Total Variation Minimization and a Class of Binary MRF Models

Antonin Chambolle

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

Abstract. We observe that there is a strong connection between a whole class of simple *binary* MRF energies and the Rudin-Osher-Fatemi (ROF) Total Variation minimization approach to image denoising. We show, more precisely, that solutions to binary MRFs can be found by minimizing an appropriate ROF problem, and vice-versa. This leads to new algorithms. We then compare the efficiency of various algorithms.

1 Introduction

The goal of this note is 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. Precisely, we will 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 $g = (g_{i,j})_{i,j}$ results from an original binary signal $(b + \theta_{i,j}(a-b))_{i,j}$ taking only values a and b, to which a Gaussian noise is added. (Here, g, a, b could be vector valued.) But, in fact, similar problems involving binary fields appear in many branches of image processing and can be used in a much more elaborate way: for instance, in [24] the authors build a tree of binary MRFs to classify images in much more than two labels (see also [20, 21]).

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.

A. Rangarajan et al. (Eds.): EMMCVPR 2005, LNCS 3757, pp. 136–152, 2005.

This has been first observed by Greig, Porteous and Seheult [14], and these techniques have been then extended to much more general problems in the recent years [15, 6, 17, 18, 19, 22].

On the other hand, problem (2) has been first proposed in image processing by Rudin, Osher and Fatemi [23], as an efficient approach to edge-preserving image denoising or reconstruction. Here, g is the sum of a "clean" image w and a Gaussian noise, and the minimizer w of the energy is supposed to be a good approximation of the original signal.

Our main point, in this paper, 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. While a first version of this note was already submitted to the EMMCVPR conference, J. F. Aujol mentioned to us the recent work of Jérôme Darbon and Marc Sigelle [12, 13], which may be seen as the probabilistic counterpart of the present work. They show essentially the same results (including, in particular, Prop. 2), with very different proofs. Although we may claim our proofs are probably simpler, they reach the same conclusions and the algorithm they derive is essentially equivalent to the dyadic algorithm we present in section 3.3.

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. A first important result is a comparison principle for the solutions of (1) (Prop. 2). From this, we can deduce that these solutions are embedded in the level sets of the solutions of appropriate problems (2) (Prop. 3). This has interesting theoretical consequences: for instance, we can deduce that solutions of (1) are generically unique. But from a practical point of view, it also shows that one can use algorithms for one problem to solve the other one. This is discussed in Section 3. We describe some implementations of graph cuts algorithms that can be designed to solve (2). We then recall, in Section 4, the algorithm proposed in [9]. Numerical experiments are eventually performed to compare these algorithms.

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. 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 $i = 1, \dots, N$, $u_i^t = \begin{cases} 1 & \text{if } u_i > t\,, \\ 0 & \text{otherwise,} \end{cases}$

that is, $u^t = \chi_{\{u>t\}}$, the characteristic function (in {0,1,...,N}) of the superlevel s of $u = (u_i)_{i=1}^N$. Observe that the one-homogeneity of J follows in fact easily 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. We now study problems of the form (P_s). A first observation, which is quite obvious, is the following:

Proposition 1. *Any solution θ of (P_s) is also a solution of the convex problem*

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

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

The proof is simple and based on the following identity: if $v_i \in [0,1]$ for any i, then $v_i = \int_0^{v_i} dt = \int_0^1 v_i^t \, dt$, so that

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

This property shows that the minimization of the binary problem (P_s) is in fact a convex minimization problem. In the sequel, problems (P_0) and (P_0') are simply denoted by (P) and (P').

2.3 Comparison for Binary MRFs

Let us now mention the following comparison property, which does not seem to be well-known. It is also proved in J. Darbon and M. Sigelle's recent papers [12, 13], by a probabilistic approach. Our proof is quite simpler, and may be seen as the finite-dimensional counterpart of the proofs we proposed in [1, 2, 7].

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

Proof. The proof of this result relies on the following Lemma:

Lemma 1. *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 $i = 1, \ldots, N$. We do not prove Lemma 1 (first mentioned to us by Bouchitté [4]). It follows from (i) the convexity and 1-homogeneity of J, and (ii) the coarea formula (3), applied to the vector $\theta + \theta' = \theta \vee \theta' + \theta \wedge \theta'$, where $\theta_i, \theta'_i \in \{0, 1\}$ for all i.

