An Augmented Lagrangian Method for TVg + L1-norm Minimization

Jonas Koko, Stéphanie Jehan-Besson

An Augmented Lagrangian Method for $TV_g + L^1$-norm Minimization

Jonas Koko [1] Stéphanie Jehan-Besson [2]

Research Report LIMOS/RR-09-07

October 6, 2009

[1]koko@isima.fr LIMOS, Université Blaise Pascal – CNRS UMR 6158, Complexe des Cézeaux, 63173 AUBIERE, FRANCE

[2]jehan@isima.fr, LIMOS, Université Blaise Pascal – CNRS UMR 6158, Complexe des Cézeaux, 63173 AUBIERE, FRANCE

Abstract

In this paper, the minimization of a weighted total variation regularization term with L^1 norm as the data fidelity term is addressed using Uzawa block relaxation methods. The unconstrained minimization problem is transformed into a saddle-point problem by introducing a suitable auxiliary unknown. Applying a Uzawa block relaxation method to the corresponding augmented Lagrangian functional, we obtain a new numerical algorithm in which the main unknown is computed using Chambolle projection algorithm. The auxiliary unknown is computed explicitly. Numerical experiments show the availability of our algorithm for salt and pepper noise removal or shape retrieval and also its robustness against the choice of the penalty parameter. This last property allows us to attain the convergence in a reduced number of iterations leading to efficient numerical schemes. Moreover, we highlight the fact that an appropriate weighted total variation term, chosen according to the properties of the initial image, may provide not only a significant improvement of the results but also a geometric filtering of the image components.

Keywords: Total variation, L^1 norm, augmented Lagrangian, Fenchel duality, Uzawa methods, salt and pepper noise removal, shape retrieval, geometric filtering.

1 Introduction

In many image processing problems, a denoising step is required to remove noise or spurious details from corrupted pictures. Variational approaches have gained a wide popularity these years due to the possible addition of well-chosen regularity terms. Among the most influential models, we can cite the total variation minimization framework introduced by Rudin and Osher [40] and Rudin, Osher and Fatemi [41]. In this framework, given a noisy image f(x), they propose to recover the original image $u(x)$ by minimizing the total variation under L^2 data fidelity:

$$E(u) = \int_\Omega |\nabla u(x)|\, dx + \lambda \int_\Omega (u(x) - f(x))^2\, dx, \tag{1.1}$$

where $\Omega \subset \mathbb{R}^2$, is the image domain and λ a positive scale parameter.
Such a minimization allows the recovery of a simple geometric description of the image u while preserving boundaries. This framework is then very efficient when denoising images with flat zones but fails in preserving texture details. It also fails in removing contrasted and isolated pixels in images corrupted by an impulse noise. Another drawback is that the minimizer presents a loss of contrast due to the L^2 data fidelity term as mentioned in [19].

Consequently, many recent works propose to investigate the minimization of a total variation regularization term with a L^1 data fidelity term:

$$E(u) = \int_\Omega |\nabla u(x)|\, dx + \lambda \int_\Omega |u(x) - f(x)|\, dx. \tag{1.2}$$

This energy is non strictly convex and thus the global minimizer is not unique on the contrary to the energy (1.1). However the L^1 norm presents some interesting properties [19] and then outperforms the L^2 norm for applications such as impulse noise removal [37, 28] or shape denoising [6, 38, 10]. The minimization of $TV + L^1$ yields a contrast invariant filter [19, 21] and well preserves contrasted features at different scales.

In [10], the authors propose to use a weighted total variation regularization term, denoted by TV_g, instead of TV and they search for the image u which minimizes:

$$E(u) = \int_\Omega g(x)|\nabla u(x)|\, dx + \lambda \int_\Omega |u(x) - f(x)|\, dx, \tag{1.3}$$

where $g : \Omega \to \mathbb{R}^+$ is a function independent of u.
When g is chosen as an edge indicator function of the input image (e.g., $g(x) = 1/(1+|\nabla f|)$), the weighted TV norm allows a better preservation of corners and sharp angles in shape denoising. More important, the introduction of such a function allows to establish a link between TV_g and the Geodesic active contours model introduced by [13, 31, 14] as an improvement of the original snakes [30]. This point will be further explained in Section 3 devoted to the geometric properties of the model.

The minimization of functionals (1.2) and (1.3) is not trivial due to their non differentiability. Recent papers addressed the minimization of $TV + L^1$ using various numerical algorithms. For example, standard calculus of variations and Euler-Lagrange equations can be used to compute the PDE that will drive the functional u towards a minimum [6, 38, 10]. This method requires

a smooth approximation of the L^1 norm and a small time step much be chosen so as to ensure the convergence. This often leads to a large number of iterations as mentioned by [10]. In [16], a MRF (Markov Random Field) model is proposed which uses the anisotropic separable approximation (i.e. $|\nabla u| = |D_x u| + |D_y u|$ where D_x and D_y are the horizontal and vertical discrete derivative operators). This approximation is also used in [22, 23] where the authors proposed an efficient graph-cut method. In all these approaches, an approximation or a smoothing of the L^1 norm is required. In [10], following the works of [18, 15, 4] and more particularly [5], an elegant fast minimization algorithm based on a dual formulation is proposed. Thanks to such approaches, they do not need any approximation or smoothing of the L^1 norm, they only use a convex regularization of the criterion as follows (first proposed by [5]):

$$E_r(u,v) = \int_\Omega g(x)|\nabla u(x)|\,dx + \frac{r}{2}\int_\Omega |u(x)+v(x)-f(x)|^2\,dx + \lambda\int_\Omega |v(x)|\,dx \qquad (1.4)$$

In their algorithm, the penalty parameter r must be chosen large enough so as to ensure that $f = u+v$ where the function u represents the geometric information (piecewise smooth regions) and the function v captures the texture information [34, 43, 5]. The choice of the penalty parameter r can then be problematic and the influence of this parameter on convergence must be deeply studied. Moreover, choosing r too large may lead to an increase computational time (ill-conditioning).

Based on this very interesting work and after a reminder of the geometric properties of this model, we propose a new numerical scheme for the minimization of (1.3) using Uzawa (dual) methods. Indeed, (1.4) is the penalty functional associated with the constrained minimization problem

$$\min_{u+v=f} E(u,v) = \int_\Omega g(x)|\nabla u(x)|\,dx + \lambda\int_\Omega |v(x)|\,dx. \qquad (1.5)$$

The "natural" improvement of the method proposed by [10] is then to associate the penalty and multiplier methods, i.e. an augmented Lagrangian method. The augmented Lagrangian method combines the features of the penalty and primal-dual approach and moderates the disadvantages of both. Moreover, convergence in augmented Lagrangian methods can usually be attained without the need to increase r to infinity, see e.g. [7]. We then propose a Uzawa block relaxation algorithm based on the augmented Lagrangian functional associated with (1.5). In each iteration of our algorithm, the main unknown u is computed using Chambolle algorithm [15], and the auxiliary unknown v is computed explicitly using Fenchel duality theory.

The numerical scheme is then tested and evaluated for salt and pepper noise removal and shape denoising in order to demonstrate the applicability of our method. We show that the proposed algorithm is robust against the choice of the penalty parameter r. The optimal choice for the penalty parameter leads to an efficient scheme in terms of number of iterations and then computational time. Besides, we also propose to study the influence of well-chosen functions g in order to improve shape retrieval or salt-and-pepper noise removal. An efficient algorithm, denoted UBR-EDGE, is then proposed and evaluated for this last application. We also give an example showing that this function can help us to perform a geometrical filtering of shape components.

The paper is organized as follows. In Section 2 we present the $TV_g + L^1$ model followed by its geometric properties in Section 3. In Section 4, we introduce the augmented Lagrangian

formulation of the $TV_g + L^1$ model. The Uzawa block relaxation method is detailed in Section 5. The convergence of the algorithm is presented in Section 6, followed by numerical experiments in Section 7.

2 Introduction of the $TV_g + L^1$-norm minimization problem

Let Ω be a two-dimensional bounded open domain of $\mathbb{R}^d$ with Lipschitz boundary. An image can be interpreted as a real function defined on Ω or a suitable discretization of this continuous image. We consider the following convex energy functional defined, for any $f \in L^1(\Omega)$, any $g : \Omega \to \mathbb{R}^+$ and any positive parameter λ:

$$E(u) = \int_\Omega g(x)|\nabla u(x)|\,dx + \lambda \int_\Omega |u(x) - f(x)|\,\mathrm{d}x \tag{2.1}$$

Our aim is the minimization of the energy functional E, i.e.

$$\min_{u \in BV(\Omega)} E(u), \tag{2.2}$$

where $BV(\Omega)$ is the subspace of functions $u \in L^1(\Omega)$ such that

$$\int_\Omega |\nabla u| := \sup \left[\int_\Omega u \nabla \cdot \varphi \,\mathrm{d}x \mid \varphi \in \mathscr{C}_c^1(\Omega, \mathbb{R}^2),\ |\varphi| \leq 1 \right] < \infty$$

with $\nabla \cdot \varphi = \mathrm{div}\varphi$. It is known that $BV(\Omega)$ is a Banach space when equipped with its "natural" norm

$$\| u \|_{BV(\Omega)} = \| u \|_{L^1(\Omega)} + \int_\Omega |\nabla u|.$$

In order to approximate (2.1) by an augmented Lagrangian, we use the following minimization problem

$$\min_{u \in V} E(u) \tag{2.3}$$

where $V = W^{1,1}(\Omega) \cap L^2(\Omega)$. In practice, discrete operators are considered.

