ECOLE POLYTECHNIQUE

CENTRE DE MATHÉMATIQUES APPLIQUÉES

UMR CNRS 7641

91128 PALAISEAU CEDEX (FRANCE). Tél: 01 69 33 41 50. Fax: 01 69 33 30 11
http://www.cmap.polytechnique.fr/

Total variation minimization and a class of binary MRF models

A. Chambolle

R.I. 578 *June,2005*

Total variation minimization and a class of binary MRF models.

Antonin Chambolle*

June 20, 2005

Abstract

We observe that there is a strong connection between a whole class of simple binary MRF models and the Rudin-Osher-Fatemi Total Variation minimization approach to image denoising. This leads to new algorithms. We then compare the efficiency of various algorithms.

1 Introduction

In this paper, we would like to study the relationship between the Rudin-Osher-Fatemi Total Variation (TV) minimization model for image denoising, and a class of simple binary MRF models. In particular, we will show that some algorithms designed to solve one type of problem can be adapted to the other. Our goal is to discuss the links between problems such as

$$\min_{\theta_{i,j}\in\{0,1\}} \lambda \sum_{i,j} |\theta_{i+1,j}-\theta_{i,j}|+|\theta_{i,j+1}-\theta_{i,j}| + \frac{1}{2}\sum_{i,j} \theta_{i,j}|g_{i,j}-a|^2 + (1-\theta_{i,j})|g_{i,j}-b|^2\,, \tag{1}$$

and

$$\min_{w_{i,j}\in\mathbb{R}} \lambda \sum_{i,j} |w_{i+1,j}-w_{i,j}| + |w_{i,j+1}-w_{i,j}| \;+\; \frac{1}{2}\sum_{i,j} |g_{i,j}-w_{i,j}|^2\,. \tag{2}$$

Here, i, j index the rows and columns of a digital image and run, for instance, from 1 to N and 1 to M, $N, M \geq 1$. Problems such as (1) arise in simplified MRF image denoising models where one assumes, for instance, that an observation g results from an original binary signal taking only values a and b, to which a Gaussian noise is added. (Here, g, a, b could be vector valued.) But, in fact, very similar problems can be used efficiently in a much more elaborate way: for instance, in [27] the authors build a tree of binary MRFs to classify images in much more than two labels (see also [23, 24]).

It is known that (1) can be solved exactly using linear programming, and more exactly by finding a minimal cut in a graph, using a max-flow algorithm. This has been first observed by Greig, Porteous and Seheult [17], and these techniques have

*CMAP (CNRS UMR 7641), Ecole Polytechnique, 91128 Palaiseau Cedex, France. antonin.chambolle@polytechnique.fr

been then extended to much more general problems in the recent years [18, 7, 20, 21, 22, 25].

On the other hand, problem (2) has been first proposed in image processing by Rudin, Osher and Fatemi [26], as an efficient approach to edge-preserving image denoising or reconstruction. We have proposed recently [11] an algorithm for solving the "isotropic" version of this problem (where, in the sum, each term $|w_{i+1,j} - w_{i,j}| + |w_{i,j+1} - w_{i,j}|$ is replaced with the standard Euclidean norm of the discrete gradient $\left((w_{i+1,j} - w_{i,j})^2 + (w_{i,j+1} - w_{i,j})^2\right)^{1/2}$), however, our algorithm is quite general and easily adapted to (2) (see Section 4 below). Our point is that problems (1) and (2) are easily derived one from the other, so that algorithms designed to solve one can be used to solve the other. We would like to discuss the consequences of these links and compare the algorithms.

This note is organized as follows. In the next section we describe an abstract framework in which both (1) and (2) enter as a particular case. We show that the solutions of these problems are closely related, in particular, we can deduce a generic uniqueness for solutions of (1). In Section 3 we discuss various implementations of graph cuts algorithms that can be derived to solve (2). We then recall, in Section 4, the algorithm proposed in [11]. Numerical experiments are then performed to compare these various algorithms.

While we were completing a first version of this note, J. F. Aujol mentioned to us the recent work of Jérôme Darbon and Marc Sigelle [14, 15], which may be seen as the probabilistic counterpart of the present work. They show essentially the same results (including, in particular, Prop. 2.2), with very different proofs. Although we may claim our proofs are probably simpler, their results are equivalent and the algorithm they derive is essentially the same as the dyadic algorithm we present in section 3.3.

Acknowledgements

The author wishes to thank Guy Bouchitté (Univ. of Toulon, France) and Vicent Caselles (Univ. Pompeu Fabra, Barcelona, Spain): this research owes a lot to their collaboration on *apparently* quite different topics. Also, though the description of the max-flow algorithm in [6] is extremely clear, this research was considerably speeded up thanks to a few C++ implementations of this algorithm that are available on the internet. We would like to thank the authors of these implementations. We tried two of these, one by Walter Bell (from Cornell University), available through his web page [4] and the other by Vladimir Kolmogorov (Cornell), available at `http://www.cs.cornell.edu/People/vnk/software.html`. We found the latter somewhat more efficient, with the additional advantage that the data type of the capacities can be set to `float` or `double`. Eventually, we would like also to thank J. F. Aujol for mentioning to us the work of J. Darbon and M. Sigelle, and the two latter for interesting and helpful discussion.

The author is supported by the CNRS, and partially supported by the "Fonds national de la science", ACI "Nouvelles interfaces des mathématiques" MULTIM.

2 The abstract framework

2.1 A class of regularizing energies

We consider a vector space $X \sim \mathbb{R}^N$ with the Euclidean scalar product $(u, v) = \sum_{i=1}^N u_i v_i$. In practice, an element in X will represent a $2D$ scalar or multichannel image, but other situations could be encountered. Let us also mention that all we will write is still valid in the infinite dimensional case, if $X = L^2(\Omega)$ for Ω a bounded, open set in $\mathbb{R}^d$, $d \geq 1$ (with some adaption in the proofs and notation). The first part of the energies that appear in problems (1) and (2) is a particular case (as we will check in Section 3) of a function $J : X \to [0, +\infty]$ which is convex (i.e., $J(tu + (1-t)v) \leq tJ(u) + (1-t)J(v)$ for any $t \in [0, 1]$, $u, v \in X$), lower semicontinuous, positively one-homogeneous (i.e., $J(tu) = tJ(u)$ for any $t \geq 0$ and $u \in X$), and that satisfies the generalized *co-area formula*:

$$J(u) \;=\; \int_{-\infty}^{+\infty} J(u^t)\, dt \tag{3}$$

where for any $i = 1, \ldots, N$,

$$u_i^t \;= \begin{cases} 1 & \text{if } u_i > t\,, \\ 0 & \text{otherwise,} \end{cases}$$

that is, $u^t = \chi_{\{u>t\}}$, the characteristic function (in {0,1,...,N}) of the superlevel s of $u = (u_i)_{i=1}^N$. Let us mention that all what will be said here is still valid if "$>$" is replaced with "$\geq$" and, whenever $J(-u) = J(u)$ and up to a change of sign in some formulas, with "$<$" or "$\leq$". Observe also that the one-homogeneity of J follows in fact from (3). Moreover, $J(u) = 0$ if $u_i = u_j$ for all i, j (otherwise the integral in (3) is always infinite).

2.2 Abstract binary MRFs

We will check later on that problem (1) can be restated in the following abstract form

$$\min_{\theta \in X, \theta_i \in \{0,1\}} \lambda J(\theta) \;+\; \sum_{i:\theta_i = 1} s - G_i \tag{P_s}$$

where $G \in X$ would be a vector depending on g, a, b and $s \in \mathbb{R}$ a level depending on a, b.

