

Numerical minimization of a second-order functional for image segmentation

Massimo Zanetti*

Department of Information and Communication Technology, University of Trento, I-38123 Trento, Italy

Valeria Ruggiero, Michele Miranda Jr

Dipartimento di Matematica e Informatica, Università di Ferrara, I-44122 Ferrara, Italy

Abstract

In this paper we address the numerical minimization of a variational approximation of the Blake-Zisserman functional given by Ambrosio, Faina and March. Our approach exploits a compact matricial formulation of the objective functional and its decomposition into quadratic sparse convex sub-problems. This structure is well suited for using a block-coordinate descent method that cyclically determines a descent direction with respect to a block of variables by few iterations of a preconditioned conjugate gradient algorithm. We prove that the computed search directions are gradient related and, with convenient step-sizes, we obtain that any limit point of the generated sequence is a stationary point of the objective functional. An extensive experimentation on different datasets including real and synthetic images and digital surface models, enables us to conclude that: (1) the numerical method has satisfying performance in terms of accuracy and computational time; (2) a minimizer of the proposed discrete functional preserves the expected good geometrical properties of the Blake-Zisserman functional, i.e., it is able to detect first and second order edge-boundaries in images; (3) the method allows the segmentation of large images.

Keywords: Blake-Zisserman functional, variational segmentation, edge-crease detection, block-coordinate descent method

1. Introduction

Segmentation is a typical and widely investigated topic in image processing. It can be defined as the process of partitioning an image into groups of pixels, called segments, in order to represent its content at an object level that potentially simplifies the interpretation of the image itself.

A class of segmentation methods recognizes the objects as the regions delimited by edge-boundaries, i.e., sets of pixels presenting sharp variations of intensity. Mathematical methods for segmentation are mainly divided into two categories: methods based on PDEs and variational methods. PDE's approaches originated in the early 80s with the isotropic scale-space noise-reduction coarsening based on heat diffusion proposed in [1]. Then, anisotropic diffusion have been introduced, which inhibits diffusion according to local properties of the image [2]. Another approach, which is a particular case of anisotropic diffusion, has the interesting property that the related PDE represents the flow generated by the minimization of the Total Variation [3]. Principal issues of PDEs approaches are mainly due to the difficult interpretation of the role that parameters play in the model and the physical meaning of solutions [4, 5].

In this perspective, the variational approach seems to be more intuitive and allows for having a proper and explicit modelization of all the components: noise-reduction, edge-detection, scale-space representation. By fully exploiting a variational framework, Mumford and Shah [6] proposed a model for image segmentation based on the minimization

*Corresponding author

Email addresses: massimo.zanetti@unitn.it (Massimo Zanetti), rgv@unife.it (Valeria Ruggiero), mrmnh1@unife.it (Michele Miranda Jr)

of the following functional

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

Here $\Omega \subset \mathbb{R}^2$ and $g \in L^\infty(\Omega)$ is the input image. The minimization is among all the functions continuously differentiable outside K , i.e. $u \in C^1(\Omega \setminus K)$, where $K \subset \bar{\Omega}$ is compact. \mathcal{H}^1 is the 1-dimensional Hausdorff measure, and α, μ are positive parameters. The minimization of the first term forces u to be smooth (a piecewise constant behavior is expected) outside K . Because of the term $\mathcal{H}^1(K \cap \Omega)$, K is a one-dimensional set with finite length. The last integral term is a distance term that forces u to be close to the original image g . The set K can be easily understood as the set of the discontinuities of u , indeed this is a typical problem belonging to a general class of problems called *free discontinuities problems*, [7]. Kawohl found a strict relationship between the Mumford-Shah and Perona-Malik approaches to segmentation [8]. In particular, he showed how the parameters of the Mumford-Shah (MS) functional can be interpreted as parameters regulating an anisotropic diffusion process applied to the image g .

From a practical point of view, the minimization of the MS functional (1) cannot be addressed because the measure term $\mathcal{H}^1(K \cap \Omega)$ is not semi-continuous with respect to any reasonable topology. As suggested in [7], by relaxing the problem into the weaker space of Special Functions of Bounded Variation $SBV(\Omega)$, the methods of Calculus of Variations can be used to prove the existence of minima [9]. The advantage of this approach is that for every $u \in SBV(\Omega)$, the discontinuity set S_u is uniquely determined by geometrical properties of the function. This results in a functional formulation of the MS problem that uniquely depends on the function u :

$$\mathcal{G}(u) = \int_{\Omega} |\nabla u|^2 dx + \alpha \mathcal{H}^1(S_u \cap \Omega) + \mu \int_{\Omega} |u - g|^2 dx, \quad (2)$$

where $u \in SBV(\Omega)$ and S_u is the complement set of Lebesgue points of u . Using compactness and lower semi-continuity theorems [10] it is showed that under mild conditions, there exists a solution such that $\mathcal{H}^1(S_u) < \infty$. Moreover, by regularity results one has that $\mathcal{H}^1(\bar{S}_u \setminus S_u) = 0$ and the couple (u, \bar{S}_u) can be identified with a minimizer of the strong formulation.

Based on this relaxed formulation, many techniques have been proposed to tackle the problem of numerically computing a minimizer. The free discontinuity term poses a serious problem. Ambrosio and Tortorelli [11], by exploiting a nice result of Modica and Mortola [12], proposed a Γ -convergence approximation via integral functionals defined on proper Sobolev spaces. In their approximation the discontinuity set is replaced by an auxiliary function that plays the role of indicator function. Numerical solutions based on the Ambrosio-Tortorelli approximation are given in the framework of Finite Element Method (FEM) in [13], and via finite-difference discretization of Euler-Lagrange equations in [14]. In [15], a Γ -convergence approximation using local integral functionals defined on a discrete space is given. Numerical implementation of the method is presented in [16]. Another minimization technique is based on a convex relaxation of the functional [17]. A level set approach to minimization is presented in [18]. With no intent of being exhaustive, we refer the interested reader to the overview on the numerical approaches for solving the MS functional given in [19].

1.1. The Blake-Zisserman model for image segmentation

Being a first-order model, the MS variational segmentation suffers of some side effects [20, 21]. The minimization of the gradient norm forces the solution to be locally constant (zero gradient). In those regions where the gradient of g is too steep, this local approximation results in a step-wise function characterized by many fictitious discontinuities. This phenomenon is well-known as *over-segmentation* of steep gradients. Moreover, the minimization of the length term results in an approximation of complex edge junctions by triple-junctions where edges meet at $2/3\pi$ wide angles. This may lead to a degradation of the real geometry of boundaries. Lastly, properly because of its first-order nature, the MS model is unable to detect second-order geometrical features such as points of gradient discontinuity, see Figure 1. Since very often such points correspond to object boundaries, the MS model has the limitation that is not capable of detecting them.

With the specific intent to overcome such problems, Blake and Zisserman proposed a variational model based on second order derivatives, free discontinuities and free gradient discontinuities [20]. In their original formulation one

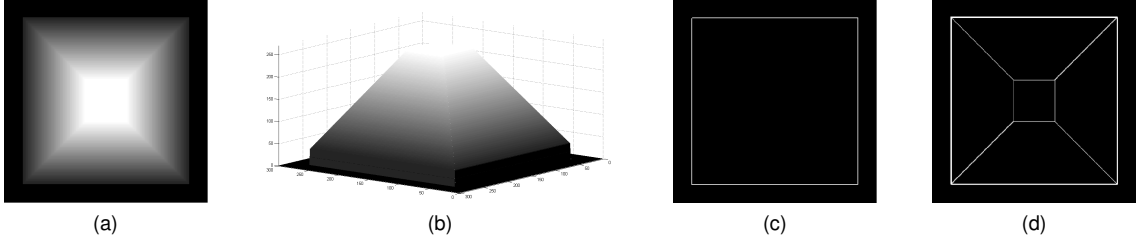


Figure 1: Limitation of the MS model of detecting second-order geometrical features. (a,b) Gray-scale image with second-order edges. (c) Edge-detection via Mumford-Shah functional compared to (d) a full theoretical exact detection of 2nd-order features.

has to minimize

$$\begin{aligned} \mathcal{BZ}(u, K_0, K_1) = & \int_{\Omega \setminus (K_0 \cup K_1)} |\nabla^2 u|^2 dx + \mu \int_{\Omega} |u - g|^2 dx \\ & + \alpha \mathcal{H}^1(K_0 \cap \Omega) + \beta \mathcal{H}^1((K_1 \setminus K_0) \cap \Omega), \end{aligned} \quad (3)$$

among all functions u that are twice differentiable (with continuity) outside $K_0 \cup K_1$ and at least differentiable (with continuity) outside K_0 . K_0 and K_1 vary among all the compact sets such that $K_0 \cup K_1$ is closed in $\bar{\Omega}$. μ, α, β are positive parameters. Here $\nabla^2 u$ denotes the Hessian matrix of u . Notice that, for an admissible solution u , discontinuities are allowed both on $K_0 \cup K_1$, whereas discontinuities of the gradient are allowed only on K_1 . α and β are contrast parameters regulating the total length of the discontinuity sets.

As in the MS case, also for the BZ functional minima existence and numerical issues can be addressed by considering a relaxation of the functional. Following [22], a relaxation in the space of Generalized Special Functions of Bounded Variation $GSBV(\Omega)$, is given by the functional

$$\mathcal{F}(u) = \int_{\Omega} (\mu |u - g|^2 + |\nabla^2 u|^2) dx + (\alpha - \beta) \mathcal{H}^1(S_u) + \beta \mathcal{H}^1(S_{\nabla u} \cup S_u), \quad (4)$$

where $u \in GSBV^2(\Omega) := \{w \in GSBV(\Omega) : \nabla w \in [GSBV(\Omega)]^2\}$. In this weaker space, a proper definition of $\nabla^2 u$ and $S_{\nabla u}$ (the theoretic discontinuity set of ∇u) as geometrical property of the function u , is possible. By regularity arguments it can be proved [23] that a minimizer of (4) can be identified with a minimizing couple of the strong formulation, provided $\beta \leq \alpha \leq 2\beta$. Thus, the optimal set $K_0 \cup K_1$ is recovered via the discontinuity set S_u and the gradient discontinuity set $S_{\nabla u}$.

A vivid research interest is devoted to the Blake-Zisserman functional as it represents the generalization of the well-known and widely used Mumford-Shah. From a theoretical point of view it is a challenging topic, well-posedness of the problem and uniqueness of the solution [24] as well as regularity properties of minimizers [25, 26, 27] are still under investigation. Recently, a concise survey of the main results about the functional have been presented [28]. Segmentation based on the Blake-Zisserman model, because of its second-order nature, is specifically suitable for addressing problems such as: (1) image inpainting [29], where the functional minimization allows for predicting partially occluded regions in an image and their contours continuation, and (2) 3D data segmentation [30], where the unique capability of the functional of tracing second-order edges (creases) allows for precisely locating planar objects (such as roof planes) in remote sensing 3D models of urban areas.

1.2. Variational approximation of the Blake-Zisserman relaxed functional

Implementing gradient descent of (3) with respect to the unknown free discontinuity sets is extremely difficult. Γ -convergence has shown to be fundamental to solve the problem of numerically computing a minimizer. This notion of convergence, suitable for functionals, has been introduced by [31]. For a deep treatment of this topic we refer to [32, 33]. The key point in Γ -convergence is that a specific functional, which may not have good properties for minimization, can be approximated by a sequence of regular functionals all admitting minimizers. The sequence of these approximate minimizers converges (in the classical sense) to a minimizer of the original objective functional.

Besides its importance as mathematical tool, Γ -convergence is very attractive also from a numerical point of view as it allows for the solution of several difficult numerical problems in Computer Vision, Physics, and many other fields. See for instance [33, 34].

Following the idea of Ambrosio and Tortorelli, in [35] a Γ -convergence result is proved for the BZ functional in dimension 1. A full proof in dimension 2 and a partial result for any dimension n is given by Ambrosio, Faina and March [36]. The authors, by properly adapting the techniques of [35] and [11], have introduced two auxiliary functions $s, z : \Omega \rightarrow [0, 1]$ (aimed at approximating the indicator functions of the discontinuity sets) to the model and proposed a Γ -convergence approximation of \mathcal{F} via the family of uniformly elliptic functionals

$$\begin{aligned} \mathcal{F}_\epsilon(s, z, u) = & \delta \int_{\Omega} z^2 |\nabla^2 u|^2 dx + \xi_\epsilon \int_{\Omega} (s^2 + o_\epsilon) |\nabla u|^2 dx \\ & + (\alpha - \beta) \int_{\Omega} \epsilon |\nabla s|^2 + \frac{1}{4\epsilon} (s - 1)^2 dx \\ & + \beta \int_{\Omega} \epsilon |\nabla z|^2 + \frac{1}{4\epsilon} (z - 1)^2 dx \\ & + \mu \int_{\Omega} |u - g|^2 dx, \end{aligned} \quad (5)$$

where $(s, z, u) \in [W^{1,2}(\Omega, [0, 1])]^2 \times W^{2,2}(\Omega) =: \mathcal{D}(\Omega)$. Here ϵ is the convergence continuous parameter, ξ_ϵ, o_ϵ are infinitesimals and the convergence is intended for $\epsilon \rightarrow 0$. To prove Γ -convergence, one has to show that for any $u \in GSBV^2(\Omega)$, $s \equiv 1, z \equiv 1$ the two following properties are verified:

Liminf inequality: for any sequence $\{(s_\epsilon, z_\epsilon, u_\epsilon)\}_{\epsilon>0} \subset \mathcal{D}(\Omega)$ that $[L^1(\Omega)]^3$ -converges to (s, z, u) it holds that $\mathcal{F}(u) \leq \liminf_{\epsilon \rightarrow 0} \mathcal{F}_\epsilon(s_\epsilon, z_\epsilon, u_\epsilon)$.

Limsup inequality: there exists a sequence $\{(s_\epsilon, z_\epsilon, u_\epsilon)\}_{\epsilon>0} \subset \mathcal{D}(\Omega)$ that $[L^1(\Omega)]^3$ -converges to (s, z, u) such that $\limsup_{\epsilon \rightarrow 0} \mathcal{F}_\epsilon(s_\epsilon, z_\epsilon, u_\epsilon) \leq \mathcal{F}(u)$.