In two-dimensional form, an image is an array of size $N \times N$. The Euclidean space $\mathbb{R}^{N \times N}$ is denoted by X and equipped with the L^2 scalar product $(u,v)_X = \sum_{1 \leq i,j \leq N} u_{ij} v_{ij}$ and the norm $\| u \|_{L^2} = \sqrt{(u,u)_X}$. The L^1 norm is defined by $\| u \|_{L^1} = \sum_{1 \leq i,j \leq N} |u_{ij}|$. If $u \in X$, ∇u is a vector in $Y = X \times X$ defined by $\nabla u = ((\nabla u_{ij})^1, (\nabla u_{ij})^2)$, with

$$(\nabla u_{ij})^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases} \qquad (\nabla u_{ij})^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}$$

A discrete version of the divergence operator must be defined by analogy with the continuous setting $(p, \nabla u)_Y = -(\mathrm{div}(p), u)_X$:

$$(\mathrm{div}(p))_{ij} = \begin{cases} p^1_{i,j} - p^1_{i-1,j} & \text{if } 1 < i < N \\ p^1_{i,j} & \text{if } i = 1 \\ -p^1_{i-1,j} & \text{if } i = N \end{cases} + \begin{cases} p^2_{i,j} - p^2_{i,j-1} & \text{if } 1 < j < N \\ p^2_{i,j} & \text{if } j = 1 \\ -p^2_{i,j-1} & \text{if } j = N \end{cases}$$

We sometimes use the notation $\nabla \cdot p$ for $\mathrm{div}(p)$.

The discrete total variation and fidelity terms are then

$$J(u) := \int_\Omega g(x)|\nabla u(x)|\,dx = \sum_{1\le i,j\le N} g_{ij}|\nabla u_{ij}|, \tag{2.4}$$

$$F(u) := \lambda \int_\Omega |u(x)-f(x)|\,dx = \lambda \sum_{1\le i,j\le N} |u_{ij}-f_{ij}|, \tag{2.5}$$

and the minimization problem (2.3) becomes

$$\min_{u\in X} E(u) = J(u)+F(u). \tag{2.6}$$

From now and through the rest of the paper, we will consider the discrete functionals (2.4)-(2.5) and the discrete minimization problem (2.6). We will sometimes use the continuous notations; however, the reader has to keep in mind that only the discrete case is considered.

3 Geometric properties of the model

In this section, we propose to remind some geometric properties of the TV, $TV+L^1$ and TV_g+L^1 functionals.

In [38], the authors propose a geometric interpretation of the energy criterion L^1+TV in terms of the level sets of u and f. We remind here for completeness the main results of their study using the notations introduced in [21].

Let us denote the lower level sets of an image as follows:

$$L^\alpha(u) = \{x, u(x) < \alpha\}, \tag{3.1}$$

and the upper level sets as:

$$U^\alpha(u) = \{x, u(x) > \alpha\}. \tag{3.2}$$

For each level α, $U^\alpha(u)$ and $L^\alpha(u)$ denote two sets of the image u. From a geometrical point of view, the co-area formula [26] states that, for any function which belongs to the space of bounded variations $\mathrm{BV}(\Omega)$, there is a relation between the TV regularization term and the perimeter $Per(U^\alpha)$ of the set U^α. Indeed, we can write for all $\alpha \in [0,1]$:

$$Per(U^\alpha(u)) = \int_\Omega |\nabla \chi_{U^\alpha(u)}|\,dx, \tag{3.3}$$

where $\chi_{U^\alpha}(u)$ stands for the characteristic function of the set $U^\alpha(u)$. In [38], the authors lighten the fact that the energy (1.2) can be written as an integration over the different level set values α of the images u and v of the energy E^α:

$$E(u) = \int_{\mathbb{R}} E^\alpha(u,f)\,d\alpha, \tag{3.4}$$

with

$$E^\alpha(u,f) = Per(U^\alpha(u)) + \lambda |U^\alpha(u)\Delta U^\alpha(f)| \tag{3.5}$$

where the second term represents the Lebesgue measure of the symmetric difference between the two sets $U^\alpha(u)$ and $U^\alpha(f)$. Such a geometrical feature may contribute to explain the properties of the $TV+L^1$ energy. Indeed, when decreasing the weight λ of the data term, components will be removed in an order determined by their size and their geometry. For example, small components will be removed first and sharp angles will be smoothed. Moreover, this criterion can be exploited to formulate efficient shape segmentation algorithms [38, 9]. Indeed, let us consider that the initial function f is a binary shape defined by $U^\alpha(f)$, the main idea is to find the minimizer u of the energy (1.2) in the space of all functions rather than in the non-convex collection of characteristic sets. In [38], the authors show that any set $U^{\alpha_1}(u)$, obtained by a simple threshold of the result function u, is a global minimizer of (3.4). The main problem of this approach lies in the choice of the level α_1 for thresholding. Note that this method can be extended to the segmentation of an image in two regions [38, 10] based on the Mumford-Shah functional [35]. It can then be considered as an alternative to geometrical PDEs (Partial Differential Equations) classically used in active contours [20, 39]. On the contrary to these approaches, it provides a global optimum but this optimum is not unique (the criterion is not strictly convex).

When dealing with the weighted TV norm, similar results can be stated [10]. Indeed, the TV_g term, when applied to a characteristic set is equivalent to a weighted perimeter

$$TV_g(U^\alpha(u)) = \int_C g(s)ds, \tag{3.6}$$

where C designates the boundary of the set $U^\alpha(u)$ and s its arc length. Such a term corresponds to the energy criterion introduced by [14] under the name of *geodesic active contours*. The introduction of the function g may then be used to minimize a weighted length that takes benefit of image properties. In [14], an edge indicator function is introduced ($g(x) = 1/(1+\beta|\nabla I|)$) in the criterion (3.6) in order to segment objects with strong boundaries in images corrupted with a Gaussian noise.

In [10], based on the results of [38], the authors propose to take benefit of the relation between TV_g and the criterion (3.6) to address the segmentation problem. Let us remind their main theorem in order to be self content.

Theorem 3.1 ([10]) *Suppose that $g(x) \in [0,1]$ and $f(x)$ the given image is a characteristic function of a bounded domain $\Omega_f \subset \Omega$, for any given $\lambda > 0$, if u(x) is any minimizer of the criterion (1.3), then for almost every $\alpha_1 \in [0,1]$ we have that the characteristic function of the set $U^{\alpha_1}(u)$ is a global minimizer of the criterion (1.3).*

Hence, minimizing $TV_g + L^1$ norm can be interpreted as the research of an optimal domain that minimizes the geodesic active contour energy with an additional data fidelity term based on the symmetric difference between shapes. In [10], the authors demonstrate that this algorithm can then be exploited for shape segmentation. Here again, the choice of the coefficient α_1 can be problematic, since for each $\alpha_1 \in [0,1]$, the set $U^{\alpha_1}(u)$ is a potential solution.

As far as image denoising or shape denoising is concerned, we propose to choose the function g according to the type of noise of the corrupted image. Indeed, an edge indicator function

is well appropriated for a Gaussian noise but not for a salt and pepper noise. In our experimental results, we test three different functions g and we study their influences on the final results. A substantial improvement of both the restoration quality and the segmentation result can be observed when using a suitable function g. Moreover it can also be used to select some image components according to their geometry as demonstrated in the last section dedicated to the experimental results.

Let us now introduce our augmented Lagrangian method for the derivation of an efficient numerical scheme for $TV_g + L^1$ minimization.

4 Augmented Lagrangian formulation

In this section we present Uzawa (dual) methods for solving (2.6). To this end, we need to transform the convex minimization problem (2.6) into a suitable saddle-point problem by introducing an auxiliary unknown.

Let us introduce the auxiliary unknown $p = f - u$ and rewrite the functional E as

$$E(u,p) = J(u) + F(p). \tag{4.1}$$

For consistency, we introduce the constraints set

$$K = \{(u,p) \in X \times X \mid u + p - f = 0 \text{ in } X\}.$$

The unconstrained minimization problem (2.6) becomes

$$\min_{(u,p)\in K} E(u,p). \tag{4.2}$$

To problem (4.2), we associate the Lagrangian functional $\mathscr{L}$, defined on $X \times X \times X$, by

$$\mathscr{L}(u,p;s) = E(u,p) + (s, u + p - f)_X. \tag{4.3}$$

In (6), $(\cdot,\cdot)_X$ denotes the $L^2(X)$ scalar product. The corresponding saddle-point problem is then

$$\sup_{s\in X} \inf_{(u,p)\in K} \mathscr{L}(u,p;s). \tag{4.4}$$

Since E is convex, proper and lower semi-continuous, a saddle point $(u^*, p^*; s^*) \in X \times X \times X$ of $\mathscr{L}$ exists and verifies

$$\mathscr{L}(u^*,p^*;s) \leq \mathscr{L}(u^*,p^*;s^*) \leq \mathscr{L}(u,p;s^*), \quad \forall (u,p,s) \in X \times X \times X.$$

We now introduce the augmented Lagrangian functional, defined by

$$\mathscr{L}_r(u,p;s) = \mathscr{L}(u,p;s) + \frac{r}{2} \parallel u + p - f \parallel_{L^2}^2 \tag{4.5}$$

where $r > 0$ is the penalty parameter. It can be proved (easily) that a saddle point of $\mathscr{L}_r$ is a saddle point of $\mathscr{L}$ and conversely. This is due to the fact that the quadratic term in $\mathscr{L}_r$ vanishes when the constraint $u + p - f = 0$ is satisfied.