We compare the energies of v^1, v^2 with the energies of $v^1 \vee v^2$ and $v^1 \wedge v^2$:

$$\begin{aligned} \lambda J(v^1) - \textstyle\sum_{i=1}^N G_i^1 v_i^1 &\leq \lambda J(v^1 \vee v^2) - \textstyle\sum_{i=1}^N G_i^1 (v_i^1 \vee v_i^2)\,, \\ \lambda J(v^2) - \textstyle\sum_{i=1}^N G_i^2 v_i^2 &\leq \lambda J(v^1 \wedge v^2) - \textstyle\sum_{i=1}^N G_i^2 (v_i^1 \wedge v_i^2)\,. \end{aligned}$$

Summing both inequalities and using the Lemma, we find

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

and since $(v_i^1 \vee v_i^2) - v_i^1 = v_i^2 - (v_i^1 \wedge v_i^2) = \max\{v_i^2 - v_i^1, 0\}$, we easily deduce that if $G_i^1 > G_i^2$ for every i, then $v_i^2 - v_i^1 \leq 0$. □

In fact, the property in Lemma 1 is equivalent to the generalized coarea formula (3) (assuming J is a convex, l.s.c., one-homogeneous function such that $J(c + \cdot) = J(\cdot)$ for any $c \in \mathbb{R}$). Functions that satisfy the thesis of Lemma 1 appear in optimization theory as "sub-modular" functions, and it is observed in [18] that they are the only two-point interactions functions of binary variables that can be minimized using graph-cut algorithms.

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 3. *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 $w^s \leq \theta \leq \overline{\overline{w}}^s$ (resp, $w^s \leq v \leq \overline{\overline{w}}^s$): that is, $w_i \geq s$ when $\theta_i = 1$, and $w_i \leq s$ when $\theta_i = 0$. 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 continuous setting, the same relationship has been observed in [8, 7]. 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 [10, 11].

Proof. We briefly sketch the proof of this result. 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. Indeed, if $s > s'$, one readily sees that any pair of solutions θ, θ' of (P_s) and $(P_{s'})$ will satisfy $\theta' \geq \theta$. Hence, letting for each $i = 1, \ldots, N$

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

it is not difficult to show that for any $s \in \mathbb{R}$, w^s is a solution of (P_s). Moreover, if $s \notin \{w_i \,:\, i = 1, \ldots, N\}$, then it is the unique solution, still because of the comparison principle. On the other hand, if $s \in \{w_i \,:\, i = 1, \ldots, N\}$, then also $\overline{\overline{w}}^s$ is a solution (and there might be other solutions in between).

We now explain why w is the solution of (4). For $v \in X$, and $s_* \leq \min_i v_i$,

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

By use of this formula and the coarea formula (3), one deduces (if $s_* \leq \min_i w_i$ as well) that the energy (4) of v must be larger than the energy of w. Hence w is the (unique) solution of the (strictly convex) problem (4). □

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 the real levels $(l_k)_{k=1}^n$ are given. 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). Then the following result is true.

Proposition 4. *Let z be a solution of* (5)*, and w be the solution of* (4)*. Then $\max_{i=1}^N |z_i - w_i| \leq \delta l/2$, provided $l_0 \leq \min_i w_i$ and $l_n \geq \max_i w_i$.*

The last condition is certain if l_0 is chosen no larger than $\min_i G_i$, and l_n no less $\max_i G_i$. This results means that the quantized problem (5) produces exactly a quantization of the solution of (4).

