




# Spatially Adaptive Regularization in Image Segmentation

Laura Antonelli <sup>1,†</sup>  0000-0002-4031-099X, Valentina De Simone <sup>2,‡</sup>  0000-0002-3357-5252 and Daniela di Serafino <sup>2,†,‡</sup>  0000-0001-8215-0771

<sup>1</sup> Institute for High Performance Computing and Networking (ICAR), CNR, Naples, Italy; laura.antonelli@cnr.it

<sup>2</sup> University of Campania “Luigi Vanvitelli”, Department of Mathematics and Physics, Caserta, Italy; daniela.diserafino@unicampania.it, valentina.desimone@unicampania.it

\* Correspondence: daniela.diserafino@unicampania.it

† Current address: Department of Mathematics and Physics, University of Campania “Luigi Vanvitelli”, viale A. Lincoln 5, I-81100 Caserta, Italy

‡ All the authors contributed equally to this work.

July 31, 2020

**Abstract:** We modify the total-variation-regularized image segmentation model proposed by Chan, Esedoğlu and Nikolova [SIAM Journal on Applied Mathematics 66, 2006] by introducing local regularization that takes into account spatial image information. We propose some techniques for defining local regularization parameters, based on the cartoon-texture decomposition of the given image, on the mean and median filters, and on a thresholding technique, with the aim of preventing excessive regularization in piecewise-constant or smooth regions and preserving spatial features in nonsmooth regions. We solve the modified model by using split Bregman iterations. Numerical experiments show the effectiveness of our approach.

**Keywords:** image segmentation; nonsmooth optimization; spatially adaptive regularization; split Bregman method

**MSC:** 94A08, 65K10.

## 1. Introduction

Image segmentation is a fundamental problem in image processing and computer vision. Its goal is to divide the given image into regions that represent different objects in the scene. Variational models for image segmentation have been widely investigated, proving to be very effective in many applications – see, e.g., [1] and the references therein. Roughly speaking, the segmentation may be obtained by minimizing a cost functional which linearly combines regularization and data fidelity terms:

$$\min_{u \in \Omega} E(u) \equiv F(u) + \lambda G(u), \quad (1)$$

where  $u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  provides a representation of the segmentation and  $\lambda > 0$  is a parameter that controls the weight of the fidelity term  $G$  versus the regularization term  $F$ . A widely-used segmentation model is the two-phase partitioning model introduced by Chan, Esedoğlu and Nikolova [2], which we refer to as CEN model:

$$\begin{aligned} & \underset{u, c_1, c_2}{\text{minimize}} && \int_{\Omega} |\nabla u| dx + \lambda \int_{\Omega} \left( (c_1 - \bar{u}(x))^2 u(x) + (c_2 - \bar{u}(x))^2 (1 - u(x)) \right) dx, \\ & \text{s.t.} && 0 \leq u \leq 1, \\ & && c_1, c_2 > 0. \end{aligned} \quad (2)$$

Here  $\bar{u}$  denotes the image to be segmented, which is assumed to take its values in  $[0, 1]$ . The CEN model allows us to obtain one of the two domains defining the segmentation,  $\Sigma$  and  $\Omega \setminus \Sigma$ , by setting

$$\Sigma = \{x : u(x) > \alpha\} \text{ for a.e. } \alpha \in (0, 1),$$

where  $u$  is the solution of problem (2). Note that (2) is the result of a suitable relaxation of the Chan-Vese model [3] leading to a convex formulation of that problem for any given  $(c_1, c_2)$ .

Here we start from a discrete version of the CEN model. Let

$$\Theta = \{(i, j) : 0 \leq i \leq m - 1, 0 \leq j \leq n - 1\}$$

be a discretization of  $\Omega$  consisting of an  $m \times n$  grid of pixels and let

$$|\nabla_x u|_{i,j} = |\delta_x^+ u|_{i,j}, \quad |\nabla_y u|_{i,j} = |\delta_y^+ u|_{i,j}$$

where  $\delta_x^+$  and  $\delta_y^+$  are the forward finite-difference operators in the  $x$ - and  $y$ -directions, with unit spacing, and the values  $u_{i,j}$  with indices outside  $\Theta$  are defined by replication. We consider the following discrete version of (2) with anisotropic discrete total variation (TV):

$$\begin{aligned} & \underset{u, c_1, c_2}{\text{minimize}} && \sum_{i,j} (|\nabla_x u|_{i,j} + |\nabla_y u|_{i,j}) + \lambda \sum_{i,j} \left( (c_1 - \bar{u}_{i,j})^2 u_{i,j} + (c_2 - \bar{u}_{i,j})^2 (1 - u_{i,j}) \right), \\ & \text{s.t.} && 0 \leq u_{i,j} \leq 1, \\ & && c_1, c_2 > 0. \end{aligned} \tag{3}$$

After the minimization problem (3) is solved, the segmentation is obtained by taking

$$\Sigma = \{(i, j) \in \Theta : u_{i,j} > \alpha\}, \tag{4}$$

for some  $\alpha \in (0, 1)$ . Problem (3) is usually solved by alternating the minimization with respect to  $u$  and the minimization with respect to  $c_1$  and  $c_2$ . In the sequel, we denote  $E(u, c_1, c_2)$  the objective function in (3), and  $F(u)$  and  $G(u, c_1, c_2)$  its discrete regularization and fidelity terms, respectively.

The selection of a parameter  $\lambda$  able to balance  $F(u)$  and  $G(u, c_1, c_2)$  and to produce a meaningful solution is a critical issue. Too large values of  $\lambda$  may produce oversegmentation, while too small values of  $\lambda$  may produce undersegmentation [4]. Furthermore, a constant value of  $\lambda$  may not be suitable for the whole image, i.e., different regions of the image may need different values. In recent years, spatially adaptive techniques have been proposed, focusing on local information, such as gradient, curvature, texture and noise estimation cues – see, e.g., [5]. Space-variant regularization has been also widely investigated in the context of image restoration, using TV and its generalizations – see, e.g., [6–10].

In this work, we propose some techniques for setting the parameter  $\lambda$  in an adaptive way based on spatial information, in order to prevent excessive regularization of smooth regions while preserving spatial features in nonsmooth areas. Our techniques are based on the so-called cartoon-texture decomposition of the given image, on the mean and median filters, and on thresholding techniques. The resulting locally adaptive segmentation model can be solved either by smoothing the discrete TV term – see, e.g., [11,12] – and applying optimization solvers for differentiable problems such as spectral gradient methods [13–16] or by using directly optimization solvers for nondifferentiable problems, such as Bregman, proximal and ADMM methods [17–23]. In this work we use an alternating minimization procedure exploiting the split Bregman (SB) method proposed in [19]. The results of numerical experiments on images with different characteristics show the effectiveness of our approach and the advantages coming from using local regularization.