5 Uzawa block relaxation methods

Uzawa block relaxation methods have been used in nonlinear mechanics for operator splitting and domain decomposition methods [27, 29, 32]. Applying Uzawa block relaxation method to the saddle point problem (4.4) we obtain the following algorithms.

Algorithm UBR

Initialization. p^{-1}, s^0 and $r > 0$ given.

$k \geq 0$. Compute successively u^k, p^k and s^k as follows.

Step 1. Find $u^k \in X$ such that

$$\mathscr{L}_r(u^k, p^{k-1}; s^k) \leq \mathscr{L}_r(v, p^{k-1}; s^k), \quad \forall v \in X. \tag{5.1}$$

Step 2. Find $p^k \in X$ such that

$$\mathscr{L}_r(u^k, p^k; s^k) \leq \mathscr{L}_r(u^k, q; s^k), \quad \forall q \in X. \tag{5.2}$$

Step 3. Update the Lagrange multiplier

$$s^{k+1} = s^k + r(u^k + p^k - f).$$

The algorithm UBR corresponds to the generic Uzawa block relaxation algorithm ALG2 (see, e.g., [27, 29]). We detail the algorithm above in the next subsections.

5.1 Solution of sub-problem (5.1)

The functional $u \mapsto \mathscr{L}_r(u, p^{k-1}; s^k)$ can be rewritten as

$$\Phi_1(u) := \frac{r}{2} \parallel u \parallel^2_{L^2} + J(u) + (\tilde{p}^{k-1}, u)_X + C \tag{5.3}$$

where $\tilde{p}^{k-1} = s^k + r(p^{k-1} - f)$ and C is a constant which does not count in the minimization.

Let $\mathscr{F}_1 : X \to \mathbb{R}$ and $\mathscr{G}_1 : Y = X \times X \to \mathbb{R}$ be defined by

$$\mathscr{F}_1(u) = \frac{r}{2} \parallel u \parallel^2_{L^2} + (\tilde{p}^{k-1}, u)_X, \tag{5.4}$$

$$\mathscr{G}_1(v) = \int_\Omega g|v| \, dx. \tag{5.5}$$

Setting $\Lambda = \nabla \in \mathscr{L}(X, Y)$, the sub-problem in u can be rewritten as

$$(P_1) \qquad \inf_{u \in X} \mathscr{F}_1(u) + \mathscr{G}_1(\Lambda u).$$

The Fenchel dual problem of (P_1) is

$$(P_1^*) \qquad \sup_{v^* \in Y} -\mathscr{F}_1^*(-\Lambda^* v^*) - \mathscr{G}_1^*(v^*),$$

where $\Lambda^* \in \mathscr{L}(Y,X)$ is the adjoint of Λ ($\Lambda^* v^* = -\mathrm{div}(v^*) = -\nabla \cdot v^*$), $\mathscr{F}_1 : X^* = X \to \mathbb{R} \cup \{+\infty\}$ and $\mathscr{G}_1^* : Y^* = Y \to \mathbb{R} \cup \{+\infty\}$ denote the Fenchel convex conjugate functionals of $\mathscr{F}_1$ and $\mathscr{G}_1$, respectively. $\mathscr{F}_1$ and $\mathscr{G}_1$ satisfy the conditions of the Fenchel duality theorem (see e.g. [25, p. 59]) and then, it follows that no duality gap occurs. The primal solution $\bar{u}$ and the dual solution $\bar{v}^*$ satisfy the extremality condition (see e.g. [25, p. 53])

$$-\Lambda^* \bar{v}^* = \nabla \cdot \bar{v}^* = \nabla \mathscr{F}_1(\bar{u}) = r\bar{u} + \tilde{p}^{k-1}, \tag{5.6}$$

since $\mathscr{F}_1$ is differentiable.

From the definition of the Fenchel convex conjugate functional, we have

$$\begin{aligned} \mathscr{F}_1^*(u^*) &= \sup_{u \in X} (u^*, u)_X - \mathscr{F}_1(u) \\ &= \frac{1}{2r} \parallel u^* - \tilde{p}^{k-1} \parallel_{L^2}^2 . \end{aligned} \tag{5.7}$$

For v^*, the Fenchel convex conjugate of $\mathscr{G}_1$ is

$$\mathscr{G}_1^*(v^*) = \sup_{v \in Y} (v^*, v)_Y - \int_\Omega g|v|dx = \begin{cases} 0 & \text{if } |v^*| \le g, \\ +\infty & \text{if } |v^*| \ge g. \end{cases} \tag{5.8}$$

Substituting (5.7) and (5.8) into (P^*), we get the dual problem

$$\begin{aligned} \sup_{v^*} -\mathscr{F}_1^*(-\Lambda^* v^*) - \mathscr{G}_1^*(v^*) &= \sup_{|v^*| \le g} -\frac{1}{2r} \parallel \nabla \cdot v^* - \tilde{p}^{k-1} \parallel_{L^2}^2, \\ &= \inf_{|v^*| \le g} \frac{1}{2r} \parallel \nabla \cdot v^* - \tilde{p}^{k-1} \parallel_{L^2}^2, \\ &= \inf_{|v^*|^2 - g^2 \le 0} \frac{1}{2r} \parallel \nabla \cdot v^* - \tilde{p}^{k-1} \parallel_{L^2}^2 . \end{aligned} \tag{5.9}$$

The Karush-Kuhn-Tucker conditions (see e.g. [8, 33]) applied to the convex problem (5.9) yield the existence of a multiplier $\mu \ge 0$ such that

$$-\frac{1}{r} \nabla (\nabla \cdot v^* - \tilde{p}^{k-1}) + 2\mu v^* = 0, \tag{5.10}$$

$$\mu(|v^*|^2 - g^2) = 0. \tag{5.11}$$

As in Chambolle [15], either the constraint is active or not, we have

$$\frac{1}{r^2} |\nabla (\nabla \cdot v^* - \tilde{p}^{k-1})|^2 - 4\mu^2 g^2 = 0$$

that is,

$$\mu = \frac{1}{2rg} |\nabla (\nabla \cdot v^* - \tilde{p}^{k-1})|. \tag{5.12}$$

Substituting (5.12) into (5.10), we obtain

$$-\nabla(\nabla\cdot v^* - \tilde{p}^{k-1}) + \frac{1}{g}|\nabla(\nabla\cdot v^* - \tilde{p}^{k-1})|v^* = 0. \tag{5.13}$$

Equation (5.13) is identical with [10, Eq.38], up to a (penalty) constant. For solving (5.13), we can then use the fixed-point procedure of Chambolle [15], $v^0 = 0$ and for any $\ell \geq 0$

$$v^{\ell+1} = \frac{v^\ell + \tau\nabla(\nabla\cdot v^\ell - \tilde{p}^{k-1})}{1 + (\tau/g)|\nabla(\nabla\cdot v^\ell - \tilde{p}^{k-1})|}, \tag{5.14}$$

where $\tau > 0$. The Chambolle procedure (5.14) can be viewed as a semi-implicit Euler scheme for computing the stationary solution of the following evolution equation

$$\frac{\partial v}{\partial t} - \nabla(\nabla\cdot v - \tilde{p}^{k-1}) + \frac{1}{g}|\nabla(\nabla\cdot v - \tilde{p}^{k-1})|v = 0$$

that is,

$$\frac{v^{\ell+1} - v^\ell}{\tau} - \nabla(\nabla\cdot v^\ell - \tilde{p}^{k-1}) + \frac{1}{g}|\nabla(\nabla\cdot v^\ell - \tilde{p}^{k-1})|v^{\ell+1} = 0.$$

Finally, with v^* computed using (5.14), we compute $\bar{u}$ using the extremality condition (5.6), i.e.

$$\bar{u} = \frac{1}{r}(\nabla\cdot v^* - \tilde{p}^{k-1}) = f - p^{k-1} + \frac{1}{r}(\nabla\cdot v^* - s^k). \tag{5.15}$$

5.2 Solution of sub-problem (5.2)

The functional $p \mapsto \mathscr{L}_r(u^k, p; s^k)$ can be rewritten as

$$\Phi_2(p) = \frac{r}{2} \parallel p \parallel_{L^2}^2 + (\tilde{u}^k, p)_X + \lambda \int_\Omega |p| dx + C,$$

where $\tilde{u}^k = s^k + r(u^k - f)$ and C is a constant which does not count in the minimization. As in previous subsection we set

$$\begin{aligned} \mathscr{F}_2(p) &= \frac{r}{2} \parallel p \parallel_{L^2}^2 + (\tilde{u}, p)_X, \quad \forall p \in X, \\ \mathscr{G}_2(q) &= \lambda \int_\Omega |q| dx, \quad \forall q \in X. \end{aligned}$$

Setting $\Lambda = Id$ (the identity operator), the sub-problem (5.2) reads

$$(P_2) \qquad \inf_{p\in X} \mathscr{F}_2(p) + \mathscr{G}_2(\Lambda p)$$

for which the dual problem is

$$(P_2^*) \qquad \sup_{q^*\in X} -\mathscr{F}_2^*(-q^*) - \mathscr{G}_2^*(q^*).$$

A straightforward calculation, using fenchel convex conjugate functional, yields

$$\begin{aligned}\mathscr{F}_2^*(p^*) &= \frac{1}{2r}\parallel p^* - \tilde{u}^k \parallel_{L^2}^2, \quad \forall p^* \in X \\ \mathscr{G}_2^*(q^*) &= \begin{cases} 0 & \text{if } |q^*| \leq \lambda, \\ +\infty & \text{if } |q^*| \geq \lambda \end{cases}, \quad \forall q^* \in X,\end{aligned}$$

with the extremality condition between the primal solution $\bar{p}$ and the dual solution $\bar{q}^*$

$$-\bar{q}^* = r\bar{p} + \tilde{u}^k.$$

Gathering the results above, we compute the solution of the sub-problem (5.2) explicitly

$$p^k = \begin{cases} 0 & \text{if } |s^k + r(u^k - f)| \leq \lambda, \\ f - u^k - \frac{1}{r}\left[s^k - \lambda \frac{s^k + r(u^k - f)}{|s^k + r(u^k - f)|}\right] & \text{if } |s^k + r(u^k - f)| \geq \lambda. \end{cases}$$

5.3 Uzawa block relaxation algorithms

With the results of the previous section, we can present our Uzawa block relaxation algorithm.

