5, 8, 12, 19, 23], cartooning [20] or inpainting [10]. This new method for solving AT redefines this functional into a discrete calculus setting and is presented in details in [11].

---

<sup>\*</sup> This work has been partly funded by CoMEDiC ANR-15-CE40-0006 research grant.

*Contributions.* The present paper is a companion to the original paper [11], and complements it on several points. First of all, the possible formulations of AT in discrete calculus are discussed and compared in more details. Secondly, we examine other convex relaxations of the Ambrosio-Tortorelli and Mumford-Shah functionals (Kee and Kim [15], Strelakovski and Cremers [20]) and we compare our approach to them. Third, we study the influence of parameters in the method, as well as the possible scale spaces induced by the model. Fourth, we test the possible application of our discrete AT model to inpainting problems and discuss its potential and drawbacks. Last, we describe its implementation in the DGtal library [22], which is an open source C++ library for image and geometry processing of digital data.

*Outline.* The paper is organized as follows. We first briefly introduce the MS and AT functionals as well as the many variants and discretizations proposed in the literature (Section 2). We then present our discrete formulations of the AT functional, which is expressed in a discrete calculus framework, and the algorithm for its numerical optimization (Section 3). Our approach is compared to several state-of-the-art convex relaxations of MS or AT in Section 4. Parameters and their associated scale spaces are discussed in Section 5. The potential of our discrete AT for inpainting is encompassed in Section 6. Section 7 concludes by describing the implementation of our AT models in the DGtal library.

## 2 The Mumford-Shah and Ambrosio-Tortorelli functionals

The Mumford-Shah (MS) model was introduced in [17] to solve image segmentation and restoration problems. This model was seminal to many important contributions in image processing and analysis, such as the Rudin-Osher-Fatemi (ROF) restoration method, the total variation (TV) denoising, Chan-Vese or Boykov-Jolly segmentation methods. The main idea is to see the input grey-level image as a function  $g \in L^\infty(\Omega)$  where  $\Omega$  is some domain of  $\mathbb{R}^2$ , and to seek the solution as a piecewise smooth function  $u$  that approaches  $g$ . The loci of discontinuities  $K$  is also a variable of the problem. Another strong idea is that the “size” of  $K$  is penalized. So the MS model is the optimal solution  $(K, u)$  to the following functional:

$$\mathcal{MS}(K, u) = \alpha \int_{\Omega \setminus K} |u - g|^2 \, dx + \int_{\Omega \setminus K} |\nabla u|^2 \, dx + \lambda \mathcal{H}^1(K \cap \Omega) \quad (1)$$

where  $\alpha, \lambda > 0$  and  $\mathcal{H}^1$  denotes the 1-dimensional Hausdorff measure. We recognize in the first term the  $L_2$  approximation to the input image, the second term requires the smoothness of  $u$ , while the last term penalizes the length of discontinuities  $K$ .

The MS functional is difficult to solve in this form, so many relaxations have been proposed to overcome this issue. We take an interest in the Ambrosio-

Tortorelli relaxation [1], defined as:

$$\text{AT}_\varepsilon(u, v) = \int_{\Omega} \alpha |u - g|^2 + v^2 |\nabla u|^2 + \lambda \varepsilon |\nabla v|^2 + \frac{\lambda}{4\varepsilon} |1 - v|^2 \, dx, \quad (2)$$

for functions  $u, v \in W^{1,2}(\Omega)$  with  $0 \leq v \leq 1$ .

In (2), function  $u$  is a smooth approximation of the input image  $g$ . Function  $v$  is a smooth approximation of the set of discontinuities, and takes value close to 0 in this set, while being close to 1 outside discontinuities. A remarkable property of this functional is that it  $\Gamma$ -converges to (a relaxation of) the MS functional as  $\varepsilon$  tends to 0 [1]. The intuition is that a large  $\varepsilon$  induces a solution with a fuzzy set of discontinuities, which is then progressively narrowed to the crisp 1-dimensional set of discontinuities as  $\varepsilon$  goes to 0.

However the minimization of (2) with standard discretization schemes such as finite differences is not possible. Indeed, to achieve convergence, parameters  $\varepsilon$ , the grid-step  $h$  and the ratio  $\frac{\varepsilon}{h}$  must all tend to zero [4]. So in practice we set  $\varepsilon \approx 5h$  but the resulting set of discontinuities is thick and fuzzy (sometime more than 6 pixels wide), and the restoration is very poor in these regions. To overcome this issue, Bourdin and Chambolle [4] proposed a finite element method with adaptive domain refinement around discontinuities. It will be shown later that this method is still not satisfactory, despite the added complexity. These difficulties explain why the AT model is not a popular approach in image processing, and why TV, ROF, or graph cuts methods have been heavily used instead.

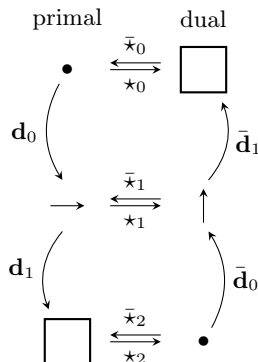
For more details about other similar variational models for image processing and more background about related works, we refer the reader to [11].