The rest of this paper is organized as follows. In Section 2 we propose our spatially adaptive techniques. In Section 3 we describe the solution of the segmentation model using those techniques by

the aforementioned SB-based alternating minimization method. In Section 4 we discuss the results obtained with our approaches on several test images, performing also a comparison with the CEN model and a segmentation model developed for images with texture. The results show the effectiveness of our approach and the advantages coming from the use of local regularization. Some conclusions are provided in Section 5.

## 2. Defining local regularization by exploiting spatial information

The regularization parameter  $\lambda$  plays an important role in the segmentation process, because it controls the tradeoff between data fidelity and regularization. In general, the smaller the parameter in (3) the smoother the image content, i.e., image details are flattened or blurred. Conversely, the larger the parameter the more the enhancement of image details, and hence noise may be retained or amplified. Therefore,  $\lambda$  should be selected according to local spatial information. A small value of  $\lambda$  should be used in the smooth regions of the image to suppress noise, while a large value of  $\lambda$  should be considered to preserve spatial features in the nonsmooth regions. In other words, a matrix  $\Lambda = (\lambda_{i,j})$  should be associated with the image, where  $\lambda_{i,j}$  weighs pixel  $(i, j)$ , as follows:

$$\begin{aligned} & \underset{u, c_1, c_2}{\text{minimize}} && \sum_{i,j} (|\nabla_x u|_{i,j} + |\nabla_y u|_{i,j}) + \sum_{i,j} \lambda_{i,j} \left( (c_1 - \bar{u}_{i,j})^2 u_{i,j} + (c_2 - \bar{u}_{i,j})^2 (1 - u_{i,j}) \right), \\ & \text{s.t.} && 0 \leq u_{i,j} \leq 1, \\ & && c_1, c_2 > 0. \end{aligned} \quad (5)$$

Furthermore, in order to avoid oversegmentation or undersegmentation, it is convenient to fix a minimum and a maximum value for the entries of  $\Lambda$ , so as to drive the level of regularization in a reasonable range, depending on the image to be segmented.

We define  $\Lambda$  as a function of the image  $\bar{u}$  to be segmented:

$$f : \bar{u}_{i,j} \rightarrow \lambda_{i,j} \in [\lambda_{min}, \lambda_{max}], \quad (6)$$

where  $0 < \lambda_{min} < \lambda_{max} < \infty$ . We propose three choices of  $f$ , detailed in the next subsections.

We note that problem (5) still has a unique solution for any fixed  $(c_1, c_2)$ , as a consequence of the next proposition.

**Proposition 1.** *For any fixed  $(c_1, c_2)$ , problem (5) is a convex problem.*

**Proof.** Since the CEN model is convex for any fixed  $(c_1, c_2)$ , the thesis immediately follows from the fact that the parameters  $\lambda_{i,j}$  are constant with respect to  $u$ .  $\square$

### 2.1. Regularization based on the cartoon-texture decomposition

We define  $f(\bar{u}_{i,j})$  by using the Cartoon-Texture Decomposition (CTD) of the image discussed in [24]. CTD splits any image  $u$  into the sum of two images,  $w$  and  $v$ , such that  $w$  represents the cartoon or geometric (piecewise-smooth) component of  $u$ , while  $v$  represents the oscillatory or textured component, i.e.,  $v$  contains essentially textures, oscillating patterns, fine details and noise. The algorithm for computing CTD acts as described next. For each image pixel, we decide whether it belongs to the cartoon or the textural part by computing a local indicator associated with an image window around the pixel. The main feature of a textured region is its high TV, which decreases very fast under low-pass filtering. This leads to the following definition of local total variation (LTV) at a pixel  $(i, j)$ :

$$LTV_\sigma(u)_{i,j} = (L_\sigma * |\nabla u|_{i,j}), \quad (7)$$

where  $L_\sigma$  is a low-pass filter,  $\sigma$  is a scale parameter,  $|\nabla u|_{i,j} = \sqrt{(\nabla_x u)_{i,j}^2 + (\nabla_y u)_{i,j}^2}$  and  $*$  denotes the convolution product. The relative reduction rate of LTV,

$$(\rho_\sigma)_{i,j} = \frac{LTV_\sigma(u)_{i,j} - LTV_\sigma(L_\sigma * u)_{i,j}}{LTV_\sigma(u)_{i,j}}, \quad (8)$$

gives the local oscillatory behavior of the image. A value of  $(\rho_\sigma)_{i,j}$  close to 0 means that there is little relative reduction of LTV by the low-pass filter, thus the pixel  $(i, j)$  belongs to the cartoon. Conversely,  $(\rho_\sigma)_{i,j}$  close to 1 indicates that the relative reduction is large and hence the pixel belongs to a textured region.

We use (8) for defining the weights  $\lambda_{i,j}$ . The basic idea is that a large regularization parameter is needed if the pixel  $(i, j)$  belongs to the cartoon, while the parameter must be reduced in texture regions. Therefore, we set the function  $f$  in (6) as

$$f(\bar{u}_{i,j}) \equiv f^{\text{CTD}}(\bar{u}_{i,j}) = \max \left\{ \frac{\lambda_{\min}}{\lambda_{\max}}, 1 - (\rho_\sigma)_{i,j} \right\} \lambda_{\max} \quad (9)$$

and  $(\rho_\sigma)_{i,j}$  is defined by using the given image  $\bar{u}$ . Following [25], we set  $L_\sigma$  equal to the Gaussian filter.

### 2.2. Regularization based on the mean and median filters

We define a technique based on spatial filters that are commonly used to enhance low-frequency details or to preserve edges [26,27]. More precisely, we combine the mean and median filters; the former aims at identifying smooth regions, where the regularization parameter can take small values, the latter aims at identifying edges, where the parameter must remain large. Mean filtering is a simple and easy-to-implement method for smoothing images, i.e., for reducing the intensity variation between a pixel and its neighbors. It also removes high-frequencies components due to the noise and the edges in the image, so the mean filter is a low-pass filter. The median filter preserves edges and useful details in the image.