Algorithm UBR

Initialization. p^{-1}, s^0 and $r > 0$ given.

Iteration $k \geq 0$. Compute successively u^k, p^k and s^k as follows.

Step 1. Set $\tilde{p}^{k-1} = s^k + r(p^{k-1} - f)$ and compute v^k with (5.14).
Compute u^k

$$u^k = f - p^{k-1} + \frac{1}{r}(\nabla \cdot v^k - s^k)$$

Step 2. Compute p^k

$$p^k = \begin{cases} 0 & \text{if } |s^k + r(u^k - f)| \leq \lambda, \\ f - u^k - \frac{1}{r}\left[s^k - \lambda \frac{s^k + r(u^k - f)}{|s^k + r(u^k - f)|}\right] & \text{if } |s^k + r(u^k - f)| \geq \lambda. \end{cases}$$

Step 3. Update the Lagrange multiplier

$$s^{k+1} = s^k + r(u^k + p^k - f).$$

We iterate until the relative error in u^k and p^k becomes sufficiently "small".

6 Convergence

We first rewrite the constrained optimization problem (4.2) in a standard form by setting

$$G(u) = J(u), \quad F(p) = \lambda \parallel p \parallel_{L^1} .$$

Let us introduce the linear and continuous operator $B : X \to X$, defined by

$$Bu = u - f.$$

We observe that the constrained minimization problem (4.2) is equivalent to unconstrained minimization problem

$$\min_{u \in X} G(u) + F(Bu).$$

The augmented Lagrangian functional (4.5) can be rewritten as

$$\mathscr{L}_r(u, p; s) = G(u) + F(p) + (s, Bu + p)_X + \frac{r}{2} \parallel Bu + p \parallel^2_{L^2} .$$

Since F and G are convex, proper and lower semi-continuous functionals and the constraint is linear, a saddle-point for $\mathscr{L}$ exists. We easily verify the the functional $(u, p) \mapsto \mathscr{L}_r(u, p; s)$ is coercive on $X \times X$, proper in u (for any fixed p and s) and proper in p (for any fixed u and s). Algorithm UBR is therefore equivalent to finite dimensional version of ALG2 described in [29, chapter 3]. We have the following convergence theorem, [29, theorem 4.2].

Theorem 6.1 (Convergence) *The sequence (u^k, p^k, s^k) generated in Algorithm UBR is such that*

$$u^k \to u^* \text{ in } X, \quad p^k \to p^* \text{ in } X, \quad s^k \to s^* \text{ in } X,$$

(u^*, p^*, s^*) being a saddle-point of $\mathscr{L}_r$.

Since we are in finite dimension, it is not necessary to assume the uniform convexity of F or of G, [29, Remark 4.4-4.6].

7 Numerical experiments

In this section, we present some numerical examples to evaluate the algorithm UBR for applications such a salt and pepper noise removal (section 7.1), shape denoising (section 7.2) or geometric filtering (section 7.3). The influence of the penalty parameter r is more particularly studied and we propose to test the robustness of our numerical scheme against variations of this parameter. The stability of our algorithm regarding with this parameter allows to obtain the convergence in a reduced number of iterations without decreasing the quality of the result. We also propose to take benefit of the function $g(x)$ to improve the denoising results by choosing an appropriate function for the different noise models. This function can also help us to perform a geometric filtering of shapes.

In all numerical experiments, the convergence of the algorithm UBR is checked using the following convergence criterion:

$$\frac{\sqrt{||u^k - u^{k-1}||_2^2 + ||p^k - p^{k-1}||_2^2}}{\sqrt{||u^k||_2^2 + ||p^k||_2^2}} \leq \varepsilon_{up} \tag{7.1}$$

Note that, each iteration of Algorithm UBR requires the convergence of the Chambolle fixed point procedure (5.14). The convergence of this loop is checked using a threshold on the normalized L^2 error on v^l. In the experiments, we choose $\varepsilon_{FP} = 0.5$ for the first iterations and $\varepsilon_{FP} = 0.1$ to end the process. According to our experiments (not reported here), increasing the accuracy of the Chambolle fixed point procedure does not improve the final result whereas it increases the computational cost of each iteration.

The numerical experiments were run in C^{++} with the library Pandore [3] developed by R. Clouard. The salt and pepper noise was generated with gmic [4] proposed by D. Tschumperle, except the images of the Figure 12 which were downloaded from the page of R. Chan [5].

7.1 Salt and pepper noise removal

Salt and pepper noise is a model that can represent the effects of bit errors in transmission or faulty memory locations. In salt and pepper noisy images, the noisy pixels can take only the minimum or maximum values in the dynamic range of image values. For such images, the use of the L^1 norm is then well suited due to its link to median filtering. It has been used by [1, 2, 3] for 1D data and by [36, 37, 28, 6] for efficient image denoising algorithms. Two-phase approaches are also proposed in [17, 11, 12] with very nice results for a high level of salt an pepper noise. In this paper, we first propose to test the robustness of our dual algorithm, named UBR, for the denoising of the image "peppers" (Figure 1.(a)). In a second step, we propose to take benefit of a dedicated function g in order to increase the quality of the results. Our algorithm is then embedded in a more complete process, named UBR-EDGE, that is evaluated for the denoising of various images corrupted with a high level of noise.

The restoration performances are classically measured by the PSNR (peak signal-to-noise ratio) defined as follows:

$$PSNR = 10 \log_{10} \frac{max^2}{\frac{1}{|\Omega|} \sum_{i,j} (I_0(i,j) - I_R(i,j))^2} \tag{7.2}$$

where max denotes the maximum value of I (for 8-bits images $max = 255$) and $|\Omega|$ is the number of pixels (i, j) of the image I_0. We note $I_0(i, j)$ and $I_R(i, j)$ the discrete values of I_0, the original image, and I_R, the restored image. This value is inversely proportional to the mean square error and so a higher value of PSNR corresponds to a better restoration result (note that this is only an overall measure that must not be used without a visualization of the results).

[3] available at http://www.greyc.ensicaen.fr/regis/Pandore/

[4] http://gmic.sourceforge.net/

[5] http://www.math.cuhk.edu.hk/ rchan/paper/impulse/

(a) Peppers (b) Lena

Figure 1: Input image: (a) Peppers (256x256), (b) Lena (512x512)

7.1.1 Robustness of UBR

Firstly, the experimental results provided in Figure 2 show the applicability of our numerical scheme (named UBR) for this application. With the function $g(x) = 1$ and $\lambda = 1.5$, we find a PSNR of 32.5 dB for the denoising of an image corrupted with a noise of 10%. Noise is correctly removed as can be observed in Figure 2.b, moreover, the noisy part is captured through the auxiliary unknown v as displayed in Figure 2.c.

(a) Noisy image (b) Final image u (c) Final image v

Figure 2: The two different images u (PSNR= 32.5dB) and v obtained after convergence with UBR and with $g(x) = 1$ ($\lambda = 1.5$, $r = 20$, $\varepsilon_{up} = 0.0001$) for the image "peppers" with a salt and pepper noise of 10%.

Secondly, we want to study the robustness of the result against the choice of the parameter r. Our experimental results show that the algorithm UBR provides the same denoised image for different values of r. This is demonstrated by the Figure 3 that displays the evolution of the PSNR according to the number of iterations for different parameters r (from 10 to 200). Such a feature then represents an improvement of the method proposed in [10] since the convergence can be obtained without the need to increase r to infinity.

We also report the number of iterations according to r (Figure 4). In this case, the optimal value in terms of iterations is obtained for $r = 30$ with 60 iterations when $\lambda = 1.5$, and for $r = 10$ with 91 iterations when $\lambda = 0.5$. Choosing a higher value for r increases the number of iterations needed to attain the convergence without improving the final result. We can then choose a small

value for r to obtain a low computational cost without decreasing the quality of the result.