### 3 Discrete calculus formulations of AT functional

In [11], a different formulation of the AT functional was proposed, which retains its main characteristics (piecewise smoothness, penalization of the length of the discontinuities), while being able to capture thin discontinuities. This reformulation uses the setting of discrete exterior calculus (DEC), as described in the context of computer graphics in [7] and image analysis in [13]. The advantage of this setting is that one can control easily where the variables are defined, and where they are coupled. In particular, the function  $v$  does not become identically one even when  $\varepsilon$  tends to zero.

*Discrete exterior calculus.* The idea is to decompose the image domain  $\Omega$  into a cell complex  $K$ . Here, the faces of  $K$  are simply the pixels of the image, the edges of  $K$  are the sides shared by two pixels, while the vertices of  $K$  are the those shared by four pixels. For technical reasons, the dual complex  $\bar{K}$  of  $K$  is also needed and is defined in the usual way: a vertex of  $K$  is thus associated one-to-one to a face of  $\bar{K}$ , etc. A *discrete  $k$ -dimensional form* is just a map that associates a scalar to a  $k$ -dimensional cell.<sup>3</sup> If the number of  $k$ -dimensional cells

<sup>3</sup> Although discrete forms can be defined without reference to the continuous setting, one can see them as the integration of  $k$ -dimensional differential form over a  $k$ -cell, resulting in a scalar per  $k$ -cell.



**Fig. 1.** Primal and dual complex and common Discrete Exterior Calculus (DEC) operators.

of  $K$  is denoted by  $n_k$ , then a discrete  $k$ -form is simply represented by a column vector of size  $n_k \times 1$ .

We then define the usual linear operators between  $k$ -forms (see Fig. 1). We denote by  $\mathbf{d}_k$  and  $\bar{\mathbf{d}}_k$  the standard discrete exterior primal and dual derivative operators. The derivative operator  $\mathbf{d}_0$  is the oriented vertex-edge incidence matrix of  $K$ , and has size  $n_1 \times n_0$ . Since the complex  $K$  is the regular grid with edge size 1, the dual derivative  $\bar{\mathbf{d}}_1$  is the transpose of  $\mathbf{d}_0$ . Similarly, the primal derivative  $\mathbf{d}_1$  is the oriented edge-face incidence matrix of  $K$ , of size  $n_2 \times n_1$ , and the dual derivative  $\bar{\mathbf{d}}_0$  is its transpose.

The Hodge star is an operator that maps a differential form onto its complementary differential form. Discrete Hodge star operators  $\star$  sends  $k$ -forms of the primal complex  $K$  onto  $n - k$ -forms of the dual complex  $\bar{K}$  (see again Fig. 1). In our setting, these operators reduce to identity operations except for a change of sign in some cases, and they also change the meaning of vectors. For instance, if  $f$  is a 0-form on the vertices of  $K$ ,  $\star f$  is a dual 2-form defined on the faces of  $\bar{K}$ . Clearly, it is also a vector of size  $n_0 \times 1$  since there are  $n_0$  faces in  $\bar{K}$ .

We define  $\mathbf{M}_{01}$  the matrix which transforms a 0-form into a 1-form by averaging the values on the two edge extremities, i.e.  $\mathbf{M}_{01} = \frac{1}{2}|\mathbf{d}_0|$ . Moreover, we use the edge laplacian defined in [13] by  $\bar{\star}\bar{\mathbf{d}}_0\star\mathbf{d}_1 + \mathbf{d}_0\bar{\star}\bar{\mathbf{d}}_1\star$ .

*Discrete formulations of AT.* We first set  $u$  and  $g$  to live on the faces and  $v$  to live on the vertices and edges. Pixels are faces, so functions  $u$  and  $g$  are 2-forms since they represent the gray levels of each pixel. On the contrary,  $v$  is an estimation of the set of discontinuities of  $u$ , and should be of null Hausdorff-1 measure when  $\varepsilon \rightarrow 0$ . Thus we set  $v$  in-between cells of non null measure, so in this case on vertices as a 0-form, and on edges by averaging with  $\mathbf{M}_{01}$ . We call this formulation  $\text{AT}_\varepsilon^{2,0}$ . Looking at (2), the DEC reformulation is straightforward, except for the second term, where  $v$  is a 0-form and  $\nabla u$  a 1-form. Hence we use

matrix  $\mathbf{M}_{01}$  to transport the 0-form  $v$  onto edges by simple averaging:

$$\begin{aligned} \text{AT}_\varepsilon^{2,0}(u, v) = & \alpha \langle u - g, u - g \rangle_2 + \langle \mathbf{M}_{01}v \odot \bar{\star} \bar{\mathbf{d}}_0 \star u, \mathbf{M}_{01}v \odot \bar{\star} \bar{\mathbf{d}}_0 \star u \rangle_1 \\ & + \lambda \varepsilon \langle \mathbf{d}_0 v, \mathbf{d}_0 v \rangle_1 + \frac{\lambda}{4\varepsilon} \langle 1 - v, 1 - v \rangle_0. \end{aligned} \quad (3)$$

where  $\odot$  denotes the point-wise multiplication.