By standard arguments of functional analysis it is possible to prove that for any $\epsilon > 0$ the functional \mathcal{F}_ϵ always admits a minimizing triplet. Let us denote it by $(s_\epsilon, z_\epsilon, u_\epsilon)$. By sending $\epsilon \rightarrow 0$, thanks to the compactness properties of the Γ -convergence, the sequence $\{(s_\epsilon, z_\epsilon, u_\epsilon)\}_{\epsilon>0}$ converges in the $[L^1(\Omega)]^3$ -norm to a triplet (s, z, u) where u is a minimizer of the limit functional \mathcal{F} and $s, z \equiv 1$ almost everywhere over Ω .¹

The constructive part of the Γ -convergence (Limsup inequality) provides us the tremendous advantage of keeping trace of the discontinuity sets S_u and $S_u \cup S_{\nabla u}$ via their regular function approximations. For a fixed $\epsilon > 0$, the two discontinuity sets, enjoying the regularity properties of $GSBV^2(\Omega)$ functions, are approximated by s_ϵ and z_ϵ (respectively) using a slicing argument and Coarea-formula for Lipschitz functions [36]. Let S be either S_u or $S_u \cup S_{\nabla u}$ and let us consider a 2-dimensional orthogonal slice of S (see Figure 2). The idea is to build a function σ_ϵ that is 0 in a tubular neighborhood of radius b_ϵ of the set S and that tends to 1 smoothly elsewhere. The tubular neighborhood shrinks as $\epsilon \rightarrow 0$. Formally the function σ_ϵ is defined as:

$$\sigma_\epsilon := \begin{cases} 0, & (S)_{b_\epsilon} \\ 1 - \eta_\epsilon, & \Omega \setminus (S)_{b_\epsilon + a_\epsilon} \\ h_\epsilon \circ \tau, & \text{elsewhere} \end{cases} \quad (6)$$

where $a_\epsilon, b_\epsilon, \eta_\epsilon$ are infinitesimals as $\epsilon \rightarrow 0$, $\tau(y) := \text{dist}(y, \bar{S})$ and $(S)_r := \{y \in \mathbb{R}^2 : \text{dist}(y, S) < r\}$. The function h_ϵ (the blue piece of function in Figure 2) is obtained as the solution of the differential problem $h' = (1 - h)/2\epsilon$, $h(b_\epsilon) = 0$, where $h(b_\epsilon + a_\epsilon) = 1 - \eta_\epsilon$. Exploiting the Schwartz inequality $a^2 + b^2 \geq 2ab$ it is possible to prove that such h_ϵ is energetically optimal in the class of the admissible functions (a general result is given in [12] and used for the approximation of discontinuity sets in [11, 36]). Because of the global minimization of \mathcal{F}_ϵ , the distance term $\mu|u_\epsilon - g|^2$ keeps the function u_ϵ close to g . High values of $|\nabla u_\epsilon|$ (associated to discontinuities of g) and high values of $|\nabla^2 u_\epsilon|$ (associated to crease points of g) force the transition of the functions s_ϵ and z_ϵ from 1 to 0. Elsewhere, the minimization of the two terms containing the differential operators causes the smoothing of g . We remark here the

¹In practice it is assumed that $\mathcal{H}^2(\{\sigma = 0\}) = 0$ and $0 \leq \mathcal{H}^1(\{\sigma = 0\}) < \infty$, for either $\sigma = s$ and $\sigma = z$.

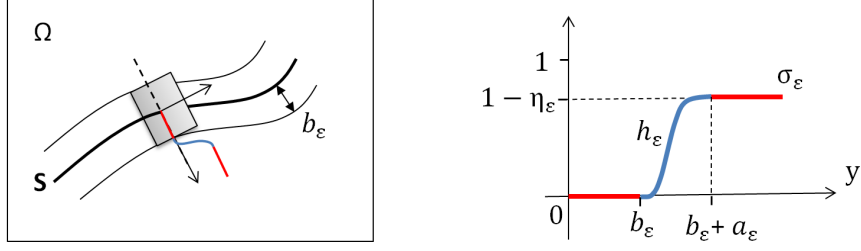


Figure 2: Slice section of the discontinuity set S and its approximation via the recovering function σ_ϵ realizing the Γ -convergence.

importance of the parameters $\delta, \mu, \alpha, \beta$, that control the ratio at which the whole mechanism described before takes place.

From the discussion above it follows that, for small values of ϵ , the computation of a minimizing triplet of (5) provides u_ϵ , an approximation of a real minimizer u of \mathcal{F} , and s_ϵ, z_ϵ , the functions that map the tubular neighborhoods of the discontinuity sets S_u and $S_{\nabla u} \cup S_u$, respectively. The price to pay for having such nice outputs is computational complexity. In the remainder of the paper we will show how the explicit minimization of (5) can be addressed in an efficient way by exploiting the nice properties of the functional and a compact formulation via finite-difference schemes enjoying good properties of convergence.

2. Numerical minimization of the Blake-Zisserman functional

In this section the numerical minimization of (5) is addressed. Firstly the functional is discretized and written in matricial form. Because of nice properties of the functional, the finite-difference discretization of the functional leads to a quadratic function with respect to each block variable when the others two are left fixed.

2.1. Discretization

A simple discretization technique, commonly used for computer vision problems (see for example [37, 38]), can be applied to the functional (5) in a straightforward way. The rectangular domain $\Omega \subset \mathbb{R}^2$ is discretized by a lattice of points $\Lambda = \{(it_x, jt_y); i = 1, \dots, N, j = 1, \dots, M\}$ with step sizes t_x and t_y on the x and y directions respectively, giving rise to a point grid Λ of size $n := NM$. Using the standard representation of grey-scale images as matrices, the values of the image g on the grid points (it_x, jt_y) are denoted g_{ij} . Similarly, the approximate values of the functions s, z, u on the grid points are denoted s_{ij}, z_{ij}, u_{ij} . Furthermore, for any function $v \in \{g, s, z, u\}$, we denote by \mathbf{v} the column vector of dimension n obtained from the corresponding matrix rearranging the elements v_{ij} by a column-wise vectorization. The function $w(i, j) := (j-1)N + i$ makes a bijective correspondence between the entry v_{ij} and its position in the vector \mathbf{v} . Shortly, $[\mathbf{v}]_{w(i,j)} = v_{ij}$. Given a vector \mathbf{v} , let us denote $\mathbf{R}_\mathbf{v}$ the diagonal matrix with diagonal entries equal to the elements of \mathbf{v} . Furthermore, we also denote \mathbf{v}^2 the vector of the squared coefficients of \mathbf{v} , i.e., $[\mathbf{v}^2]_i = ([\mathbf{v}]_i)^2$ and $\mathbf{e} := (1, 1, \dots, 1)^T$. The maximum value of the entries of a vector is denoted by $\|\mathbf{v}\|_\infty := \max_i [\mathbf{v}]_i$.

The first and second order differential operators appearing in the functional can be approximated via finite difference-schemes as follows

$$\begin{aligned}
 \partial_x v_{ij} &:= \frac{v_{i+1,j} - v_{i,j}}{t_x} = [\mathbf{D}_x \mathbf{v}]_{w(i,j)} \\
 \partial_y v_{ij} &:= \frac{v_{i,j+1} - v_{i,j}}{t_y} = [\mathbf{D}_y \mathbf{v}]_{w(i,j)} \\
 \partial_{xx} v_{ij} &:= \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{t_x^2} = [\mathbf{D}_{xx} \mathbf{v}]_{w(i,j)} \\
 \partial_{yy} v_{ij} &:= \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{t_y^2} = [\mathbf{D}_{yy} \mathbf{v}]_{w(i,j)} \\
 \partial_{xy} v_{ij} &:= \frac{1}{t_y} \left(\frac{v_{i+1,j+1} - v_{i,j+1}}{t_x} - \frac{v_{i+1,j} - v_{i,j}}{t_x} \right) = [\mathbf{D}_{xy} \mathbf{v}]_{w(i,j)}
 \end{aligned} \tag{7}$$

for $i = 1, \dots, N$ and $j = 1, \dots, M$. By assuming zero boundary conditions ($v_{0,j} = v_{N+1,j} = v_{i,0} = v_{i,M+1} = 0$) as in [36], the above matrices $\mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_{xx}, \mathbf{D}_{yy}$ are given by

$$\begin{aligned}\mathbf{D}_x &:= \frac{1}{t_x} \mathbf{I}_M \otimes \mathbf{A}_N^1 & \mathbf{D}_y &:= \frac{1}{t_y} \mathbf{A}_M^1 \otimes \mathbf{I}_N \\ \mathbf{D}_{xx} &:= \frac{1}{t_x^2} \mathbf{I}_M \otimes \mathbf{A}_N^2 & \mathbf{D}_{yy} &:= \frac{1}{t_y^2} \mathbf{A}_M^2 \otimes \mathbf{I}_N \\ \mathbf{D}_{xy} &:= \mathbf{D}_y \mathbf{D}_x = \mathbf{D}_x \mathbf{D}_y\end{aligned}$$

where \otimes is the Kronecker product. Here \mathbf{I}_K denotes the identity matrix of dimension K and $\mathbf{A}_K^1, \mathbf{A}_K^2$ are square matrices of order $K > 0$, representing a forward-scheme approximating first-derivative and a central-scheme approximating second-derivative respectively²:

$$\mathbf{A}_K^1 := \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & -1 \end{pmatrix}, \quad \mathbf{A}_K^2 := \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}. \quad (8)$$

By using the following approximations over each grid point

$$\begin{aligned}|\nabla^2 v_{ij}|^2 &= ([\mathbf{D}_{xx} \mathbf{v}]_{w(i,j)})^2 + ([\mathbf{D}_{yy} \mathbf{v}]_{w(i,j)})^2 + 2([\mathbf{D}_{xy} \mathbf{v}]_{w(i,j)})^2, \\ |\nabla v_{ij}|^2 &= ([\mathbf{D}_x \mathbf{v}]_{w(i,j)})^2 + ([\mathbf{D}_y \mathbf{v}]_{w(i,j)})^2,\end{aligned}$$

we can approximate the integral over Ω with a simple 2-D composite rectangular rule, obtaining the following discrete form of the functional (5):

$$\begin{aligned}F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u}) &:= t_x t_y \left\{ \delta [\mathbf{u}^T \mathbf{D}_{xx}^T \mathbf{R}_{z^2} \mathbf{D}_{xx} \mathbf{u} + \mathbf{u}^T \mathbf{D}_{yy}^T \mathbf{R}_z \mathbf{D}_{yy} \mathbf{u} + 2\mathbf{u}^T \mathbf{D}_{xy}^T \mathbf{R}_{z^2} \mathbf{D}_{xy} \mathbf{u}] + \right. \\ &+ \xi_\epsilon [\mathbf{u}^T \mathbf{D}_x^T \mathbf{R}_{s^2} \mathbf{D}_x \mathbf{u} + \mathbf{u}^T \mathbf{D}_y^T \mathbf{R}_{s^2} \mathbf{D}_y \mathbf{u}] + \\ &+ (\alpha - \beta) [\epsilon (\mathbf{s}^T \mathbf{D}_x^T \mathbf{D}_x \mathbf{s} + \mathbf{s}^T \mathbf{D}_y^T \mathbf{D}_y \mathbf{s}) + \frac{1}{4\epsilon} (\mathbf{s} - \mathbf{e})^T (\mathbf{s} - \mathbf{e})] + \\ &+ \beta [\epsilon (\mathbf{z}^T \mathbf{D}_x^T \mathbf{D}_x \mathbf{z} + \mathbf{z}^T \mathbf{D}_y^T \mathbf{D}_y \mathbf{z}) + \frac{1}{4\epsilon} (\mathbf{z} - \mathbf{e})^T (\mathbf{z} - \mathbf{e})] + \\ &\left. + \mu (\mathbf{u} - \mathbf{g})^T (\mathbf{u} - \mathbf{g}) \right\}. \quad (9)\end{aligned}$$

Here, with abuse of notation, \mathbf{R}_{s^2} is the diagonal matrix composed by the elements of the vector $\mathbf{s}^2 + o_\epsilon$ (instead of \mathbf{s}^2). Globally this functional is not convex, but it is quadratic with respect to each block of variables $\mathbf{s}, \mathbf{z}, \mathbf{u}$. The terms of F_ϵ containing \mathbf{s} or \mathbf{z} depend only on \mathbf{u} . On the other hand, the terms containing \mathbf{u} depend on \mathbf{s} and \mathbf{z} . Indeed, by fixing the variable \mathbf{u} or the other two variables \mathbf{s} and \mathbf{z} , we can write

$$\begin{aligned}F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u}) &= t_x t_y \left\{ \frac{1}{2} (\mathbf{s}^T \mathbf{z}^T) \begin{pmatrix} \mathbf{A}_s & 0 \\ 0 & \mathbf{A}_z \end{pmatrix} \begin{pmatrix} \mathbf{s} \\ \mathbf{z} \end{pmatrix} - (\mathbf{s}^T \mathbf{z}^T) \begin{pmatrix} \mathbf{b}_s \\ \mathbf{b}_z \end{pmatrix} + \mathbf{c}_{sz} \right\} \\ F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u}) &= t_x t_y \left\{ \frac{1}{2} \mathbf{u}^T \mathbf{A}_u \mathbf{u} - \mathbf{u}^T \mathbf{b}_u + \mathbf{c}_u \right\}\end{aligned} \quad (10)$$

²The implementation of homogeneous Neumann boundary conditions follows straightforwardly by replacing the entries: $[\mathbf{A}_K^1]_{K,K} = 0$ and $[\mathbf{A}_K^2]_{1,1} = [\mathbf{A}_K^2]_{K,K} = -1$.

where $\mathbf{A}_s = \mathbf{A}_s(\mathbf{u})$, $\mathbf{A}_z = \mathbf{A}_z(\mathbf{u})$, $\mathbf{A}_u = \mathbf{A}_u(\mathbf{s}, \mathbf{z})$ and $\mathbf{b}_s, \mathbf{b}_z, \mathbf{b}_u$ are given by

$$\begin{aligned}
\mathbf{A}_s &= 2\xi_\epsilon \mathbf{R}_{|\nabla \mathbf{u}|^2} + 2\epsilon(\alpha - \beta)(\mathbf{D}_x^T \mathbf{D}_x + \mathbf{D}_y^T \mathbf{D}_y) + \frac{\alpha - \beta}{2\epsilon} \mathbf{I} \\
\mathbf{b}_s &= \frac{\alpha - \beta}{2\epsilon} \mathbf{e} \\
\mathbf{A}_z &= 2\delta \mathbf{R}_{|\nabla^2 \mathbf{u}|^2} + 2\epsilon\beta(\mathbf{D}_x^T \mathbf{D}_x + \mathbf{D}_y^T \mathbf{D}_y) + \frac{\beta}{2\epsilon} \mathbf{I} \\
\mathbf{b}_z &= \frac{\beta}{2\epsilon} \mathbf{e} \\
\mathbf{A}_u &= 2\delta(\mathbf{D}_{xx}^T \mathbf{R}_z \mathbf{D}_{xx} + \mathbf{D}_{yy}^T \mathbf{R}_z \mathbf{D}_{yy} + 2\mathbf{D}_{xy}^T \mathbf{R}_z \mathbf{D}_{xy}) + 2\xi_\epsilon(\mathbf{D}_x^T \mathbf{R}_{s^2} \mathbf{D}_x + \mathbf{D}_y^T \mathbf{R}_{s^2} \mathbf{D}_y) + 2\mu \mathbf{I} \\
\mathbf{b}_u &= 2\mu \mathbf{g}
\end{aligned} \tag{11}$$

with $|\nabla \mathbf{u}|^2 := (\mathbf{D}_x \mathbf{u})^2 + (\mathbf{D}_y \mathbf{u})^2$ and $|\nabla^2 \mathbf{u}|^2 := (\mathbf{D}_{xx} \mathbf{u})^2 + (\mathbf{D}_{yy} \mathbf{u})^2 + 2(\mathbf{D}_{xy} \mathbf{u})^2$. Vectors \mathbf{c}_{sz} and \mathbf{c}_u are constant, thus irrelevant for the minimization. In view of the terms $\frac{\alpha - \beta}{2\epsilon} \mathbf{I}$, $\frac{\beta}{2\epsilon} \mathbf{I}$ and $2\mu \mathbf{I}$, with $\epsilon, \mu, \beta, \alpha - \beta > 0$, the matrices $\mathbf{A}_s, \mathbf{A}_z, \mathbf{A}_u$ are symmetric and positive definite. Furthermore, these matrices are very sparse and structured: \mathbf{A}_s and \mathbf{A}_z are block tridiagonal matrices where the diagonal blocks are tridiagonal and the off-diagonal blocks are diagonal. \mathbf{A}_u is a block five matrix, with at most 13 nonzero entries for each row.