Based on these considerations, we define a weight function as follows:

$$\omega_{i,j} = \begin{cases} |\bar{u}_{i,j} - (L_{h_1} * \bar{u})_{i,j}| & \text{if } |(L_{h_1} * \bar{u})_{i,j} - (M_{h_2} * \bar{u})_{i,j}| < t, \\ 1 & \text{otherwise.} \end{cases} \quad (10)$$

where  $h_1$  is the window size of the mean filter  $L_{h_1}$ ,  $h_2$  is the window size of the median filter  $M_{h_2}$ , and  $t$  is a threshold value acting as a cutoff between the two filters. Note that  $0 \leq \omega_{i,j} \leq 1$  and the pixels in homogeneous regions have  $\omega_{i,j}$  close to 1. The function  $f$  in (6) is set as

$$f(\bar{u}_{i,j}) \equiv f^{\text{MM}}(\bar{u}_{i,j}) = \max \left\{ \frac{\lambda_{\min}}{\lambda_{\max}}, 1 - \omega_{i,j} \right\} \lambda_{\max}, \quad (11)$$

where MM stands for ‘‘Mean and Median’’.

### 2.3. Regularization based on thresholding

This approach implicitly exploits the definition of  $\Sigma$  in (4) to set  $\lambda_{i,j}$ . The idea is to use large values of  $\lambda_{i,j}$  when  $u_{i,j}$  is close to 1 and small values when  $u_{i,j}$  is close to 0. Therefore, the parameters  $\lambda_{i,j}$  are not defined in terms of the given image  $\bar{u}$  only. If the function  $u$  identifying the segmentation were known a priori, we could define  $\lambda_{i,j} = f(u_{i,j})$  as follows:

$$f(u_{i,j}) \equiv f^{\text{THR}}(u_{i,j}) = 10^{\eta_{i,j}} \in [\lambda_{\min}, \lambda_{\max}], \quad (12)$$

where

$$\eta_{i,j} = e_{\max} - (1 - u_{i,j})(e_{\max} - e_{\min}),$$

$emax = \log_{10} \lambda_{max}$  and  $emin = \log_{10} \lambda_{min}$ .

Since the function  $u$  must be computed by minimizing (3) and this is done by using an iterative procedure, we decided to update  $\lambda_{i,j}$  at each iteration, using the current value of  $u_{i,j}$ . On the other hand, in this case evaluating  $f$  is computationally cheaper than in the previous approaches, which apply two-dimensional convolution operators; thus the cost of the iterative update of  $\lambda_{i,j}$  is practically negligible.

### 3. Solution by split Bregman iterations

As mentioned in Section 1, we use an alternating minimization method to solve problem (5). Given  $u_{i,j}^{k-1}$ , by imposing the first-order optimality conditions with respect to  $c_1$  and  $c_2$ , we get

$$c_1^k = \frac{\sum_{i,j} \lambda_{i,j} \bar{u}_{i,j} u_{i,j}^{k-1}}{\sum_{i,j} \lambda_{i,j} u_{i,j}^{k-1}}, \quad c_2^k = \frac{\sum_{i,j} \lambda_{i,j} \bar{u}_{i,j} (1 - u_{i,j}^{k-1})}{\sum_{i,j} \lambda_{i,j} (1 - u_{i,j}^{k-1})}. \quad (13)$$

For the solution of (5) with respect to  $u$  we use the split Bregman (SB) method [19]. Let

$$G_{loc}(u, c_1, c_2) = \sum_{i,j} \lambda_{i,j} \left( (c_1 - \bar{u}_{i,j})^2 - (c_2 - \bar{u}_{i,j})^2 \right) u_{i,j} = \sum_{i,j} r_{i,j} u_{i,j}, \quad (14)$$

where

$$r_{i,j} = \lambda_{i,j} \left( (c_1 - \bar{u}_{i,j})^2 - (c_2 - \bar{u}_{i,j})^2 \right).$$

Following [28], we reformulate the minimization problem as follows:

$$\begin{aligned} & \underset{u, d_x, d_y}{\text{minimize}} && \|d_x\|_1 + \|d_y\|_1 + G_{loc}(u, c_1, c_2), \\ & \text{s.t.} && 0 \leq u_{i,j} \leq 1, \\ & && d_x = \nabla_x u, \\ & && d_y = \nabla_y u. \end{aligned} \quad (15)$$

Given  $c_1^k$  and  $c_2^k$ , the SB method applied to (15) reads:

$$\begin{aligned} u^k &= \underset{0 \leq u_{i,j} \leq 1}{\text{argmin}} G_{loc}(u, c_1^k, c_2^k) + \frac{\mu}{2} \|d_x^{k-1} - \nabla_x u - b_x^{k-1}\|_2^2 + \frac{\mu}{2} \|d_y^{k-1} - \nabla_y u - b_y^{k-1}\|_2^2, \\ d_x^k &= \underset{d_x}{\text{argmin}} \|d_x\|_1 + \frac{\mu}{2} \|d_x - \nabla_x u^k - b_x^{k-1}\|_2^2, \\ d_y^k &= \underset{d_y}{\text{argmin}} \|d_y\|_1 + \frac{\mu}{2} \|d_y - \nabla_y u^k - b_y^{k-1}\|_2^2, \\ b_x^k &= b_x^{k-1} + \mu(\nabla_x u^k - d_x^k), \\ b_y^k &= b_y^{k-1} + \mu(\nabla_y u^k - d_y^k), \end{aligned} \quad (16)$$

where  $\mu > 0$ .

Closed-form solutions of the minimization problems with respect to  $d_x$  and  $d_y$  can be computed using the soft-thresholding operator:

$$d_x^k = \mathcal{S} \left( \nabla_x u^k + b_x^k, \frac{1}{\mu} \right), \quad d_y^k = \mathcal{S} \left( \nabla_y u^k + b_y^k, \frac{1}{\mu} \right),$$

where, for any  $v = (v_{i,j})$  and any scalar  $\gamma > 0$ ,

$$\mathcal{S}(v, \gamma) = z = (z_{i,j}), \quad z_{i,j} = \frac{z_{i,j}}{|z_{i,j}|} \max \{ |z_{i,j}| - \gamma, 0 \}.$$

Finally, an approximate solution to the minimization problem with respect to  $u$  can be obtained by applying Gauss-Seidel (GS) iterations to the following system, as explained in [28]:

$$-\Delta u_{i,j} = \frac{r_{i,j}}{\mu} + (\nabla_x(b_x^k - d_x^k)_{i,j}) + (\nabla_y(b_y^k - d_y^k)_{i,j}), \quad (17)$$

where  $\Delta$  is the classical finite-difference discretization of the Laplacian. If the solution to (17) lies outside  $[0,1]^{m \times n}$ , then it is projected onto that set. We denote  $\mathcal{P}_{[0,1]}$  the corresponding projection operator.

The overall solution method is outlined in Algorithm 1. Note that when the approach in Section 2.3 is used, the values  $\lambda_{i,j}$  must be updated at each iteration  $k$ , using (12) with  $u = u^k$ .