A second possibility is to define  $u$  and  $g$  on the vertices and  $v$  on the edges. We denote this formulation  $\text{AT}_\varepsilon^{0,1}$ . Contrary to the previous formulation, the gray levels are seen as point mass on the center of pixels, so that functions  $u$  and  $g$  are both 0-forms. An alternative interpretation is to say that pixel values are dual 2-forms, while  $v$  is a dual 1-form in between  $u$ . It follows:

$$\begin{aligned} \text{AT}_\varepsilon^{0,1}(u, v) = & \alpha \langle u - g, u - g \rangle_0 + \langle v \odot \mathbf{d}_0 u, v \odot \mathbf{d}_0 u \rangle_1 \\ & + \lambda \varepsilon \langle (\mathbf{d}_1 + \bar{\star} \bar{\mathbf{d}}_1 \star) v, (\mathbf{d}_1 + \bar{\star} \bar{\mathbf{d}}_1 \star) v \rangle_1 + \frac{\lambda}{4\varepsilon} \langle 1 - v, 1 - v \rangle_1. \end{aligned} \quad (4)$$

We further extend (3) and (4) to the vectorial case by associating to each component of the input image  $n$  forms  $\{g_1, \dots, g_n\}$ , and the corresponding forms  $\{u_1, \dots, u_n\}$ , and summing over the coordinates. In this paper, we will use  $n = 1$  for gray-level images and  $n = 3$  for RGB color images.

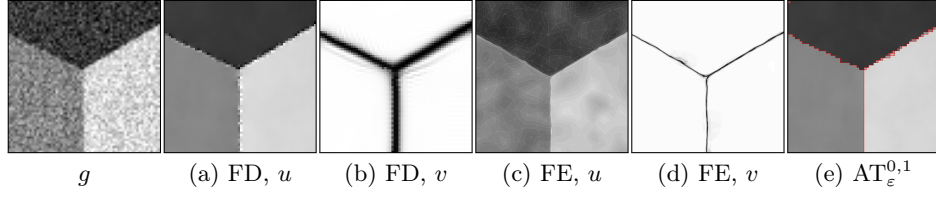
*Optimization.* The functionals  $\text{AT}_\varepsilon^{2,0}$  and  $\text{AT}_\varepsilon^{0,1}$  are both sums of quadratic terms, independently, but not simultaneously, convex in  $u$  and  $v$ . They must have null derivative at optimum. We thus propose to alternatively solve for  $u$ , then  $v$ . The derivatives can be given explicitly as linear systems. To simplify notations, let  $\mathbf{A} := \mathbf{d}_0$ ,  $\mathbf{B} := \mathbf{d}_1$ ,  $\mathbf{A}' := \bar{\star} \bar{\mathbf{d}}_1 \star$  and  $\mathbf{B}' := \bar{\star} \bar{\mathbf{d}}_0 \star$ . It is worth to note  $\mathbf{A}'$  and  $\mathbf{B}'$  are the respective transpose of  $\mathbf{A}$  of  $\mathbf{B}$ , except on the boundary of the image. We get at optimum:

$$\left\{ \begin{array}{l} \left[ \alpha \mathbf{Id} - \mathbf{B}'^\top \text{diag}(\mathbf{M}_{01}v)^2 \mathbf{B}' \right] u = \alpha g, \\ \left[ \frac{\lambda}{4\varepsilon} \mathbf{Id} + \lambda \varepsilon \mathbf{A}^\top \mathbf{A} + \mathbf{M}_{01}^\top \text{diag}(\mathbf{B}'u)^2 \mathbf{M}_{01} \right] v = \frac{\lambda}{4\varepsilon} \mathbf{1}. \end{array} \right. \quad (5)$$

for the derivative of  $\text{AT}_\varepsilon^{2,0}$ , and

$$\left\{ \begin{array}{l} \left[ \alpha \mathbf{Id} - \mathbf{A}^\top \text{diag}(v)^2 \mathbf{A} \right] u = \alpha g, \\ \left[ \frac{\lambda}{4\varepsilon} \mathbf{Id} + \lambda \varepsilon (\mathbf{A}'^\top \mathbf{A}' + \mathbf{B}^\top \mathbf{B}) + \text{diag}(\mathbf{A}u)^2 \right] v = \frac{\lambda}{4\varepsilon} \mathbf{1}. \end{array} \right. \quad (6)$$

for the derivative of  $\text{AT}_\varepsilon^{0,1}$ . Since all matrices are symmetric, definite and positive, we use a Cholesky factorization to solve alternatively the two equations of (5) (resp. (6)). Because of the  $\mathbf{Id}$  additive term, the left-hand side is full rank, yielding a unique minimum at each iteration. It is a known result in convex analysis linked to block coordinate descent algorithms [3, Prop. 2.7.1], that these iterations must converge to a stationary point of  $\text{AT}_\varepsilon^{2,0}$  ( resp.  $\text{AT}_\varepsilon^{0,1}$ ). In the optimization process, we start with a large enough  $\varepsilon_1$ , in order to capture the