A first observation, which is quite obvious, is the following:

Proposition 2.1 *Any solution θ of (P_s) is also a solution of*

$$\min_{v \in X, v_i \in [0,1]} \lambda J(v) \;+\; \sum_{i=1}^N (s - G_i) v_i\,. \tag{P_s'}$$

Conversely, if v is a solution of (P_s'), then for any $t \in (0, 1)$ v^t is a solution of (P_s)

Proof. Without loss of generality we assume $s = 0$ and denote by (P), (P') the problems (P_0) and respectively (P_0'). It is enough to observe that if $v_i \in [0, 1]$ for

any i, we have $v_i = \int_0^{v_i} dt = \int_0^1 v_i^t\, dt$. We deduce

$$\lambda J(v) \;-\; \sum_{i=1}^N G_i v_i \;=\; \int_0^1 \left(\lambda J(v^t) \;-\; \sum_{i:v_i^t=1} G_i \right) dt\,.$$

We easily deduce that the minimal value m of both problems (P) and (P') is the same, and, clearly, that a solution θ of (P) is also a solution of (P'). Let v solve (P'). One has

$$\int_0^1 \left(\lambda J(v^t) \;-\; \sum_{i:v_i^t=1} G_i \right) \;-\; m\, dt \;=\; 0$$

hence for a.e. $t \in [0,1]$, $\lambda J(v^t) \;-\; \sum_{i:v_i^t=1} G_i = m$. By the lower semicontinuity of J, for any $t < 1$ this remains true. □

This property shows that the minimization of the binary problem (P_s) is in fact a convex minimization problem.

2.3 Comparison for binary MRFs

Let us now observe the following comparison property, which does not seem to be well-known. It is already mentioned in J. Darbon and M. Sigelle's recent papers, where it is proved using a probabilistic approach. Our proof, which we claim is quite simpler, is the finite-dimensional counterpart of the proof we proposed in [2, 3, 8].

Proposition 2.2 *Assume $G^2 > G^1$, i.e., for any $i = 1, \ldots, N$, $G_i^2 > G_i^1$. For $\alpha = 1, 2$, let v^α be solutions of (P') with G replaced with G^α. Then $v^2 \geq v^1$.*

Proof. The proof relies on the following Lemma (mentioned to us first by Bouchitté [5])

Lemma 2.3 *Let $v, w \in X$. Then $J(v \wedge w) + J(v \vee w) \leq J(v) + J(w)$.*

Here $(v \wedge w)_i = \min\{v_i, w_i\}$ and $(v \vee w)_i = \max\{v_i, w_i\}$, for any $i = 1, \ldots, N$.
Proof. By (3), we see that it is enough to prove the inequality when v, w are characteristic functions, that is, when $v_i, w_i \in \{0, 1\}$ for any $i = 1, \ldots, N$. In this case, one has for any i

$$v_i + w_i \;=\; v_i \vee w_i + v_i \wedge w_i \;=\; \begin{cases} 0 & \text{if } w_i = v_i = 0\,, \\ 1 & \text{if } w_i = 1, v_i = 0 \text{ or } w_i = 0, v_i = 1\,, \\ 2 & \text{if } w_i = v_i = 1\,. \end{cases}$$

Hence, we see that $\{v + w > 0\} = \{v \vee w = 1\}$ while $\{v + w = 2\} = \{v \wedge w = 1\}$, so that by (3),

$$\begin{aligned} J(v + w) \;&=\; \int_0^2 J((v + w)^s)\, ds \\ &=\; \int_0^1 J(v \vee w)\, ds \;+\; \int_1^2 J(v \wedge w)\, ds \;=\; J(v \vee w) + J(v \wedge w)\,. \end{aligned}$$

Since $J(v + w) = 2J((v + w)/2) \leq 2(J(v)/2 + J(w)/2) = J(v) + J(w)$, Lemma 2.3 is true. □

Let us mention that, in fact, the property in Lemma 2.3 is equivalent to the generalized coarea formula (3) (assuming J is a convex, l.s.c., one-homogeneous function with $J(c+\cdot) = J(\cdot)$ for any constant $c \in \mathbb{R}$). Functions that satisfy the thesis of Lemma 2.3 appear in optimization theory as "sub-modular" functions, and it is observed in [21] that they are the only two-point interactions functions of binary variables that can be minimized using graph-cut algorithms.

We now return to the proof of Proposition 2.2. We follow the proof of a similar result in [2, 3, 8]. We observe that

$$\lambda J(v^1) - \sum_{i=1}^N G_i^1 v_i^1 \;\leq\; \lambda J(v^1 \wedge v^2) - \sum_{i=1}^N G_i^1 (v_i^1 \wedge v_i^2)\,,$$

$$\lambda J(v^2) - \sum_{i=1}^N G_i^2 v_i^2 \;\leq\; \lambda J(v^1 \vee v^2) - \sum_{i=1}^N G_i^2 (v_i^1 \vee v_i^2)\,.$$

Summing both inequalities and using Lemma 2.3, we find that

$$\sum_{i=1}^N G_i^2 (v_i^1 \vee v_i^2 - v_i^2) \;\leq\; \sum_{i=1}^N G_i^1 (v_i^1 - v_i^1 \wedge v_i^2)\,.$$

Now, for any i, $v_i^1 \vee v_i^2 - v_i^1 = v_i^1 - v_i^1 \vee v_i^2 = (v_i^1 - v_i^2)^+ = v_i^1 - v_i^2$ if $v_i^1 > v_i^2$ and 0 otherwise. Hence

$$\sum_{i=1}^N (G_i^2 - G_i^1)(v_i^1 - v_i^2)^+ \;\leq\; 0\,.$$

We find that if $G^2 \geq G^1$, then for any i such that $G_i^2 > G_i^1$, one must have $v_i^1 \leq v_i^2$. In particular, this yields Proposition 2.2. □

Remark 2.4 If $v_i^\alpha \in \{0,1\}$ for $\alpha = 1,2$ and any i, that is, if v^α solves (P) (with G replaced with G^α), we find that when $G^2 > G^1$, then $\{v^1 = 1\} \subseteq \{v^2 = 1\}$. In fact, in general, one has $\{v^1 > 0\} \subseteq \{v^2 = 1\}$.

Remark 2.5 If $G^1 = G^2 = G$ and v^1, v^2 are two different solutions, the same proof will show that necessarily, $J(v^1) + J(v^2) = J(v^1 \wedge v^2) + J(v^1 \vee v^2)$ and both $v^1 \wedge v^2$, $v^1 \vee v^2$ are also solutions of (P').

Remark 2.6 If $G^2 \geq G^1$, by approximating G^2 by $G^2 + \epsilon$, $\epsilon > 0$ and letting $\epsilon \to 0$, one shows that there exist *a* solution v^2 of (P') with G replaced with G^2 such that $v^2 \geq v^1$ (for any choice of v^1) – or, conversely, there exists a v^1 with $v^2 \geq v^1$ for any choice of v^2.

2.4 The abstract ROF model

Let us now introduce the following minimization problem, which is the abstract version of (2):

$$\min_{w \in X} J(w) \;+\; \frac{1}{2\lambda} \|G - w\|^2\,. \tag{4}$$

Our main result is the following.