---

**Algorithm 1:** SB-based method for spatially adaptive segmentation

---

**Input:**  $\bar{u}, \lambda_{min}, \lambda_{max}, f, \mu, \alpha$  (with  $f$  defined in (9) or (11) or (12))

**Output:**  $u, c_1, c_2$

Set  $u^0 = \bar{u}, d_x^0 = 0, d_y^0 = 0, b_x^0 = 0, b_y^0 = 0$

Compute  $\Lambda = f(u^0)$

**for**  $k = 1, 2, \dots$  **do**

Compute  $c_1^k$  and  $c_2^k$  by (13)

Compute  $u^k$  by applying GS iterations to system (17)

$u^k = \mathcal{P}_{[0,1]}(u^k)$

$d_x^k = \mathcal{S}(\nabla_x u^k + b_x^k, 1/\mu)$

$d_y^k = \mathcal{S}(\nabla_y u^k + b_y^k, 1/\mu)$

Update  $b_x^k$  and  $b_y^k$  according to (16)

**end**

Set  $\Sigma = \{(i, j) \in \Theta : u_{i,j}^k > \alpha\}$

---

#### 4. Results and comparisons

The three spatially adaptive regularization techniques were implemented in MATLAB, using the Image Processing Toolbox. Algorithm 1 was implemented by modifying the *Fast Global Minimization Algorithm for Active Contour Models* by X. Bresson, available from [http://htmlpreview.github.io/?https://github.com/xbresson/old\\_codes/blob/master/codes.html](http://htmlpreview.github.io/?https://github.com/xbresson/old_codes/blob/master/codes.html). This is a C code with a MATLAB MEX interface.

$L_\sigma$  in (7) is defined as a rotationally symmetric Gaussian filter with size 3 and standard deviation  $\sigma = 2$ . The mean and median filters use windows of size 3 and 7, respectively, and the parameter  $t$  in (10) is set as  $t = 0.5$ . The parameter  $\alpha = 0.5$  is used to identify the domain  $\Sigma$  according to (4).

In the original and modified codes, the SB iterations are stopped as follows:

$$|\text{diff}^k - \text{diff}^{k-1}| \leq \text{tol} \quad \text{and} \quad k > \text{maxit}, \quad (18)$$

where

$$\text{diff}^l = \frac{\text{sd}(u^l)}{\text{sd}(u^l) \cdot \text{sd}(u^{l-1})}, \quad \text{sd}(u^l) = \sum_{i,j} (u_{i,j}^l - u_{i,j}^{l-1})^2,$$

$\text{tol}$  is a given tolerance, and  $\text{maxit}$  denote the maximum number of outer iterations. The stopping criterion for the GS iterations is

$$E(u^k) = 1 - \frac{|\text{msd}^k - \text{msd}^1|}{\text{msd}^1} \leq \text{tol}_{\text{GS}} \quad \text{and} \quad k > \text{maxit}_{\text{GS}}$$

where

$$\text{msd}^l = \frac{1}{mn} \sum_{i,j} (u_{i,j}^l - u_{i,j}^{l-1})^2,$$

and  $\text{tol}_{\text{GS}}$  and  $\text{maxit}_{\text{GS}}$  are the tolerance and the maximum number of iterations for the GS method, respectively. In our experiments we set  $\text{tol} = 10^{-6}$ ,  $\text{maxit} = 30$ ,  $\text{tol}_{\text{GS}} = 10^{-2}$  and  $\text{maxit}_{\text{GS}} = 50$ .

The adaptive models are compared with the original CEN model on different images widely used in image processing tests, listed in Table 1 and shown in Figures 1 to 4. In particular, the images *bacteria*, *bacteria2*, *brain*, *cameraman*, *squirrel* and *tiger* are included in Bresson's code distribution, *flowerbed* has been downloaded from the Berkeley segmentation dataset [29] available from <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html> (image #86016), and *ninetyeight* is available from <https://tineye.com/query/2817cf0d186fbfe263a188952829a3b5e699d839>. We note that *tiger* is included in Bresson's code as a test problem for a segmentation model specifically developed for images with texture [30], which is also implemented in the code. This model uses the well-known Kullback-Leibler divergence function regularized with a TV term. The model is solved with the SB method. We perform the segmentation of *tiger* with the CEN model, the textural segmentation model and our approaches, to investigate whether our locally adaptive model can be also suitable for textural images. The *cameraman* image is perturbed with additive Gaussian noise, with zero mean and two values of standard deviation,  $\sigma = 15, 25$ , with the aim of evaluating the behavior of our adaptive approaches on noisy images. The noisy images are called *cameraman15* and *cameraman25*.

For the images provided with Bresson's code, the values of  $\lambda$  and  $\mu$  associated with the original CEN model are set as in that code. For the remaining images, the values of  $\lambda$  and  $\mu$  are set by trial and error, following the empirical rule reported in Bresson's code. The values of  $\lambda_{\min}$  and  $\lambda_{\max}$  used in the spatially adaptive approaches are chosen so that the corresponding  $\lambda$  in the original CEN model is in  $[\lambda_{\min}, \lambda_{\max}]$ , with few exceptions. The associated values of  $\mu$  are set as in the non-adaptive case. The values of  $\lambda$ ,  $\lambda_{\min}$ ,  $\lambda_{\max}$  and  $\mu$  used for each image are specified in Tables 2 to 4. The values of  $\lambda_{i,j}$  in the adaptive models are initialized as specified in (9), (11) and (12). As described in Section 2.3, in the strategy based on thresholding those values change at each iteration  $k$  of Algorithm 1. It is worth noting that our approach also simplifies the choice of the regularization parameter, which may be a time-consuming task.

The initial approximation  $u^0$  is set equal to the image  $\bar{u}_{i,j}$ , which takes its values in  $[0, 1]$ , as specified in Section 1. This is used to compute the starting values of  $c_1$  and  $c_2$ , and  $s_{i,j} = (c_1 - \bar{u}_{i,j})^2 - (c_2 - \bar{u}_{i,j})^2$  for all  $(i, j)$ . In the original non-adaptive code, the value of  $\lambda$  is scaled using the difference between the maximum and the minimum value of  $s_{i,j}$ ; the same scaling is applied to  $\lambda_{\min}$  and  $\lambda_{\max}$  in the implementations of the adaptive approaches.

We run the tests on an Intel Core i7 processor with clock frequency of 2.6 GHz, 8 GB of RAM, and a 64-bit Linux system.