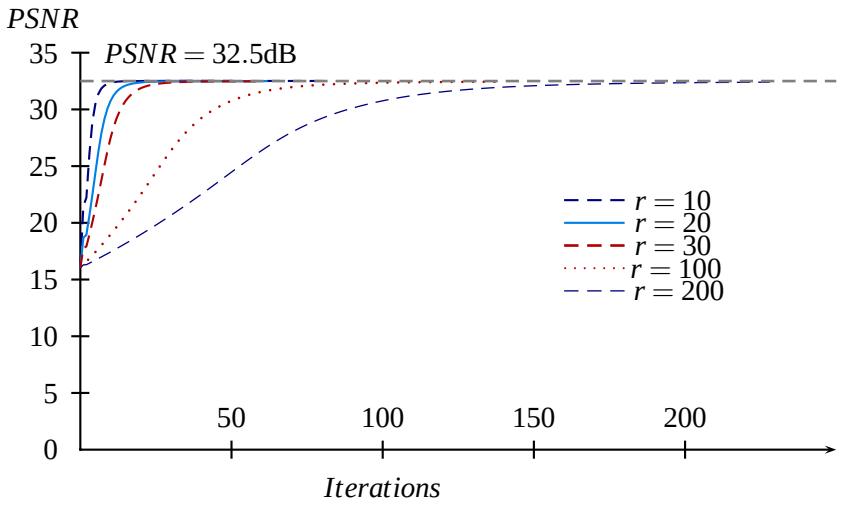


Figure 3: Algorithm UBR (g=1) : Evolution of PSNR during iterations ($\lambda = 1.5$) with $r = 10, 20, 30, 100, 200$ ($\varepsilon_{up} = 0.0001$) for the image "peppers" with a salt and pepper noise of 10% (the function is $g(x) = 1$).

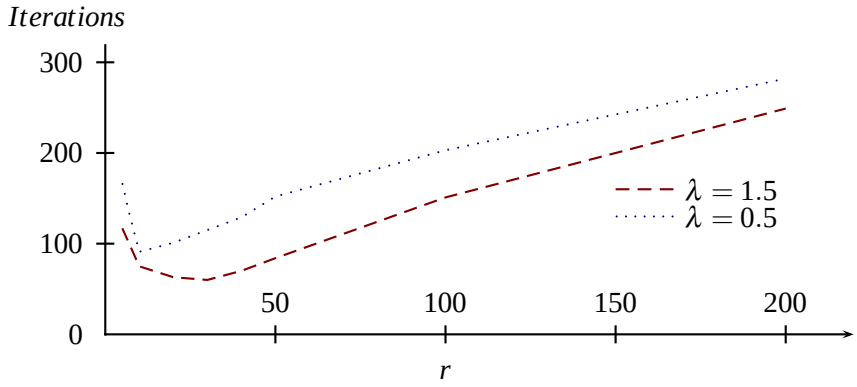


Figure 4: Algorithm UBR (g=1) : Number of iterations for convergence according to the parameter r with $\lambda = 0.5$ and $\lambda = 1.5$ for the image "Peppers" with a salt and pepper noise of 10% ($g(x) = 1$).

7.1.2 Improvement using an appropriate function g

Thirdly, we propose to take benefit of the fact that the dynamic range of the noise is known. We propose to replace the edge indicator function used in [10] for the function $g(x)$ by a regularized version of the following mask function:

$$m(x) = \begin{cases} \alpha_n & \text{if} \quad f(x) = min \quad \text{or} \quad max \\ \alpha & \text{elsewhere} \end{cases} \tag{7.3}$$

where *min* and *max* are respectively the minimum and maximum intensity values of the noisy image f. We choose $\alpha_n = 1.5$ and $\alpha = 0.5$ in order to uppermost smooth the corrupted pixels. We then take $g(x) = m_\sigma(x)$ where $m_\sigma(x) = G_\sigma * m(x)$ is a slight regularized version of m ($\sigma = 0.5$). Figure 5 displays the different values of PSNR and the resulting images obtained while setting $g(x) = 1$ (first row) and $g(x) = m_\sigma(x)$ (second row). Final images are provided for different values of the regularization parameter λ. For each parameter, we observe a significant increase of 2 to 4dB in the final PSNR. The best value of PSNR is 34.5 dB obtained for $\lambda = 1.2$. The scale effect of the parameter λ is also less visible due to the fact that we restrict the regularization term to the extreme values of intensities corresponding to the corrupted pixels.

(a) $\lambda = 0.5, g = 1$ PSNR= 25.5 dB | (b) $\lambda = 1, g = 1$ PSNR= 30.3 dB | (c) $\lambda = 1.5, g = 1$ PSNR= 32.5 dB

(a) $\lambda = 0.5, g = m_\sigma$ PSNR= 29.6 dB | (b) $\lambda = 1, g = m_\sigma$ PSNR= 34.3 dB | (c) $\lambda = 1.5, g = m_\sigma$ PSNR= 34.9 dB

Figure 5: Experimental results with the algorithm UBR for different smoothing values of λ ($r = 20$,$\varepsilon_{up} = 0.0001$) for the image "peppers" with a salt and pepper noise of 10%. The first row displays the results obtained with $g(x) = 1$ while the second row displays the result obtained using $g(x) = m_\sigma(x)$.

In Figure 6, we report the variation of the PSNR according to the parameter λ for $g(x) = 1$ and $g(x) = m_\sigma(x)$. In Table 7, we give the PSNR values for different noise levels and the corresponding computational costs (with $g(x) = m_\sigma(x)$ and $r = 20$). A good quality of restoration is obtained at a low computational cost (from 1.6 seconds for a noise of 10% to 4.3 seconds for a noise of 70% with a computer of $3GHz$ and $2Gb$ of RAM), which confirms the efficiency of our numerical scheme UBR. We use this feature to design our salt and pepper noise removal algorithm detailed thereafter. Note that the parameter r is set to the same value for all the experiments of Table 7. Choosing automatically the value of r in order to obtain the lower computational cost

at each noise level or each parameter λ is an open question that remains difficult to solve.

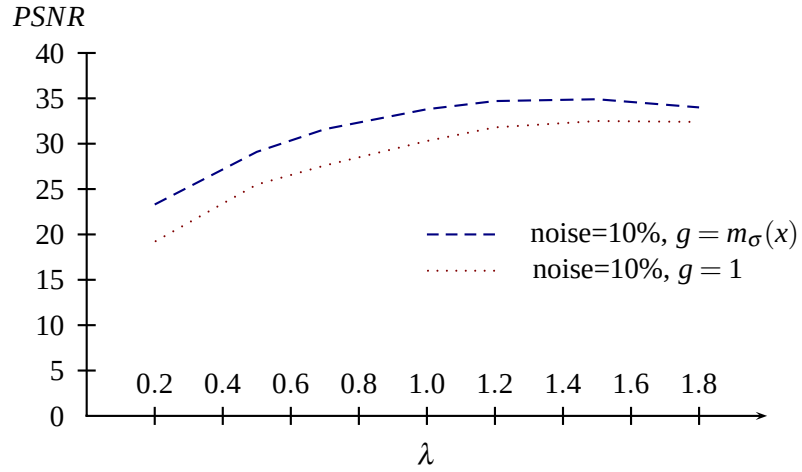


Figure 6: Algorithm UBR : Final PSNR value obtained according to the parameter λ with $r = 20$ for the image "peppers" corrupted with a salt and pepper noise of 10% (with $g(x) = 1$ and with $g(x) = m_\sigma(x)$).

Algorithm UBR ($g = m_\sigma$) for Peppers (256x256)				
Noise level	PSNR	λ	Iterations	Computational cost
10	34.9	1.5	63	1.6 s
20	31.4	1.2	139	2.8 s
30	29.0	1.2	158	3.3 s
40	27.3	1.2	161	3.9 s
50	25.5	1.2	167	4.3 s
70	21.9	1.2	167	4.3 s

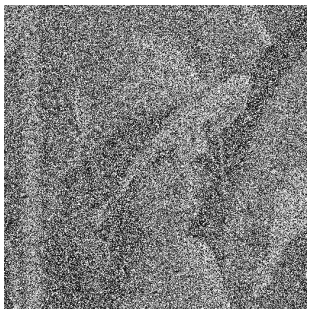
Figure 7: PSNR according to the salt and pepper noise level for the image "peppers" using $g(x) = m_\sigma(x)$ ($r = 20, \varepsilon_{up} = 0.0001$). The computational cost for convergence is obtained with a computer of $3GHz$ with $2Gb$ of RAM.

7.1.3 UBR-EDGE: an algorithm for salt and pepper noise removal

The use of the function g provides a significant increase of the quality of the final results. However, even if the algorithm $TV_g + L^1$ well performs for low noise values, it gives very smoothed results for higher noise values. Indeed, in order to remove large noisy patches, we must decrease the parameter λ and so increase the smoothing of the whole image. In order to improve the results for very noisy images, we propose to first decrease the size of unknown values using a median filter (of half-size 1). The pixels that are still unknown after this first pass are estimated by computing a mean on the known 4-connexity neighbours (i.e. we only take the known values

to compute the mean). The aim of this first pass is to correct the bias introduced by the extreme intensity values of the noisy pixels (*min* or *max*). This first estimation is then corrected using the $TV_g + L^1$ algorithm which is able to smooth differently noisy pixels from uncorrupted ones through the g function. At the end of the process, we apply a very simple edge smoother also known as EDDI [24] usually used in deinterlacing process for electronic devices. In this efficient edge smoother, the unknown intensity value is estimated by computing the mean between the two opposite pixels that have the nearest value of intensity in a 4-neighborhood. We apply this simple filtering scheme only on pixels that are detected as corrupted pixels in the input image.

In Figure 8, we show the different steps of our process for the restoration of the image "Lena" with a salt and pepper noise of 70%. The Figure 8.(c) displays the image obtained after the pre-processing step (median filter + mean). This image is processed as an input of our algorithm UBR using $g(x) = m_\sigma(x)$ and the result of our UBR algorithm is given in Figure 8.(d). The EDGE smoother EDDI is then applied to this result giving the final image Figure 8.(e) which is the result of our UBR-EDGE algorithm.



