Primal-dual Algorithm
for Convex Markov Random Fields

Vladimir Kolmogorov

Microsoft Research
Cambridge, UK
vnk@microsoft.com

September, 2005
Technical Report
MSR-TR-2005-117

Computing maximum a posteriori configuration in a first-order Markov Random Field has become a routinely used approach in computer vision. It is equivalent to minimizing an energy function of discrete variables. In this paper we consider a subclass of minimization problems in which unary and pairwise terms of the energy function are convex. Such problems arise in many vision applications including image restoration, total variation minimization, phase unwrapping in SAR images and panoramic image stitching.

We give a new algorithm for computing an exact solution. Its complexity is $K \cdot T(n,m)$ where K is the number of labels and $T(n,m)$ is the time needed to compute a maximum flow in a graph with n nodes and m edges. This is the fastest maxflow-based algorithm for this problem: previously best known technique takes $T(nK, mK^2)$ time for general convex functions. Our approach also needs much less memory ($O(n+m)$ instead of $O(nK+mK^2)$).

Experimental results show for the panoramic stitching problem our method outperforms other techniques.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
`http://www.research.microsoft.com`

1 Introduction

Many early vision problems can be naturally formulated in terms of energy minimization where the energy function has the following form:

$$E(\mathbf{x}) = \sum_{u \in \mathcal{V}} D_u(x_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(x_v - x_u) \tag{1}$$

Here $(\mathcal{V}, \mathcal{E})$ is an undirected graph. Set $\mathcal{V}$ usually corresponds to pixels; x_u denotes the label of pixel $u \in \mathcal{V}$ which must belong to a finite set of integers $\{0, 1, \ldots, K-1\}$. For motion or stereo, the labels are disparities, while for image restoration they represent intensities. Terms $D_u(\cdot)$ encode unary data penalty functions, and terms $V_{uv}(\cdot)$ are pairwise interaction potentials. This energy is often derived in the context of Markov Random Fields [11]: a minimum of E corresponds to a *maximum a-posteriori* labeling $\mathbf{x}$.

In this paper we consider a subclass of energy functions in which terms $D_u(\cdot)$ and $V_{uv}(\cdot)$ are convex. We refer to such energies as "convex MRF functions". They arise in many vision applications. Our work was inspired by one of them, namely panoramic image stitching [20, 26]. Given different portions of the same scene with some overlap, the goal is to generate an output image which is similar to the original images and does not have a visible seam. The approach of [20, 26] is to compute the image whose gradients match the gradients of the two input images. This is done by minimizing an objective function of the form of equation (1). The resulting algorithm, GIST1 under l_1 norm, is shown to outperform many other stitching techniques [20, 26]. Functions E with convex terms are also used in other vision applications such as image restoration [7], minimization of total variation [8], phase unwrapping in SAR images [4].

Algorithms We now discuss algorithms for minimizing convex MRF functions. We use notation n, m for the number of nodes and edges, respectively. We concentrate on the case when the number of labels K is small compared to n.

It is well known that the problem can be solved exactly by reducing it to a linear program whose size is polynomial in n, m and K. General LP solvers, however, do not exploit the special structure of the convex MRF problem, and therefore are not very efficient.

Ishikawa [15] and Ahuja et al. [2] reduce convex MRF to a minimum s-t cut problem in a graph with $O(nK)$ nodes and $O(mK^2)$ edges. In the important case when terms $V_{uv}(\cdot)$ are piecewise linear functions with a constant number of breakpoints, the number of edges is reduced to $O(mK)$. A disadvantage of this approach is that that it needs a large amount of memory (either $O(mK^2)$ or $O(mK)$).

Algorithm in [15, 2] can be characterized as *primal* since it directly solves convex MRF minimization problem. An alternative is to solve the *dual* problem instead. This dual is known as *convex minimum cost network flow* problem. Several dual algorithms were proposed by Karzanov et al. [16] and Ahuja et al. [1]. The worst-case complexity of the latter algorithm is $O(nm \log(n^2/m) \log(nK))$, which is the best known for this problem.