Proof. 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 $\theta_i^k = 1$ 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,

$$\|G - z\|^2 \;=\; \sum_{i=1}^{N} (G_i - l_0)^2 \;+\; 2\delta l \sum_{k=1}^{n} \sum_{i=1}^{N} \left(\frac{l_k + l_{k-1}}{2} - G_i \right) \theta_i^k \,,$$

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

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

where the min is on the binary fields $(\theta^k)_{k=1}^n$, 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, 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*. By Proposition 3, a solution of (P_{s_k}) is given by w^s, where w solves (4) – and any solution is between w^s and $\overline{\overline{w}}^s$. 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 deduce that any solution z of (5) satisfies $|z_i - w_i| = \min_{k=0}^{n} |l_k - w_i|$ for any i: in particular, we deduce the thesis of Proposition 4. □

2.6 Observation

One may check that all that has been said up to now can be adapted to problems of the form

$$\min_{w} J(w) \;+\; \sum_{i=1}^{N} H(i, w_i) \tag{6}$$

where for all i, $H(i, \cdot)$ is strictly convex, by replacing $s - G_i$ with $\partial_{w_i} H(i, \cdot)_{|s}$ in (P_s). The strict convexity of H (that is, the fact $\partial_{w_i} H(i, \cdot)$ is increasing) is important here, as it allows to use Proposition 2; however, many of our results remain valid with some adaption when the convexity is not strict.

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 [25].)

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 $\{l_0, \dots, l_n\}$), 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 (see [10, 11]), 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

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

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

$$\mathcal{E} \;=\; \{(i,j) \,:\, 1 \le i, j \le N \, i \ne j \,, \alpha_{i,j} > 0\} \\ \cup \{(s,i) \,:\, 1 \le i \le N\} \cup \{(i,t) \,:\, 1 \le i \le N\} \,.$$

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, with $s \in \mathcal{V}_s$ and $t \in \mathcal{V}_t$, 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 [14, 6, 18, 5, 15], 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

$$E(\mathcal{V}_s, \mathcal{V}_t) \;=\; J(\theta) - \sum_{i=1}^{N} G_i\theta_i \,+\, \text{constant} \tag{7}$$

for any cut $(\mathcal{V}_s, \mathcal{V}_t)$. Indeed, 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 (7).

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 [5] for a very clear description of the method, and of an algorithm.

3.2 Minimization of (4) Using Graph Cuts

3.3 First Method

According to the discussion in section 2, one deduces the following method for minimizing (4) using the max flow algorithm for computing graph cuts. It 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 [12, 13]. 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 running from 0 to $D - 1$, we segment at each level s_k for $k = (2p - 1)2^{D-1-d}$, $p = 1, \ldots, 2^d$. 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 $\theta_i = 0$, $w_i \leq s_k$ (by Proposition 3). We let $K_i = k = 2^{D-1}$ when $\theta_i = 1$ and we leave the value 0 when $\theta_i = 0$.

Next, 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

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

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 $\mathcal{V}\backslash\{s, t\}$ 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)

We briefly mention alternative approach to solve (5) using a max flow algorithm. It is based on a representation due to Ishikawa (see [15, 16]). 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 $\Theta \in Y$,

$\Theta = (\Theta_i^k)_{i=1,\dots,N}^{k=1,\dots,n}$ 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)

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

then one easily checks that for any $\Theta \in Y$, if $F(\Theta) < +\infty$ then Θ_i^k is nonincreasing with respect to k, so that Θ is in correspondence with an admissible $z \in X$, $z_i \in \{l_0, \dots, l_n\}$. In this case, one has

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

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.

Let us observe that this construction is quite general: in [15], it is shown that as soon as J is convex an energy such as F 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 (6) where the function H *needs not be convex* in w_i. This is of particular interest in some situations, like for instance stereo correspondence problems.

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 [9], 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

$$J_{iso}(w) \;=\; \sum_{i,j} \sqrt{(w_{i+1,j}-w_{i,j})^2 + (w_{i,j+1}-w_{i,j})^2}\,, \tag{8}$$

that to our knowledge are not 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 [9], in the particular setting of a 2D, nearest-neighbours interaction energy. Let us briefly recall how the algorithm is implemented.

The energy J can be written

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

where $(\nabla w)_{i,j} = ((\nabla^x w)_{i,j}, (\nabla^y w)_{i,j}) \in X \times X$ is defined by $(\nabla^x w)_{i,j} = w_{i+1,j} - w_{i,j}$ when $i < N$ and 0 if $i = N$, and $(\nabla^y w)_{i,j} = w_{i,j+1} - w_{i,j}$ when $j < M$ and 0 if $j = M$. If both X and $X \times X$ are endowed with the standard Euclidean scalar product, then a discrete divergence is given by $\text{div} = -\nabla^*$, that is

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

(It is easily computed, see [9].)