**Table 1.** Test images and their sizes.

image	size (pixels)
<i>bacteria</i>	233 × 256
<i>bacteria2</i>	380 × 391
<i>brain</i>	210 × 210
<i>cameraman</i>	204 × 204
<i>flowerbed</i>	321 × 481
<i>ninetyeight</i>	300 × 225
<i>squirrel</i>	167 × 230
<i>tiger</i>	321 × 481



For each of the six images *bacteria*, *bacteria2*, *brain*, *flowerbed*, *ninetyeight* and *squirrel* we show the results obtained with the spatially adaptive strategy yielding the best segmentation for that image. The corresponding (unscaled) values of  $\lambda$ ,  $\lambda_{min}$ ,  $\lambda_{max}$ , the value of  $\mu$ , as well as the number of outer iterations and the mean number of GS iterations per outer iteration are reported in Table 2. Note that for *squirrel* we use two values of  $\lambda_{min}$ , one equal to the value of  $\lambda$  in the CEN model and the other greater than that, obtained by trial and error. Both values of  $\lambda_{min}$  produce the same segmentation, but the larger value of  $\lambda_{max}$  reduces the number of outer and GS iterations. The segmentations corresponding to the data in Table 2 are shown in Figures 1 to 3. For *squirrel* we display the CTD segmentation computed by using the larger value of  $\lambda_{min}$ .

**Table 2.** Segmentation of six test images by the CEN and spatially adaptive models: parameters and iterations. The value of  $\mu$  is the same for all the models, thus it is reported only once per image.

	$\lambda$	$\mu$	it	it <sub>GS</sub>	$\lambda_{min}$	$\lambda_{max}$	it	it <sub>GS</sub>
<b>image</b>	<b>CEN</b>				<b>CTD</b>			
<i>brain</i>	0.7e+3	0.1e+4	5	19.2	0.9e+3	0.4e+5	5	14.8
<i>squirrel</i>	0.118e+3	0.1e+4	6	50	0.118e+3	0.1e+4	15	50
					0.2e+3	0.1e+4	5	45
	<b>CEN</b>				<b>MM</b>			
<i>bacteria2</i>	0.1e+3	0.1e+4	2	50	0.1e+3	0.1e+4	10	50
<i>ninetyeight</i>	0.2e+2	0.1e+3	4	38.5	0.1e+2	0.5e+2	3	50
	<b>CEN</b>				<b>THR</b>			
<i>bacteria</i>	0.5e+4	0.1e+5	6	27.5	0.4e+4	0.8e+4	7	30.7
<i>flowerbed</i>	0.1e+2	0.1e+4	2	27	0.3e+2	0.1e+3	2	16.5

We see that on the selected problems CTD reduces the number of outer and GS iterations with respect to the CEN model; on the other hand, the setup of the regularization parameters is computationally more expensive. In terms of iterations, there is no clear winner between CEN and MM and between CEN and THR. The models based on the spatially adaptive techniques are slightly more expensive than the CEN model in this case too. A significant result is that the segmentations obtained with the adaptive techniques appear better than those obtained with the non-adaptive model, i.e., the spatially adaptive models can better outline boundaries between objects and foreground. This is clearly visible by looking at the segmentations of *brain*, *bacteria2* (see the upper right corner), *ninetyeight*, *bacteria* (see the shape of the bacteria). It is also worth noting that the adaptive model based on THR removes textural details that do not belong to the *flowerbed* in the homonym image.

The latter observation is confirmed by the experiments performed on *tiger*. The corresponding model and algorithmic details are reported in Table 3, while the segmentations are shown in Figure 4 along with (visual) information on quantities used to define  $\lambda_{i,j}$  (see (8) and (10)). We see that the CTD and MM strategies produce satisfactory results, although they have been obtained by generalizing a non-textural model.

Finally, we show the results obtained on *cameraman* and its noisy versions by using CEN, CTD, MM, and THR. The methods perform comparable numbers of inner and GS iterations (see Table 4), but the spatially adaptive model THR yields some improvement over the CEN model on the noisy images (Figure 5).

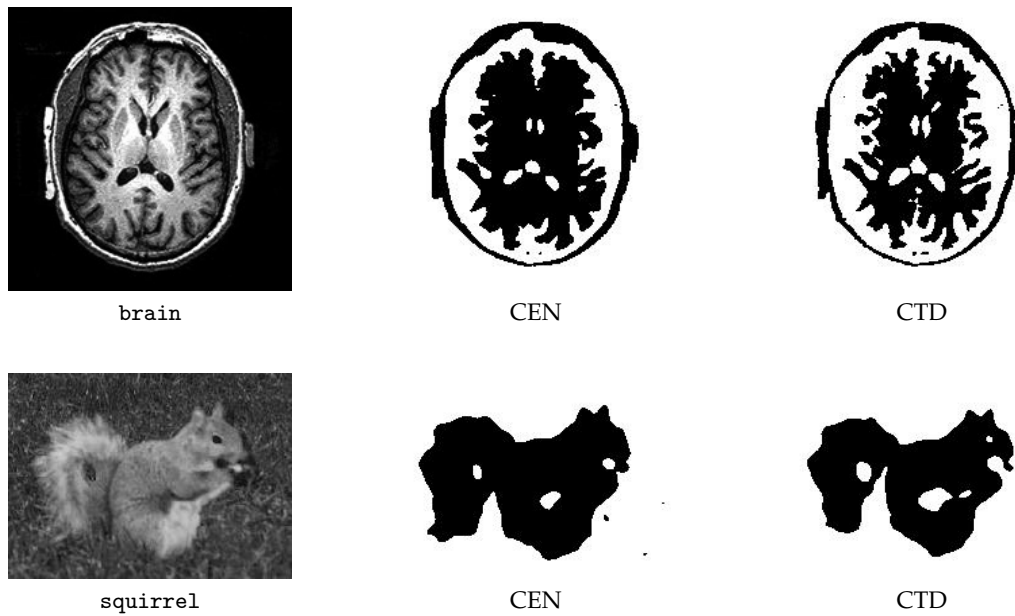
## 5. Conclusions

We introduced spatially adaptive regularization in a well-established variational segmentation model with the aim of improving the segmentation of images by suitably taking into account their smooth and nonsmooth regions. To this aim, we introduced three techniques, based on the application of suitable spatial filters or thresholding. The locally adaptive models, solved via an alternating