Proposition 2.7 *Let w solve (4). Then, for any $s \in \mathbb{R}$, both*

$$w_i^s = \begin{cases} 1 & \text{if } w_i > s\,, \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \overline{\overline{w}}_i^s = \begin{cases} 1 & \text{if } w_i \geq s\,, \\ 0 & \text{otherwise} \end{cases}$$

solve (P_s). Conversely, if θ solves (P_s) (resp, v solves (P'_s)), then $w^s \leq \theta \leq \overline{\overline{w}}_i^s$ (resp, $w^s \leq v \leq \overline{\overline{w}}^s$). In particular, if $w^s = \overline{\overline{w}}^s$ (which is true for all but a finite number of levels s), then the solution of (P_s) and (P'_s) is unique.

This means that solutions to the whole family of problems (P_s), $s \in \mathbb{R}$, could be computed by solving just one convex problem (4), and conversely, that (4) can be minimized by solving an appropriate family of binary problems (P_s). We will explain later on how this is done. In the case $J(w)$ is the total variation of $w \in X = L^2(\Omega)$, Ω open subset of $\mathbb{R}^d$, then $J(\theta)$ is the *perimeter* of the set $\{\theta = 1\}$ and this kind of equivalence has been observed many years ago in minimal surfaces problems (see for instance [16]). It has been extended recently to problems such as (1), in [10, 8]. In image processing, the observation that (1) can be solved by finding the appropriate superlevel of the solution of (4) was also mentioned recently in [12, 13].

Proof. The fact that the solutions of (P_s), for $s \in \mathbb{R}$, can be seen as the level sets of a $w \in X$, follows from Proposition 2.2. Indeed, if $s > s'$, one readily sees that any pair of solutions θ, θ' of (P_s) and $(P_{s'})$ will satisfy $\theta' \geq \theta$. One can let for each $i = 1, \ldots, N$

$$w_i \;=\; \sup\{s \in \mathbb{R} \,:\, \exists \theta \text{ solving } (P_s) \text{ with } \theta_i = 1\}\,.$$

If $s - G_i \leq 0$ for all i, then $\theta \equiv 1$ solves (P_s), and if $s - G_i \geq 0$ for all i, then $\theta \equiv 0$ is the solution: one deduces that $\min_j G_j \leq w_i \leq \max_j G_j$ for all i. By the comparison principle, we see that for any $s < w_i$, $\theta_i = 1$ for any solution of (P_s), while if $s > w_i$, it must be that $\theta_i = 0$ for any solution. Hence if $s \notin \{w_i \,:\, i = 1, \ldots, N\}$, w^s is a solution of (P_s). Moreover, it is the unique solution, because of the comparison principle and because if s' is close enough to s, $w^s = w^{s'}$. On the other hand, if $s \in \{w_i \,:\, i = 1, \ldots, N\}$, one has $w^s = \lim_{n\to\infty} w^{s+1/n}$ while $\overline{\overline{w}}^s = \lim_{n\to\infty} w^{s-1/n}$, showing that both are (different) solutions of (P_s). Let us now show that w is the solution of (4). Consider $v \in X$, and let $s_* \leq \min_i v_i$. One has

$$\int_{s_*}^{+\infty} (s - G_i) v_i^s \, ds \;=\; \int_{s_*}^{v_i} s - G_i \, ds \;=\; \frac{1}{2}\left((v_i - G_i)^2 - (s_* - G_i)^2\right)\,,$$

hence (since $J(v^s) = 0$ for any $s < s_*$)

$$\frac{1}{\lambda}\int_{s_*}^{+\infty} \left(\lambda J(v^s) + \sum_{i : v_i^s = 1} s - G_i\right) ds$$

$$= \; J(v) + \frac{1}{2\lambda}\|G - v\|^2 - \frac{1}{2\lambda}\sum_{i=1}^N (s_* - G_i)^2\,.$$

If, also, $s_* \leq \min_i w_i$, we find (by the minimality of w^s in (P_s) for all s)

$$J(v) + \frac{1}{2\lambda}\|G - v\|^2 \;\geq\; J(w) + \frac{1}{2\lambda}\|G - w\|^2\,,$$

so that w is the (unique) solution of the (strictly convex) problem (4). □

Remark 2.8 We have used, here and earlier, an obvious stability property of problems (P_s), (P'_s), and (4), which results from the lower semicontinuity of J: if G^n, or s^n, go to G, or respectively s, as $n \to \infty$, and if v_n denote solutions of the corresponding approximate problems, then any cluster point v of the sequence $(v_n)_{n\geq 1}$ is a solution of the limit problem.

Remark 2.9 It is important to observe that a consequence of Proposition 2.7 is the generic uniqueness (in some sense) of the solution of the binary problem (P_s).

2.5 Quantized ROF model

We now consider the following quantized variant of (4):

$$\min\left\{ J(z) \ + \ \frac{1}{2\lambda}\|G-z\|^2 \ : \ z\in X\,,\ z_i \in \{l_0\ \ldots, l_n\}\,\forall i=1,\ldots,N \right\} \qquad (5)$$

where $(l_k)_{k=1}^n$ are given real values. That is, we minimize (4) only among functions that take values in a prescribed, finite set. Without loss of generality, we assume that $l_0 < l_1 < \cdots < l_n$, and for simplicity that for all $k=1,\ldots,n$, $l_k - l_{k-1} = \delta l$ (adaption to other cases is straightforward). For a z admissible, we can write

$$z \ = \ l_0 \ + \ \sum_{k=1}^n (l_k - l_{k-1})\theta^k \ = \ l_0 \ + \ \delta l \sum_{k=1}^n \theta^k$$

where for each $k\geq 1$, θ^k is the binary vector defined by $\theta^k_i = 1$ iff $z_i \geq l_k$. Then, the fact $\theta^k \leq \theta^{k-1}$ for any $k\geq 2$, and the co-area formula (3), yield $J(z) = \sum_{k=1}^n \delta l\, J(\theta^k)$. On the other hand,

$$\begin{aligned}
\|G-z\|^2 \ &= \ \sum_{i=1}^N (G_i - z_i)^2 \\
&= \ \sum_{i=1}^N \left((G_i - l_0)^2 \ + \ \sum_{k=1}^n \left((G_i - l_k)^2 - (G_i - l_{k-1})^2 \right) \theta^k \right) \\
&= \ \sum_{i=1}^N (G_i - l_0)^2 \ + \ 2\delta l \sum_{k=1}^n \sum_{i=1}^N \left(\frac{l_k + l_{k-1}}{2} - G_i \right) \theta^k_i \,,
\end{aligned}$$

hence, up to a constant, problem (5) is the same as

$$\min_{\theta^k} \sum_{k=1}^n \left(\lambda\, J(\theta^k) \ + \ \sum_{i=1}^N \left(\frac{l_k + l_{k-1}}{2} - G_i \right) \theta^k_i \right) \,,$$

where the min is on the $(\theta^k)_{k=1}^n$, with the constraint that $\theta^k \leq \theta^{k-1}$ for any $k = 2,\ldots,n$. Each term in the sum is the energy that appears in problem (P_{s_k}), for $s_k = (l_k + l_{k-1})/2$. Now, by Lemma 2.2, if for each $k=1,\ldots,n$, θ^k minimizes (P_{s_k}), then since $s_k > s_{k-1}$, $\theta^k \leq \theta^{k-1}$: hence the minimum problem above is in fact unconstrained. Moreover, by Proposition 2.7, a solution of (P_{s_k}) is given by w^s, where w solves (4). We find that a solution z of (5) is given by $z_i = l_0$ if

$w_i \leq (l_1 + l_0)/2$, $z_i = l_k$, $k = 2, \ldots, n-1$, if $(l_k + l_{k-1})/2 < w_i \leq (l_{k+1} + l_k)/2$, and $z_i = l_n$ if $w_i > (l_n + l_{n-1})/2$.