By standard duality arguments, it is shown in [9] that the solution of (2) is given by $\overline{w} = g + \lambda \text{div}\,\overline{\xi}$ where $\overline{\xi}$ is a solution to

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

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 related to the flow computed by the max-flow algorithm of the previous sections.

The adaption of the iterative algorithm of [9] to problem (9) is as follows: we let $\xi^0 = 0$, and for all $n \geq 0$ we let

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

where $\tau > 0$ is a fixed "time-step". One shows as in [9] 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:

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

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 (being $\nabla \mathrm{div}$ singular). Experiments show convergence up to $\tau \leq 1/4$, however, $\tau = 1/4$ seems not optimal, and a better convergence is obtained for $.24 \lesssim \tau \lesssim .249$.

We eventually observe that the error between w^n and the solution $\overline{w}$ of (2) can be estimated: indeed, since $w^n = g + \lambda \mathrm{div}\, \xi^n$ and $\overline{w} = g + \lambda \mathrm{div}\, \overline{\xi}$, one shows

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

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).

We first 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 ~ 90 – SNR $\simeq 13$), and of a smaller subimage of size $256 \times 256 = 65\,536$ pixels, see Figure 1. We chose a value of $\lambda = 20$ for the large image and 16 for the smaller. 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 [5], implemented by Vladimir Kolmogorov and that we downloaded from his web page. The type of the capacities was set to `double`, and it is likely that the results can be slightly improved

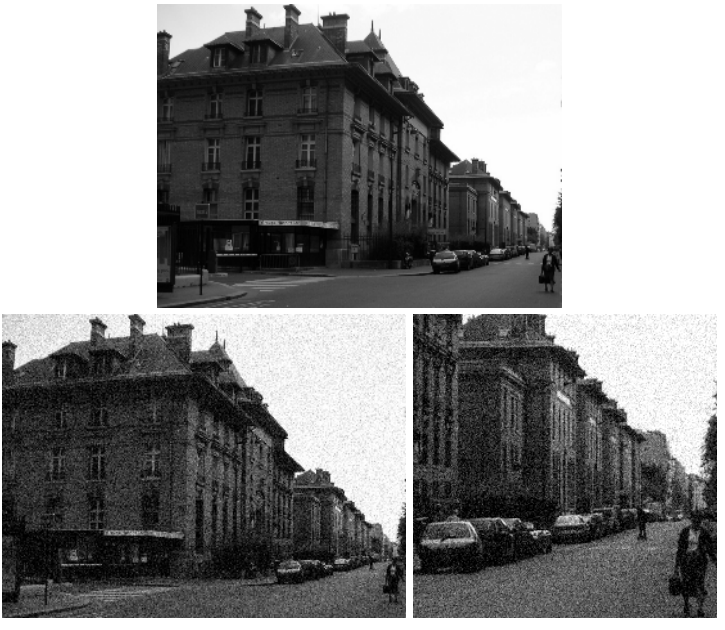

Fig. 1. The original and noisy image used in the experiments

Fig. 2. The results

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 $(1/2) \times 255/(2^{16}-1)$). 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

Table 1. Comparisons for the small 256×256 image

method	time (s)	iter.	RMSE	Abs. Diff. [0-255]
graph-cut, depth = 8	.43		.0011	.5 (theor.)
graph-cut, depth = 12	.84		.000067	.031 (theor.)
graph-cut, depth = 16	1.25		0	.002 (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

("proj. grad.") algorithm (11), 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 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.