In the following, for notation convenience, a generic point in \mathbb{R}^{3n} is represented by either \mathbf{y} or $(\mathbf{s}, \mathbf{z}, \mathbf{u})$. This makes a simple correspondence of the type: $\mathbf{y}_1 = \mathbf{s}$, $\mathbf{y}_2 = \mathbf{z}$ and $\mathbf{y}_3 = \mathbf{u}$. Accordingly, throughout the paper a similar correspondence is used for denoting operators/vectors related to a specific block of variables. For example: $\mathbf{A}_s = \mathbf{A}_1$, $\mathbf{A}_z = \mathbf{A}_2$, $\mathbf{A}_u = \mathbf{A}_3$, and $\mathbf{b}_s = \mathbf{b}_1$, $\mathbf{b}_z = \mathbf{b}_2$, $\mathbf{b}_u = \mathbf{b}_3$ etc. Furthermore, we denote the gradient of F_ϵ with respect to the generic block of variables \mathbf{y}_i , computed at $\bar{\mathbf{y}}$, by $\nabla_i F_\epsilon(\bar{\mathbf{y}}) = \mathbf{A}_i \bar{\mathbf{y}}_i - \mathbf{b}_i$.

2.2. Minimization method

We address here the minimization of the function $F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u})$. Firstly we can observe that the objective function is continuously differentiable, and in view of the positive definiteness of matrices \mathbf{A}_i , $i = 1, 2, 3$, it is strictly convex with respect to each block component \mathbf{y}_i , when the others are left fixed. Let us prove that F_ϵ is also coercive.

Lemma 1. *The function $F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u})$ is coercive in \mathbb{R}^{3n} .*

Proof. Given a sequence $\mathbf{y}^k = (\mathbf{s}^k, \mathbf{z}^k, \mathbf{u}^k) \in \mathbb{R}^{3n}$ such that $\lim_{k \rightarrow \infty} \|\mathbf{y}^k\| = +\infty$ the lemma is proved if we show that $\lim_{k \rightarrow \infty} F_\epsilon(\mathbf{y}^k) = +\infty$. The hypothesis on \mathbf{y}^k implies that there exists a coordinate index $j \in \{1, 2, \dots, 3n\}$ such that $\lim_{k \rightarrow \infty} |\mathbf{y}_j^k| = +\infty$. If $1 \leq j \leq n$, then j corresponds to an index i in the \mathbf{s} block and we have that $\lim_{k \rightarrow \infty} |\mathbf{s}_i^k| = +\infty$. In particular $\lim_{k \rightarrow \infty} (\mathbf{s}_i^k - 1)^2 = +\infty$ and since $(\mathbf{s}_i^k - 1)^2 \leq (\mathbf{s}^k - \mathbf{e})^T (\mathbf{s}^k - \mathbf{e})$ we also have that $\lim_{k \rightarrow \infty} F_\epsilon(\mathbf{y}^k) = +\infty$. A similar argument works in the case of $n+1 \leq j \leq 2n$, where j corresponds to an index in the \mathbf{z} block. If $2n+1 \leq j \leq 3n$ then j corresponds to an index i in the \mathbf{u} block. From $\lim_{k \rightarrow \infty} |\mathbf{u}_i^k| = +\infty$ we have that $\lim_{k \rightarrow \infty} (\mathbf{u}_i^k - \mathbf{g}_i)^2 = +\infty$ and, since $(\mathbf{u}_i^k - \mathbf{g}_i)^2 \leq (\mathbf{u}^k - \mathbf{g})^T (\mathbf{u}^k - \mathbf{g})$, then we have again $\lim_{k \rightarrow \infty} F_\epsilon(\mathbf{y}^k) = +\infty$. \square

By using a truncation argument we have that the functions \mathbf{s} and \mathbf{z} that minimize the objective functional F_ϵ belong to a specific compact subset of \mathbb{R}^{3n} . In fact, given $\tau(\mathbf{v}) := 0 \vee \mathbf{v} \wedge 1$, i.e., the function that truncates \mathbf{v} at 0 and 1, one can see that for any triplet $(\mathbf{s}, \mathbf{z}, \mathbf{u}) \in \mathbb{R}^{3n}$

$$F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u}) \geq F_\epsilon(\tau(\mathbf{s}), \tau(\mathbf{z}), \mathbf{u}) \tag{12}$$

holds (in fact, the truncation of \mathbf{s} and \mathbf{z} does not increase their gradients). It follows that for a minimizer $(\mathbf{s}, \mathbf{z}, \mathbf{u})$ the functions $\mathbf{s}, \mathbf{z} \in [0, 1]^n$. A similar argument is used for the Mumford-Shah functional in [11] to prove that the optimal u is such that $\|u\|_\infty \leq \|g\|_\infty$ (maximum principle). Unfortunately, the Hessian component of the Blake-Zisserman functional does not allow us to exploit the maximum principle and an explicit bound for the function \mathbf{u} cannot be calculated, [28].

The structure of the function $F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u})$ justifies the use of a block decomposition method, such as the **block non-linear Gauss-Seidel (GS)** method. Starting from $(\mathbf{s}^0, \mathbf{z}^0, \mathbf{u}^0)$, in view of (10), the method has the following form:

$$\begin{cases} \mathbf{s}^{k+1} = \arg \min_{\mathbf{s}} F_\epsilon(\mathbf{s}, \mathbf{z}^k, \mathbf{u}^k) \\ \mathbf{z}^{k+1} = \arg \min_{\mathbf{z}} F_\epsilon(\mathbf{s}^{k+1}, \mathbf{z}, \mathbf{u}^k) \\ \mathbf{u}^{k+1} = \arg \min_{\mathbf{u}} F_\epsilon(\mathbf{s}^{k+1}, \mathbf{z}^{k+1}, \mathbf{u}) \end{cases} . \quad (13)$$

Because of the the block diagonal structure of the matrix related to the quadratic functional obtained by fixing \mathbf{u} in the first subproblem in (10), \mathbf{s}^{k+1} and \mathbf{z}^{k+1} can be obtained by further subdividing this subproblem into two independent tasks.

Theorem 6.2 in [39] assures that the algorithm generates a sequence $\{\mathbf{s}^k, \mathbf{z}^k, \mathbf{u}^k\}$ such that every limit point is a stationary point of F_ϵ . Because of coercivity, the level sets $\mathcal{L}_\alpha = \{(\mathbf{s}, \mathbf{z}, \mathbf{u}) : F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u}) \leq \alpha\}$ are compact for every $\alpha > 0$. Since in particular \mathcal{L}_{α^0} is compact, where $\alpha^0 = F_\epsilon(\mathbf{s}^0, \mathbf{z}^0, \mathbf{u}^0)$, the theorem also guarantees that $\nabla F_\epsilon(\mathbf{s}^k, \mathbf{z}^k, \mathbf{u}^k) \rightarrow 0$ as $k \rightarrow \infty$ and there exists at least a limit point that is a stationary point of F_ϵ .

Nevertheless, any step of the non-linear Gauss-Seidel method requires the solution of three large and sparse systems. Although such systems can be efficiently solved by the Preconditioned Conjugate Gradient (PCG) algorithm, the whole method could be too expensive, above all for large images.

Therefore, we propose to solve our minimization problem with a **block coordinate descent algorithm (BCDA)**, based on the line search technique described in [39]. The basic idea of the method is to cyclically determine for each block variable a descent direction \mathbf{d}_i by few iterations of an iterative solver; then by an Armijo-type procedure a suitable step size is devised to assure a sufficient decrease of the objective function along this direction with respect to the i -th block variable, when the remaining variables are fixed.

In view of the special structure of $F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u})$, for each subproblem in (10) we can cyclically obtain a descent direction by few iterations of the PCG method applied to the linear system $\mathbf{A}_i^k \mathbf{d}_i = \mathbf{b}_i - \mathbf{A}_i^k \mathbf{y}_i^k$. In the first subproblem, \mathbf{d}_s^k and \mathbf{d}_z^k can be independently obtained. Furthermore, in view of the quadratic structure of the objective function with respect to each block of variables when the others are fixed, the step-lengths along the computed descent directions can be determined without having to use an Armijo-type procedure. Indeed, it is well known that, for a symmetric positive definite quadratic function, a sufficient decrease is assured when the step size α_i^k is chosen as $\gamma_i \frac{-(\mathbf{A}_i^k \mathbf{y}_i^k - \mathbf{b}_i)^T \mathbf{d}_i^k}{\mathbf{d}_i^{kT} \mathbf{A}_i^k \mathbf{d}_i^k} = \gamma_i \frac{-\nabla_i F_\epsilon(\mathbf{y}^k)^T \mathbf{d}_i^k}{\mathbf{d}_i^{kT} \mathbf{A}_i^k \mathbf{d}_i^k}$, with $0 < \gamma_i < 2$; in particular, for $\gamma_i = 1$, we obtain the exact one-dimensional minimizer of the quadratic function along the direction \mathbf{d}_i^k . As consequence, we can devise a specialized version of the block-coordinate descent algorithm for $F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u})$; such scheme is outlined in Algorithm 1.

2.2.1. Gradient related search directions

In order to obtain convergence results for BCDA, the vectors \mathbf{d}_i^k , $i = 1, 2, 3$, have to be chosen so that they are *gradient related search directions*. Equivalently, they have to satisfy the following assumption:

- (a) $\mathbf{d}_i^k = 0$ if and only if $\nabla_i F_\epsilon(\mathbf{y}^k) = 0$,
- (b) there exists a forcing function $\sigma_i : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ such that:

$$\frac{\nabla_i F_\epsilon(\mathbf{y}^k)^T \mathbf{d}_i^k}{\|\mathbf{d}_i^k\|} \leq -\sigma_i(\|\nabla_i F_\epsilon(\mathbf{y}^k)\|) \quad (14)$$

for all k satisfying $\nabla_i F_\epsilon(\mathbf{y}^k) \neq 0$.

In order to determine a gradient related search direction when $\mathbf{A}_i^k \mathbf{y}_i^k - \mathbf{b}_i \neq 0$, we can execute several iterations of PCG method for the symmetric positive linear system $\mathbf{A}_i^k \mathbf{d}_i = \mathbf{b}_i - \mathbf{A}_i^k \mathbf{y}_i^k$ by stopping the algorithm when the residual $\mathbf{r}^\ell = \mathbf{b}_i - \mathbf{A}_i^k \mathbf{y}_i^k - \mathbf{A}_i^k \mathbf{d}_i^\ell$ of the system at the ℓ -th iteration satisfies the rule

$$\|\mathbf{r}^\ell\| \leq \eta_i \|\mathbf{A}_i^k \mathbf{y}_i^k - \mathbf{b}_i\| \quad \eta_i \leq \frac{c}{\sqrt{K((\mathbf{A}_i^k)^{-1})}} \quad (15)$$

Algorithm 1 BCDA

Step 0: Given $\mathbf{s}^0, \mathbf{z}^0, \mathbf{u}^0, \rho_{sz} > 0, \rho_u > 0, \gamma_s \in (0, 2), \gamma_z \in (0, 2), \gamma_u \in (0, 2)$;

Step 1: $k = 0$;

Step 2: Inexact minimization with respect to \mathbf{s} and \mathbf{z} :

- compute the search directions \mathbf{d}_s^k and \mathbf{d}_z^k ;
- compute $\alpha_s^k = \gamma_s \frac{-(A_s^k \mathbf{s}^k - \mathbf{b}_s)^T \mathbf{d}_s^k}{\mathbf{d}_s^{kT} A_s^k \mathbf{d}_s^k}$, $\alpha_z^k = \gamma_z \frac{-(A_z^k \mathbf{z}^k - \mathbf{b}_z)^T \mathbf{d}_z^k}{\mathbf{d}_z^{kT} A_z^k \mathbf{d}_z^k}$
- update $\mathbf{s}^{k+1} = \mathbf{s}^k + \alpha_s^k \mathbf{d}_s^k$; $\mathbf{z}^{k+1} = \mathbf{z}^k + \alpha_z^k \mathbf{d}_z^k$.

Step 3: Inexact minimization with respect to \mathbf{u} :

- compute the search directions \mathbf{d}_u^k ;
- compute $\alpha_u^k = \gamma_u \frac{-(A_u^k \mathbf{u}^k - \mathbf{b}_u)^T \mathbf{d}_u^k}{\mathbf{d}_u^{kT} A_u^k \mathbf{d}_u^k}$
- update $\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha_u^k \mathbf{d}_u^k$.

Step 4: Set $k = k + 1$ and go to Step 2;

where $K((A_i^k)^{-1})$ is the spectral condition number of $(A_i^k)^{-1}$ and $c < 1$. We observe that $K((A_i^k)^{-1}) = K(A_i^k)$ is bounded by a positive constant L in \mathcal{L}_{α^0} . Then, we set $\mathbf{d}_i^{\bar{\ell}} = \mathbf{d}_i^k$. We can prove that $\mathbf{d}_i^{\bar{\ell}}$ satisfies the assumption (14). Indeed, recalling that $\|\cdot\|_{A^{-1}}$ denotes the A^{-1} -norm (that is $\|\mathbf{x}\|_{A^{-1}} = \sqrt{\mathbf{x}^T A^{-1} \mathbf{x}}$), for $\mathbf{d}_i^{\bar{\ell}} = \mathbf{d}_i^k$ we have

$$\begin{aligned} \frac{\nabla_i F_\epsilon(\mathbf{y}^k)^T \mathbf{d}_i^k}{\|\mathbf{d}_i^k\|} &\leq \frac{(\mathbf{A}_i^k \mathbf{y}_i^k - \mathbf{b}_i)^T (A_i^k)^{-1} A_i^k \mathbf{d}_i^{\bar{\ell}}}{\|\mathbf{d}_i^{\bar{\ell}}\|} + \\ &\quad \frac{1}{2\|\mathbf{d}_i^{\bar{\ell}}\|} (\|\mathbf{r}^{\bar{\ell}} + \nabla_i F_\epsilon(\mathbf{y}^k)\|_{(A_i^k)^{-1}}^2 + \|\nabla_i F_\epsilon(\mathbf{y}^k)\|_{(A_i^k)^{-1}}^2 - \|\nabla_i F_\epsilon(\mathbf{y}^k)\|_{(A_i^k)^{-1}}^2) \\ &= \frac{1}{2\|\mathbf{d}_i^{\bar{\ell}}\|} \left(\|\mathbf{r}^{\bar{\ell}}\|_{(A_i^k)^{-1}}^2 - \|\nabla_i F_\epsilon(\mathbf{y}^k)\|_{(A_i^k)^{-1}}^2 \right) \\ &\leq \frac{1}{2\|\mathbf{d}_i^{\bar{\ell}}\|} \left(\lambda_{\max}((A_i^k)^{-1}) \|\mathbf{r}^{\bar{\ell}}\|^2 - \lambda_{\min}((A_i^k)^{-1}) \|\nabla_i F_\epsilon(\mathbf{y}^k)\|^2 \right) \end{aligned} \quad (16)$$

$$\leq \frac{1}{2\|\mathbf{d}_i^{\bar{\ell}}\|} \left(\lambda_{\max}((A_i^k)^{-1}) \eta_i^2 - \lambda_{\min}((A_i^k)^{-1}) \right) \|\nabla_i F_\epsilon(\mathbf{y}^k)\|^2 \quad (17)$$

where (16) follows from the well-known inequalities

$$\lambda_{\min}((A_i^k)^{-1}) \|\mathbf{x}\|^2 \leq \mathbf{x}^T (A_i^k)^{-1} \mathbf{x} \leq \lambda_{\max}((A_i^k)^{-1}) \|\mathbf{x}\|^2$$

for any \mathbf{x} , and the inequality (17) follows from (15). Furthermore, the bound on η_i implies $(\lambda_{\max}((A_i^k)^{-1}) \eta_i^2 - \lambda_{\min}((A_i^k)^{-1})) < 0$; then, since $\mathbf{d}_i^{\bar{\ell}} = (\mathbf{A}_i^k)^{-1} (-\nabla_i F_\epsilon(\mathbf{y}^k) - \mathbf{r}^{\bar{\ell}})$, we have

$$\begin{aligned} \|\mathbf{d}_i^{\bar{\ell}}\| &\leq \|(\mathbf{A}_i^k)^{-1} (\|\nabla_i F_\epsilon(\mathbf{y}^k)\| + \|\mathbf{r}^{\bar{\ell}}\|)\| \\ &\leq \lambda_{\max}((\mathbf{A}_i^k)^{-1}) (1 + \eta_i) \|\nabla_i F_\epsilon(\mathbf{y}^k)\| \end{aligned}$$