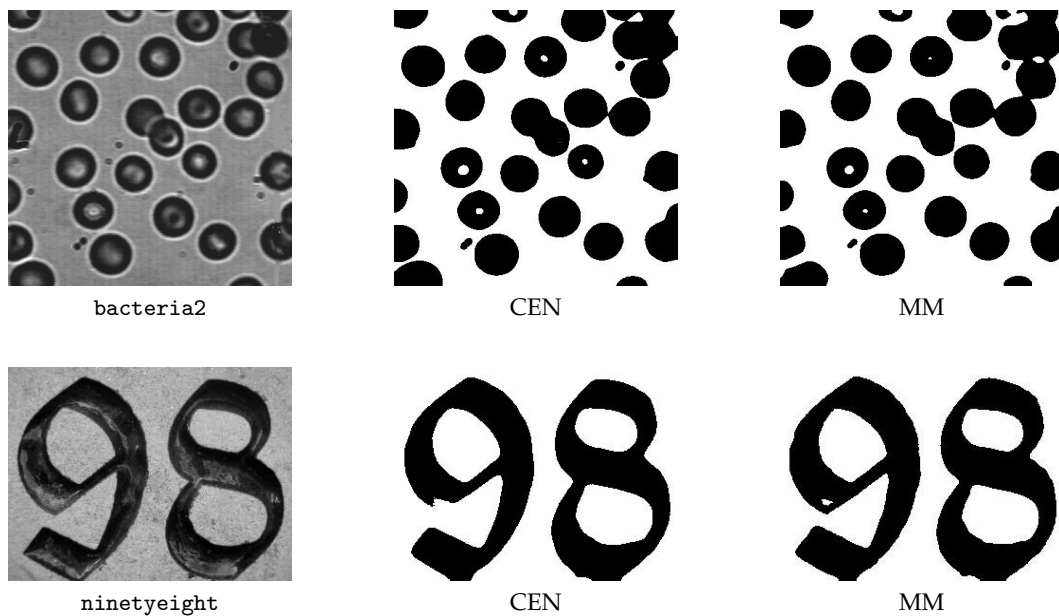


minimization method using split Bregman iterations, showed the effectiveness of our approaches on several images, including also textural and noisy images. We also believe that the proposed models may simplify the setup of the regularization parameter.

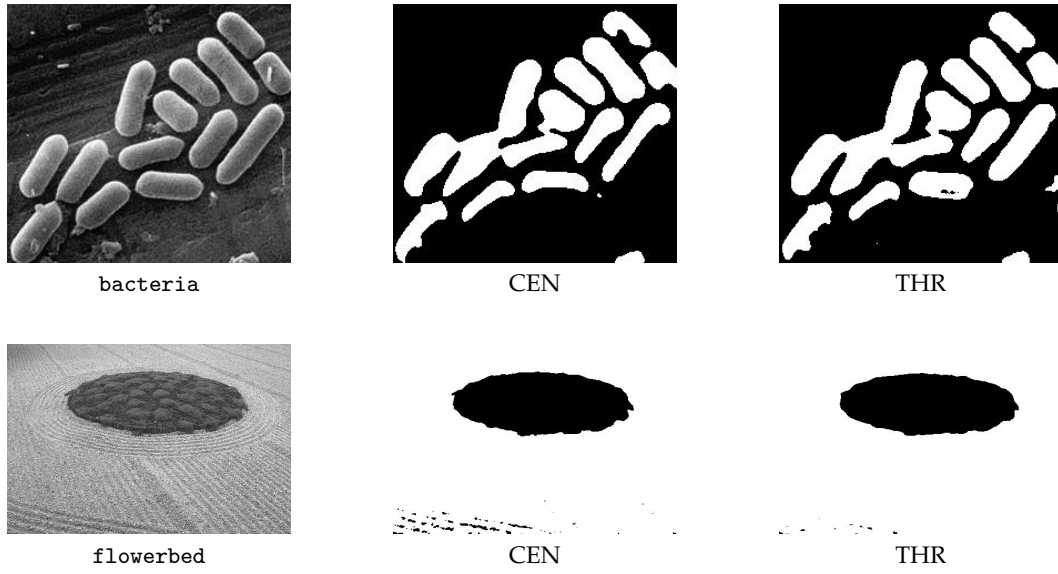
Future work can include the extension of our spatially adaptive strategies to other segmentation models and the development of further adaptive techniques.



**Figure 1.** Segmentations of brain and squirrel by using CEN and CTD.



**Figure 2.** Segmentations of bacteria2 and ninetyeight by using CEN and MM.



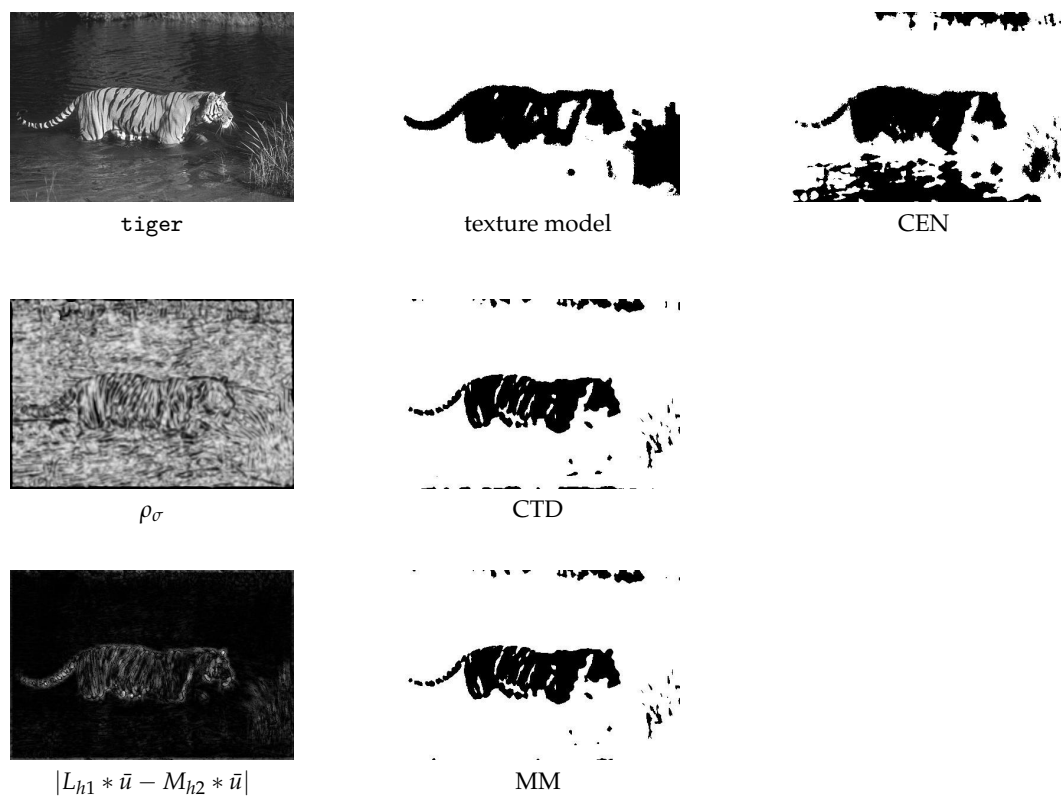
**Figure 3.** Segmentations of bacteria and flowerbed by using CEN and THR.