(a) Input image (70%)

(b) Original image

(c) Step 1: pre-processing
PSNR= 19.6 dB

(d) Step 2: UBR
PSNR= 30.1 dB

(e) Final : UBR + EDGE
PSNR= 30.6 dB

Figure 8: Salt and pepper noise removal using the algorithm UBR-EDGE for the image Lena corrupted by a noise of 70%. The result is given for each step of the process. The image obtained after the pre-processing (median+mean) is given in (c). This image is used as an input of the algorithm UBR and the result is given in (d). A last post-processing is applied to the image which yields to the final result given in (e).

Some visual results are provided in Figure 9 for "Lena" (512x512) and in Figure 10 for

"Peppers" (256x256). Thanks to these visual results and to the associated PSNR values and computational costs reported for all the noise levels in Table 11, we can conclude that our algorithm provides good visual results at a low computational cost. The PSNR values obtained for the image "Lena" can be compared with the PSNR values reported in [17] for many different algorithms. Compared to the values computed in this paper, our algorithm gives comparable PSNR results to the best algorithm (i.e. algorithm III) even for a high noise level. Indeed, the PSNR value reported for the restoration of the image "Lena" corrupted by a noise of 70% is 29.3 dB using the algorithm III [17] and 31.4 dB using our algorithm. For a noise of 90%, they find a PSNR of 25.4 dB while our algorithm gives a PSNR of 26.6 dB. We also report the visual results and associated PSNR values for the noisy images of the web page of R. Chan [6]. For such images our algorithm gives good quality results with a PSNR value that is near to the one found by the algorithm [17] even if a little smaller (with a difference of less than 1 dB). As far as the computational cost is concerned, it is difficult to compare the two computational costs since the algorithm III is programmed using Matlab. However, our algorithm seems to provide a lower computational cost especially for a high level of noise (see Table 11).

7.2 Shape retrieval

The second example concerns shape retrieval with the image ("circle") corrupted by a Gaussian noise of variance 10 (Figure 13). In order to take benefit of the algorithm UBR for segmentation purposes, we apply the same procedure as in [10]. The algorithm is processed until convergence and the final function u is thresholded in order to display the set $U^{\alpha}(u)$ with $\alpha = 0.5$. In all experiments, we display both the characteristic function of this set and its final boundary in white on the initial image. The accuracy of the segmentation result is evaluated using the Dice Coefficient defined as follows between two shapes S_1 and S_{ref}:

$$DC(S_1, S_{ref}) = 2\frac{|S_1 \cap S_{ref}|}{|S_1| + |S_{ref}|}. \tag{7.4}$$

Note that, for a perfectly segmented shape, we have $DC = 1$. Here, S_{ref} is the circle shape of Figure 13.a and $S_1 = U^{0.5}(u)$ (upper level set of u where u is the result of our algorithm UBR).

Let us first test the availability of our scheme for segmentation purposes. In Figure 14 (first and second row) the final results obtained using different values of λ and with $g(x) = 1$ are reported. The boundary of the set $U^{0.5}(u)$ is displayed in white on the noisy image and the extracted shape is represented on the second row using a binary image. Our algorithm allows to properly segment the shape. However, for small values of λ, the TV regularization term smoothes the corners and removes some small components of the shape. In order to avoid this scale space effect and to improve the DC value, we can take benefit of a classical edge indicator function:

$$g(x) = 1/(1 + \beta G_{\sigma} * |\nabla f|).$$

where G_{σ} is a Gaussian kernel of 0-mean and variance σ (we take $\sigma = 0.1$ and $\beta = 10$). Thanks to this function, the DC coefficient is significantly improved as can be observed in Figure 14

[6] http://www.math.cuhk.edu.hk/ rchan/paper/impulse/

(third and fourth rows) and is less dependent on the value of λ (see also the Figure 16). These experiments confirm the interest of the weighted total variation term for segmentation using dual approaches as mentioned in [10].

As far as the robustness against the parameter r is concerned, we again visualize the number of iterations (Figure 15.a) and the DC value (Figure 15.b) according to the parameter r. The experimental results tend to prove that the algorithm converges towards the same final result for each value of r. The number of iterations is provided for two different values of λ (0.5 and 1).

7.3 Geometric filtering

Finally, we give here an example of the applicability of $TV_g + L^1$ minimization for geometric filtering of shapes according to the orientation of their gradients. Such geometric filters are usually designed in the framework of mathematical morphology in order to remove some shapes from a set using their geometric properties [42]. In this paper, we take one example to demonstrate the potential use of $TV_g + L^1$ for such an application. Let us consider that we want to remove the horizontal ellipses from the binary image of Figure 17.a. We then define the regularized mask $m_\varepsilon(x, \theta_{ref}) = \delta_\varepsilon(\theta(x) - \theta_{ref})$, where $\theta(x)$ represents the orientation of the gradient of f and δ_ε is a regularized dirac function equal to 1 when $\theta = \theta_{ref}$ and almost 0 elsewhere. We then use the following function for g:

$$g(x) = 1/(1 + \beta m_\varepsilon(x, \theta_{ref})|\nabla f|).$$

As can be observed in Figure 17, when applying $TV_g + L^1$ using the function defined above for $\theta_{ref} = \pi/2[\pi]$ (horizontal values of the gradient), we make the vertical shapes disappear from the initial image (the resulting binary shape and boundaries are shown in Figure 17).

8 Conclusion

In this paper, we propose to minimize a $TV_g + L^1$ criterion using an augmented Lagrangian method which combines the features of the penalty and primal-dual approach and moderates the disadvantages of both. We propose a Uzawa Block Relaxation (UBR) scheme and we more particularly study the robustness of the algorithm against the penalty parameter r. Experimental results tend to prove that the convergence can be attained without increasing r to infinity. This parameter can then be chosen so as to decrease the number of iterations and therefore the computational cost. We also study the influence of the function g for different applications such as salt and pepper noise removal, shape retrieval or geometric filtering. An appropriate choice for this function improves the final results for both salt and pepper noise removal and shape retrieval. We also show that it can be used to select some shape components according to their geometric properties. Using this function, we propose a whole algorithm for salt and pepper noise removal (UBR-EDGE) that is able to handle high noise levels at a low computational cost. As far as the perspectives are concerned, we can remark that choosing automatically the value of the penalty parameter in order to obtain the lower computational cost for each image is an open question that remains difficult to solve. Our on going research is directed towards this issue and towards the design and evaluation of some other functions g for geometric filtering.

References

[1] Alliney, S.: Digital filters as absolute norm regularizers. IEEE Transactions on Signal Processing **40**(6), 1548–1562 (1992)

[2] Alliney, S.: Recursive median filters of increasing order: a variational approach. IEEE Transactions on Signal Processing **44**(6), 1346–1354 (1996)

[3] Alliney, S.: A property of the minimum vectors of a regularizing functional defined by means of the absolute norm. IEEE Transactions on Signal Processing **45**(4), 913–917 (1997)

[4] Aujol, J.F., Chambolle, A.: Dual norms and image decomposition models. International Journal of Computer Vision **63**(1), 85–104 (2005)

[5] Aujol, J.F., Gilboa, G., Chan, T.F., Osher, S.: Structure-texture image decomposition - modeling, algorithms, and parameter selection. International Journal of Computer Vision **67**(1), 111–136 (2006)

[6] Bar, L., Sochen, N.A., Kiryati, N.: Image deblurring in the presence of salt-and-pepper noise. In: R. Kimmel, N.A. Sochen, J. Weickert (eds.) Scale-Space, Lecture Notes in Computer Science, vol. 3459, pp. 107–118. Springer (2005)

[7] Bertsekas, D.: Constrained Optimization and Lagrange Multipliers Methods. Academic Press, New York (1982)

[8] Bonnans, J.F., Gilbert, J., Lemaréchal, C., Sagastizabal, C.: Numerical Optimization: Theoretical and Numerical Aspects. Springer-Verlag (2003)

[9] Bresson, X., Chan, T.F.: Active contours based on chambolle's mean curvature motion. In: ICIP (1), pp. 33–36. IEEE (2007)

[10] Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.P., Osher, S.: Fast global minimization of the active contour/snake model. Journal of Mathematical Imaging and Vision **28**, 151–167 (2007)

[11] Cai, J., Chan, R., Nikolova, M.: Two-phase methods for deblurring images corrupted by impulse plus gaussian noise. Inverse Probl. Imaging **2**, 187–204 (2008)

[12] Cai, J., Chan, R., Nikolova, M.: Fast two-phase image deblurring under impulse noise. Journal of Mathematical Imaging and Vision (2009)

[13] Caselles, V., Catte, F., Coll, T., Dibos, F.: A geometric model for active contours. Numerische Mathematik **66**, 1–31 (1993)

[14] Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision **22**, 61–79 (1997)

[15] Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision **20**(1-2), 89–97 (2004)

[16] Chambolle, A.: Total variation minimization and a class of binary MRF models. In: Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 136–152 (2005)

[17] Chan, R., Hu, C., Nikolova, M.: An iterative procedure for removing random-valued impulse noise. IEEE Signal Processing Letters pp. 921–924 (2004)

[18] Chan, T., Golub, G., P.Mulet: A nonlinear primal-dual method for total variation-based image restoration. SIAM Journal of Scientific Computing **20**(6), 1964–1977 (1999)

[19] Chan, T.F., Esedoglu, S.: Aspects of total variation regularized L1 function approximation. SIAM Journal of Applied Mathematics **65**(5), 1817–1837 (2004)