**Fig. 2.** Minimization of AT with finite differences (FD), finite elements (FE) from [4] and our approach  $AT_\varepsilon^{2,0}$  on a triple point image with PSNR= 20.3dB.

discontinuities, and after convergence for a given  $\varepsilon$ , we decrease it by dividing it by  $\varepsilon_r$ , until  $\varepsilon = \varepsilon_2$ . We denote this process by  $\varepsilon = \varepsilon_1 \searrow \varepsilon_2$ . The main loop stops when  $v$  reaches stability. Fig. 2 illustrates that our AT model are able to capture discontinuities as a 1-dimensional set, since  $v$  is defined on 0- and 1-dimensional cells.

```

OPTIMIZE-AT(  $(g_1, \dots, g_n)$ : 2-forms (resp. 0-forms),  $(\alpha, \lambda, \varepsilon_1, \varepsilon_2, \varepsilon_r)$ : reals );
Var  $(u_1, \dots, u_n)$ : 2-forms (resp. 0-forms),  $(v, v')$ : 0-forms (resp. 1-forms),  $\varepsilon$ : real
;
begin
  foreach  $i \in \{1, \dots, n\}$  do  $u_i \leftarrow g_i$ ;
   $v \leftarrow \mathbf{1}$ ,  $\varepsilon \leftarrow \varepsilon_1$ ;
  while  $\varepsilon \geq \varepsilon_2$  do
1   repeat
       $v' \leftarrow v$ ;
      Solve 1st line of (5) (resp. (6)) for all  $u_i$ ;
      Solve 2nd line of (5) (resp. (6)) for  $v$ ;
      until  $\|v - v'\|$  is small;
       $\varepsilon \leftarrow \varepsilon / \varepsilon_r$  ;
  end
  return  $((u_1, \dots, u_n), v)$  ;
end

```

*Limit cases.* As is the case with many image restoration methods, our approach assumes stationary noise and blur. Some spatially variant noise occurs in nature, for instance Poisson noise [21]. As is illustrated on Fig. 3, when the noise is spatially variant, suboptimal results are achieved. This is due to the parameters  $\alpha$  and  $\lambda$  being constant for the entire image. For future work, we plan to let  $\alpha$  and  $\lambda$  depend on  $x$  and thus choose locally the optimal pair  $(\alpha, \lambda)$ , and to adapt AT to use data fidelity terms that are appropriate for some non-stationary noise models [14].



**Fig. 3.** In this figure we have added a spatially-variant noise to the Mandrill image. In this case it is not possible to find an optimal  $(\alpha, \lambda)$  pair that denoises and segments the image appropriately with our current model.

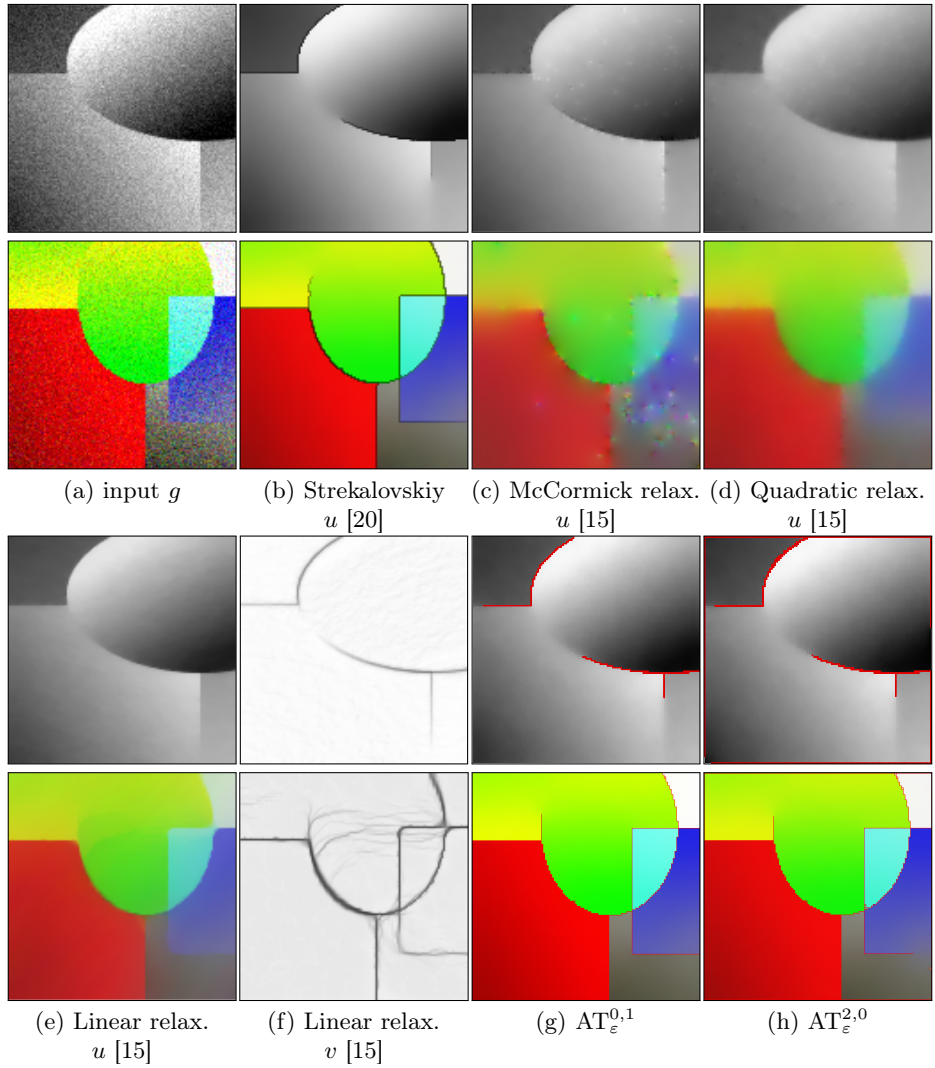
#### 4 Comparisons with other relaxations of MS and AT functionals