**Table 3.** Segmentation of tiger by the texture [30], CEN, CTD and MM models: parameters and iterations.

image	$\lambda$	$\mu$	it	it <sub>GS</sub>	$\lambda_{min}$	$\lambda_{max}$	$\mu$	it	it <sub>GS</sub>
<b>Texture Model</b>					<b>CTD</b>				
tiger	0.3e+1	0.1e+3	5	12	0.1e+1	0.1e+2	0.1e+2	8	50
<b>CEN</b>					<b>MM</b>				
tiger	0.3e+2	0.1e+3	7	49.6	0.1e+1	0.1e+2	0.1e+2	8	50

**Table 4.** Segmentations of cameraman and its noisy versions by using CEN and the spatially adaptive models: parameters and iterations. The value of  $\mu$  is the same for all the models, thus it is reported only once per image.

image	CEN				CTD			
	$\lambda$	$\mu$	it	it <sub>GS</sub>	$\lambda_{min}$	$\lambda_{max}$	it	it <sub>GS</sub>
cameraman	0.8e+3	0.1e+3	4	2.8	0.5e+3	0.1e+4	4	3
cameraman15	0.3e+3	0.1e+3	5	4.2	0.3e+3	0.1e+4	5	4.2
cameraman25	0.17e+3	0.1e+3	6	7	0.17e+3	0.8e+3	6	7
image	MM				THR			
	$\lambda_{min}$	$\lambda_{max}$	it	it <sub>GS</sub>	$\lambda_{min}$	$\lambda_{max}$	it	it <sub>GS</sub>
cameraman	0.5e+3	0.1e+4	4	3	0.5e+3	0.1e+4	4	2.8
cameraman15	0.3e+3	0.1e+4	4	4.2	0.3e+3	0.1e+4	5	3.6
cameraman25	0.17e+3	0.5e+3	6	7	0.17e+3	0.8e+3	7	5.7



**Figure 4.** Segmentations of tiger obtained by using the textural, CEN, CTD and MM models. A representation of quantities used to define  $\lambda_{i,j}$  (see (8) and (10)) is also provided.

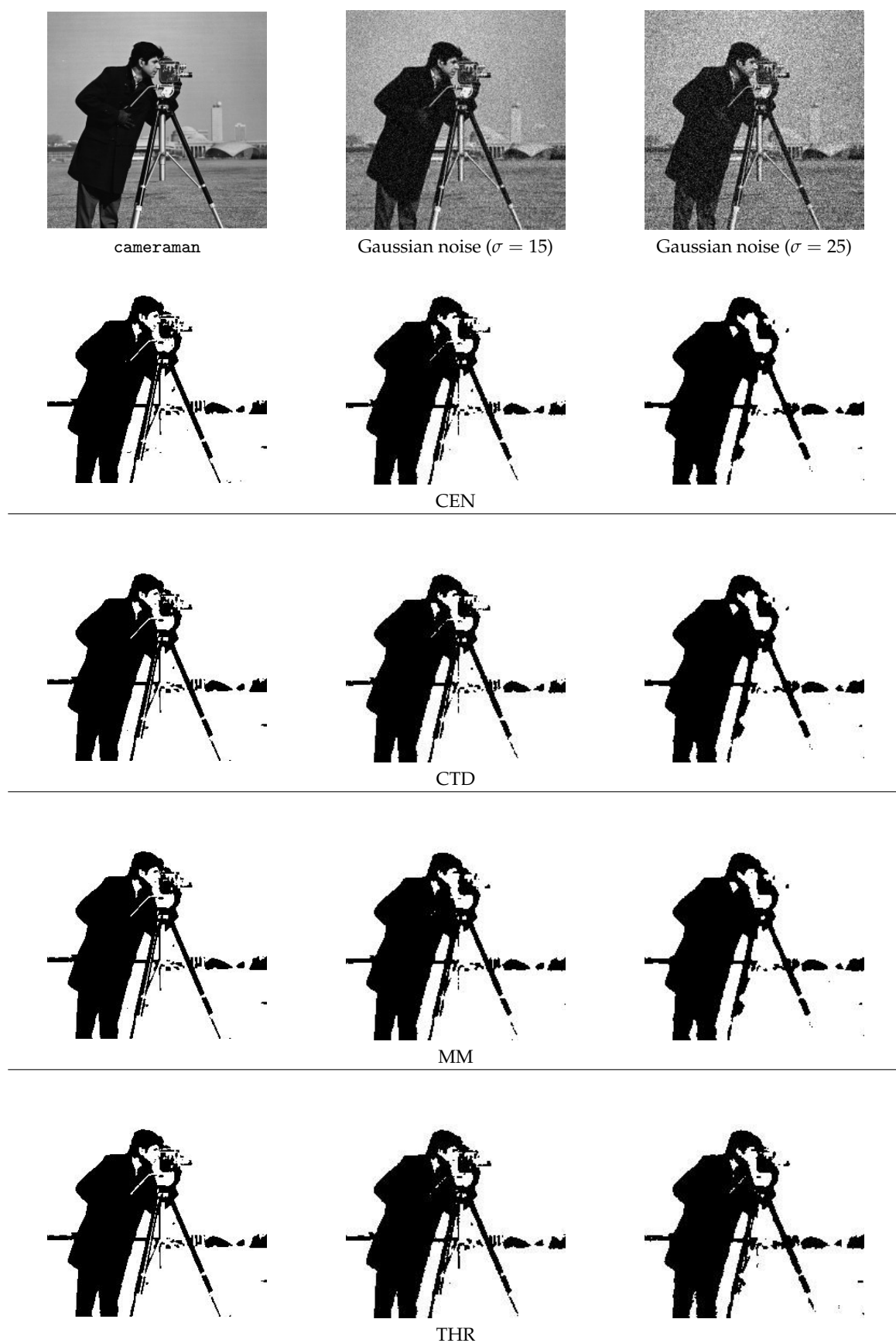


Figure 5. Segmentations of cameraman and its noisy versions by using the CEN, CTD, MM, and THR models.