Using this inequality in (17), we can conclude that

$$\begin{aligned} \frac{\nabla_i F_\epsilon(\mathbf{y}^k)^T \mathbf{d}_i^k}{\|\mathbf{d}_i^k\|} &\leq \frac{1}{2(1 + \eta_i)} \left(\eta_i^2 - \frac{1}{K((\mathbf{A}_i^k)^{-1})} \right) \|\nabla_i F_\epsilon(\mathbf{y}^k)\| \\ &\leq \frac{c^2 - 1}{2L} \|\nabla_i F_\epsilon(\mathbf{y}^k)\|, \end{aligned} \quad (18)$$

where $c^2 - 1 < 0$. Consequently, in the level set \mathcal{L}_{α^0} , the search directions \mathbf{d}_s^k , \mathbf{d}_z^k and \mathbf{d}_u^k generated by a PCG method with stopping rule (15) are *gradient related search directions*. Therefore, by Theorem 7.1 in [39], we can affirm that for BCDA the same convergence results hold as for block non-linear Gauss Seidel method, that is $\nabla F_\epsilon(\mathbf{u}^k, \mathbf{z}^k, \mathbf{z}^k) \rightarrow 0$ as $k \rightarrow \infty$ and there exists at least a limit point in \mathcal{L}_{α^0} that is a stationary point of F_ϵ .

2.2.2. Algorithm parameters and preconditioning

We observe that from the practical point of view the computation of the condition number of $(\mathbf{A}_i^k)^{-1}$ (which equals that of \mathbf{A}_i^k), can be avoided. Indeed, in view of the condition (15) on η_i , it is sufficient to have an upper bound for $K(\mathbf{A}_i^k)$. Then, using the inequalities between the matrix norms, we have $\lambda_{\max}(\mathbf{A}_i^k) \leq \|\mathbf{A}_i^k\|_\infty$. Since \mathbf{A}_s and \mathbf{A}_z are strictly diagonally dominant, the first Gerschgorin's theorem [40] enables us to determine as lower bound for the minimum eigenvalue the intersection between the union of the Gerschgorin circles and the x -axis of \mathbb{R}^2 , given by $\tilde{\lambda}_i^k = \min_t((\mathbf{A}_i^k)_{tt} - \sum_{v \neq t} |(\mathbf{A}_i^k)_{vt}|)$ for $i = 1, 2$. For the matrix \mathbf{A}_3 , a lower bound for the minimum eigenvalue is the value $\tilde{\lambda}_3^k = 2\mu$. Therefore, we have

$$K(\mathbf{A}_i^k) = \frac{\lambda_{\max}(\mathbf{A}_i^k)}{\lambda_{\min}(\mathbf{A}_i^k)} \leq \frac{\|\mathbf{A}_i^k\|_\infty}{\tilde{\lambda}_i^k}, \quad (19)$$

consequently, we can set

$$\eta_i = \sqrt{\frac{\tilde{\lambda}_i^k}{\|\mathbf{A}_i^k\|_\infty}}. \quad (20)$$

From the computational point of view, the quadratic structure of F_ϵ restricted to any block variable \mathbf{y}_i implies that the condition related to the Armijo rule that has to be verified to accept the step size $\alpha_i^k = \max_{j \geq 0} \{\delta_i^j \Delta_i^k\}$ can be

$$\alpha_i^k \leq 2(\gamma_i - 1) \frac{(A_i^k \mathbf{y}_i^k - \mathbf{b}_i)^T \mathbf{d}_i^k}{(\mathbf{d}_i^k)^T A_i^k \mathbf{d}_i^k}. \quad (21)$$

Furthermore, since \mathbf{A}_s and \mathbf{A}_z are block tridiagonal matrices, an inexpensive diagonal preconditioner enable us to satisfy the stopping rule (15) with a very few iterations, as shown by numerical experiments in the Section 3.

For the linear system related to matrix \mathbf{A}_u we can use a diagonal preconditioner or a block diagonal preconditioner. In this last case, each diagonal block is a tridiagonal matrix that can be easily factorized by the Cholesky algorithm. Although the factorization can be calculated in advance, PCG requires the factorization of the preconditioner and, at each iteration, the solution of bidiagonal lower and upper systems; thus PCG coupled with block preconditioner becomes effective with respect to the version with diagonal preconditioner for large order of the system and only when high accuracy is required. Furthermore, unless for the first outer iteration, we use as starting vector of the PCG algorithm in Step 2 and 3 of BCDA the direction computed at the previous outer iteration. For the first iteration, the starting vector is the null vector.

In the numerical experiments, standard values equal to 1 are set for γ_s and γ_z ; we set $\gamma_u = 1.5$, since for this value we obtain a slightly better performance.

2.2.3. Initialization and stopping criteria

The objective functional to minimize is non-convex; thus, the significance of the solution returned by the iterative method (a stationary point) strongly depends on the choice of the first iterates. Using prior knowledge on the properties of the theoretical solution, an effective choice of the initial values can be made [36]. Being the functions $s, z \equiv 1$

almost everywhere over Ω (in the limit case), we set the corresponding variables to $\mathbf{s}^0, \mathbf{z}^0 \equiv \mathbf{e}$. Since the function u is an approximation of the input image g , we assume that an energetically convenient initialization of the corresponding variable is $\mathbf{u}^0 = \mathbf{g}$.

In the experiments described in the next section, the algorithms are stopped at the iteration k such that the relative variation of the energy satisfies the condition

$$\left| \frac{F_\epsilon(\mathbf{u}^k, \mathbf{s}^k, \mathbf{z}^k) - F_\epsilon(\mathbf{u}^{k-1}, \mathbf{s}^{k-1}, \mathbf{z}^{k-1})}{F_\epsilon(\mathbf{u}^k, \mathbf{s}^k, \mathbf{z}^k)} \right| < TOL_F, \quad (22)$$

where TOL_F is a fixed tolerance.

3. Numerical results

In this section we present the results of an extensive numerical experimentation aimed at assessing different properties of the proposed block-coordinate descent method applied to (9). In Section 3.1 the proposed BCDA is compared with the GS in order to evaluate its performance both in terms of efficiency and accuracy. In Section 3.2, by focusing the attention on the noise reduction properties of the model, we compare the performance of the BCDA when a Point-Diagonal and a Block-Diagonal preconditioner is used for the solution of the PCG related to the linear systems involving \mathbf{A}_u . In Section 3.3 we make a discussion on the choice of the Γ -convergence parameter ϵ , which may critically affect the quality of the detection of first and second order discontinuities.

Very different datasets are considered in the tests, including both real and synthetic images and also Digital Surface Models (DSMs) obtained from remote sensing LiDAR (Light Detection and Ranging) data [41]. DSMs are obtained from airborne LiDAR point clouds by interpolation over a regular planimetric grid. The value of the DSM on each pixel (grid point) corresponds to the height of the object hit by the laser pulse. In particular DSMs are very attractive as they represent the *real geometry* of the objects instead of the *light geometry* provided by gray-scale images.

All tests are performed using MATLAB[®] on a standard workstation. Hardware is Intel(R) Core(TM) i5-4750 CPU @ 3.20 GHz, 8.00 GB Ram. For all numerical tests that follow, some common parameters to control convergence of outer/inner iterations are used. The algorithms are always stopped at the iteration k such that the corresponding relative variation of the energy (22) is less than $TOL_F = 10^{-3}$. A maximum number of outer iterations is also fixed as stopping criterion to 30. It is worth noting that in all computations this bound has never been reached. Regarding the solution of internal PCGs, we fixed a maximum number of iterations to 1000. It has been observed in a very large number of tests that, the linear systems involving $\mathbf{A}_s, \mathbf{A}_z$ are solved within an inner tolerance TOL_{PCG} in no more than 3 iterations in the case of GS, and in only 1 iteration in the case of BCDA. Other parameters and tolerances are explicitly specified in the tests.

3.1. Comparison of GS and BCDA performance

In this section we show how the proposed BCDA produces accurate solutions by also significantly reducing computational time if compared with a GS method. In order to compare the effectiveness of the two methods, we compute an *ideal* solution $\mathbf{s}^*, \mathbf{z}^*, \mathbf{u}^*$ by performing a lot of iterations of the GS method, i.e., by running GS until it reaches stagnation since all PCGs (with a very strong relative tolerance $TOL_{PCG} = 10^{-10}$) do not make any progress. For the inner PCGs in the GS method, a strong relative tolerance $TOL_{PCG} = 10^{-8}$ is required; smaller tolerances have never resulted in lower minimizers. Tolerances for the solution of inner iterations of the PCGs in the BCDA are theoretically defined by (15) and (20). Furthermore, we propose also an hybrid version of the BCDA, to which we will refer to as BCDA_c, where the number of iterations for solving the inner PCGs is capped at 10 (in view of a previous remark this affects only the solution of the system involving \mathbf{A}_u). The main idea behind this choice it to show that actually just few steps of the inner solvers are needed to reach satisfying results at lower computational cost, even though a small (negligible) amount of accuracy is payed. Since the performance of the method when a block-diagonal preconditioner is used is analyzed in detail in Section 3.2, in all tests conducted in this section a diagonal preconditioner is used for the PCGs.

We consider as test problems the four datasets represented in Figure 3. The first image is a 600×600 portion of the oil painting ‘‘Girl with a Pearl Earring’’ by Johannes van der Meer. We refer this dataset to as *pearl*. Then, two

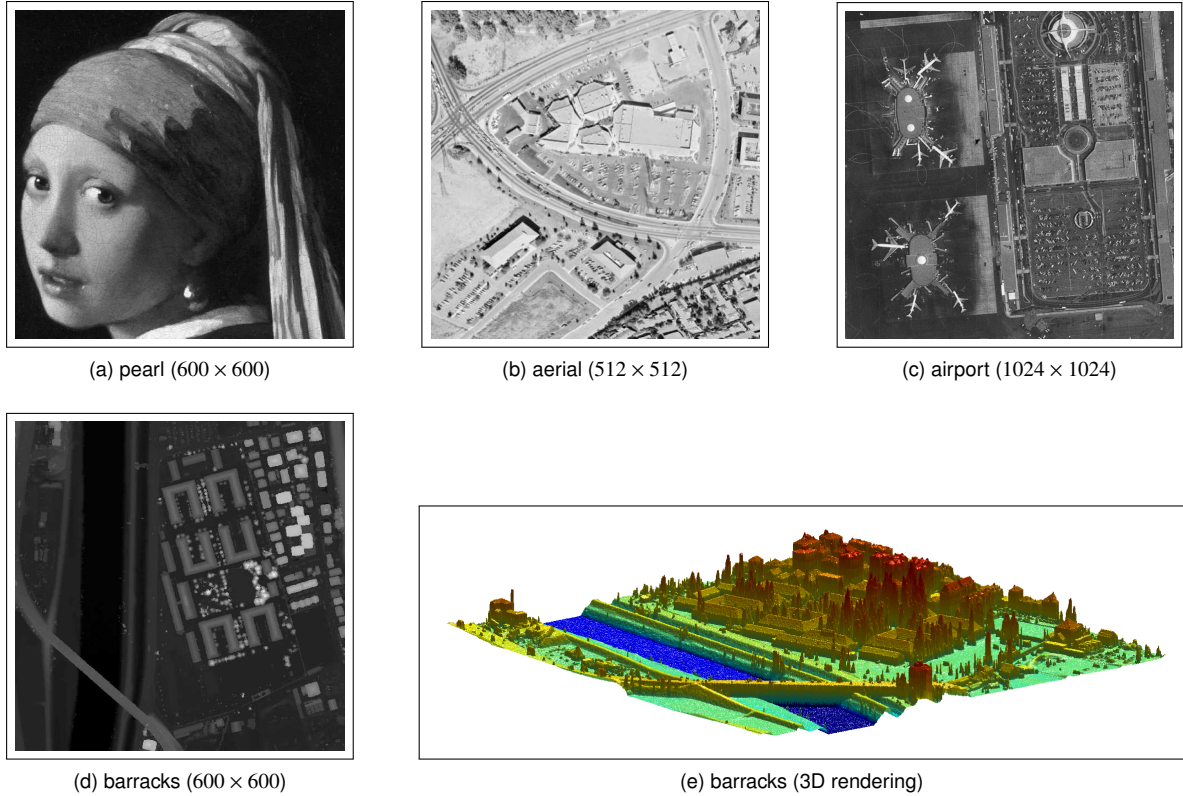


Figure 3: Datasets of the experiment including three gray-scale images and a digital surface model obtained from airborne LiDAR points acquired over Trento, Italy.

gray-scale images³ are considered. The first image is called *aerial* and it has size of 512×512 pixels. The second image is called *airport* and it has size of 1024×1024 pixels. The last dataset is a subset of a DSM of Trento⁴, in Trentino Alto-Adige, Italy. The considered scene presents some old barracks and surrounding area. The size of the grid is 600×600 , spatial resolution is 1mt. We refer this dataset to as *barracks*. In Figure 3e a 3D rendering of the surface model is shown. Functional parameters for the minimization on these datasets are set as follows:

- *pearl*: $\epsilon = 0.01, \delta = 3, \alpha = 2, \beta = 1, \mu = 0.07, t = 1$;
- *aerial*: $\epsilon = 0.01, \delta = 1, \alpha = 2, \beta = 1, \mu = 0.05, t = 1$;
- *airport*: $\epsilon = 0.01, \delta = 1, \alpha = 2, \beta = 1, \mu = 0.05, t = 1$;
- *barracks*: $\epsilon = 0.01, \delta = 30, \alpha = 2, \beta = 1, \mu = 1, t = 1$.

In Table 1 we report the total number of outer (k) and inner (totiter) iterations of GS, BCDA and BCDA_c together with the total time in seconds (the mean of ten runs) required to compute an approximate solution, with $TOL_F = 10^{-3}$ in (22). Computational performance of GS, BCDA, and BCDA_c is better illustrated in Figure 4, where the value of the objective function at each outer iteration is plotted against the cumulative execution time.

