FAST DUAL MINIMIZATION OF WEIGHTED TV + $L^1$-NORM FOR SALT AND PEPPER NOISE REMOVAL

S. Jehan-Besson
CNRS, UMR 6158, LIMOS, F-63173, Aubière, France

Jonas Koko
Clermont Université, Université Blaise Pascal, LIMOS, BP 10448, F-63000 Clermont-Ferrand, France
CNRS, UMR 6158, LIMOS, F-63173, Aubière, France

Keywords: Total variation, $L^1$ norm, Augmented Lagrangian, Fenchel duality, Uzawa methods, Salt and pepper noise removal.

Abstract: In this paper, the minimization of a weighted total variation regularization term (denoted $TV_u$) with $L^1$ norm as the data fidelity term is addressed using the Uzawa block relaxation method. Numerical experiments show the availability of our algorithm for salt and pepper noise removal and its robustness against the choice of the penalty parameter. This last property is useful to attain the convergence in a reduced number of iterations leading to efficient numerical schemes. The specific role of the function $g$ in the weighted total variation term is also investigated and we show that an appropriate choice leads to a significant improvement of the final denoising results. 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.

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 (Rudin and Osher, 1994) and Rudin, Osher and Fatemi (Rudin et al., 1992). 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:

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

where $\Omega \subset \mathbb{R}^2$, is the image domain and $\lambda$ 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 a salt and pepper noise. For such images, the $L^1$ norm is better adapted due to its link to median filtering. It has been used by (Alliney, 1997) and by (Nikolova, 2004; Fu et al., 2006; Bar et al., 2005; Chan et al., 2004; Chan et al., 2005; Cai et al., 2008; Cai et al., 2009) for efficient image denoising algorithms.

In this paper, we choose to investigate the relevance of the $L^1$ norm for salt and pepper noise removal through the minimization of the following functional where the regularization term is a weighted total variation:

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

where $g : \Omega \rightarrow \mathbb{R}^+$ is a function independent of $u$. Such a criterion has been first investigated in (Bresson et al., 2007) for shape denoising. The function $g$ was chosen as an edge indicator function of the input image (e.g., $g(x) = 1/(1 + |\nabla f|)$), which allows a better preservation of corners and sharp angles for shape denoising in images corrupted by a Gaussian noise. In order to use such a criterion for salt and pepper noise removal, we have to consider two main issues: the minimization scheme and the choice of an appropriate function $g$.

Concerning the first issue, let us remind that the minimization of the functional (1.2) is not trivial due...
its 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 (Bar et al., 2005; Nikolova et al., 2006; Bresson et al., 2007). 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 (Bresson et al., 2007). In (Chambolle, 2005), the MRF (Markov Random Field) model is based on 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 (Darbon and Sigelle, 2006a; Darbon and Sigelle, 2006b) where the authors proposed an efficient graph-cut method. In all the approaches mentioned above, an approximation or a smoothing of the $L^1$ norm is required. Recently, in (Bresson et al., 2007), following the works of (Chan et al., 1999; Chambolle, 2004; Aujol and Chambolle, 2005) and more particularly (Aujol et al., 2006), 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 rather take benefit of a convex regularization of the criterion which was first proposed by (Aujol et al., 2006).

Following this very interesting work, we propose a new numerical scheme for the minimization of (1.2) using dual methods. From the criterion (1.2), an augmented Lagrangian formulation (Fortin and Glowinski, 1983) with a penalty term is introduced and solved using the block relaxation method of Uzawa. Our algorithm (named UBR) presents the advantage to be more robust to the choice of the penalty parameter than the algorithm proposed by (Bresson et al., 2007). This parameter can then be chosen so as to decrease the number of iterations and consequently the computational cost.

The second contribution of this paper lies in the proposition of a novel algorithm for salt and pepper noise removal. Taking benefit of the weighted total variation term $TVg$, we propose to study the influence of well-chosen functions $g$ in order to improve the denoising results. An efficient algorithm, denoted UBR-EDGE, is finally proposed for salt and pepper noise removal. Thanks to the nice properties of UBR applied to the weighted TV, our algorithm is able to handle high noise levels at a low computational cost. Experimental results are provided to attest the availability of our 3-steps algorithm.

The paper is organized as follows. In Section 2, we present the $TV_g + L^1$ model and the Uzawa block relaxation method. The role of the weighted TV for salt and pepper removal and the algorithm UBR-EDGE are presented in section 3 and illustrated with some experimental results.