It is known [3] that any convex MRF problem can be reduced to a minimum cost network flow (MCNF) problem on a graph with $O(rm)$ edges where $r \leq K$ is the maximum number of breakpoints of piecewise-linear functions $D_u(\cdot)$, $V_{uv}(\cdot)$. Therefore, it is possible to use any existing MCNF method. One of them, primal-dual MCNF algorithm of Ford and Fulkerson [9, 10], is related to the technique that we develop in this paper. In particular, the two algorithms are equivalent in a special case when terms $D_u(\cdot)$ are linear, $r = 1$ and variables $\mathbf{x}$ are unconstrained integers. However, if $r > 1$ then the techniques are different: our algorithm works with a graph with $O(m)$ rather than $O(rm)$ edges.

Our contributions In this paper we describe two algorithms for convex MRF problem - *primal* and *primal-dual*. The first one computes a global minimum by at most $2K$ calls to a minimum cut procedure on a graph with n nodes and m edges. Its name stems from the fact that it maintains only primal variables, namely configuration $\mathbf{x}$. The method is very similar to that of Bioucas-Dias et al. [4] which was originally applied to the case of functions without unary terms $D_u(\cdot)$. It is also similar to the steepest descent algorithm of Murota for minimizing $L^\natural$-*convex* functions [21, 23]. Our major contribution here is that we prove a tight bound on the maximum number of steps, while bounds in [4, 23] are much weaker. Our proof is applicable not only to convex MRF functions, but to arbitrary $L^\natural$-convex functions.

One drawback of the primal algorithm is that it solves different min cut/max flow problems independently. However, these problems are strongly related. Thus, a natural idea for speeding up computations is to use maximum flow obtained in one step as an initialization for the next step. This is a motivation of our primal-dual method. It maintains both primal variables (configuration $\mathbf{x}$) and dual variables (flow). It makes at most $2K$ calls to a maximum flow algorithm and a shortest path procedure.

The worse-case complexity of both algorithms is $O(K) \cdot T(n, m)$ where $T(n, m)$ is the running time of one maximum flow computation on a graph with n nodes and m edges. This is worse than the complexity of the algorithm in [1]. However, our techniques have a practical advantage: they rely only on a maximum flow algorithm, which is more readily available. For example, it is possible to use maxflow algorithm that was specifically tuned to vision problems [6].

Other related work Hochbaum [13] gives a very efficient algorithm for a special case of convex MRF problem. Namely, if pairwise terms are linear ($V_{uv}(x_v - x_u) = \lambda_{uv}|x_v - x_u|$) then the technique in [13] has the same worst-case complexity as that of a single maxflow computation on a graph with n nodes and m edges. Similar ideas appear in [8]. The method is applicable to the image restoration problem [7] and total variation minimization [8].

If the number of labels K is large compared to n then some of the algorithms described above become pseudo-polynomial. In this case it is possible to apply proximity scaling technique of Hochbaum et al. [14] to get an algorithm polynomial in $\log K$ rather than K (see [2]). In particular, combining proximity scaling technique with our algorithms yields complexity $n \log K \cdot T(n, m)$.

Finally, we mention some algorithms for minimizing more general functions. If terms $D_u(\cdot)$ are arbitrary and $V_{uv}(\cdot)$ are convex, then the problem can be solved exactly in time $T(nK, mK^2)$ or $T(nK, mK)$, depending on the structure of terms V_{uv} [15, 2]. Similar construction exists for *submodular* functions (see section 2.1 for definition) with unary and pairwise terms [19]. Note that pairwise terms do not have to be functions of the differences of labels. If both $D_u(\cdot)$ and $V_{uv}(\cdot)$ are arbitrary then the problem becomes NP-hard. Boykov et al. [7], Kleinberg et al. [17] and Komodakis et al. [18] give constant factor approximation algorithms in the case when terms $V_{uv}(\cdot)$ are *metrics*.