[20] Chan, T.F., Vese, L.A.: Active contour without edges. IEEE Transactions on Image Processing **10**, 266–277 (2001)

[21] Darbon, J.: Total variation minimization with L1 data fidelity as a contrast invariant filter. In: International Symposium on Image and Signal Processing and Analysis, pp. 221–226. Zagreb, Croatia (2005)

[22] Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation part I: Fast and exact optimization. Journal of Mathematical Imaging and Vision **26**(3), 261–271 (2006)

[23] Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation part II: Levelable functions, convex priors and non convex cases. Journal of Mathematical Imaging and Vision **26**(3), 277–291 (2006)

[24] De Haan, G., Lodder, R.: De-interlacing of video data using motion vectors and edge information. In: International Conference on Consumer Electronics, pp. 70–71 (2002)

[25] Ekland, I., Temam, R.: Convex Analysis and Variational Problems. SIAM, Philadelphia (1999)

[26] Fleming, W., Rishel, R.: An integral formula for total gradient variation. Archiv der Mathematik **11**, 218–222 (1960)

[27] Fortin, M., Glowinski, R.: Augmented Lagrangian Methods: Application to the Numerical Solution of Boundary-Value Problems. North-Holland, Amsterdam (1983)

[28] Fu, H., Ng, M.K., Nikolova, M., Barlow, J.L.: Efficient minimization methods of mixed l2-l1 and l1-l1 norms for image restoration. SIAM J. Scientific Computing **27**(6), 1881–1902 (2006)

[29] Glowinski, R., Tallec, P.L.: Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics. SIAM, Philadelphia (1989)

[30] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1**, 321–332 (1988)

[31] Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Gradient flows and geometric active contour models. In: International Conference on Computer Vision, pp. 810–815. Boston, USA (1995)

[32] Koko, J.: Uzawa block relaxation domain decomposition method for the two-body contact problem with Tresca friction. Comput. Methods. Appl. Mech. Engrg. **198**, 420–431 (2008)

[33] Luenberger, D.: Linear and Nonlinear Programming. Addison Wesley, Reading, MA (1989)

[34] Meyer, Y.: Oscillating patterns in image processing and nonlinear evolution equations. University Lecture Series **22** (2001)

[35] Mumford, D., Shah, J.: Boundary detection by minimizing functionals. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 22–26 (1985)

[36] Nikolova, M.: Minimizers of cost-functions involving nonsmooth data-fidelity terms. application to the processing of outliers. SIAM Journal on Numerical Analysis **40**(3), 965–994 (2002)

[37] Nikolova, M.: A variational approach to remove outliers and impulse noise. Journal of Mathematical Imaging and Vision **20**(1-2), 99–120 (2004)

[38] Nikolova, M., Esedoglu, S., Chan, T.F.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM Journal of Applied Mathematics **66**(5), 1632–1648 (2006)

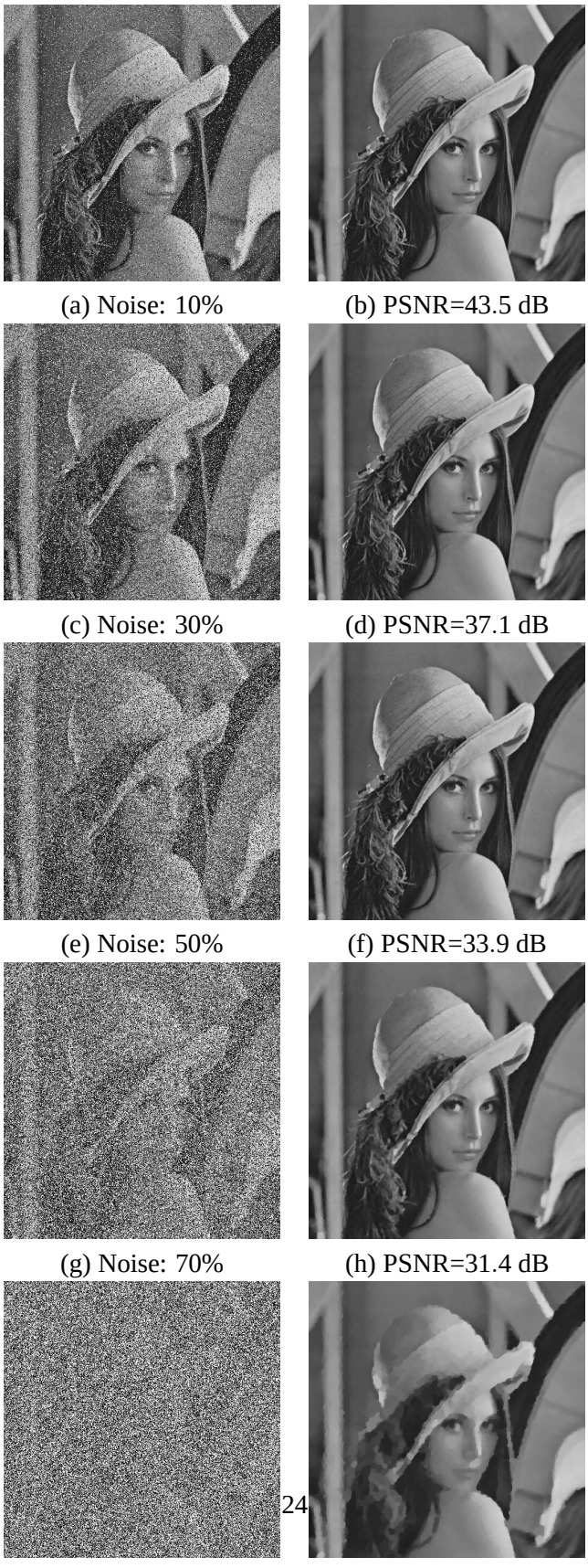
[39] Osher, S.J., Paragios, N.: Geometric Level Set Methods in Imaging, Vision, and Graphics. Springer-Verlag (2003)

[40] Rudin, L., Osher, S.: Total variation based image restoration with free local constraints. In: ICIP, vol. 1, pp. 31–35. Austin, Texas (1994)

[41] Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D. **60**, 259–268 (1992)

[42] Soille, P.: Morphological Image Analysis. Springer-Verlag, Berlin, Heidelberg (1999)

[43] Vese, L., Osher, S.: Modeling textures with total variation minimization and oscillating patterns. Journal of Scientific Computing **19**, 553–572 (2003)



(a) Noise: 10% (b) PSNR=43.5 dB

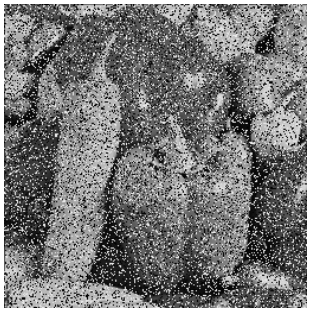
(c) Noise: 30% (d) PSNR=37.1 dB

(e) Noise: 50% (f) PSNR=33.9 dB

(g) Noise: 70% (h) PSNR=31.4 dB

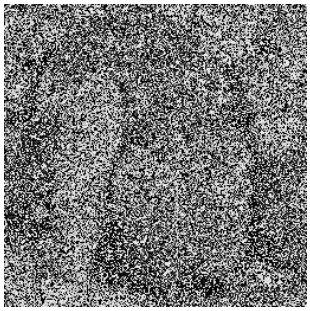
(i) Noise: 90% (j) PSNR=26.6 dB

Figure 9: Salt and pepper noise removal using the algorithm UBR-EDGE for the image Lena (512x512). The input images are given with the associated results.



(a) Peppers, Noise: 30%

(b) UBR-EDGE, PSNR=34.5 dB



(c) Peppers, Noise: 70%

(d) UBR-EDGE PSNR=27.7 dB

Figure 10: Salt and pepper noise removal using the algorithm UBR-EDGE for the image "Peppers" (256x256). For the result obtained in (b), λ=2 and for the result in (d), $\lambda = 1.5$.

	Algorithm UBR-EDGE			
	Lena (512x512)		**Peppers (256x256)**	
Noise	PSNR	Computational cost (s)	PSNR	Computational cost (s)
10	43.4	2.7	40.6	0.4
20	39.7	3.9	37.3	0.7
30	37.1	5.3	34.5	1.1
40	35.3	6.6	32.2	1.4
50	33.9	8.1	30.6	1.7
70	31.4	17.1	27.7	2.3
90	26.6	41.4	23.1	20.1

Figure 11: PSNR according to the salt and pepper noise level for the image "peppers" (256x256) and "Lena" (512x512) using the algorithm UBR-EDGE ($r = 200$, $\varepsilon_{up} = 0.0001$). For a noise level between 10% and 50%, we choose the same value of $\lambda = 2$. For a noise level of 70%, $\lambda = 1.5$ and for 90%, $\lambda = 0.7$. The computational cost for convergence is given with a computer of 3GHz with 2Gb of RAM, it includes the pre- and post- processing steps.