A frequent criticism about MS and AT functionals is that they are non convex. Current optimization techniques thus do not extract the global optimum, and results are subject to initialization. This is for instance why TV models have gained so much popularity, despite known problems (most notably staircasing instead of smoothing). Other people have followed a different path and proposed convex approximations of MS or AT. Strelakovski and Cremers have proposed a truncated quadratic potential to approach directly the MS model [20], which leads them to a fast algorithm for segmentation and cartooning. Kee and Kim proposed in [15] three convex relaxations of AT : a quadratic relaxation, a linear relaxation, and the last one based on a factorization theorem due to McCormick [16]. They used a gradient descend algorithm to compute the minimizers, and observed that the best results are obtained with the linear relaxation. We compare in Fig. 4 the results of Kee and Kim, the results of Strelakovski and Cremers, and our approach on the images used in [15].

We can see that the McCormick relaxation (Fig. 4 (c)) retains many artifacts while the quadratic relaxation (Fig. 4 (d)) leads to blurry reconstructions. The results obtained with the linear relaxation (Fig. 4 (e)) are piecewise smooth, but we observe a poor reconstruction of angles in the vectorial case. Furthermore we see on Fig. 4 (f) that they obtain diffuse and sometimes false contours.

The convex relaxation of Strelakovski and Cremers leads to piecewise smooth results (Fig. 4 (b)), with sharp discontinuities, similar to our approaches (Fig. 4 (g)-(h)).

In Fig. 5, we compare the results of TV model implemented in [9], the results of Strelakovski *et. al.* [20], and our approach on synthetic images. For all methods, we set the parameters that maximize the PSNR (see Tab. 1). We observe the characteristic staircasing effect with the TV method. Strelakovski *et. al.* and our method are robust to noise and yield piecewise smooth results.



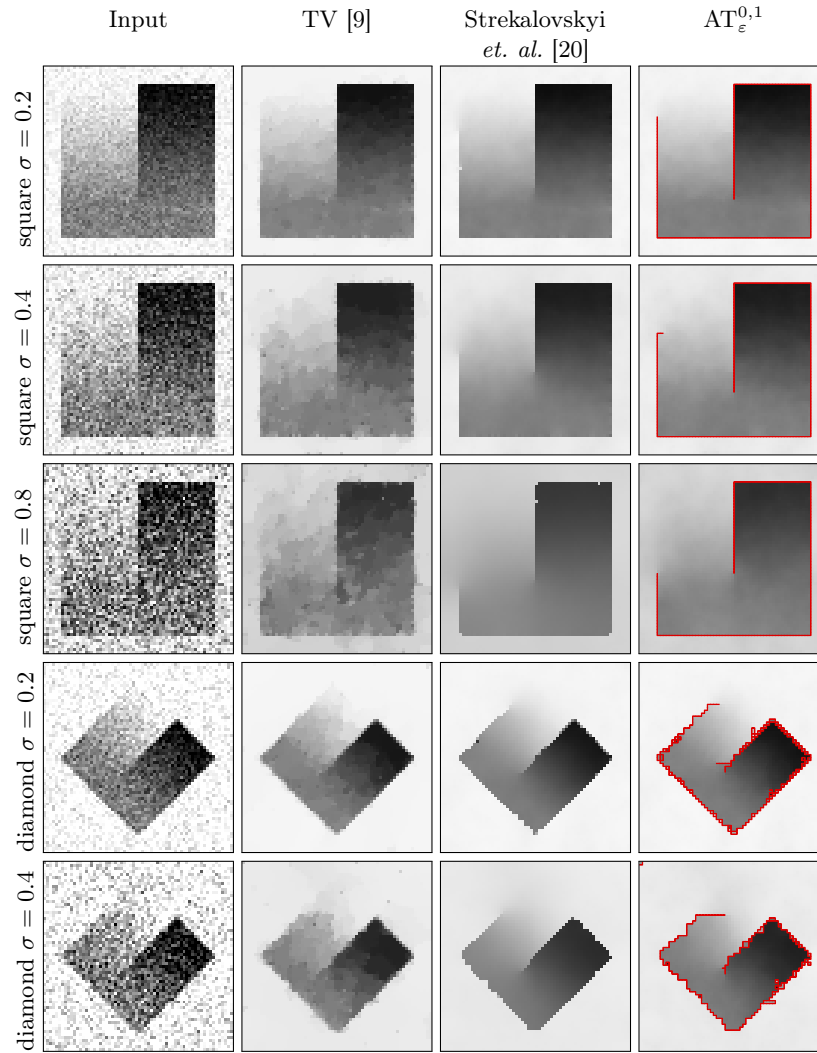
**Fig. 4.** Comparison of the minimization of convex relaxations of AT (extracted from [15]), MS [20] and our approach on images extracted from [15]. In all cases, parameters were fixed to get best possible SNR.