Outline In section 2 we describe the first algorithm (primal) and prove a bound on the number of steps. In section 3 we review the dual problem and present the second algorithm (primal-dual). In section 4 we compare the speed of several algorithms on the panoramic image stitching problem.

2 Primal algorithm

Before proceeding, let us make two comments about our notation.

- Graph $(\mathcal{V}, \mathcal{E})$ is undirected, so only one out of the two pairs (u, v), (v, u) is present in the sum (1). We use the convention that $V_{uv}(x_{uv}) = V_{vu}(-x_{uv})$. Sometimes it will be convenient to treat the graph as *directed*, then we write it as $(\mathcal{V}, \mathcal{A})$. By definition, $|\mathcal{A}| = 2 \cdot |\mathcal{E}|$.

- Let $\mathcal{X}$ be the set of all configurations: $\mathcal{X} = \{\mathbf{x} \in \mathbb{Z}^n | K_u^{\min} \le x_u \le K_u^{\max}\}$. (Note that we generalized the problem slightly: we allow the range of variables to be different for different nodes.) It is convenient to set $D_u(x_u) = +\infty$ if $x_u \notin [K_u^{\min}, K_u^{\max}]$. Similarly, we set $V_{uv}(x_{uv}) = +\infty$ if $x_{uv} \notin [K_v^{\min} - K_u^{\max}, K_v^{\max} - K_u^{\min}]$. Therefore, function $E(\mathbf{x})$ is now defined for all $\mathbf{x} \in \mathbb{Z}^n$.

Input: initial configuration $\mathbf{x} \in \mathcal{X}$.

1. Set $\mathsf{SuccessUp} := \mathsf{false}$, $\mathsf{SuccessDown} := \mathsf{false}$.
2. Do UP or DOWN in any order until $\mathsf{SuccessUp} = \mathsf{SuccessDown} = \mathsf{true}$:

   UP (do only if SuccessUp is false):
   - Compute $\mathbf{y} := \arg\min\{E(\mathbf{y}) \mid \mathbf{y} \in \mathcal{M}^{\text{up}}(\mathbf{x})\}$
   - If $E(\mathbf{y}) = E(\mathbf{x})$, set $\mathsf{SuccessUp} := \mathsf{true}$
   - Set $\mathbf{x} := \mathbf{y}$

   DOWN (do only if SuccessDown is false):
   - Compute $\mathbf{y} := \arg\min\{E(\mathbf{y}) \mid \mathbf{y} \in \mathcal{M}^{\text{down}}(\mathbf{x})\}$
   - If $E(\mathbf{y}) = E(\mathbf{x})$, set $\mathsf{SuccessDown} := \mathsf{true}$
   - Set $\mathbf{x} := \mathbf{y}$

Figure 1: **Primal algorithm.** *For definitions of sets* $\mathcal{M}^{\text{up}}(\mathbf{x})$ *and* $\mathcal{M}^{\text{down}}(\mathbf{x})$ *see text.*

As we have mentioned earlier, the first algorithm is very similar to the techniques in [4] and [21]. It iteratively invokes the following subroutine: given current configuration $\mathbf{x}$, compute a global minimum of function $\hat{E}(\mathbf{b}) = E(\mathbf{x} + \sigma\mathbf{b})$ where $\sigma = \pm 1$ is fixed and elements of vector $\mathbf{b}$ are allowed to take only binary values $b_u \in \{0, 1\}$. Function $\hat{E}$ can be written as a sum of unary and pairwise terms:

$$\hat{E}(\mathbf{b}) = \sum_{u \in \mathcal{V}} \hat{D}_u(b_u) + \sum_{(u,v) \in \mathcal{E}} \hat{V}_{uv}(b_u, b_v)$$

where

$$\hat{D}_u(b_u) = D_u(x_u + \sigma b_u) \ , \quad \hat{V}_{uv}(b_u, b_v) = V_{uv}((x_v + \sigma b_v) - (x_u + \sigma b_u))$$

Due to convexity of terms V_{uv} pairwise terms $\hat{V}_{uv}$ satisfy

$$\hat{V}_{uv}(0,1) + \hat{V}_{uv}(1,0) \geq \hat{V}_{uv}(0,0) + \hat{V}_{uv}(1,1)$$

Therefore, function $\hat{E}$ is *submodular* and can be minimized in polynomial time by computing maximum flow in an appropriately constructed graph (see [5], for example).