The solutions obtained with all the proposed methods are very similar each other, in such a way that they cannot be distinguished only visually. The accuracy (and the similarity) of the solutions obtained with GS, BCDA and BCDA_c

³Images are downloadable at <http://sipi.usc.edu/database/database.php?volume=misc#top>

⁴DSMs are downloadable at <http://www.territorio.provincia.tn.it/portal/server.pt/community/lidar/847/lidar/23954>

dataset	method	k	totiter (s-z-u)	time
pearl 600×600	GS	18	43-49-6435	190.3
	BCDA	14	14-14-690	38.9
	BCDA _c	14	14-14-106	25.0
aerial 512×512	GS	19	49-55-4592	105.9
	BCDA	15	15-15-492	25.8
	BCDA _c	11	11-11-62	14.4
airport 1024×1024	GS	20	51-55-4937	460.1
	BCDA	16	16-16-493	107.9
	BCDA _c	12	12-12-72	62.1
barracks 600×600	GS	14	31-39-3906	119.1
	BCDA	12	12-12-463	29.3
	BCDA _c	10	10-10-69	17.3

Table 1: Outer/inner iterations and execution time (seconds) observed in the run of the proposed algorithms.

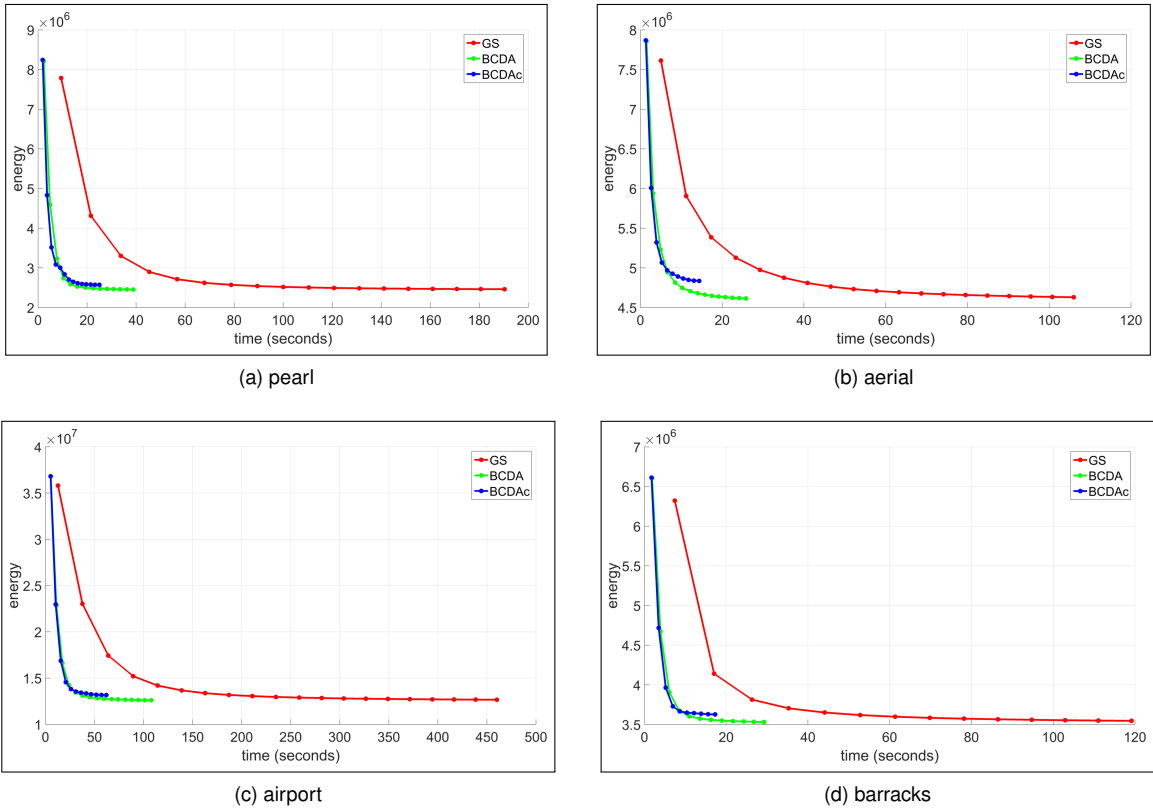


Figure 4: Energy-versus-time at every outer iteration for all three datasets with GS, BCDA and BCDAc. Algorithms are stopped by criterion (22) with tolerance $TOL_F = 10^{-3}$.

can be measured in terms of their distance to the *ideal* solutions. To this aim we defined a *normalized* distance based on the L^1 norm as follows. In view of (12), the solutions of the algorithms are such that $\mathbf{s}, \mathbf{z} \in [0, 1]^n$. Given $H := [0, h]^n$, with $h > 0$, we easily have that

$$d(h) := \max_{\mathbf{x}, \mathbf{y} \in H} \|\mathbf{x} - \mathbf{y}\|_1 = hn. \quad (23)$$

Therefore, we can define a normalized distance function $d_h : H \times H \rightarrow [0, 1]$ by setting $d_h(\mathbf{x}, \mathbf{y}) := \frac{1}{d(h)} \|\mathbf{x} - \mathbf{y}\|_1$. As a consequence, the maximum possible distance in $H := [0, 1]^n$ is 1, and the value $100 \cdot d_h(\mathbf{x}, \mathbf{y})$ can be interpreted

dataset	method	$d_1(\mathbf{s}, \mathbf{s}^*)$	$d_1(\mathbf{z}, \mathbf{z}^*)$	$d_G(\mathbf{u}, \mathbf{u}^*)$
pearl	GS	$3.40e-03$	$6.45e-03$	$4.02e-04$
	BCDA	$4.00e-03$	$7.59e-03$	$4.93e-04$
	BCDA _c	$1.38e-02$	$1.84e-02$	$4.78e-03$
aerial	GS	$9.76e-03$	$1.27e-02$	$1.13e-03$
	BCDA	$9.44e-03$	$1.22e-02$	$1.14e-03$
	BCDA _c	$3.33e-02$	$3.18e-02$	$8.35e-03$
airport	GS	$5.41e-03$	$7.46e-03$	$6.59e-04$
	BCDA	$5.74e-03$	$7.80e-03$	$7.43e-04$
	BCDA _c	$1.86e-02$	$1.83e-02$	$6.88e-03$
barracks	GS	$2.47e-03$	$1.65e-02$	$3.57e-05$
	BCDA	$2.47e-03$	$1.62e-02$	$3.63e-05$
	BCDA _c	$6.48e-03$	$2.62e-02$	$1.80e-04$

Table 2: Accuracy in the approximations given by the GS, BCDA and BCDA_c with respect to the *ideal* solution for the considered datasets.

as the percentage of image content which is changed between \mathbf{x} and \mathbf{y} . Thus, the distances for the functions \mathbf{s} and \mathbf{z} are given in terms of d_1 . As already mentioned in Section 2.2, it is not possible to define an explicit bound for the values of the function \mathbf{u} . Although our method produces a sequence of iterates $(\mathbf{s}^k, \mathbf{z}^k, \mathbf{u}^k) \in \mathcal{L}_{\alpha^0}$, the inequality $F_\epsilon(\mathbf{s}^k, \mathbf{z}^k, \mathbf{u}^k) \leq F_\epsilon(\mathbf{e}, \mathbf{e}, \mathbf{g})$ does not necessarily imply that $\|\mathbf{u}^k\|_\infty \leq \|\mathbf{g}\|_\infty$. However, in all numerical experiments presented in this paper (and in many other experiments performed by the authors) it is observed that the optimal \mathbf{u} satisfies $\|\mathbf{u}^k\|_\infty \leq \|\mathbf{g}\|_\infty$, thus it is meaningful to compute the distances of the solutions \mathbf{u} by means of d_G , where $G := \|\mathbf{g}\|_\infty$.

The distances between the solutions of GS, BCDA, BCDA_c with respect to the *ideal* solutions are given in Table 2. By analyzing the results we see that the BCDA algorithm, if compared to GS, significantly decreases the time of computation of approximately 75% and returns solutions that do not differ more than 1% with respect to the *ideal* solution. BCDA and GS always resulted in the same accuracy. In the case of BCDA_c, the time of computation further decreases of 10% and the difference of the solutions with respect to the *ideal* solution are never greater than 1%. Notice that the obtained solutions have a lower accuracy as the normalized distances are about 10 times those obtained by the other two methods. In Figure 5, for each dataset we plot the smooth approximation \mathbf{u} , the edge-map \mathbf{s} and the edge/crease-map \mathbf{z} obtained by the BCDA.

Second-order segmentation provides a piecewise linear approximation of the input image. Therefore, if compared to first order models, it avoids the problem of over-segmentation and the real geometry of objects is followed properly. For instance, in the *pearl* image both the noise and the *craquelure* are removed while the geometry of shadows is preserved. By taking a look at the particular showed in Figure 6, we see that around the nostril the variation of grey level in the shadowed area is over-segmented by the \mathbf{s} function (shadow-like trait in the \mathbf{s} map) but it is correctly outlined by the \mathbf{z} function. In fact, boundaries of shadows are characterized by a transient zone of luminance variation which is a ramp and not as sharp as a jump. The *aerial* image is smoothed out and the contrast between the ground and human-made objects is more evident in the segmented image. A similar behavior is observed for *airport*, where the smoothing removes the noise but is able to keep the geometry of cars in the parking area. By looking at the particular in Figure 7 we see that almost every trait in \mathbf{s} seems to be *doubled* in \mathbf{z} . Again, this happens because in luminance images the transition of intensity between two areas with different values is usually not purely a jump. The doubled trait is due to the fact that the \mathbf{z} function is able to detect both sides of the transition ramp.

In the *barracks* dataset, the capability of the model of detecting second-order edge boundaries is clear. In the surface models the geometry is real and the structure of many man-made objects is linear (buildings for instance). By looking at the particulars in Figure 8 we see that the noise is removed and the edges that define the roof planes are preserved and correctly detected by the \mathbf{z} function. Notice also the substantial difference between the functions \mathbf{s} and \mathbf{z} . This difference is not so evident in the images as variations of gradient of luminance are usually as not as sharp as variations of gradient of height in surface models of urban areas. This means that discontinuities that are purely of second-order are difficult to find in images, but not in DSMs. Lastly, we remark here that in a discrete setting a jump is also a crease. In view of the second-order differential operators used in (8) a jump is traced along 1-pixel wide curve in \mathbf{s} , whereas it is traced along a 2-pixel wide curve by \mathbf{z} (cfr. Section 3.3).

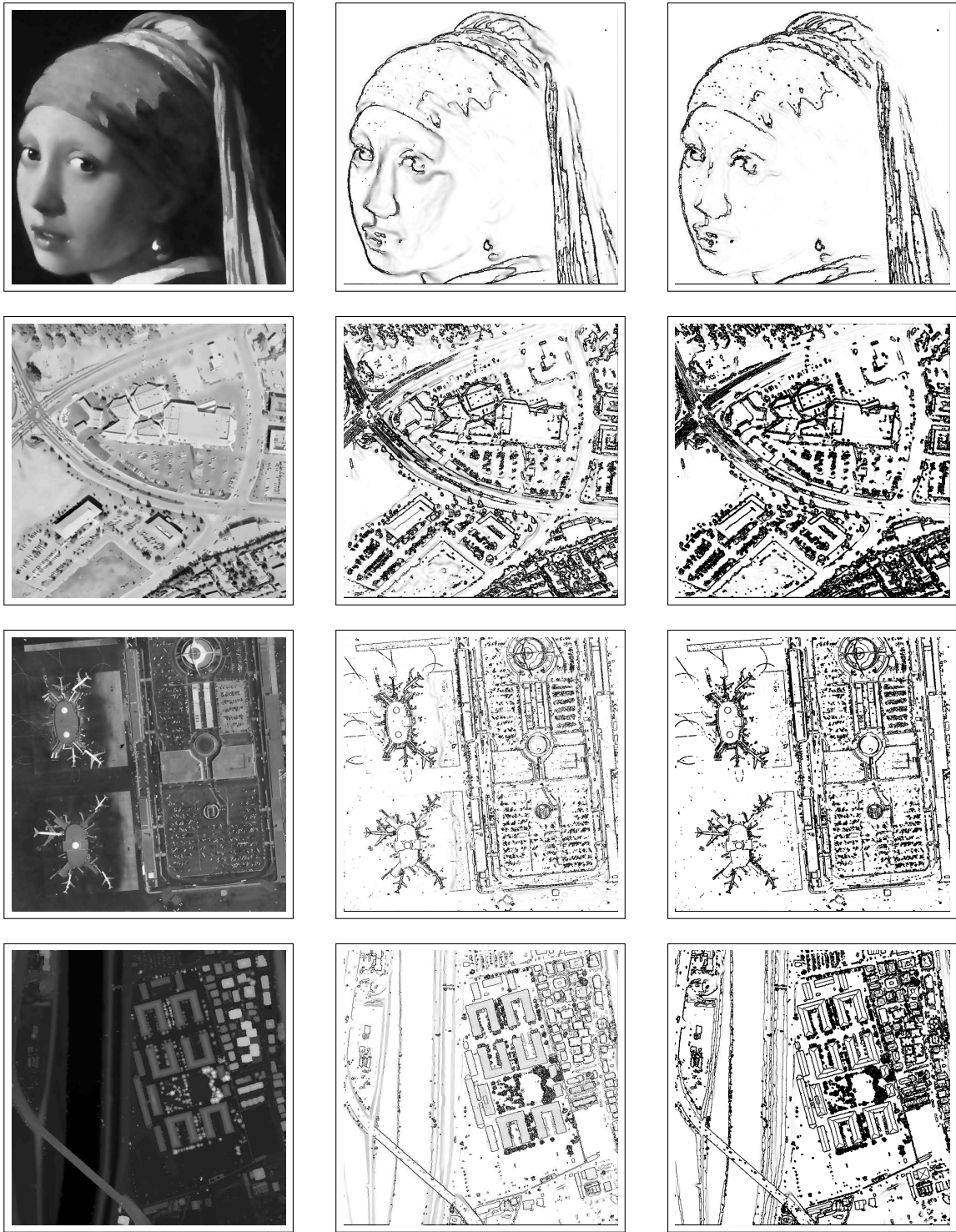


Figure 5: Results of the BCDA method for the considered datasets. First column is the smooth approximation u , second column is the edge-map s and third column is the edge/crease-map z .



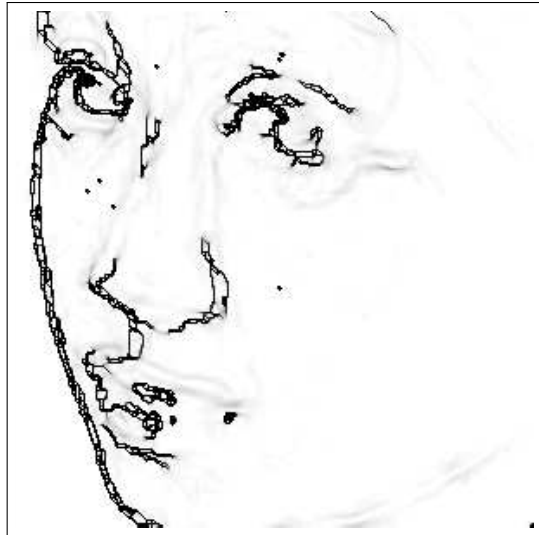
g



u



s



z

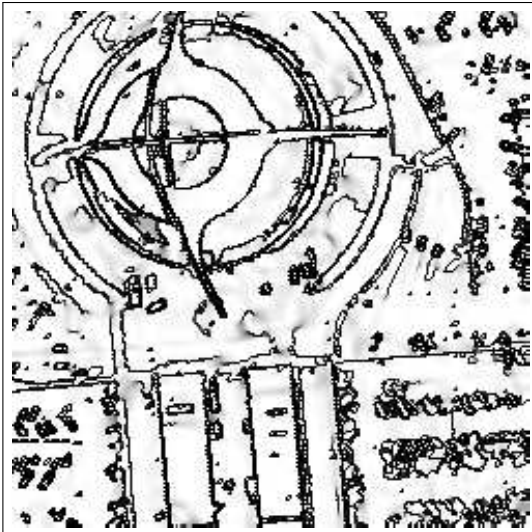
Figure 6: Particulars of the segmentation for the dataset pearl.