In the two previous experiments the SNR is quite high and the value of the regularizing parameter λ is tuned so that the output remains very close to the initial data. In this range, the superiority of the graph-cut approach is not so clear. We have run another experiment on a much noiser image (see Figure 3)

Table 2. Comparisons for the large 800×600 image

method	time (s)	iter.	RMSE	Abs. Diff. [0-255]
graph-cut, depth = 8	3.9		.0011	.5 (theor.)
graph-cut, depth = 12	8.6		.000067	.031 (theor.)
graph-cut, depth = 16	13.6		0	.002 (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

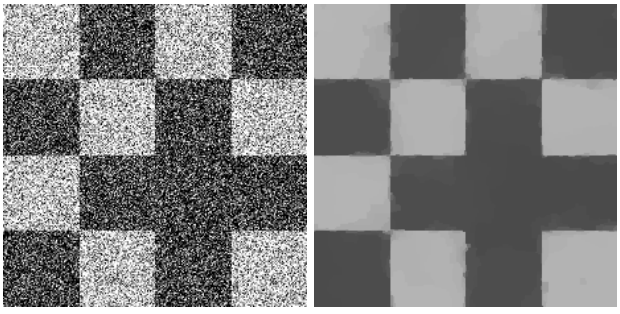

Fig. 3. The noisy checkerboard and the reconstruction

Table 3. Comparisons for the (very) noisy 230×230 checkerboard

method	time (s)	iter.	RMSE	Abs. Diff. [0-255]
graph-cut, depth = 8	0.73		.0011	.5 (theor.)
graph-cut, depth = 12	1.42		.000067	.031 (theor.)
graph-cut, depth = 16	2.15		0	.002 (theor.)
fixed point, err = .01	1.97	625	.0045	4.3
fixed point, err = .005	5.48	1752	.0019	2.0
proj. grad., err = .01	1.33	453	.0021	3.8
proj. grad., err = .005	3.68	1235	.00087	1.6

of size $230 \times 230 = 52900$ pixels, of amplitude ~ 50 (without noise) and ~ 85 with a noise of standard deviation ~ 65 (SNR $\simeq 2$). For this image, $\lambda = 80$. In this range, the graph-cut approach clearly outperforms the more classical methods, as shown in Table 3. Again, the projected gradient method is sligthly more efficient than the fixed point.

Eventually, 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 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 [12, 13], 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 [9] (or, even

better, the simple gradient-descent/projection method given by (11)) yields comparable results, in the range of coefficients useful for the processing of not too noisy images. It is clearly outperformed when the SNR becomes very poor. On the other hand, it is easily adapted to process color or multidimensional images, and can be made more isotropic (using the penalization J_{iso} given by (8)) at no expense.

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 [5] is extremely clear, we were much helped thanks to a few C++ implementations of this algorithm that are available on the internet. We would like to thank the authors of these, Walter Bell [3] and Vladimir Kolmogorov. The latter's, that we found somewhat more efficient, is available at `http://www.cs.cornell.edu/People/vnk/software.html`. 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.

References

1. 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.
2. 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.
3. 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.
4. 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.
5. Y. Boykov and V. 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.
6. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *International Conference on Computer Vision*, pages 377–384, September 1999.
7. V. Caselles and A. Chambolle. Anisotropic curvature-driven flow of convex sets. Technical Report 528, CMAP, Ecole Polytechnique, 2004.
8. A. Chambolle. An algorithm for mean curvature motion. *Interfaces Free Bound.*, 6(2):195–218, 2004.

9. 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.
10. T. F. Chan and S. Esedoglu. Aspects of total variation regularized L^1 function approximation. Technical Report 04-07, UCLA CAM, February 2004.
11. 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.
12. 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.
13. 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.
14. 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.
15. H. Ishikawa. Exact optimization for Markov random fields with convex priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003.
16. H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 125–131, 1998.
17. V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision*, volume 3, pages 82–96, may 2002.
18. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(26):147–159, 2004.
19. S. Paris, F. Sillion, and L. Quan. A surface reconstruction method using global graph cut optimization. *International Journal of Computer Vision*, 2005. to appear.
20. G. Poggi and A. R. P. Ragozini. Image segmentation by tree-structured Markov random fields. *IEEE Signal Processing Letters*, 6:155–157, 1999.
21. 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.
22. S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *ICCV*, pages 492–502, 1998.
23. L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
24. 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.
25. B. A. Zalesky. Network flow optimization for restoration of images. *J. Appl. Math.*, 2(4):199–218, 2002.