The following notation will be useful. For configuration $\mathbf{x} \in \mathcal{X}$ let $\mathcal{M}^{\text{up}}(\mathbf{x})$ be the set of all configurations that can be obtained by adding binary vector $\mathbf{b} \in \{0, 1\}^n$ to $\mathbf{x}$: $\mathcal{M}^{\text{up}}(\mathbf{x}) = \{\mathbf{x} + \mathbf{b} \mid \mathbf{b} \in \{0, 1\}^n\}$. Similarly we define the set $\mathcal{M}^{\text{down}}(\mathbf{x}) = \{\mathbf{x} - \mathbf{b} \mid \mathbf{b} \in \{0, 1\}^n\}$. These sets appeared in the work of Veksler [24] where they were used for approximate minimization of functions with *Potts* interaction terms. Following terminology of [24] we say that $\mathbf{y}$ is within 1-jump move from $\mathbf{x}$ if $\mathbf{y} \in \mathcal{M}^{\text{up}}(\mathbf{x})$, or within (-1)-jump move from $\mathbf{x}$ if $\mathbf{y} \in \mathcal{M}^{\text{down}}(\mathbf{x})$.

With these conventions, we can now present the primal algorithm (see Fig. 1). Its difference from the method of Bioucas-Dias et al. [4] is very minor: the latter uses only procedure UP. In addition, the algorithm in [4] is formulated for the case without unary terms $D_u(\cdot)$ (and therefore procedure DOWN is not needed).

The algorithm in Fig. 1 is also very similar to the steepest descent algorithm of Murota [21, 23] for minimizing $L^\natural$*-convex* functions[1]. The difference between the two is in the minimization algorithm used in the inner loop and in the schedule of procedures UP and DOWN. In general, our algorithm needs a fewer number of calls to these procedures, as will be clear from the analysis of the algorithm.

This analysis is a major contribution of our paper with regard to the primal algorithm. It leads to a tight bound on the number of steps improving bounds of [4] and [23]. Bioucas-Dias et al. show that if configuration $\mathbf{x}$ is not a global minimum of the energy, then one step of their algorithm is guaranteed to decrease $E(\mathbf{x})$. This gives a non-polynomial

[1] $L^\natural$-convex functions can be viewed as a generalization of convex MRF functions. They do not necessarily decompose as a sum of terms depending on at most two variables. See [22] for an extensive treatment of $L^\natural$-convex functions.

bound of $E(\mathbf{x})$ on the number of steps (assuming that E is integer-valued positive function). Murota proves that the algorithm in [21] will terminate in at most $O(nK)$ steps assuming a particular tie-breaking rule for choosing between multiple minimizers. (Here $K = \max_u(K_u^{\max} - K_u^{\min} + 1)$ denotes the maximum range of variables).

We now prove that our primal algorithm will terminate in at most $2K$ steps. Clearly, in some cases this bound is tight. Indeed, consider a function with constant ranges $[0, K-1]$ such that there is unique global minimum $\mathbf{x}^*$ with $(x_u^*, x_v^*) = (0, K-1)$ for some nodes u, v. If we choose initial configuration $\mathbf{x}$ with $(x_u, x_v) = (K-1, 0)$ then global minimum cannot be reached before $2K$ steps.

Our proof relies on the properties of *submodular functions*. The next section gives some background on such functions. It also introduces some additional notation.