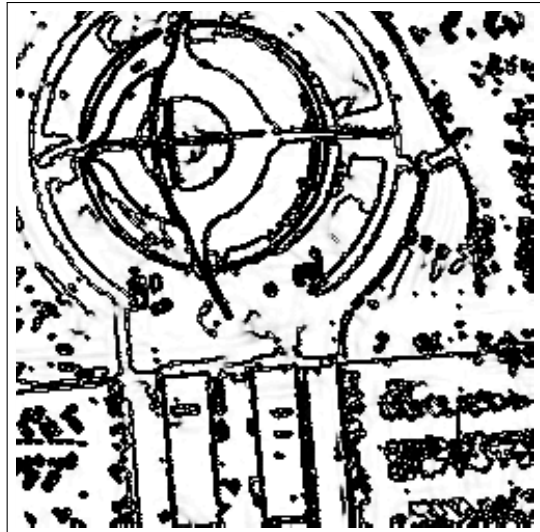
g



u

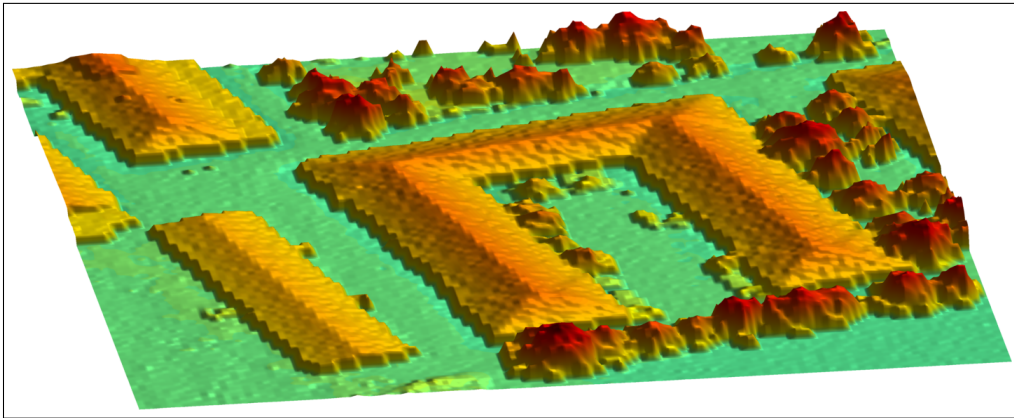


s

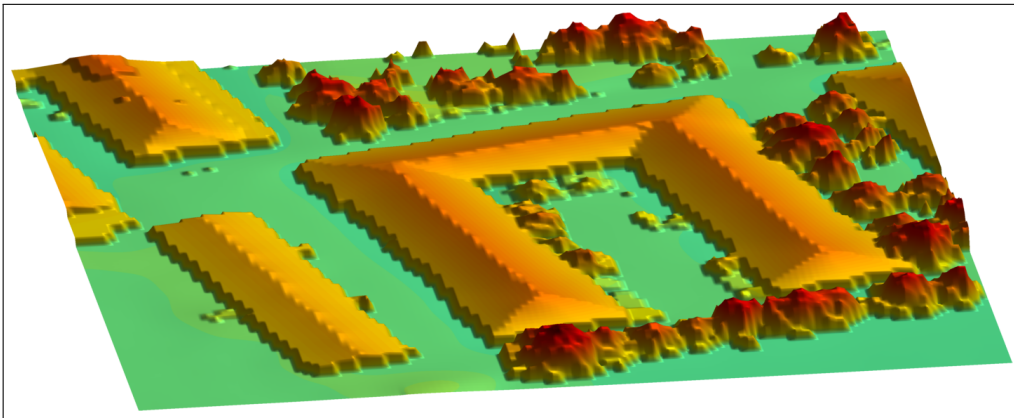


z

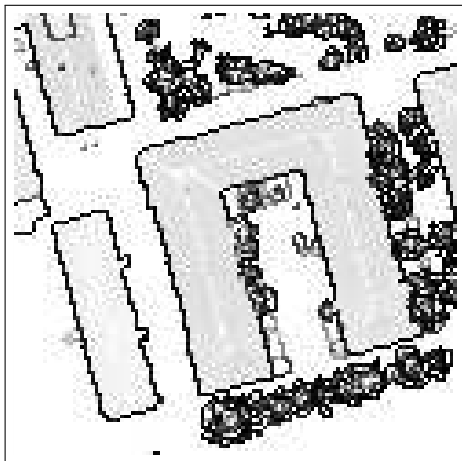
Figure 7: Particulars of the segmentation for the dataset airport.



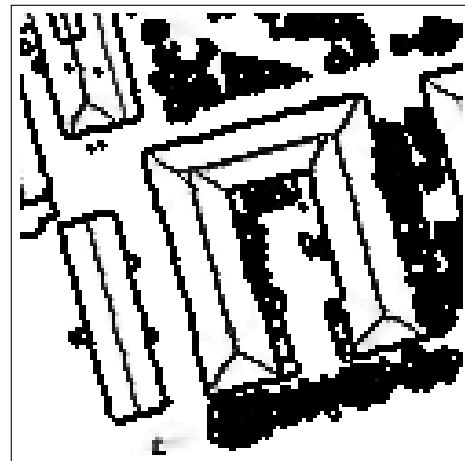
g



u



s



z

Figure 8: Particulars of the segmentation for the dataset barracks. Above, there are the 3D renderings of the surface model g and its smooth approximation u . In the bottom, there are the edge map s and the edge+crease map z .

3.2. Noise reduction and effect of Block-Preconditioner for the BCDA

The Step 3 in Algorithm 1 attempts to minimize the objective function $F_\epsilon(\mathbf{s}, \mathbf{z}, \mathbf{u})$ with respect to the variable block \mathbf{u} . The performance of this step depends on both the size and the noise level of the image. The inverse of the parameter μ of the objective function represents the scale at which variations in \mathbf{g} are considered as noise. At each outer iteration k , variations that are in order to be smoothed out are signaled by the points where $\mathbf{s}^k, \mathbf{z}^k$ are close to 1. Smoothing takes place when the system involving \mathbf{A}_μ is solved and returns \mathbf{u}^k . The computational burden related to this step is determined by the scale of the noise in the image: the greater the variations are related to the noise, the more effort is required to smooth out such variations. Assuming that interesting features (edge-boundaries) and noise are at different scales in the image, the parameter μ must be chosen at an intermediate level in such a way that the former ones are preserved by the model whereas the latter ones are smoothed out. However, from (11) one can see that small values of μ reduce the definite positiveness of matrix \mathbf{A}_μ . Thus, μ should be optimal, i.e., as greater as possible but not such that noise is preserved.

The first test we present concerns with the relationship between the choice of μ , the noise reduction and the performance of the minimization performed by the proposed BCDA. The basic element of the datasets considered in the following is a synthetic 100×100 grey-scale image presenting geometrical features of first and second order consisting in a truncated pyramid. Images of greater dimensions are obtained by assembling several basic elements of the same type. Test images are corrupted by artificial additive Gaussian noise with 0 mean and varying standard deviation σ (top-left image in Figure 9 represents one pyramidal element, in the same row other elements with added noise). Along the experiments, functional parameters except μ are fixed to $\alpha = 2, \beta = 1, \delta = 30, \epsilon = 0.01, t = 1$. After running the algorithm several times, we observed that for values $\mu < 0.05$ the smoothing also affected the interesting features of the image (pyramid edges), whereas for $\mu > 0.15$ no smoothing at all was observed. Therefore, the results given in the following relate to $0.05 \leq \mu \leq 0.15$. Performance of the BCDA with respect to noise and μ is evaluated in two experiments.

- In the first experiment the minimization is performed on one pyramidal element for different levels of noise $\sigma = 0, 0.5, 1, 2$ ($\sigma = 0$ means that no noise is added) and for $\mu = 0.15$. In Figure 9, the input image \mathbf{g} , the difference between \mathbf{g} and its smooth approximation \mathbf{u} , the edge-detection function \mathbf{s} and the edge/crease-detection function \mathbf{z} are showed. Notice from the plots of $\mathbf{g} - \mathbf{u}$ how the noise is uniformly detected and removed by the model in all cases. The detection of first and second order features of the image is sufficiently accurate, and only for $\sigma = 2$ the scale of the noise slightly affects the detection of pyramid edges. Let us denote \mathbf{g}_σ and \mathbf{u}_σ the input image and its smooth approximation for different values of σ . Quantitatively, the capability of the model of removing the noise is given in terms of the distance between the smooth approximations \mathbf{u}_σ and the original noise-free image \mathbf{g}_0 . We obtained $d_G(\mathbf{g}_0, \mathbf{u}_0) = 2.90e - 03$, $d_G(\mathbf{g}_0, \mathbf{u}_{0.5}) = 2.81e - 03$, $d_G(\mathbf{g}_0, \mathbf{u}_1) = 2.88e - 03$, $d_G(\mathbf{g}_0, \mathbf{u}_2) = 3.13e - 03$, where we used $G = \|\mathbf{g}_0\|_\infty$.
- In the second experiment, the effect of varying μ on the performance of the minimization method is tested with respect to different levels of noise. Test images have size of 1000×1000 pixels. Times of computation and average number of iterations of the PCG related to \mathbf{u} are given in Figure 10. As we can see, for a fixed value of μ the time required for smoothing the data increases with σ . Moreover, since the average number of PCG iterations for \mathbf{u} does not significantly change with σ , we conclude that more outer iterations are needed to smooth out the noise. Notice also that, as expected, increasing values of μ resulted in less computational time due to the fact that the definite positiveness of matrix \mathbf{A}_μ increases.

As we have seen in the previous experiments, the noise removal task can be very expensive. Moreover, the execution time also depends on the size of the input image. The second test we present aims at showing that the general performance of BCDA can be enhanced if a block-diagonal preconditioner, instead of a point-diagonal one, is used. In the experiments, the algorithm is run on synthetic images composed by several pyramidal elements corrupted with noise with different variances. Analyzed images have a number of pixels ranging from $2 \cdot 10^5$ to $4 \cdot 10^6$. From Figure 11 we see that the execution time linearly increases with the size of the image. Moreover, the use of a block-preconditioner reduces the time of approximately 14% regardless the noise variance. It is also confirmed from the graphs of total and average number of PGC iterations that, for increasing sizes of the image the number of iterations does not significantly change. The computational burden is instead in the time for completing each iteration.

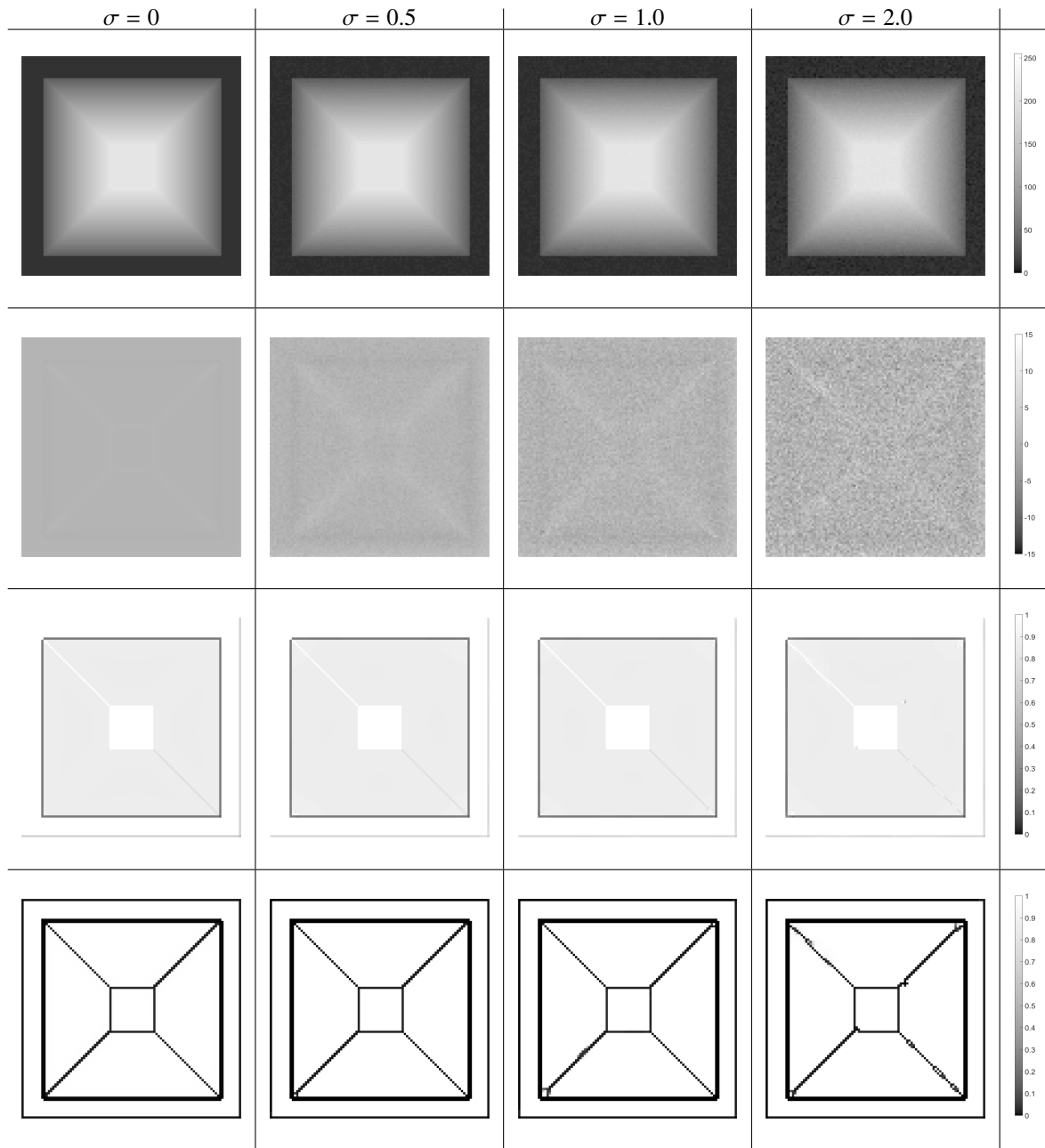


Figure 9: Smoothing on synthetic images with different noise levels, image size is 100×100 . Row 1: input noisy images \mathbf{g} . Row 2: difference $\mathbf{g} - \mathbf{u}$. Row 3: edge-detection functions \mathbf{s} . Row 4: edge/crease-detection functions \mathbf{z} .