(a) UBR-EDGE: PSNR=22.2 dB

(b) UBR-EDGE: PSNR=33.3 dB

(c) UBR-EDGE: PSNR=26.0 dB

(d) UBR-EDGE: PSNR=25.7 dB

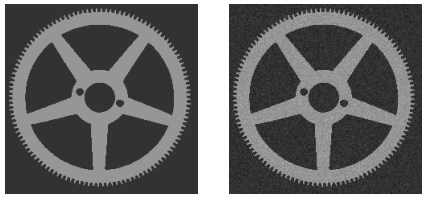
(e) UBR-EDGE: PSNR=29.1 dB

(f) UBR-EDGE: PSNR=23.8 dB

(g) UBR-EDGE: PSNR=31.8 dB

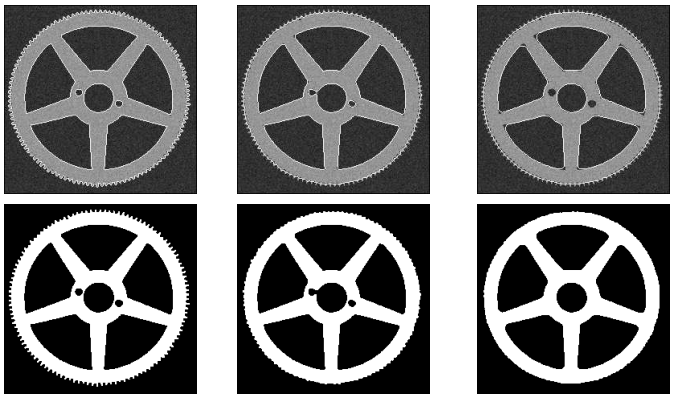
(h) UBR-EDGE: PSNR=35.4 dB

Figure 12: Salt and pepper noise removal using the algorithm UBR-EDGE for different images of the Berkeley database corrupted with a salt and pepper noise of 70%. For all the results, we take $\lambda = 2$.

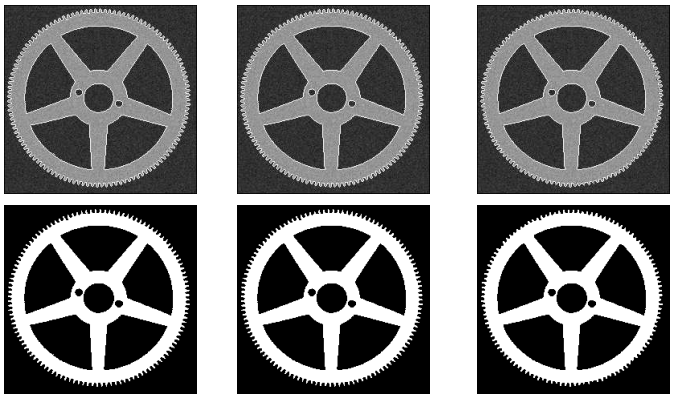


(a) (b)

Figure 13: Input image (a) with a Gaussian noise of 10% (b)



(a) $\lambda = 1, g = 1$ DC= 0.99 (b) $\lambda = 0.5, g = 1$ DC= 0.97 (c) $\lambda = 0.2, g = 1$ DC= 0.95



(a) $\lambda = 1$ DC= 0.999 (b) $\lambda = 0.5$ DC= 0.999 (c) $\lambda = 0.2$ DC= 0.994

Figure 14: Experimental results of shape segmentation with UBR for different smoothing values of λ ($r = 20, \varepsilon_{up} = 0.0001$) for the image "circle" with a Gaussian noise of 10%. The two first rows display the results obtained with $g(x) = 1$ while the two last rows display the result obtained using an appropriate function $g = \varphi(|\nabla I|)$. For each value of λ, we show both the set $U^{0.5}(u)$ and its boundary in white.

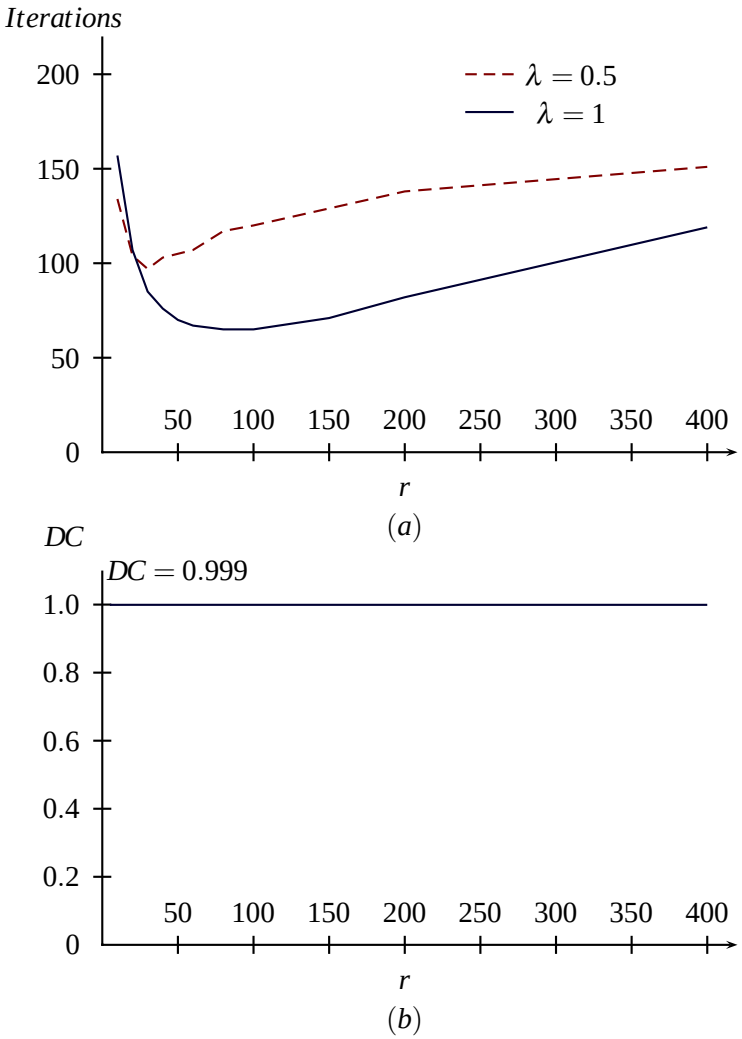


Figure 15: Algorithm UBR with $g(x) = \varphi(|\nabla I(x)|)$: Number of iterations for convergence (a) and dice coefficient (b) according to the parameter r with $\lambda = 0.5$ and $\lambda = 1$ for the segmentation of the image "circle" corrupted with a Gaussian noise of 10% ($\varepsilon_{up} = 0.0001$).

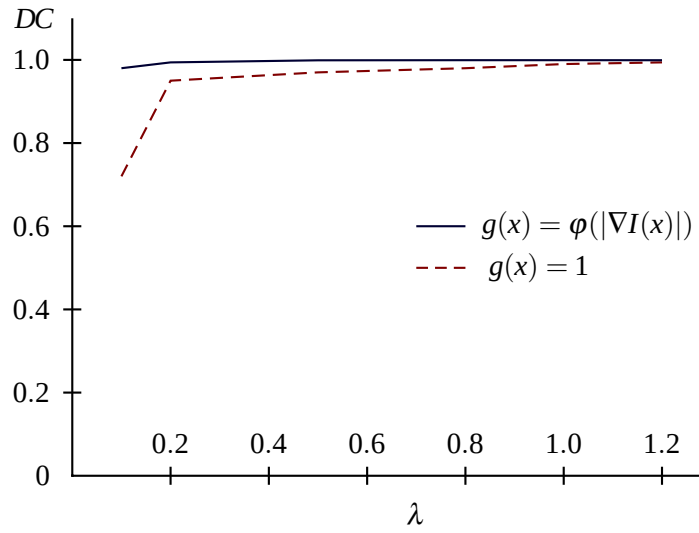
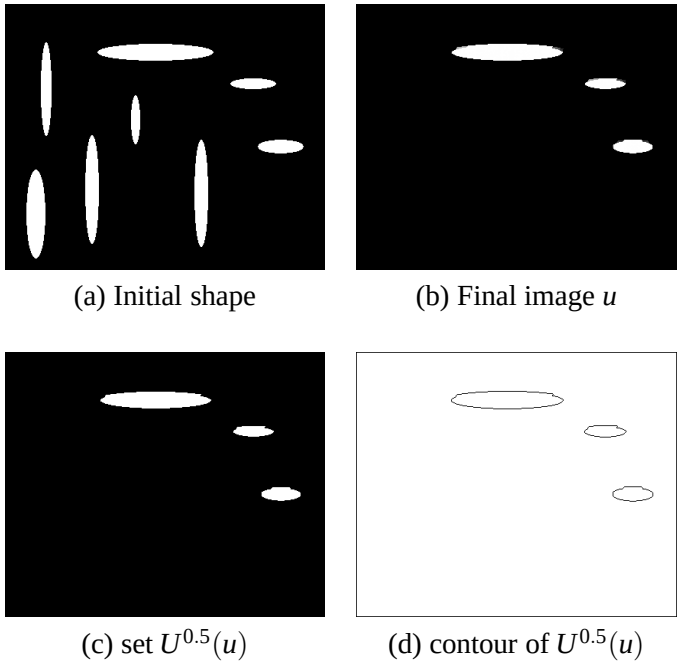


Figure 16: Algorithm UBR: Dice coefficient according to the parameter λ with $g(x) = 1$ and g chosen as a function of the image gradient for the segmentation of the image "circle" corrupted with a Gaussian noise of 10% ($r = 20$,$\varepsilon_{up} = 0.0001$).



(a) Initial shape (b) Final image u

(c) set $U^{0.5}(u)$ (d) contour of $U^{0.5}(u)$

Figure 17: Experimental results of geometrical filtering (selection of horizontal ellipses) with UBR ($r = 20$, $\lambda = 0.05$, $\varepsilon_{up} = 0.0001$) for the image "ellipses". We show the initial image (a), the final image without thresholding (b) the set $U^{0.5}(u)$ (c) and its boundary (d).