We also have that any solution z of (5) satisfies $|z_i - w_i| = \min_{k=0}^{n} |l_k - w_i|$ for any i: in particular, one has $\max_{i=1}^{N} |z_i - w_i| \leq \delta l/2$ as soon as $l_0 \leq \min_i w_i$ and $l_n \geq \max_i w_i$ (which is certain if l_0 is chosen no larger than $\min_i G_i$, and l_n no less $\max_i G_i$). In conclusion, the quantized problem (5) produces exactly a quantization of the solution of (4).

3 Algorithms

The interaction energy appearing in (1), (2) is of the form

$$J(w) \;=\; \sum_{1 \leq i < j \leq N} \alpha_{i,j} |w_i - w_j| \,.$$

The weights $\alpha_{i,j}$ are always assumed to be nonnegative. We also introduce $\alpha_{j,i} = \alpha_{i,j}$, notice however that all the discussion that follows is still valid for the more general form of the energy $\sum_{i \neq j} \alpha_{i,j} (w_i - w_j)^+$, with directional and possibly different interaction weights $\alpha_{i,j}$ and $\alpha_{j,i}$ (we define $x^+ = \max\{x, 0\}$, $x^- = (-x)^+$ for any real number x). We will assume in the rest of the discussion that $\lambda = 1$, without loss of generality. Since for any two real numbers a, b, $|a - b| = \int_{-\infty}^{+\infty} |\chi_{a>s} - \chi_{b>s}| \, ds$, J clearly satisfies (3). (The same observation appears in a recent paper by B. A. Zalesky [28].)

The consequences of the previous discussion are, on one hand, that it is possible to solve a TV minimisation problem such as (4) by solving either binary MRF problems of type (P_s) for each level s (or rather for s in a finite, reasonably large set of levels $\{l_0, \ldots, l_n\}$), or by solving directly a quantized problem of type (5). All these can be solved by graph-flow techniques. Conversely, it is possible to find a solution of a binary problem such as (1) (or (P_s)) by solving an appropriate TV minimization problem, and thresholding the result. We will not discuss this alternative in the present paper, although it might be interesting for finding solutions to the whole family of problems (P_s) in one single pass. Let us first describe the "graph-cut" techniques for solving binary MRFs.

3.1 Algorithms for binary MRFs

It has been observed first by Greig, Porteous and Seheult that a problem such as (1) or (P_s) is equivalent to finding a partition of an appropriate graph into two sets. We consider the problem written in the form (P) (remember (P) denotes problem (P_0)). Consider the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ made of vertices

$$\mathcal{V} \;=\; \{i \,:\, i = 1, \ldots, N\} \cup \{t\} \cup \{s\}$$

where the "terminals" s and t are called respectively the source and the sink, and of (oriented) edges

$$\begin{aligned} \mathcal{E} \;=\; & \{(i,j) \,:\, 1 \leq i, j \leq N \, i \neq j \,, \alpha_{i,j} > 0\} \\ & \cup \{(s,i) \,:\, 1 \leq i \leq N\} \cup \{(i,t) \,:\, 1 \leq i \leq N\} \,. \end{aligned}$$

The first two sets of edges represent the interactions between values, necessary to represent the potential J. The last set, that links each value to both terminals, is used to represent the potential $-\sum_i G_i\theta_i$ that appears in Problem (P). Now, assume each edge $e \in \mathcal{E}$ has a "capacity" $C(e)$. (For technical reasons, these capacities need be nonnegative.) Then, given a "cut" $(\mathcal{V}_s, \mathcal{V}_t)$ of the graph, that is, a partition of $\mathcal{V}$ into two sets, one containing s and the other containing t, one can define the energy of the cut by

$$E(\mathcal{V}_s, \mathcal{V}_t) \;=\; \sum_{\substack{e=(\alpha,\beta)\in\mathcal{E} \\ \alpha\in\mathcal{V}_s, \beta\in\mathcal{V}_t}} C(e)$$

As shown in [17], there is a way to associate capacities to the graph $\mathcal{G}$ so that if we let $\theta_i = 1$ if $i \in \mathcal{V}_s$ and $\theta_i = 0$ otherwise, then

$$E(\mathcal{V}_s, \mathcal{V}_t) \;=\; J(\theta) - \sum_{i=1}^{N} G_i\theta_i \;+\; \text{constant} \tag{6}$$

for any cut $(\mathcal{V}_s, \mathcal{V}_t)$. Let us describe how these capacities are assigned. To an edge $e = (i, j) \in \mathcal{E}$, we simply let $C(e) = \alpha_{i,j}$. Then, choosing $\overline{G} \geq \max_i G_i$, we let $C(s, i) = \overline{G}$ and $C(i, t) = \overline{G} - G_i$. It is then straightforward to check (6).

Now, it is possible to find, in polynomial time, an optimal cut (that is, a cut minimizing its total energy E) in such a graph, giving a solution to our binary MRF model. The idea is to find a "maximal flow" along the edges of the graph, from s to t. The equivalence between both problems is a duality result, due to Ford and Fulkerson. We refer to [6] for a very clear description of the method, and of an algorithm.

3.2 Minimization of (4) using graph cuts

According to the discussion in section 2, one deduces essentially two methods for minimizing (4) using the max flow algorithm for computing graph cuts.

3.3 First method (Darbon and Sigelle's algorithm)

The first method consists simply in fixing $n + 1$ levels $l_0, \ldots, l_n$, with $l_0 = \min_i G_i$ and $l_k = \max_i G_i$, and $l_k - l_{k-1} = (l_n - l_0)/n = \delta l$, and to find a solution z of the quantized problem (5). Practically, one solves problem (P_{s_k}) for $s_k = (l_k + l_{k-1})/2$, for each $k = 1, \ldots, n$: the result is a field θ^k with $\theta^k = 1$ when $z > s_k$ and 0 else. One easily rebuilds z from the θ^k's. Now, there is a lot of redundancy in this method. Indeed, since $\theta^k \leq \theta^{k-1}$, once problem $(P_{s_{k-1}})$ is solved one should not need to reprocess the areas where $\theta^{k-1} = 0$ (since there, $\theta^k = 0$ is already known).

This observation yields a more efficient method for solving (4), up to an arbitrary, fixed precision. The algorithm that we propose here has already been presented, in a slightly different way, in two papers by J. Darbon and M. Sigelle [14, 15]. We denote by w the (unique) solution of (4). Given a "depth" $D \geq 1$, we fix a dyadic number of (increasing) thresholding levels s_k, for $k = 1, \ldots, n = 2^D - 1$. We introduce an array (K_i), $i = 1, \ldots, N$, of integers, whose meaning will be, at the end of the process, the following: if $K_i = k$, then $s_k \leq w_i \leq s_{k+1}$ (letting by convention $s_0 = -\infty$ and $s_{2^D} = +\infty$). We initialize this array with the value 0. Then, for $d = 0, \ldots, D - 1$,

we segment at each level s_k for $k = (2p-1)2^{D-1-d}$, $p = 1, \dots, 2^d$. First, for $d = 0$, we segment at level s_k for $k = 2^{D-1}$, by solving problem (P_{s_k}), and we get a map θ such that if $\theta_i = 1$, $w_i \geq s_k$, whereas if $\theta_i = 0$, $w_i \leq s_k$. Hence we let $K_i = k$ when $\theta_i = 1$ and we leave the value 0 when $\theta_i = 0$.