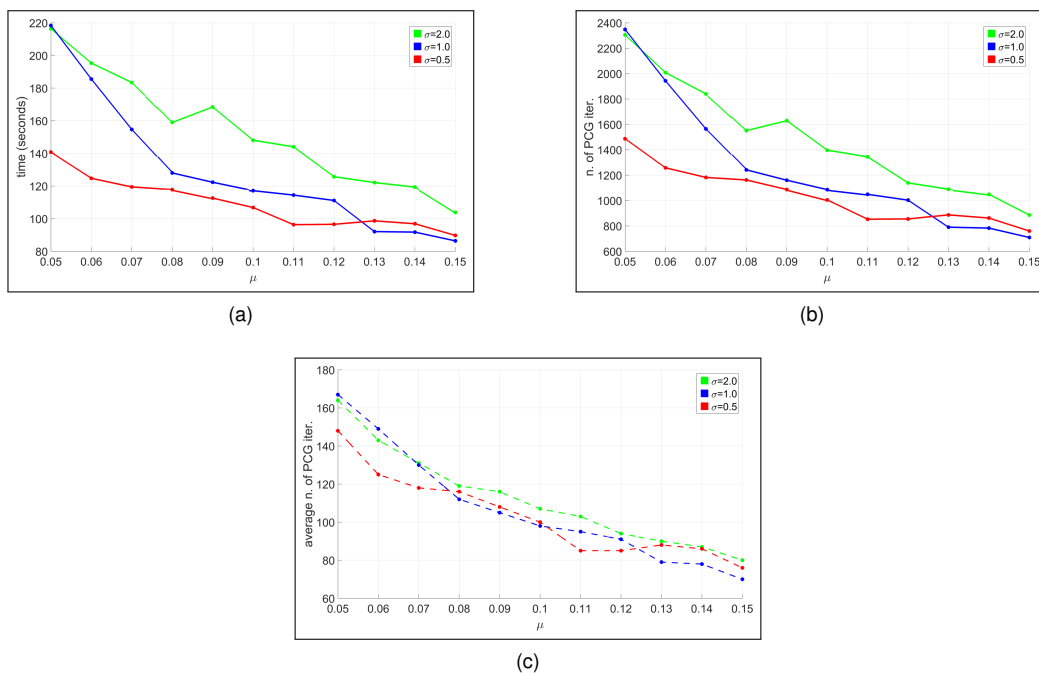


Figure 10: Performance details of BCDA by varying μ and for different variances of the noise. Test images are 1000×1000 pixels compositions of the pyramidal elements showed in Figure 9. For each value of μ we plot (a) the execution time, (b) the total number, and (c) the average number of iterations of the PCG solver related to \mathbf{u} .

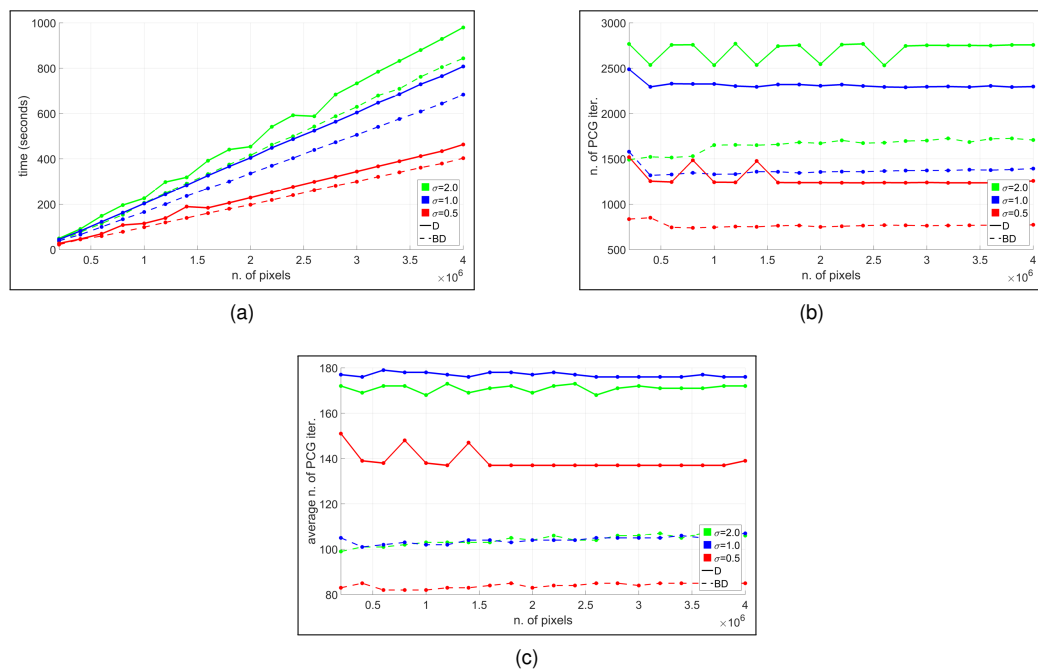


Figure 11: Performance of BCDA with Diagonal (D) and Block-Diagonal (BD) preconditioners for the PCG solver related to \mathbf{u} , versus the size of \mathbf{g} . We plot (a) the execution time, (b) the total number, and (c) the average number of iterations of the PCG solver related to \mathbf{u} .

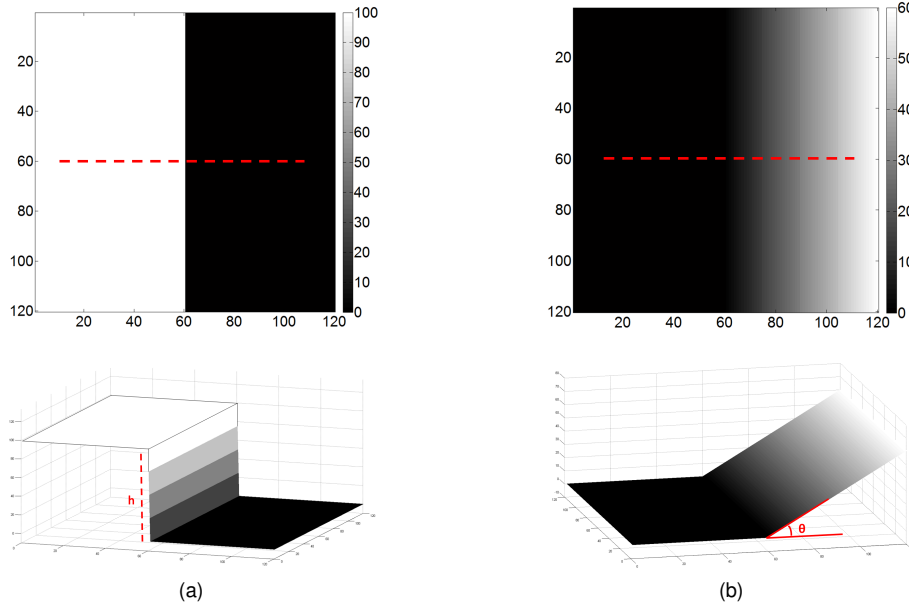


Figure 12: Test images. (a) Image with jump of variable height h . (b) Image with a crease of variable slope θ . The slices of functions \mathbf{s}, \mathbf{z} plotted in Figures 13 and 14 are located in correspondence of the red dashed lines.

3.3. Parameter ϵ , grid resolution and sensitivity of the discontinuity functions

The explicit construction of a recovery sequence in the Γ -convergence proof of [36] allows for having an essential prediction of geometrical properties of the discontinuity functions s_ϵ, z_ϵ approximating the discontinuity sets $S_u, S_{\nabla u}$, for $\epsilon > 0$ (see Figure 2 and related discussion). The geometrical behavior of s_ϵ, z_ϵ is theoretically determined by the convergence parameter ϵ and geometrical features of the input image g . However, no analytical expression of the functions in proximity of the discontinuities is given. These facts give rise to some numerical problems that must be taken into account when the functional \mathcal{F}_ϵ is minimized and the numerical approximations of s_ϵ, z_ϵ (given by \mathbf{s}, \mathbf{z} , respectively) are computed. On the one hand, the discrete sampling of the domain Ω with steps t_x, t_y must resolve the tubular neighborhoods of the discontinuity sets, which shrink as $\epsilon \rightarrow 0$. On the other hand, increasing values of $|\nabla g|$ and $|\nabla^2 g|$ will force sharper transitions of the discontinuity functions from 1 to 0, whereas decreasing values will result in softer transitions, therefore limiting the capability of detecting edge boundaries.

In order to have a clear understanding of the geometrical behavior of functions \mathbf{s}, \mathbf{z} , in the following the sensitivity of the model with respect to the parameter ϵ and the variations of $|\nabla g|$ and $|\nabla^2 g|$ is tested. In the tests the two synthetic images showed in Figure 12 are used. The first image contains a uniform jump (discontinuity) of grey value with variable height h . The second image contains a crease (gradient discontinuity) between a flat area and a uniform slope with variable angle θ .

Let us discuss the results obtained on the test image with a jump (Figure 12a), firstly by varying parameter ϵ , then by varying h . In the first experiment all the functional and discretization parameters, except ϵ , are fixed to $\delta = 30$, $\alpha = 2$, $\beta = 1$, $\mu = 1$, $t_x = t_y = 1$, the step height is $h = 90$. A sufficiently wide range of behaviors of the discontinuity functions can be depicted by minimizing the functional F_ϵ for values of ϵ ranging from 10^{-5} to 5. In Figures 13a and 13b the plots of slices of the minimizing functions \mathbf{s} and \mathbf{z} in correspondence of the jump, are given. As we can see, the width of the tubular neighborhoods of the discontinuity sets increases with ϵ . For $\epsilon = 10^{-5}$ the grid is too coarse for detecting the transition phase of both \mathbf{s} and \mathbf{z} , which are uniformly 1. The optimal choice of the Γ -convergence parameter is $\epsilon = 10^{-2}$, as it corresponds to a detection of the jump which is 1 grid-point wide in the case of \mathbf{s} , and two grid-point wide in case of \mathbf{z} (detection is optimal in view of the differential discrete operators described in Section 2.1). For greater values of ϵ the detection of the jump is increasingly over-estimated. In particular, notice the difference in the rate at which \mathbf{s} and \mathbf{z} become flat in the neighbors of the jump.

In the second experiment, the same parameters as before are used and ϵ is fixed at the optimal value $\epsilon = 0.01$. Slices

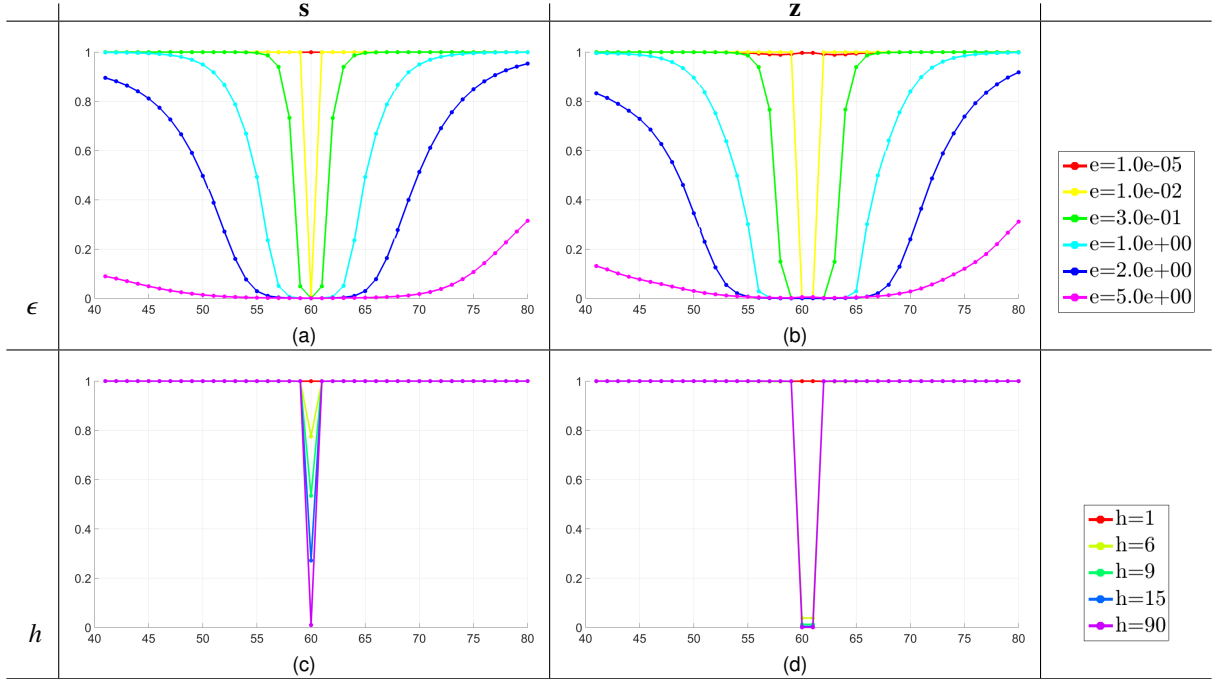


Figure 13: Slices of functions \mathbf{s} and \mathbf{z} obtained by minimizing F_ϵ in the case of test image with a jump (Figure 12a) for different values of ϵ (a,b) and h (c,d).

of discontinuity functions are plotted in Figures 13c and 13d for values of h ranging from 1 to 90. In the case of $h = 90$ the gradient ∇g is very high, therefore functions \mathbf{s} and \mathbf{z} are forced to inhibit the costly gradient contribution to the energy by taking the 0 value in correspondence of the jump. By decreasing h , the discontinuity functions become less sensitive to the jump and their values approach 1 gradually. In the limit case, $h = 1$, the step is not detected at all. Since here we are studying the behavior of minimizers as ∇g changes, we have to remark that a similar behavior is observed if h is fixed and the discretization parameters t_x and t_y are changed; this corresponds to a re-scaling of the image that only changes derivatives.

Now, we present the results obtained on the test image with a gradient discontinuity (Figure 12b), again by varying parameter ϵ , and then by varying the inclination angle θ . Functional parameters are the same as in the previous tests, expect that $\delta = 300$ (crease detection is enforced).

In the first experiment the range of values of ϵ is the same as in the previous one, and $\tan(\theta) = 1$ is fixed. In Figures 14a and 14b the plots of slices of the minimizing functions \mathbf{s} and \mathbf{z} in correspondence of the crease, are given. For $\epsilon > 10^{-2}$ the ramp is over-segmented by \mathbf{s} and \mathbf{z} . In particular, the larger is ϵ , the softer is the transition of \mathbf{z} from 1 to 0. The value $\epsilon = 10^{-2}$ is again optimal as \mathbf{s} is uniformly 1 (no over-segmentation) and the crease is correctly detected with sharp variation of \mathbf{z} from 1 to 0. Again, for smaller values of ϵ the tubular neighborhood of the discontinuity of the gradient is too thin for being resolved by the grid.

In the second experiment we tested the sensitivity of the discontinuity functions with respect to the slope of the ramp. Parameters are the same as in the previous experiment. Results are plotted in Figures 14c and 14d for values of $\tan(\theta)$ ranging from 0.2 to 5. The steepest is the ramp, the better is the detection of the gradient discontinuity given by \mathbf{z} . Notice that for $\tan(\theta) = 5$, the function \mathbf{s} tends to over-segment the ramp. This fact can be explained by noticing that for such θ , the difference of grey value from one pixel to an adjacent one in the ramp is in the order of a detectable jump. Since $t_x = 1$, the jump in this case is 5. Compare the value of function \mathbf{s} with the value of \mathbf{s} in Figure 13c corresponding to a jump of height 6: they are both close to 0.8. As a last remark we point out that, as in the previous example, a change in the step widths t_x, t_y has the only obvious effect of re-scaling the derivatives in the image, thus the behavior of \mathbf{s} and \mathbf{z} is similar to the one presented here with variable θ .