2 EFFICIENT MINIMIZATION OF $TV_g + L^1$-NORM

Let $\Omega$ be a two-dimensional bounded open domain of $\mathbb{R}^d$ with Lipschitz boundary. We consider the following convex energy functional defined, for any $f \in L^1(\Omega)$, any $g : \Omega \rightarrow \mathbb{R}^+$ and any positive parameter $\lambda$: 

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

(2.1)

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

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

(2.2)

where $BV(\Omega)$ is the subspace of functions $u \in L^1(\Omega)$ of bounded variations.

2.1 An Augmented Lagrangian Method

In order to approximate (2.1) by an augmented Lagrangian and to present our dual method of resolution, we need to transform the convex minimization problem 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

$$E(u, p) = \int_{\Omega} g(x)|\nabla u(x)| \, dx + \lambda \int_{\Omega} |p(x)| \, dx$$

(2.3)

The unconstrained minimization problem becomes

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

(2.4)

where $K = \{(u, p) \in X \times X \mid u + p - f = 0 \text{ in } X\}$, with the Euclidean space $X = \mathbb{R}^{N \times K}$ equipped with the $L^2$ scalar product $(u, v)$. To problem (2.4), we associate the augmented Lagrangian functional (see (Koko and Jehan-Besson, 2009) for details) defined by:

$$\mathcal{L}_\lambda(u, p; s) = E(u, p) + (s, u + p - f) + \frac{r}{2} \|u + p - f\|^2,$$

(2.5)

where $r > 0$ is the penalty parameter and $s$ the Lagrange multiplier. This minimization problem can be solved using Uzawa block relaxation methods which
have been used in nonlinear mechanics for operator
splitting and domain decomposition methods (Fortin
and Glowinski, 1983; Glowinski and Tallec, 1989; Koko, 2008). Applying the block relaxation method
to the problem defined above, we obtain the following
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.** Find \( u^k \in X \) such that

\[
\mathcal{L}_r(u^k, p^{k-1}; s^k) \leq \mathcal{L}_r(v, p^{k-1}; s^k), \quad \forall v \in X.
\]

(2.6)

**Step 2.** Find \( p^k \in X \) such that

\[
\mathcal{L}_r(u^k, p^k; s^k) \leq \mathcal{L}_r(u^k, q; s^k), \quad \forall q \in X.
\]

(2.7)

**Step 3.** Update the Lagrange multiplier

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

The algorithm UBR corresponds to the generic
block relaxation algorithm ALG2 (see, e.g., (Fortin
and Glowinski, 1983; Glowinski and Tallec, 1989)).
Let us now detail the explicit solutions of the different
steps (proofs are given in (Koko and Jehan-Besson,
2009)).

**Proposition 2.1** The solution of Step 1 can be given by:

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

where \( v^* \) is the solution of:

\[
-\nabla(\nabla \cdot v^*) + \frac{1}{8}(|\nabla(\nabla \cdot v^*) - p^{k-1}|)v^* = 0.
\]

(2.8)

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

For solving (2.8), we can use the fixed-point proce-
dure of Chambolle (Chambolle, 2004), \( v^0 = 0 \) and for
yany \( \ell \geq 0 \)

\[
v^{\ell+1} = v^\ell + \tau(\nabla \cdot v^\ell - p^{\ell-1})
\]

(2.9)

\[
= \frac{v^\ell + \tau(\nabla \cdot v^\ell - p^{\ell-1})}{1 + (\tau/\|\nabla(\nabla \cdot v^\ell - p^{\ell-1})\|)}.
\]

where \( \tau > 0 \).

The solution of Step 2 is detailed in (Koko and Jehan-
Besson, 2009) and reminded below in the whole
description of the 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 \( \tilde{p}^{k-1} = \tilde{s}^k + r(u^k - f) \) and compute

\[
v^k \text{ with (2.9)}.
\]

Compute \( u^k \)

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

**Step 2.** Compute \( p^k \)

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

**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”. The convergence of the
algorithm UBR is checked using the following con-
vergence criterion:

\[
\sqrt{||u^k - u^{k-1}||^2 + ||p^k - p^{k-1}||^2} \leq \epsilon_u.
\]

The discrete divergence and gradient operators are
given in (Chambolle, 2004).

Note that, each iteration of Algorithm UBR requires
the convergence of the Chambolle fixed point
procedure (2.9). The convergence of this loop is
checked using a threshold on the normalized \( L^2 \) error on \( v^k \).

### 2.2 Applicability and Robustness

We first test the availability of our UBR algorithm for
salt and pepper noise removal taking classically \( g = 1 \)
which corresponds to the minimization of \( TV + L^1 \).
The experimental results provided in Figure 1 demon-
strate that noise is correctly removed. Moreover,
the noisy part is captured through the auxiliary unknown \( v \) as displayed in Figure 1.c. With the function
\( g(x) = 1 \) and \( \lambda = 1.5 \), we find a PSNR of 32.5 dB for the
denoising of a noise of 10\%. The parameter \( \lambda \) is
a classical smoothing parameter. Choosing a smaller
value leads to a higher blurring of image components.
The influence of this parameter is less sensitive when
using the \( TV \) regularization term as demonstrated in the
next section.