For $d = 1$, let us consider the levels s_k, $k = 2^{D-2}$, and $s_{k'}$, $k' = 3 \times 2^{D-2}$. If θ solve (P_{s_k}), we know that each time $K_i = 2^{D-1}$, then $\theta_i = 1$, in the same way, if θ' solve $(P_{s_{k'}})$, each time $K_i = 0$, then $\theta'_i = 0$. Thus, $(\theta_i)_{i:K_i=0}$ solves the problem

$$\min_{\theta_i \in \{0,1\}} \sum_{i<j, K_i = K_j = 0} \alpha_{i,j} |\theta_i - \theta_j| \; + \sum_{K_i = 0 \neq K_j} \alpha_{i,j}(1-\theta_i) \; + \sum_{K_i = 0} (s_k - G_i)\theta_i \,,$$

while $(\theta'_i)_{i:K_i \neq 0}$ solves

$$\min_{\theta'_i \in \{0,1\}} \sum_{i<j, K_i = K_j \neq 0} \alpha_{i,j} |\theta'_i - \theta'_j| \; + \sum_{K_i \neq 0 = K_j} \alpha_{i,j}\theta'_i \; + \sum_{K_i \neq 0} (s_{k'} - G_i)\theta'_i \,.$$

These two problems can be solved independently, but they can also be merged in the following way: we let $\hat{\theta}_i = \theta_i$ when $K_i = 0$, and $\hat{\theta}_i = \theta'_i$ when $K_i = 2^{D-1}$, then $\hat{\theta}$ solves the problem

$$\min_{\hat{\theta}_i \in \{0,1\}} \sum_{i<j, K_i = K_j} \alpha_{i,j} |\hat{\theta}_i - \hat{\theta}_j| \; + \sum_{K_i = 0 \neq K_j} (\alpha_{i,j}(1-\hat{\theta}_i) + \alpha_{j,i}\hat{\theta}_j) \\ + \sum_{i=1}^{N} (s_{K_i + 2^{D-2}} - G_i)\hat{\theta}_i \,.$$

This problem is easily written on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}')$ where $\mathcal{E}' \subset \mathcal{E}$ contains only the edges (i,j) with $K_i = K_j$: of course, this is fictitious in the sense that $\mathcal{V} \setminus \{s,t\}$ is now completely disconnected, and (at least) two different independent problems are solved. After $\hat{\theta}$ is computed, $(K_i)_{i=1}^N$ is updated as follows: if $K_i = 0$, then we let $K_i = 0$ if $\hat{\theta}_i = 0$ and $K_i = k = 2^{D-2}$ else, and if $K_i = 2^{D-1}$, we let $K_i = 2^{D-1}$ if $\hat{\theta}_i = 0$ and $K_i = k' = 3 \times 2^{D-2}$ else. Hence, K_i is updated according to the following rule:

$$K_i \; \leftarrow \; K_i + 2^{D-2}\hat{\theta}_i \,.$$

Now, the subsequent steps ($d \geq 2$) are processed exactly in the same way. One solves the binary problem

$$\min_{\hat{\theta}_i \in \{0,1\}} \sum_{i<j, K_i = K_j} \alpha_{i,j} |\hat{\theta}_i - \hat{\theta}_j| \; + \sum_{K_i < K_j} (\alpha_{i,j}(1-\hat{\theta}_i) + \alpha_{j,i}\hat{\theta}_j) \\ + \sum_{i=1}^{N} (s_{K_i + 2^{D-1-d}} - G_i)\hat{\theta}_i \,.$$

Again, this is a disjoint union of at least 2^d problems, that can be solved on a graph with the same vertices as before (and less edges). One then updates K_i according to the rule

$$K_i \; \leftarrow \; K_i + 2^{D-1-d}\hat{\theta}_i \,.$$

At the end of the process, one finds an array (K_i) of values between 0 and $2^D - 1$, such that if $K_i = k$, then: if $k = 0$, $w \leq s_1$, if $k = n = 2^D - 1$, $w \geq s_n$, and in all

other cases, $s_k \leq w \leq s_{k+1}$. This provides, of course, an approximation of w, with a precision controlled by 2^{-D}. In particular, we get an exact solution z of (5) for the levels $l_0 < l_1 < \cdots < l_n$, $n = 2^D - 1$, if for each $k \geq 1$, $s_k = (l_k + l_{k-1})/2$ and we let at the end of the process $z_i = l_{K_i}$.

3.4 Second method (Ishikawa's representation)

An alternative approach to solve (5) is to use the representation of Ishikawa (see [18, 19]). The idea is to introduce an additional dimension and represent the field $z \in X$, $z_i \in \{l_0, \ldots, l_n\}$ for all i, in the following way: we let $Y = X^n$ and consider all binary fields $\Theta \in Y$, $\Theta = (\Theta_i^k)_{i=1,\ldots,N}^{k=1,\ldots,n}$ such that $\Theta_i^k \leq \Theta_i^{k-1}$ for $2 \leq k \leq n$, and any i. Then, there is a one-to-one correspondence between admissible z for problem (5) and these binary fields, if to any such z we associate Θ given by $\Theta_i^k = 1$ if $z_i \geq l_k$, and 0 otherwise. If we define the energies (assuming, to simplify, that $l_k - l_{k-1} = \delta l$ is independent on k)

$$F_1(\Theta) = \delta l \sum_{k=1}^{n} \sum_{1 \leq i < j \leq N} \alpha_{i,j} |\Theta_i^k - \Theta_j^k| + \sum_{i=1}^{N} \sum_{k=2}^{n} \infty \cdot (\Theta_i^k - \Theta_i^{k-1})^+ + \sum_{i=1}^{N} \sum_{k=1}^{n} \frac{(l_k - G_i)^2 - (l_{k-1} - G_i)^2}{2} \Theta_i^k$$

and

$$F_2(\Theta) = \delta l \sum_{k=1}^{n} \sum_{1 \leq i < j \leq N} \alpha_{i,j} |\Theta_i^k - \Theta_j^k| + \sum_{i=1}^{N} \sum_{k=2}^{n} \left(\infty \cdot (\Theta_i^k - \Theta_i^{k-1})^+ + \frac{(l_{k-1} - G_i)^2}{2} (\Theta_i^k - \Theta_i^{k-1})^- \right) + \sum_{i=1}^{N} \frac{(l_n - G_i)^2}{2} \Theta_i^n - \frac{(l_0 - G_i)^2}{2} \Theta_i^1 ,$$

then one easily checks that for any $\Theta \in Y$, $F_1(\Theta) = F_2(\Theta)$. Moreover, when this value is finite, then Θ_i^k is nonincreasing with respect to k, so that Θ is in correspondence with an admissible $z \in X$, $z_i \in \{l_0, \ldots, l_n\}$. In this case, one has

$$F_1(\Theta) = F_2(\Theta) = J(z) + \frac{1}{2} \sum_{i=1}^{N} (z_i - G_i)^2 - N \frac{(l_0 - G_i)^2}{2} .$$

Hence, up to a constant, the energy of Θ is the same as the energy of z. The consequence is that problem (5) can be solved by finding the (unique) optimal cut in the graph associated to the energy F_1 or F_2. Our experiments seem to show that the max flow algorithm of [6] performs better on the graph associated to F_1 than on the one associated to F_2.

Let us observe that this construction is quite general: in [18], it is shown that as soon as J is convex an energy such as F_1 or F_2 can be found, whose minimization gives a solution to the initial problem. We will see that in our case this method is

(by far) less efficient than the algorithm proposed in Section 3.3. However, it can be used for energies much more general than (4)-(5). In particular, it is important to notice that it will solve any (quantized) problem such as

$$\min_z J(z) \; + \; \sum_{i=1}^{N} H(i, z_i)$$

where the function H *needs not be convex* in z_i (whereas the method in Section 3.3 is easily adapted to solve this problem as long as $z_i \mapsto H(i, z_i)$ is convex (just replace $s - G_i$ by $\partial_{z_i} H(i, \cdot)_{|s}$ in problem (P_s)), but will not work otherwise). This is of particular interest in stereo correspondence problems, or for the computation of optical flows.