## 5 Influence of parameters and scale spaces

The  $\Gamma$ -convergence parameter  $\varepsilon$  controls the thickness of the contours. Large values of  $\varepsilon$  convexify the AT functional and help to detect the discontinuities. Then, as  $\varepsilon$  goes to 0, the contours become thinner and thinner (see Fig. 6).

On the other hand, for a large enough and fixed  $\lambda$ , the minimization of AT is equivalent to a diffusion with a small perturbation dependent on  $\alpha$ . Hence

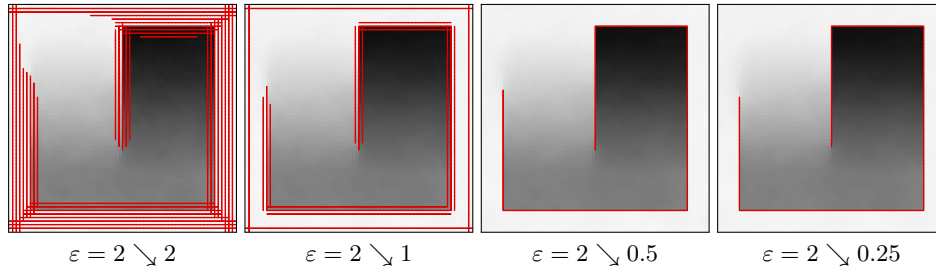




**Fig. 5.** Comparison with competing methods. Noise level is specified in the first column.

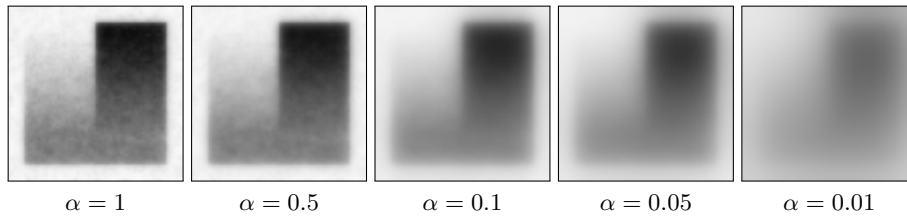
**Table 1.** Best PSNRs results are in bold.

	TV [9]	Strek. <i>et. al.</i> [20]	$AT_{\epsilon}^{2,0}$	$AT_{\epsilon}^{0,1}$
square $\sigma = 0.2$	29.1528	30.3717	30.7517	<b>30.8439</b>
square $\sigma = 0.4$	23.7067	24.7827	24.8512	<b>25.2425</b>
square $\sigma = 0.8$	17.6858	17.3795	17.1204	<b>18.3949</b>
diamond $\sigma = 0.2$	27.3675	26.6616	<b>27.7706</b>	27.2441
diamond $\sigma = 0.4$	<b>21.8341</b>	21.1176	21.7719	21.6394



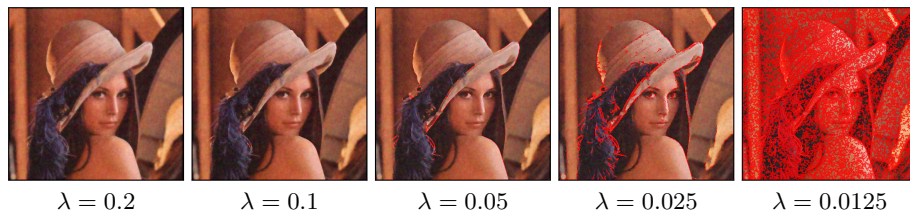
**Fig. 6.**  $\varepsilon$  scale space, for fixed  $\lambda = 0.006$  and  $\alpha = 0.1$ .

parameter  $\alpha$  can be chosen for an initial blur of the input data : the smaller  $\alpha$  is, the stronger the initial blur is (see Fig. 7).



**Fig. 7.**  $\alpha$  scale space, for fixed  $\lambda = 1.0$  and  $\varepsilon = 2 \setminus 0.25$ .

Finally, for fixed  $\alpha$  (chosen so as to remove noise), the parameter  $\lambda$  controls the length of discontinuities. Hence, for a large enough  $\lambda$ , the set of discontinuities is empty because the minimization of  $AT$  is equivalent to a diffusion. As the parameter  $\lambda$  decreases, more and more discontinuities are detected and delineated, as we can see in Fig. 8. Note that results of  $AT_{\varepsilon}^{0,1}$  and  $AT_{\varepsilon}^{2,0}$  are almost identical for this image.