These tests described the relationship between the geometry of the discontinuity functions returned by the mini-

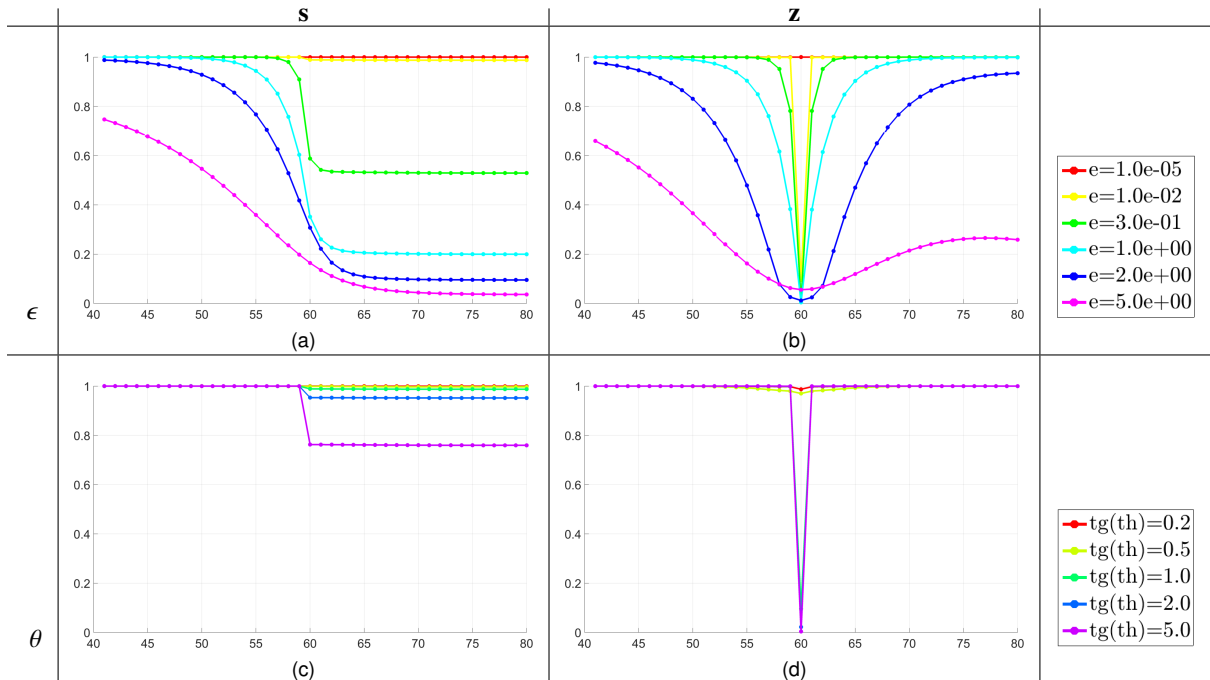


Figure 14: Slices of functions s and z obtained by minimizing F_ϵ in the case of test image with a gradient discontinuity (Figure 12b) for different values of ϵ (a,b) and $\tan(\theta)$ (c,d).

mization of F_ϵ , the geometry of the image g and the Γ -convergence parameter ϵ . By increasing the parameter ϵ the width of the tubular neighborhood of the discontinuity sets increases, until reaching a full over-segmentation of the image. Conversely, for small values of ϵ , the tubular neighborhood is narrower and narrower, until it is not resolved by the grid. A correct value of the Γ -convergence parameter should be chosen accordingly. The parameters that affect gradient and Hessian are the height of the jumps, crease angles, and also the grid steps t_x, t_y . By varying such quantities the width of the tubular neighborhoods does not change. However, functions s and z are sensitive to their variations and they approach 0 and 1 at different rates. We conclude that the width of the tubular neighborhood of the discontinuity sets depends only on ϵ , whereas other geometrical properties of the image affect the rate at which these functions approach 0 and 1.

4. Conclusions and future work

In this work we proposed an efficient block-coordinate descent method for the numerical minimization of a variational approximation of the Blake-Zisserman functional given by Ambrosio, Faina and March. The Blake-Zisserman variational model for segmentation is a second-order model based on free discontinuities and free gradient discontinuities that is able to both detect first and second order edge-boundaries in images and produce a piecewise linear approximation of the input image. Therefore, the model presents several advantages if compared with other methods for segmentation. The proposed algorithm exploits the structure of the minimization problem, allowing the segmentation of large images, with a satisfying performance in terms of accuracy and computational time. In particular, it outperforms a standard Gauss-Seidel of 75%. Moreover, the use of a block-diagonal preconditioner always increases performance of about 14%. Numerical experiments presented in the paper involve very different types of datasets. We consider also the segmentation of Digital Surface Models (DSMs), showing that the application of the Blake-Zisserman functional allows to reconstruct, and locate the boundary, of planar objects. In the experimental part we also focused the attention on the geometrical parameter of the functional related to noise reduction and the behaviour of the discontinuity functions with respect to the Γ -convergence parameter ϵ and geometrical properties of the input image.

Preliminary tests showed that the proposed method can be combined with a domain decomposition technique, enabling the segmentation of huge images by a tiling strategy. Future work will concern the deepening of this attractive feature in the framework of the multiprocessing computing with the aim to design a code for modern parallel architectures.

Acknowledgements

This research is partially supported by the project *Learning meets time: A new computational approach for learning in dynamic systems*, contract RBFR12M3A. The project is funded by the Italian Ministry of Education, University and Research *Futuro in Ricerca 2012* program. The authors are particularly thankful to the anonymous reviewers for helping us in improving the work.

References

- [1] A. P. Witkin, Scale-space filtering: A new approach to multi-scale description, in: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84., Vol. 9, IEEE, 1984, pp. 150–153.
- [2] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 12 (7) (1990) 629–639.
- [3] L. I. Rudin, S. Osher, Total variation based image restoration with free local constraints, in: *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, Vol. 1, IEEE, 1994, pp. 31–35.
- [4] J. Weickert, *Anisotropic diffusion in image processing*, European Consortium for Mathematics in Industry, B. G. Teubner, Stuttgart, 1998.
- [5] F. Catté, P.-L. Lions, J.-M. Morel, T. Coll, Image selective smoothing and edge detection by nonlinear diffusion, *SIAM J. Numer. Anal.* 29 (1) (1992) 182–193. doi:10.1137/0729012.
URL <http://dx.doi.org/10.1137/0729012>
- [6] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, *Comm. Pure Appl. Math.* 42 (5) (1989) 577–685. doi:10.1002/cpa.3160420503.
URL <http://dx.doi.org/10.1002/cpa.3160420503>
- [7] E. De Giorgi, Free discontinuity problems in calculus of variations, in: *Frontiers in pure and applied mathematics*, North-Holland, Amsterdam, 1991, pp. 55–62.
- [8] B. Kawohl, From Mumford-Shah to Perona-Malik in image processing, *Math. Methods Appl. Sci.* 27 (15) (2004) 1803–1814. doi:10.1002/mma.564.
URL <http://dx.doi.org/10.1002/mma.564>
- [9] E. De Giorgi, M. Carriero, A. Leaci, Existence theorem for a minimum problem with free discontinuity set, *Arch. Rational Mech. Anal.* 108 (3) (1989) 195–218. doi:10.1007/BF01052971.
URL <http://dx.doi.org/10.1007/BF01052971>
- [10] L. Ambrosio, N. Fusco, D. Pallara, *Functions of bounded variation and free discontinuity problems*, Oxford Mathematical Monographs, The Clarendon Press, Oxford University Press, New York, 2000.
- [11] L. Ambrosio, V. M. Tortorelli, Approximation of functionals depending on jumps by elliptic functionals via Γ -convergence, *Comm. Pure Appl. Math.* 43 (8) (1990) 999–1036. doi:10.1002/cpa.3160430805.
URL <http://dx.doi.org/10.1002/cpa.3160430805>
- [12] L. Modica, S. Mortola, Un esempio di Γ -convergenza, *Bollettino Unione Matematica Italiana* 14 (B) (1977) 285–299.
- [13] G. Bellettini, A. Coscia, Discrete approximation of a free discontinuity problem, *Numer. Funct. Anal. Optim.* 15 (3-4) (1994) 201–224. doi:10.1080/01630569408816562.
URL <http://dx.doi.org/10.1080/01630569408816562>
- [14] P. D'Ambra, G. Tartaglione, Solution of Ambrosio–Tortorelli model for image segmentation by generalized relaxation method, *Communications in Nonlinear Science and Numerical Simulation* 20 (3) (2015) 819–831.
- [15] A. Chambolle, G. Dal Maso, Discrete approximation of the Mumford-Shah functional in dimension two, *M2AN Math. Model. Numer. Anal.* 33 (4) (1999) 651–672. doi:10.1051/m2an:1999156.
URL <http://dx.doi.org/10.1051/m2an:1999156>
- [16] B. Bourdin, A. Chambolle, Implementation of an adaptive finite-element approximation of the Mumford-Shah functional, *Numer. Math.* 85 (4) (2000) 609–646. doi:10.1007/PL00005394.
URL <http://dx.doi.org/10.1007/PL00005394>
- [17] T. Pock, D. Cremers, H. Bischof, A. Chambolle, An algorithm for minimizing the Mumford-Shah functional, in: *Computer Vision, 2009 IEEE 12th International Conference on*, IEEE, 2009, pp. 1133–1140.
- [18] T. F. Chan, L. A. Vese, A level set algorithm for minimizing the Mumford-Shah functional in image processing, in: *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, IEEE, 2001, pp. 161–168.
- [19] A. Braides, *Approximation of free-discontinuity problems*, Vol. 1694 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1998.
- [20] A. Blake, A. Zisserman, *Visual reconstruction*, MIT Press Series in Artificial Intelligence, MIT Press, Cambridge, MA, 1987.
- [21] M. Carriero, A. Farina, I. Sgura, Image segmentation in the framework of free discontinuity problems, in: *Calculus of variations: topics from the mathematical heritage of E. De Giorgi*, Vol. 14 of Quad. Mat., Dept. Math., Seconda Univ. Napoli, Caserta, 2004, pp. 86–133.
- [22] M. Carriero, A. Leaci, F. Tomarelli, A second order model in image segmentation: Blake & Zisserman functional, in: *Variational methods for discontinuous structures (Como, 1994)*, Vol. 25 of Progr. Nonlinear Differential Equations Appl., Birkhäuser, Basel, 1996, pp. 57–72.

- [23] M. Carriero, A. Leaci, F. Tomarelli, Strong minimizers of Blake & Zisserman functional, *Ann. Scuola Norm. Sup. Pisa Cl. Sci.* (4) 25 (1-2) (1997) 257–285 (1998), dedicated to Ennio De Giorgi.
URL http://www.numdam.org/item?id=ASNSP_1997_4_25_1-2_257_0
- [24] T. Boccellini, F. Tomarelli, Generic uniqueness of minimizer for Blake & Zisserman functional, *Rev. Mat. Complut.* 26 (2) (2013) 361–408.
doi:10.1007/s13163-012-0103-1.
URL <http://dx.doi.org/10.1007/s13163-012-0103-1>
- [25] M. Carriero, A. Leaci, F. Tomarelli, Uniform density estimates for Blake & Zisserman functional, *Discrete Contin. Dyn. Syst.* 31 (4) (2011) 1129–1150. doi:10.3934/dcds.2011.31.1129.
URL <http://dx.doi.org/10.3934/dcds.2011.31.1129>
- [26] M. Carriero, A. Leaci, F. Tomarelli, Euler equations for Blake and Zisserman functional, *Calc. Var. Partial Differential Equations* 32 (1) (2008) 81–110. doi:10.1007/s00526-007-0129-2.
URL <http://dx.doi.org/10.1007/s00526-007-0129-2>
- [27] M. Carriero, A. Leaci, F. Tomarelli, A candidate local minimizer of Blake and Zisserman functional, *J. Math. Pures Appl.* (9) 96 (1) (2011) 58–87. doi:10.1016/j.matpur.2011.01.005.
URL <http://dx.doi.org/10.1016/j.matpur.2011.01.005>
- [28] M. Carriero, A. Leaci, F. Tomarelli, A survey on the Blake–Zisserman functional, *Milan Journal of Mathematics* 83 (2) (2015) 397–420.
- [29] M. Carriero, A. Leaci, F. Tomarelli, Image inpainting via variational approximation of a Dirichlet problem with free discontinuity, *Adv. Calc. Var.* 7 (3) (2014) 267–295. doi:10.1515/acv-2012-0020.
URL <http://dx.doi.org/10.1515/acv-2012-0020>
- [30] M. Zanetti, A. Vitti, The Blake-Zisserman model for digital surface models segmentation, in: *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. II-5/W2, 2013, pp. 355–360. doi:10.5194/isprsannals-II-5-W2-355-2013.
- [31] E. De Giorgi, T. Franzoni, Su un tipo di convergenza variazionale, *Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur.* (8) 58 (6) (1975) 842–850.
- [32] G. Dal Maso, An introduction to Γ -convergence, *Progress in Nonlinear Differential Equations and their Applications*, 8, Birkhäuser Boston, Inc., Boston, MA, 1993. doi:10.1007/978-1-4612-0327-8.
URL <http://dx.doi.org/10.1007/978-1-4612-0327-8>
- [33] A. Braides, Γ -convergence for beginners, Vol. 22 of *Oxford Lecture Series in Mathematics and its Applications*, Oxford University Press, Oxford, 2002. doi:10.1093/acprof:oso/9780198507840.001.0001.
URL <http://dx.doi.org/10.1093/acprof:oso/9780198507840.001.0001>
- [34] G. Del Piero, G. Lancioni, R. March, A variational model for fracture mechanics: numerical experiments, *J. Mech. Phys. Solids* 55 (12) (2007) 2513–2537. doi:10.1016/j.jmps.2007.04.011.
URL <http://dx.doi.org/10.1016/j.jmps.2007.04.011>
- [35] G. Bellettini, A. Coscia, Approximation of a functional depending on jumps and corners, *Boll. Un. Mat. Ital. B* (7) 8 (1) (1994) 151–181.
- [36] L. Ambrosio, L. Faina, R. March, Variational approximation of a second order free discontinuity problem in computer vision, *SIAM J. Math. Anal.* 32 (6) (2001) 1171–1197 (electronic). doi:10.1137/S0036141000368326.
URL <http://dx.doi.org/10.1137/S0036141000368326>
- [37] P. C. Hansen, J. G. Nagy, D. P. O’Leary, Deblurring images, Vol. 3 of *Fundamentals of Algorithms*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006, matrices, spectra, and filtering. doi:10.1137/1.9780898718874.
URL <http://dx.doi.org/10.1137/1.9780898718874>
- [38] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, T. Pock, An introduction to total variation for image analysis, in: *Theoretical foundations and numerical methods for sparse recovery*, Vol. 9 of *Radon Ser. Comput. Appl. Math.*, Walter de Gruyter, Berlin, 2010, pp. 263–340. doi:10.1515/9783110226157.263.
URL <http://dx.doi.org/10.1515/9783110226157.263>
- [39] L. Grippo, M. Sciandrone, Globally convergent block-coordinate techniques for unconstrained optimization, *Optim. Methods Softw.* 10 (4) (1999) 587–637. doi:10.1080/10556789908805730.
URL <http://dx.doi.org/10.1080/10556789908805730>
- [40] R. A. Horn, C. R. Johnson, *Matrix analysis*, 2nd Edition, Cambridge University Press, Cambridge, 2013.
- [41] G. Forlani, C. Nardinocchi, M. Scaioni, P. Zingaretti, Complete classification of raw lidar data and 3d reconstruction of buildings, *Pattern Analysis and Applications* 8 (4) (2006) 357–374.