2.1 Submodular functions

Suppose that set $\mathcal{X}$ is a *distributive lattice*, i.e. there are binary operations $\vee, \wedge : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$ satisfying certain axioms [22]. Function $E : \mathcal{X} \to \mathbb{R}$ is called *submodular* if for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ it satisfies

$$E(\mathbf{x}) + E(\mathbf{y}) \geq E(\mathbf{x} \vee \mathbf{y}) + E(\mathbf{x} \wedge \mathbf{y})$$

In our case operations $\vee, \wedge$ correspond to taking component-wise maximum and minimum, respectively:

$$(\mathbf{x} \vee \mathbf{y})_u = \max\{x_u, y_u\}\ , \quad (\mathbf{x} \wedge \mathbf{y})_u = \min\{x_u, y_u\} \qquad \forall\, \mathbf{x}, \mathbf{y} \in \mathcal{X}$$

It is easy to check that convex MRF functions are submodular.

Let $OPT(E)$ be the set of global minima of submodular function E; by definition, $OPT(E) \subseteq \mathcal{X}$. The following properties are well-known [22]:

(P1) For any $\mathbf{y} \in \mathcal{X}, \mathbf{x}^* \in OPT(E)$ there holds $E(\mathbf{y} \vee \mathbf{x}^*) \leq E(\mathbf{y})$, $E(\mathbf{y} \wedge \mathbf{x}^*) \leq E(\mathbf{y})$.

(P2) $OPT(E)$ is closed under operations $\vee, \wedge$ (and thus it's a distributive lattice).

(P3) There exist unique minimal configuration $\mathbf{x}^{\min} \in OPT(E)$ and maximal configuration $\mathbf{x}^{\max} \in OPT(E)$ s.t. $\mathbf{x}^{\min} \leq \mathbf{x} \leq \mathbf{x}^{\max}$ for any $\mathbf{x} \in OPT(E)$.

Proof. By definition of submodularity, we have

$$E(\mathbf{y}) \geq E(\mathbf{y} \vee \mathbf{x}^*) + E(\mathbf{y} \wedge \mathbf{x}^*) - E(\mathbf{x}^*)$$

The first two terms on the RHS are greater or equal than $E(\mathbf{x}^*)$. This establishes property (P1). If we choose configuration $\mathbf{y}$ to be a global minimum of E ($\mathbf{y} \in OPT(E)$) then we immediately get property (P2). Finally, (P3) is a property of any distributive lattice. □

The last property will allow us to define quantities $\rho^+(\mathbf{x})$ and $\rho^-(\mathbf{x})$ for configuration $\mathbf{x} \in \mathcal{X}$. These quantities will play a crucial role in our analysis.

First, let us define sets

$$\mathcal{X}^+ = \{\mathbf{y} \in \mathcal{X} \mid \mathbf{y} \geq \mathbf{x}\}\ , \quad \mathcal{X}^- = \{\mathbf{y} \in \mathcal{X} \mid \mathbf{y} \leq \mathbf{x}\}$$

It is easy to verify that $\mathcal{X}^+$ is a distributive lattice. Therefore, restriction of E onto set $\mathcal{X}^+$ is a submodular function $E^+ : \mathcal{X}^+ \to \mathbb{R}$. Let $\mathbf{x}^+$ be the minimal configuration in $OPT(E^+)$. By construction, $\mathbf{x}^+ \geq \mathbf{x}$ and $\mathbf{x}^+ \in \mathcal{X}$. Quantity $\rho^+(\mathbf{x})$ is now defined as

$$\rho^+(\mathbf{x}) = ||\mathbf{x}^+ - \mathbf{x}||_\infty = \max_{u \in \mathcal{V}}\{x_u^+ - x_u\}$$

Similarly, let E^- be the restriction of E onto set $\mathcal{X}^-$, and let $\mathbf{x}^-$ be the maximal configuration in $OPT(E^-)$. By construction, $\mathbf{x}^- \leq \mathbf{x}$ and $\mathbf{x}^- \in \mathcal{X}$. We define

$$\rho^-(\mathbf{x}) = ||\mathbf{x} - \mathbf{x}^-||_\infty = \min_{u \in \mathcal{V}}\{x_u - x_u^-\}$$