**Fig. 8.**  $\lambda$  scale space of  $AT_{\varepsilon}^{2,0}$ , for fixed  $\alpha = 0.48$  and  $\varepsilon = 2 \setminus 0.25$ .

## 6 Potential of the model in image inpainting

It is easy to adapt the AT models for image inpainting. It suffices to have a parameter  $\alpha$  that varies across the image. More precisely, it is set to 0 wherever the user selected a damaged zone while it is kept to its normal value elsewhere. Furthermore, forms  $u$  are initialized with random values at these places.

We have checked if our discrete AT models can be interesting for image inpainting. As illustrated on Fig. 9, it appears that model  $AT_\varepsilon^{0,1}$  introduces too many artefacts on artificial examples. Indeed, it optimizes the  $L1$ -length of discontinuities, which is not a natural inpainting for a human eye. The problem is that the  $L1$ -length has a very flat minimum, which is composed of all the discrete paths in some rectangle joining two constraints. Model  $AT_\varepsilon^{0,1}$  picks any one of these paths: this is governed by the order of cells in the linear systems.

However, model  $AT_\varepsilon^{2,0}$  seems much more interesting. It is able to detect the two triple points that optimize a four region contact as well as the 120 angle for a three region contact. The reason is that this model averages the  $L1$ -length of discontinuities especially when  $\varepsilon$  is large. Hence, the diffused  $L1$ -length is closer to the natural  $L2$ -length and the model chooses a better  $L1$ -path for large  $\varepsilon$ . As  $\varepsilon$  decreases, discontinuities stay at the correct location while getting crispier. This explains why triple points are correctly detected by  $AT_\varepsilon^{2,0}$ , if we start from a large enough  $\varepsilon$ , which must be proportional to the width of the damaged zone. In Fig. 9, setting  $\varepsilon = 4 \searrow 0.25$  was good for all examples.

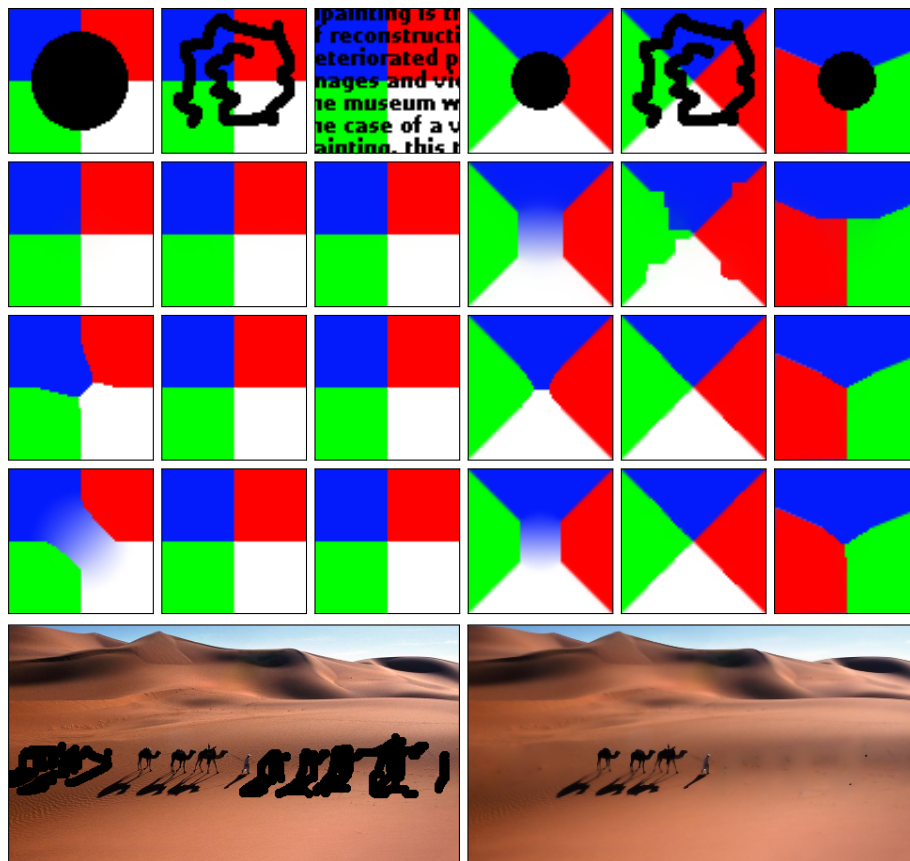
Of course, variational methods for image inpainting should be combined with some non-local/texture filtering to compete with state-of-the-art approaches, but we may conclude that  $AT_\varepsilon^{2,0}$  is a candidate for such a hybrid inpainting method.

## 7 Implementation in the DGtal library

All presented material and programs have been implemented as a tool in the DGtal library [22], which is an open-source C++ library, bundled with a set of tools for digital geometry and image processing. Our discrete AT implementation relies heavily on the Discrete Exterior Calculus package. This package provides data-structures and algorithms for manipulating discrete forms on cellular complexes, linear operators such as derivatives, and wrappers for linear system solvers.