Eventually, we would like to point out the fact that the *same* representation appeared recently in the continuous setting, for the study of functionals of the form $u \mapsto \int f(x, u, \nabla u)$ – possibly with an additional jump term (like the Mumford-Shah functional). A necessary condition of minimality was derived from this representation, related to the existence of a vector field that maximizes its flow through the graph of u, in a certain class. See [1, 9].

4 TV minimization

We now consider the case where our vector space X represents the grey-level values of a bidimensional image, that is, $X = \mathbb{R}^{N \times M}$, and a vector $w \in X$ is a matrix $(w_{i,j})_{i=1,\dots,N, j=1,\dots,M}$. We consider the simplest, anisotropic discretization of the TV, given by

$$J(w) \; = \; \sum_{i,j} |w_{i+1,j} - w_{i,j}| + |w_{i,j+1} - w_{i,j}|$$

where in the sum all terms that are well-defined appear. Extension of what will be said to more complex interactions (not nearest-neighbours, or nonuniform) is obvious.

We see that problem (2) is of the form (4). On the other hand, if a, b and $g_{i,j}$ are scalar quantities in (1), then clearly this problem is a particular 2-levels case of (5), with G simply given by g and λ by $\lambda/(b-a)$ (assuming $b > a$). If those quantities are vectorial, on the other hand, one can also rewrite (1) as

$$\min_{\theta_{i,j} \in \{0,1\}} \lambda \sum_{i,j} J(\theta) \; + \; \sum_{i,j} \left(\frac{b^2 - a^2}{2} - (b-a) \cdot g_{i,j} \right) \theta_{i,j} \; + \; \text{constant} \, ,$$

which is of the form (P_s).

We will compare the two algorithms described in sections 3.3 and 3.4 to the algorithm introduced in [11], for minimizing (2). Let us briefly recall this algorithm. First of all, we mention that also this algorithm could be described in the more general abstract setting of the previous section. However, it does not seem to be much more efficient than the first algorithm in section 3.3. Its strength, on the other hand, is that it also works with interaction energies of the form

$$J_{iso}(w) \; = \; \sum_{i,j} \sqrt{(w_{i+1,j} - w_{i,j})^2 + (w_{i,j+1} - w_{i,j})^2} \, ,$$

that to our knowledge are not easily handled by the methods described previously. (In particular, J_{iso} does not satisfy (3), but it may be seen as a discretization of the "true" total variation, which satisfies, in the continuous setting, the co-area formula.) Also, it is easily generalized to the case where $w_{i,j}$ is vectorial (case of color/multispectral images). For these reasons, we prefer to stick to the description in [11], in the particular setting of a 2D, nearest-neighbours interaction energy. Let us briefly recall how the algorithm is implemented.

The energies J and J_{iso} can be written

$$J(w) \;=\; \sum_{i,j} |(\nabla^x w)_{i,j}| + |(\nabla^y w)_{i,j}| \;\text{ and }\; J_{iso}(w) \;=\; \sum_{i,j} |(\nabla w)_{i,j}|$$

where the vector $(\nabla w)_{i,j} = ((\nabla^x w)_{i,j}, (\nabla^y w)_{i,j}) \in X \times X$ is defined by

$$(\nabla^x w)_{i,j} \;= \begin{cases} w_{i+1,j} - w_{i,j} & \text{if } \; 1 \leq i < N, 1 \leq j \leq M \\ 0 & \text{if } \; i = N, 1 \leq j \leq M, \end{cases}$$

$$(\nabla^y w)_{i,j} \;= \begin{cases} w_{i,j+1} - w_{i,j} & \text{if } \; 1 \leq i \leq N, 1 \leq j < M \\ 0 & \text{if } \; 1 \leq i \leq N, j = M. \end{cases}$$

When $(x, y) \in \mathbb{R}^2$, $|(x, y)|$ denotes the Euclidean norm. If both X and $X \times X$ are endowed with the standard Euclidean scalar product, then a discrete divergence is given by $\mathrm{div} = -\nabla^*$, that is

$$(\mathrm{div}\, \xi, w)_X \;=\; -(\xi, \nabla w)_{X \times X} \;\forall w \in X, \xi \in X \times X.$$

(It is easily computed, see [11].)

By standard duality arguments, it is shown in [11] that the solution of

$$\min_{w \in X} J_{iso}(w) \;+\; \frac{1}{2\lambda} \|w - g\|^2 \tag{7}$$

is given by $\overline{w} = g + \lambda \mathrm{div}\, \overline{\xi}$ where $\overline{\xi}$ is a solution to

$$\min\{\|g + \lambda \mathrm{div}\, \xi\|^2 \,:\, \xi = (\xi^x, \xi^y) \in X \times X \,, |\xi_{i,j}| \leq 1 \forall i, j\}\,. \tag{8}$$

Moreover, one has $\overline{\xi}_{i,j} \cdot (\nabla \overline{w})_{i,j} = |(\nabla \overline{w})_{i,j}|$ for all i, j. The same proof will show that the solution of (2) is given by the same formula, with now $\overline{\xi}$ a solution to

$$\min\{\|g + \lambda \mathrm{div}\, \xi\|^2 \,:\, \xi \in X \times X \,, |\xi^x_{i,j}| \leq 1 \text{ and } |\xi^y_{i,j}| \leq 1 \forall i, j\}\,. \tag{9}$$

and $\overline{\xi}_{i,j} \cdot (\nabla \overline{w})_{i,j} = |(\nabla^x \overline{w})_{i,j}| + |(\nabla^y \overline{w})_{i,j}|$ for all i, j. We observe that the field $\overline{\xi}$ which is found here is closely related to the flow computed by the max-flow algorithm of the previous sections.

In [11], the following iterative algorithm is proposed to solve (8). We let $\xi^0 = 0$, and for all $n \geq 0$ we let

$$\begin{cases} w^n \;=\; g \;+\; \lambda \mathrm{div}\, \xi^n \\ \xi^{n+1}_{i,j} \;=\; \dfrac{\xi^n_{i,j} \;+\; (\tau/\lambda)(\nabla w^n)_{i,j}}{1 + (\tau/\lambda)|(\nabla w^n)_{i,j}|} \end{cases} \tag{10}$$

where $\tau > 0$ is a fixed "time-step". It is shown in [11] that as $n \to \infty$, $w^n \to \overline{w}$, provided $\tau \leq 1/8$ (in fact, experimental convergence is observed as long as $\tau \leq 1/4$). The following variant, which is a simple gradient descent/reprojection method, seems to perform better:

$$\left\{ \begin{array}{rcl} w^n & = & g + \lambda \mathrm{div}\, \xi^n \\ \xi_{i,j}^{n+1} & = & \dfrac{\xi_{i,j}^n + (\tau/\lambda)(\nabla w^n)_{i,j}}{\max\{1, |\xi_{i,j}^n + (\tau/\lambda)(\nabla w^n)_{i,j}|\}} \end{array} \right. \tag{11}$$

It is easy to show the stability of this scheme up to $\tau \leq 1/4$ (indeed, $\xi^n \mapsto \xi^{n+1}$ is 1-Lipschitz); convergence is also probably true but not straigthforward (since $\nabla \mathrm{div}$ is singular). Experiments show that it converges as long as $\tau \leq 1/4$, however, $\tau = 1/4$ is not optimal, and a better convergence is obtained for $.24 \lesssim \tau \lesssim .249$.