2.2 Complexity of the primal algorithm

The analysis of the algorithm is based on the theorem below. We use the following notation in the theorem: for configurations $\mathbf{x}, \mathbf{y}$ we define

$$\mathcal{M}^{\rm up}(\mathbf{x},\mathbf{y}) = \{\mathbf{z} \in \mathcal{M}^{\rm up}(\mathbf{x}) \mid \mathbf{z} \le \mathbf{y}\}\ , \quad \mathcal{M}^{\rm down}(\mathbf{x},\mathbf{y}) = \{\mathbf{z} \in \mathcal{M}^{\rm down}(\mathbf{x}) \mid \mathbf{z} \ge \mathbf{y}\}$$

Theorem 1. *1. Suppose that $\mathbf{y} \in \mathcal{M}^{\rm up}(\mathbf{x})$.*

(a) If $E(\mathbf{y}) = \min\{E(\mathbf{z}) \mid \mathbf{z} \in \mathcal{M}^{\rm up}(\mathbf{x},\mathbf{y})\}$ then $\rho^+(\mathbf{y}) \le \rho^+(\mathbf{x})$, $\rho^-(\mathbf{y}) \le \rho^-(\mathbf{x})$.

(b) If $E(\mathbf{y}) = \min\{E(\mathbf{z}) \mid \mathbf{z} \in \mathcal{M}^{\rm up}(\mathbf{x})\}$ and $\rho^+(\mathbf{x}) > 0$ then $\rho^+(\mathbf{y}) < \rho^+(\mathbf{x})$.

2. Suppose that $\mathbf{y} \in \mathcal{M}^{\rm down}(\mathbf{x})$.

(a) If $E(\mathbf{y}) = \min\{E(\mathbf{z}) \mid \mathbf{z} \in \mathcal{M}^{\rm down}(\mathbf{x},\mathbf{y})\}$ then $\rho^+(\mathbf{y}) \le \rho^+(\mathbf{x})$, $\rho^-(\mathbf{y}) \le \rho^-(\mathbf{x})$.

(b) If $E(\mathbf{y}) = \min\{E(\mathbf{z}) \mid \mathbf{z} \in \mathcal{M}^{\rm down}(\mathbf{x})\}$ and $\rho^-(\mathbf{x}) > 0$ then $\rho^-(\mathbf{y}) < \rho^-(\mathbf{x})$.

3. If $\rho^+(\mathbf{x}) = \rho^-(\mathbf{x}) = 0$, then $\mathbf{x}$ is a global minimum of E.

Using this theorem, it is not difficult to see that the primal algorithms terminates in at most $2K$ steps and yields a global minimum upon termination. Indeed, we have $\rho^+(\mathbf{x}) \le K-1$, $\rho^-(\mathbf{x}) \le K-1$ in the beginning. Quantities $\rho^+(\mathbf{x})$ and $\rho^-(\mathbf{x})$ never increase according to 1(a) and 2(a). If $\rho^+(\mathbf{x}) > 0$ in the beginning of procedure UP then $\rho^+(\mathbf{x})$ will decrease (part 1(a)). Note that in this case we must have $E(\mathbf{y}) < E(\mathbf{x})$, otherwise configuration $\mathbf{y} = \mathbf{x}$ could be a valid outcome of the procedure, which contradicts to condition $\rho^+(\mathbf{y}) < \rho^+(\mathbf{x})$. Therefore, flag SuccessUp will remain false. If $\rho^+(\mathbf{x}) = 0$ then $\mathbf{x}^+ = \mathbf{x}$, so the energy cannot decrease in the procedure UP. Thus, SuccessUp will be set to true. Similar argumentation holds for procedure DOWN.

Note that it is easy to construct initial configuration $\mathbf{x}$ such that $\rho^+(\mathbf{x}) + \rho^-(\mathbf{x}) \le K-1$. For example, we could take $\mathbf{x}$ to be the minimal configuration in $\mathcal{X}$ (i.e. $x_u = K_u^{\min}$ for all nodes u). Then $\rho^-(\mathbf{x}) = 0$, therefore the algorithm will then terminate in at most K steps without using procedure DOWN.

In order to prove the theorem, we will need the following lemma.