We provide two command-line tools in the `imageProcessing` package of `DGtalTools` (<http://dgtal.org/doc/tools/nightly/imageProcessing>), one called `at-u0-v1` that implements  $AT_\varepsilon^{0,1}$ , the other called `at-u2-v0` that implements  $AT_\varepsilon^{2,0}$ . They provide exactly the same functionalities and options.

Among functionalities, these tools can process grey-level or color images given in PBM format. You may either restore the whole image or perform inpainting by giving a mask image specifying damaged zone. SNR computation is provided if you provide ground truth. Both tools export the restored image  $u$  as well as an (possibly zoomed) image displaying both  $u$  and the set of discontinuities  $v$ . You may set parameter  $\alpha$ , the range for parameter  $\varepsilon$ , and either a value or a range



**Fig. 9.** Using discrete AT functionals for image inpainting. Top row: input images. Second row: results of  $AT_{\varepsilon}^{0,1}$  with  $\varepsilon = 4 \searrow 0.25$ . Third row: results of  $AT_{\varepsilon}^{2,0}$  with  $\varepsilon = 4 \searrow 0.25$ . Fourth row: results of  $AT_{\varepsilon}^{2,0}$  with  $\varepsilon = 1 \searrow 0.25$ . Fifth row: inpainting of real picture with  $AT_{\varepsilon}^{0,1}$  and  $\varepsilon = 4 \searrow 0.25$ .

of values for parameter  $\lambda$ . In the latter case, computations are performed in descending  $\lambda$  order, delineating progressively more and more discontinuities. All options are described in pages associated to these tools. Note also that several generic classes have been created to facilitate the development of further image tools that use a discrete exterior calculus formulation.

For further works, we plan to develop an on-line demo of these image restoration methods, which uses the IPOL (Image Processing On-Line) platform. This will facilitate even more the reproducibility of our research as well as fair comparisons with other restoration methods.

## References

1. Luigi Ambrosio and Vincenzo Maria Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.
2. Leah Bar, Tony F. Chan, Ginmo Chung, Miyoung Jung, Nahum Kiryati, Rami Mohieddine, Nir Sochen, and Luminita A. Vese. *Mumford and Shah Model and its Applications to Image Segmentation and Image Restoration*, pages 1095–1157. Springer New York, New York, NY, 2011.
3. Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific, 2nd edition, 1999.
4. Blaise Bourdin and Antonin Chambolle. Implementation of an adaptive finite-element approximation of the mumford-shah functional. *Numerische Mathematik*, 85(4):609–646, 2000.
5. Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.
6. Tony F Chan and Luminita A Vese. Active contours without edges. *Image processing, IEEE transactions on*, 10(2):266–277, 2001.
7. M. Desbrun, A. N. Hirani, M. Leok, and J. E. Marsden. Discrete exterior calculus. *arXiv preprint math/0508341*, 2005.
8. Jinming Duan, Wenqi Lu, Zhenkuan Pan, and Li Bai. New second order mumford–shah model based on  $\gamma$ -convergence approximation for image processing. *Infrared Physics & Technology*, 76:641–647, 2016.
9. Joan Duran, Bartomeu Coll, and Catalina Sbert. Chambolle’s projection algorithm for total variation denoising. *Image Processing On Line*, 2013:311–331, 2013.
10. Selim Esedoglu and Jianhong Shen. Digital inpainting based on the mumford–shah–euler image model. *European Journal of Applied Mathematics*, 13(04):353–370, 2002.
11. Marion Foare, Jacques-Olivier Lachaud, and Hugues Talbot. Image restoration and segmentation using the ambrosio-tortorelli functional and discrete calculus. In *Proceedings of the IAPR International Conference on Pattern Recognition (ICPR2016), Cancun, Mexico*, 2016.
12. Leo Grady and Christopher V Alvino. The piecewise smooth mumford–shah functional on an arbitrary graph. *IEEE Transactions on Image Processing*, 18(11):2547–2561, 2009.
13. Leo J. Grady and Jonathan Polimeni. *Discrete calculus: Applied analysis on graphs for computational science*. Springer, 2010.
14. Anna Jezierska, Emilie Chouzenoux, Jean-Christophe Pesquet, Hugues Talbot, et al. A convex approach for image restoration with exact poisson-gaussian likelihood. 2013.

15. Youngwook Kee and Junmo Kim. A convex relaxation of the ambrosio–tortorelli elliptic functionals for the mumford-shah functional. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4074–4081, 2014.
16. Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i – convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.
17. David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.
18. Mila Nikolova, Selim Esedoglu, and Tony F. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.*, 66(5):1632–1648, 2006.
19. Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.
20. Evgeny Strekalovskiy and Daniel Cremers. Real-time minimization of the piecewise smooth mumford-shah functional. In *Computer Vision–ECCV 2014*, pages 127–141. Springer, 2014.
21. H. Talbot, H. Phelippeau, M. Akil, and S. Bara. Efficient poisson denoising for photography. In *Proceedings of IEEE ICIP 2009*, pages 3881–3883, Cairo, Egypt, 2009.
22. DGTAL: Digital Geometry tools and algorithms library. <http://dgtal.org>.
23. Luminita A Vese and Carole Le Guyader. *Variational methods in image processing*. CRC Press, 2015.