We observe that the error between w^n and the solution $\overline{w}$ of (7) is easily estimated: indeed, since $w^n = g + \lambda \mathrm{div}\, \xi^n$ and $\overline{w} = g + \lambda \mathrm{div}\, \overline{\xi}$, one has

$$\|w^n - \overline{w}\|^2 \;=\; \lambda(\mathrm{div}\, \xi^n - \mathrm{div}\, \overline{\xi}, w^n - \overline{w})_X \;=\; -\lambda(\xi^n - \overline{\xi}, \nabla w^n - \nabla \overline{w})_{X \times X}\,.$$

Since $(\xi, \nabla w)_{X\times X} \leq J_{iso}(w)$ for any $w \in X$ and $\xi \in X \times X$ with $|\xi_{i,j}| \leq 1$ for all i, j, and since $(\overline{\xi}, \nabla \overline{w})_{X\times X} = J_{iso}(\overline{w})$, we find that

$$\|w^n - \overline{w}\|^2 \;\leq\; \lambda J_{iso}(w^n) - (\xi^n, \nabla w^n)_{X\times X}. \tag{12}$$

The situation is the same with problem (2) and its dual (9): the fixed point scheme (10) is replaced with

$$\left\{ \begin{array}{rcl} w^n & = & g + \lambda \mathrm{div}\, \xi^n \\ (\xi_{i,j}^{n+1})^x & = & \dfrac{(\xi_{i,j}^n)^x + (\tau/\lambda)(\nabla^x w^n)_{i,j}}{1 + (\tau/\lambda)|(\nabla^x w^n)_{i,j}|}\,, \\ (\xi_{i,j}^{n+1})^y & = & \dfrac{(\xi_{i,j}^n)^y + (\tau/\lambda)(\nabla^y w^n)_{i,j}}{1 + (\tau/\lambda)|(\nabla^y w^n)_{i,j}|}\,, \end{array} \right. \tag{13}$$

while the gradient descent/projection scheme (11) becomes

$$\left\{ \begin{array}{rcl} w^n & = & g + \lambda \mathrm{div}\, \xi^n \\ (\xi_{i,j}^{n+1})^x & = & \dfrac{(\xi_{i,j}^n)^x + (\tau/\lambda)(\nabla^x w^n)_{i,j}}{\max\{1, |(\xi_{i,j}^n)^x + (\tau/\lambda)(\nabla^x w^n)_{i,j}|\}}\,, \\ (\xi_{i,j}^{n+1})^y & = & \dfrac{(\xi_{i,j}^n)^y + (\tau/\lambda)(\nabla^y w^n)_{i,j}}{\max\{1, |(\xi_{i,j}^n)^y + (\tau/\lambda)(\nabla^y w^n)_{i,j}|\}}\,. \end{array} \right. \tag{14}$$

The estimate (12) remains true, with now J_{iso} replaced with J.

5 Comparisons

We have compared four programs based on the two algorithms in Sections 3.3 and 3.4, and the two variants (fixed-point and gradient descent/projection) of the algorithm in Section 4, for the anisotropic problem (2).

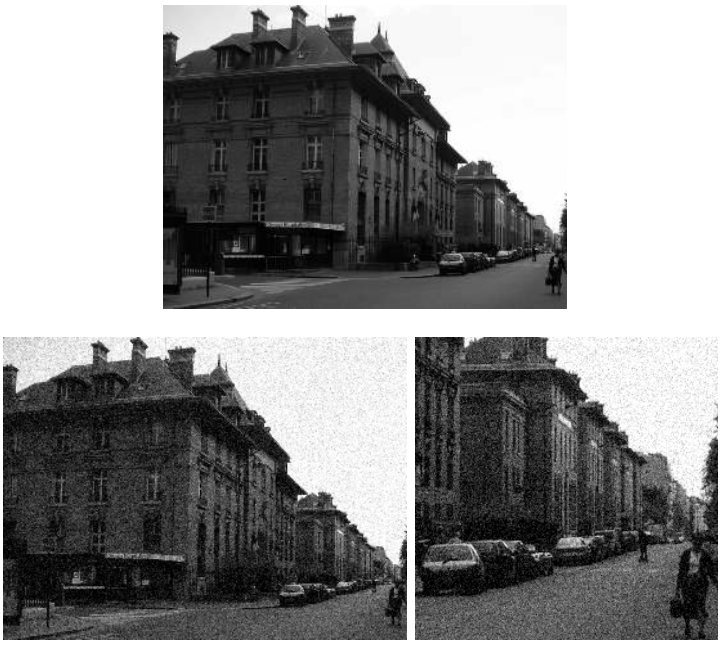

Figure 1: The original and noisy image used in the experiments

We performed the denoising of an image of $800 \times 600 = 480\,000$ pixels (corrupted with a noise of standard deviation 20, for original values in $[0, 255]$, with deviation $\sim 90 - \text{SNR} \sim 20$), and of a smaller subimage of size $256 \times 256 = 65\,536$ pixels, see Figure 1. We chose a value of $\lambda = .1 \times (\pi/4)$ for the large image and $.08 \times (\pi/4)$ for the smaller (the factor $\pi/4$ is introduced since when, roughly speaking, the "directions" are uniformly distributed in an image w then $J_{iso}(w) = (\pi/4)J(w)$: this allows to compare better the results obtained with J and J_{iso}). The results are shown in figure 2.

All algorithms were programmed in C/C++ and were run on a 3.20 GHz-Pentium 4 Linux 2.6 system with 1 Mb of cache. The max flow algorithm program was the `maxflow-v2.2` implementation of [6], implemented by Vladimir Kolmogorov and that we downloaded from his web page. The type of the capacities was set to

Figure 2: The results

`double`, and it is likely that the results can be slightly improved by letting it to `short` or `int` and appropriately quantifying the values. This max-flow program was then linked with an appropriate C++ program organizing the dyadic decomposition of the levels. The execution times of the programs was measured using the `times()` C command.

Table 1 compares the algorithms for the small image. The RMSE and Absolute Difference are with respect to the "true" solution, computed using the dyadic graph-cut algorithm at depth 16 (precision $255/(2^{-16}-1)$). The RMSE is renormalized (divided by 255) whereas the absolute difference is in pixel values (in $[0-255]$). For the fixed point algorithm and the projected gradient-descent ("proj. grad.")

method	time (s)	iter.	RMSE	Abs. Diff. [0-255]
graph-cut, depth = 8	.46			1. (theor.)
graph-cut, depth = 12	.87			.0625 (theor.)
graph-cut, depth = 16	1.32			.0039 (theor.)
fixed point, err = .01	.13	32	.0047	5.7
fixed point, err = .005	.35	88	.0022	3.0
proj. grad., err = .01	.17	37	.000717	4.0
proj. grad., err = .005	.38	81	.000371	2.2