**Author Contributions:** All the authors contributed equally to this work. They have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by Istituto Nazionale di Alta Matematica - Gruppo Nazionale per il Calcolo Scientifico (INdAM-GNCS), Italy. L. Antonelli was also supported by the Italian Ministry of University and Research under grant no. PON03PE\_00060\_5.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

1. Wang, F.; Zhao, C.; Liu, J.; Huang, H. A variational image segmentation model based on normalized cut with adaptive similarity and spatial regularization. *SIAM J. Imaging Sci.* **2020**, *13*, 651–684. doi:10.1137/18M1192366.
2. Chan, T.F.; Esedoğlu, S.; Nikolova, M. Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models. *SIAM J. Appl. Math.* **2006**, *66*, 1632–1648.
3. Chan, T.F.; Vese, L.A. Active contours without edges. *IEEE Trans. Image Process.* **2001**, *10*, 266–277.
4. Candemir, S.; Akgül, Y.S. Adaptive Regularization Parameter for Graph Cut Segmentation. Image Analysis and Recognition. ICIAR 2010; Campilho, A.; Kamel, M., Eds. Springer, Berlin, Heidelberg, 2010, Vol. 6111, *Lecture Notes in Computer Science*, pp. 117–126. doi:10.1007/978-3-642-13772-3\_13.
5. Rao, J.; Abugharbieh, R.; Hamarneh, G. Adaptive Regularization for Image Segmentation Using Local Image Curvature Cues. Computer Vision – ECCV 2010; Daniilidis, K.; Maragos, P.; Paragios, N., Eds. Springer, Berlin, Heidelberg, 2010, Vol. 6314, *Lecture Notes in Computer Science*, pp. 651–665. doi:10.1007/978-3-642-15561-1\_47.
6. Fong, W.L.; Ng, M.K. On selection of spatial-varying regularization parameters in total variation image restoration. The 2011 International Workshop on Multidimensional (nD) Systems, 2011, pp. 1–5.
7. Bredies, K.; Dong, Y.; Hintermüller, M. Spatially dependent regularization parameter selection in total generalized variation models for image restoration. *Int. J. Comput. Math.* **2013**, *90*, 109–123.
8. Chan, R.H.; Lanza, A.; Morigi, S.; Sgallari, F. An adaptive strategy for the restoration of textured images using fractional order regularization. *Numer. Math. Theory Methods Appl.* **2013**, *6*, 276–296.
9. Ma, T.H.; Huang, T.Z.; Zhao, X.L. Spatially dependent regularization parameter selection for total generalized variation-based image denoising. *Comp. Appl. Math.* **2018**, *37*, 277–296.
10. Calatroni, L.; Lanza, A.; Pragliola, M.; Sgallari, F. A flexible space-variant anisotropic regularization for image restoration with automated parameter selection. *SIAM J. Imaging Sci.* **2019**, *12*, 1001–1037. doi:10.1137/18M1227937.
11. Antonelli, L.; De Simone, V. Comparison of minimization methods for nonsmooth image segmentation. *Communications in Applied and Industrial Mathematics* **2018**, *9*, 68–86.
12. di Serafino, D.; Landi, G.; Viola, M. ACQUIRE: an inexact iteratively reweighted norm approach for TV-based Poisson image restoration. *Appl. Math. Comput.* **2020**, *364*, 124678.
13. Birgin, E.G.; Martínez, J.M.; Raydan, M. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.* **2000**, *10*, 1196–1211.
14. Antonelli, L.; De Simone, V.; di Serafino, D. On the Application of the Spectral Projected Gradient Method in Image Segmentation. *J. Math. Imaging Vis.* **2016**, *54*, 106–116. doi:10.1007/s10851-015-0591-y.
15. di Serafino, D.; Ruggiero, V.; Toraldo, G.; Zanni, L. On the steplength selection in gradient methods for unconstrained optimization. *Appl. Math. Comput.* **2018**, *318*, 176–195.
16. Crisci, S.; Porta, F.; Ruggiero, V.; Zanni, L. Spectral properties of Barzilai-Borwein rules in solving singly linearly constrained optimization problems subject to lower and upper bounds. *SIAM J. Optim.* **2020**, *30*, 1300–1326.
17. Bregman, L.M. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *U.S.S.R. Comput. Math. and Math. Phys.* **1967**, *7*, 200–217.
18. Osher, S.; Burger, M.; Goldfarb, D.; Xu, J.; Yin, W. An iterative regularization method for total variation-based image restoration. *Multiscale Model. and Simul.* **2005**, *4*, 460–489.
19. Goldstein, T.; Osher, S. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.* **2009**, *2*, 323–343.

20. Campagna, R.; Crisci, S.; Cuomo, S.; Marcellino, L.; Toraldo, G. Modification of TV-ROF denoising model based on split Bregman iterations. *Appl. Math. Comput.* **2017**, *315*, 45–467. doi:<https://doi.org/10.1016/j.amc.2017.08.001>.
21. Parikh, N.; Boyd, S. Proximal Algorithms. *Foundations and Trends in Optimization* **2014**, *1*, 127–239. doi:10.1561/2400000003.
22. Beck, A.; Teboulle, M. Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems. *IEEE Trans. Image Proc.* **2009**, *18*, 2419–2434. doi:10.1109/TIP.2009.2028250.
23. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* **2011**, *3*, 1–122. doi:10.1561/2200000016.
24. Buades, A.; Le, T.M.; Morel, J.; Vese, L.A. Fast Cartoon + Texture Image Filters. *IEEE Transactions on Image Processing* **2010**, *19*, 1978–1986. doi:10.1109/TIP.2010.2046605.
25. Buades, A.; Le, T.M.; Morel, J.; Vese, L.A. Cartoon+Texture Image Decomposition. *Image Processing On Line* **2011**, *1*, 200–207. doi:10.5201/ipol.2011.blmv\_ct.
26. Jain, A.K. *Fundamentals of Digital Image Processing*; Prentice-Hall: Englewood Cliffs, 1989.
27. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 2nd ed.; Addison-Wesley Longman Publishing Co., Inc.: USA, 2001.
28. Goldstein, T.; Bresson, X.; Osher, S. Geometric applications of the split Bregman method: segmentation and surface reconstruction. *J. Sci. Comput.* **2010**, *45*, 272–293.
29. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour Detection and Hierarchical Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 898–916. doi:10.1109/TPAMI.2010.161.
30. Houhou, N.; Thiran, J.P.; Bresson, X. Fast Texture Segmentation Based on Semi-Local Region Descriptor and Active Contour. *Numerical Mathematics: Theory, Methods and Applications* **2009**, *2*, 445–468. doi:<https://doi.org/10.4208/nmtma.2009.m9007s>.