In a second step, we want to study the robustness of
the result against the choice of the parameter \( r \). Our
experimental results tend to prove that the algorithm
UBR provides the same denoised images for different
values of \( r \). This is demonstrated by the Figure 2 that
displays the evolution of the PSNR according to the
number of iterations for different parameters \( r \) (from
Moreover, the noisy part is provided in Figure 1, which demonstrates that noise is removed. We also report the number of iterations according to the parameter $r$ with $r = 10$ for the image “peppers” with a salt and pepper noise of 10%. Such a feature then represents an improvement of the method proposed in (Bresson et al., 2007) since the convergence can be obtained without increasing $r$ to infinity. We also report the number of iterations according to $r$ (Figure 3). In this case, the optimal value in terms of iterations is obtained for $r = 30$ with 60 iterations when $\lambda = 1.5$, and for $r = 20$, the convergence can be obtained 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.

3 SALT AND PEPPER NOISE REMOVAL

In this section, we first propose to take benefit of the weighted TV regularization term and of a dedicated function $g$ in order to increase the quality of the denoising results. Our algorithm UBR is then embedded in a more complete process specified for salt and pepper noise removal and named UBR-EDGE.

3.1 The Role of the Weighted TV

A first improvement of the denoising results can be obtained using the fact that the dynamic range of the noise is known. Indeed, corrupted pixels take the values $\text{min}$ or $\text{max}$ that correspond respectively to the minimum and maximum values of intensity. In order to embed this information in the function $g$, we introduce the following mask function:

$$m(x) = \begin{cases} \alpha_0 & \text{if } f(x) = \text{min or max} \\ \alpha & \text{elsewhere} \end{cases}$$

(3.1)

We choose $\alpha_0 = 1.5$ and $\alpha = 0.5$ in order to upmost 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 $m$ ($G$ is a Gaussian of 0-mean and variance $\sigma = 0.5$).

Figure 4 displays the resulting images and the corresponding values of PSNR 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 $\lambda$. For each parameter, we observe a significant increase of 2 to 4dB in the final PSNR. The best value of PSNR is 34.9 dB obtained for $\lambda = 1.5$. The scale effect of the parameter $\lambda$ 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.

Moreover, these first results are obtained at a low computational cost (from 1.6 seconds for a noise of 10% to 4.3 seconds for a noise of 70% on the image Peppers (256x256) with a computer of 3GHz and 2Gb of RAM). This confirms the efficiency of our numerical scheme UBR and attests its availability for the design of our 3-steps salt and pepper noise removal algorithm detailed thereafter.
The use of the weighted TV provides a significant increase in the quality of the final results. However, even if the algorithm $TV_x + L^1$ 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 $\lambda$ and so increase the smoothing of the whole image. In order to circumvent such a problem, we propose both a pre and post-processing to UBR. As a first step (pre-processing), we propose to decrease the size of uncorrupted pixels. This first pass are estimated by computing a mean on pixels that are still corrupted after the median filter (of half-size $1$). The pixels that are still corrupted after this first pass are estimated by computing a mean on known values (corrupted pixels) using a median filter processing), we propose to decrease the size of uncorrupted pixels as well.

### 3.2 The 3-steps Algorithm UBR-EDGE

The use of the weighted TV provides a significant increase in the quality of the final results. However, even if the algorithm $TV_x + L^1$ 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 $\lambda$ and so increase the smoothing of the whole image. In order to circumvent such a problem, we propose both a pre and post-processing to UBR. As a first step (pre-processing), we propose to decrease the size of uncorrupted pixels. This first pass are estimated by computing a mean on pixels that are still corrupted after the median filter (of half-size $1$). The pixels that are still corrupted after this first pass are estimated by computing a mean on known values (corrupted pixels) using a median filter processing), we propose to decrease the size of uncorrupted pixels as well.

#### Algorithm UBR-EDGE

**Step 1.** Pre-processing

Let us remark that the first functional $f_1$ only acts as an initial condition of the algorithm UBR in order to give a first rough estimate for the corrupted pixels. The last edge smoother is applied only on the corrupted pixels.

**Step 2.** Algorithm UBR

Let us remark that the first functional $f_2$ only acts as an initial condition of the algorithm UBR in order to give a first rough estimate for the corrupted pixels. The last edge smoother is applied only on the corrupted pixels as well.

In Figure 5, final results of the different steps of the process are given for the restoration of the image “Lena” corrupted by a salt and pepper noise of $70\%$. The Figure 5(b) 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 5(c). The EDGE smoother EDDI is then applied which gives the final image of Figure 5(d).