Table 1: Comparisons for the small 256×256 image

algorithm (14), the estimate (12) (renormalized in order to be an RMSE estimate) was used as a stopping criterion. In the tables, "err=xxx" gives the corresponding value. We observe that for the projected gradient algorithm, the RMSE that is actually reached is about 7% of the stopping criterion, while it is almost 50% for the fixed point algorithm, the total number of iterations remaining of the same order: it shows that the projected gradient is more efficient. As a matter of fact, for a stopping criterion of .01, oscillations remain visible in the output of the fixed-point method, while they are much attenuated in the output of the projected gradient method. This algorithm seems to be the most efficient, however, the control of the error is more precise with the graph-cut algorithm. Another important observation is that the projected gradient algorithm is quite straightforward to implement. The

method	time (s)	iter.	RMSE	Abs. Diff. [0-255]
graph-cut, depth = 8	4.0			1. (theor.)
graph-cut, depth = 12	8.7			.0625 (theor.)
graph-cut, depth = 16	13.9			.0039 (theor.)
fixed point, err = .01	1.60	50	.0047	6.6
fixed point, err = .005	4.90	154	.0022	3.0
proj. grad., err = .01	2.62	79	.000683	3.3
proj. grad., err = .005	5.50	163	.000387	2.4

Table 2: Comparisons for the large 800×600 image

comparisons for the larger image, in Table 2, yield the same conclusions. In both cases, both iterative algorithms had a "time-step" $\tau = .249$. Experiments with

$\tau = .24$ show that the projected gradient iterations stop much earlier: after 37 iterations and 1.17 seconds for a stopping value of .01 and 110 iterations and 3.73 seconds for the stopping value of .005. However, in both cases, the RMSE that is attained is also proportionally higher than with $\tau = .249$: .001162 in the first case and .000535 in the second case. Still, this seems to show that it works better than the fixed-point iteration.

We also have run Ishikawa's algorithm of Section 3.4. Obviously, it is much slower than our dyadic graph-cut method (and gives exactly the same answer). Is also requires a huge amount of memory, so that we did not run it with 256 levels. With 16, 32, and 64 levels, it ran in respectively 1.61, 3.65 and 8.15 seconds, on the small image (using the graph representing the energy F_1, and 2.93, 7.4 and 19.6 seconds for the graph representing F_2) while the dyadic graph-cut method at depths 4, 5 and 6 ran respectively in .16, .21 and .31 seconds. However, we recall again that Ishikawa's method can be used in much more difficult (nonconvex) problems.

6 Conclusion

We compared three different techniques for solving the (anisotropic) Rudin-Osher-Fatemi minimization problem. One, based on the exact resolution of binary MRFs by integer optimization methods (and which is already found in [14, 15], has the advantage that it yields an exact solution of the problem, up to a known precision. However, it seems that the method proposed in [11] or the simple gradient-descent/projection method given by (14) yield comparable results (the later being more efficient). We observe that all of these methods solve, in fact, an appropriate dual problem.

Providing a sharp a posteriori L^2 or, even better, L^∞ error estimate for the scheme (14) would be a considerable improvement.

We also observe that our iterative schemes can easily be made "more" isotropic by using J_{iso} instead of J (cf. scheme (11)). On the other hand, improving the rotational invariance of the algorithm in Section 3.3 requires the use of energies with more interactions (next-nearest-neighbours, at least), at the cost of efficiency. The advantage of iteration (11) over (14) is illustrated in Figure 3.

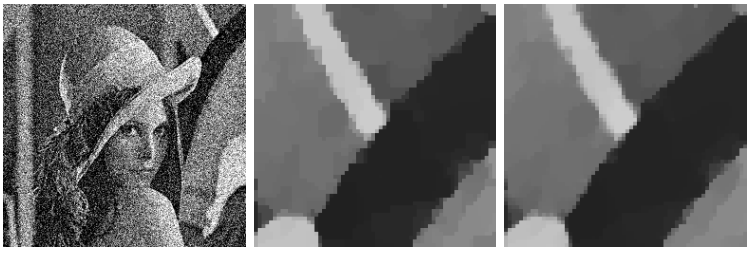

Figure 3: A noisy image and a detail of the denoised image, first with the anisotropic nearest-neighbours interaction potential J and then with J_{iso}

References

[1] G. Alberti, G. Bouchitté, and G. Dal Maso. The calibration method for the Mumford-Shah functional and free-discontinuity problems. *Calc. Var. Partial Differential Equations*, 16(3):299–333, 2003.

[2] F. Alter, V. Caselles, and A. Chambolle. A characterization of convex calibrable sets in $\mathbb{R}^N$. *Math. Ann.*, 332(2):329–366, June 2005.

[3] F. Alter, V. Caselles, and A. Chambolle. Evolution of characteristic functions of convex sets in the plane by the minimizing total variation flow. *Interfaces Free Bound.*, 7(1):29–53, 2005.

[4] W. Bell. A C++ implementation of a Max Flow-Graph Cut algorithm. available at `http://www.cs.cornell.edu/vision/wbell/`, November 2001. Computer Science Dept, Cornell University.

[5] G. Bouchitté. Recent convexity arguments in the calculus of variations. (lecture notes from the 3rd Int. Summer School on the Calculus of Variations, Pisa), 1998.

[6] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004.

[7] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *International Conference on Computer Vision*, pages 377–384, September 1999.

[8] V. Caselles and A. Chambolle. Anisotropic curvature-driven flow of convex sets. Technical Report 528, CMAP, Ecole Polytechnique, 2004.

[9] A. Chambolle. Convex representation for lower semicontinuous envelopes of functionals in L^1. *J. Convex Anal.*, 8(1):149–170, 2001.

[10] A. Chambolle. An algorithm for mean curvature motion. *Interfaces Free Bound.*, 6(2):195–218, 2004.

[11] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20(1-2):89–97, 2004. Special issue on mathematics and image analysis.

[12] T. F. Chan and S. Esedoglu. Aspects of total variation regularized L^1 function approximation. Technical Report 04-07, UCLA CAM, February 2004.

[13] T. F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. Technical Report 04-54, UCLA CAM, September 2004.

[14] J. Darbon and M. Sigelle. Exact optimization of discrete constrained total variation minimization problems. In R. Klette and J. Zunic, editors, *Tenth International Workshop on Combinatorial Image Analysis*, volume 3322 of *LNCS*, pages 548–557, December 2004.

[15] J. Darbon and M. Sigelle. A fast and exact algorithm for total variation minimization. In J. S. Marques, N. Pérez de la Blanca, and P. Pina, editors, *2nd Iberian Conference on Pattern Recognition and Image Analysis*, volume 3522 of *LNCS*, pages 351–359, June 2005.

[16] E. Giusti. *Minimal surfaces and functions of bounded variation*, volume 80 of *Monographs in Mathematics*. Birkhäuser Verlag, Basel, 1984.

[17] D.M. Greig, B.T. Porteous, and A.H. Seheult. Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc. B*, 51:271–279, 1989.

[18] H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003.

[19] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 125–131, 1998.

[20] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision*, volume 3, pages 82–96, may 2002.

[21] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(26):147–159, 2004.

[22] Sylvain Paris, François Sillion, and Long Quan. A surface reconstruction method using global graph cut optimization. *International Journal of Computer Vision*, 2005. to appear.

[23] G. Poggi and A. R. P. Ragozini. Image segmentation by tree-structured Markov random fields. *IEEE Signal Processing Letters*, 6:155–157, 1999.

[24] M. Rivera and J.C. Gee. Two-level MRF models for image restoration and segmentation. In *Proc. British Machine Vision Conference, London*, volume 2, pages 809–818, Sept. 2004.

[25] Sebastien Roy and Ingemar J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *ICCV*, pages 492–502, 1998.

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

[27] G. Scarpa, G. Poggi, and J. Zerubia. A binary tree-structured MRF model for multispectral satellite image segmentation. Rapport de recherche RR-5062, INRIA Sophia Antipolis, December 2003.

[28] B. A. Zalesky. Network flow optimization for restoration of images. *J. Appl. Math.*, 2(4):199–218, 2002.