### 3.3 Experimental Results

Some visual results are provided in Figure 6 for “Lena” (512x512) and in Figure 7 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 8, we can conclude that our algorithm provides good visual results.

### Table 8

<table>
<thead>
<tr>
<th>Noise Level</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>34</td>
</tr>
<tr>
<td>5%</td>
<td>32</td>
</tr>
<tr>
<td>10%</td>
<td>30</td>
</tr>
<tr>
<td>20%</td>
<td>28</td>
</tr>
<tr>
<td>30%</td>
<td>26</td>
</tr>
<tr>
<td>40%</td>
<td>24</td>
</tr>
</tbody>
</table>

In Figure 6, some visual results are provided in Figure 6 for “Lena” (512x512) and in Figure 7 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 8, we can conclude that our algorithm provides good visual results.

### Figure 4

(a) $\lambda = 1, g = 1$ PSNR = 30.3 dB

(b) $\lambda = 1.5, g = 1$ PSNR = 32.5 dB

(c) $\lambda = 1, g = m_\sigma$ PSNR = 34.3 dB

(d) $\lambda = 1.5, g = m_\sigma$ PSNR = 34.9 dB
Figure 5: 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 (b). This image is used as an input of the algorithm UBR and the result is given in (c). A last post-processing is applied to the image which yields to the final result given in (d).

at a low computational cost. The PSNR values obtained for the image “Lena” can be compared with the PSNR values reported in (Chan et al., 2004; Chan et al., 2005) 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. For completeness, we report the values given by (Chan et al., 2005) for the denoising of Lena (512x512) with a noise of 70%. With the classical Median filter, the PSNR is 23.2 dB and with an improved switching median (ISM) filter, the PSNR is 23.4 dB. Using the algorithm III proposed in (Chan et al., 2005), the PSNR is 29.3 dB. We find a PSNR of 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 run our algorithm on the input noisy images provided in the web page of R. Chan 1. Experimental results reported in (Koko and Jehan-Besson, 2009) show that our algorithm gives good quality results with a PSNR value that is a little smaller than the one found by the algorithm III (Chan et al., 2004) with a difference of less than 1 dB. More precisely, for the denoising of the first image in Figure 6.a (noise of 70%), they find a PSNR of 23.07 dB while our PSNR is 22.2 dB. For the image 6.b, they find a PSNR of 34.16 dB while our is 33.3 dB. For the image 6.c, they find a PSNR of 26.78 dB while our is 26.0 dB. So their algorithm gives better PSNR for these images but 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 8).

4 CONCLUSIONS

In this paper, our contribution is twofold. First, we propose a new efficient and robust minimization scheme for the minimization of a $TV_g + L^1$ criterion using Uzawa Block Relaxation (UBR) method. We more particularly study the robustness of the algorithm against the penalty parameter $r$. Secondly, we investigate the role of the weighted TV to improve salt and pepper noise removal and we embed our algorithm in an efficient 3-steps process dedicated to high noise levels. Our algorithm gives comparable

---

1http://www.math.cuhk.edu.hk/ rchan/paper/impulse/
Algorithm UBR-EDGE

<table>
<thead>
<tr>
<th>Noise</th>
<th>Lena (512x512)</th>
<th>Peppers (256x256)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>time(s)</td>
</tr>
<tr>
<td>10</td>
<td>43.4</td>
<td>2.7</td>
</tr>
<tr>
<td>20</td>
<td>39.7</td>
<td>3.9</td>
</tr>
<tr>
<td>30</td>
<td>37.1</td>
<td>5.3</td>
</tr>
<tr>
<td>40</td>
<td>35.3</td>
<td>6.6</td>
</tr>
<tr>
<td>50</td>
<td>33.9</td>
<td>8.1</td>
</tr>
<tr>
<td>70</td>
<td>31.4</td>
<td>17.1</td>
</tr>
<tr>
<td>90</td>
<td>26.6</td>
<td>41.4</td>
</tr>
</tbody>
</table>

Figure 7: Salt and pepper noise removal using the algorithm UBR-EDGE for “Peppers”. For the result obtained in (b), \( \lambda = 2 \) and for the result in (d), \( \lambda = 1.5 \).

Figure 8: 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%, \( \lambda = 0.7 \).

ACKNOWLEDGEMENTS

The numerical experiments were run in C++ with the library Pandore\(^2\). The salt and pepper noise was generated with gmic\(^3\).

REFERENCES

Alliney, S. (1997). A property of the minimum vectors of a regularizing functional defined by means of the abso-

\(^2\)available at http://www.greyc.ensicaen.fr/regis/Pandore/
\(^3\)http://gmic.sourceforge.net/