Lemma 2. *1. Suppose that $\rho^+(\mathbf{x}) > 0$. Define configuration $\mathbf{x}^{\rm up}$ as follows:*

$$x_u^{\rm up} = \max\{x_u, x_u^+ - \rho^+(\mathbf{x}) + 1\}$$

Then $\mathbf{x}^{\rm up} \in \mathcal{M}^{\rm up}(\mathbf{x})$ and $E(\mathbf{x}^{\rm up}) < E(\mathbf{x})$.

2. Suppose that $\rho^-(\mathbf{x}) > 0$. Define configuration $\mathbf{x}^{\rm down}$ as follows:

$$x_u^{\rm down} = \min\{x_u, x_u^- + \rho^-(\mathbf{x}) - 1\}$$

Then $\mathbf{x}^{\rm down} \in \mathcal{M}^{\rm down}(\mathbf{x})$ and $E(\mathbf{x}^{\rm down}) < E(\mathbf{x})$.

A proof of this lemma is given in the Appendix; we use the fact that convex MRF functions are $L^\natural$-convex. In the remainder of this section we prove the theorem using lemma 2 and properties of submodular functions. The following facts will be useful (they follow directly from properties (P1)-(P3) and definitions of $\mathbf{x}^+$ and $\mathbf{x}^-$).

(P^+) If $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, $\mathbf{y} \ge \mathbf{x}$ then

- There holds $E(\mathbf{y} \vee \mathbf{x}^+) \le E(\mathbf{y})$, $E(\mathbf{y} \wedge \mathbf{x}^+) \le E(\mathbf{y})$.
- If $E(\mathbf{y}) \le E(\mathbf{x}^+)$ then $E(\mathbf{y}) = E(\mathbf{x}^+)$ and $\mathbf{y} \ge \mathbf{x}^+$.

(P^-) If $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, $\mathbf{y} \le \mathbf{x}$ then

- There holds $E(\mathbf{y} \vee \mathbf{x}^-) \le E(\mathbf{y})$, $E(\mathbf{y} \wedge \mathbf{x}^-) \le E(\mathbf{y})$.
- If $E(\mathbf{y}) \le E(\mathbf{x}^-)$ then $E(\mathbf{y}) = E(\mathbf{x}^-)$ and $\mathbf{y} \le \mathbf{x}^-$.

We now proceed with the proof of theorem 1. We consider only parts 1 and 3; part 2 is completely analogous to part 1.

First statement of part 1(a): $\rho^+(\mathbf{y}) \le \rho^+(\mathbf{x})$ Let us prove that $\mathbf{y}^+ \le \mathbf{x}^+ \vee \mathbf{y}$. This will imply the desired result since for any node u

$$y_u^+ - y_u \le \max\{x_u^+, y_u\} - y_u = \max\{x_u^+ - y_u, 0\} \le \max\{x_u^+ - x_u, 0\} = x_u^+ - x_u$$

It is easy to see that $\mathbf{x}^+ \wedge \mathbf{y} \in \mathcal{M}^{\text{up}}(\mathbf{x}, \mathbf{y})$, therefore by assumption $E(\mathbf{y}) \le E(\mathbf{x}^+ \wedge \mathbf{y})$. Let us denote $\mathbf{z} = \mathbf{x}^+ \vee \mathbf{y}$. From submodularity of E we get

$$E(\mathbf{z}) \le E(\mathbf{x}^+) + E(\mathbf{y}) - E(\mathbf{x}^+ \wedge \mathbf{y}) \le E(\mathbf{x}^+)$$

There holds $\mathbf{z} \ge \mathbf{x}$, so from construction of $\mathbf{x}^+$ we must have $E(\mathbf{z}) = E(\mathbf{x}^+)$.

Since $\mathbf{y}^+ \ge \mathbf{x}$, by property (P$^+$) we have $E(\mathbf{y}^+) \ge E(\mathbf{x}^+)$. Therefore, $E(\mathbf{y}^+) \ge E(\mathbf{z})$. There holds $\mathbf{z} \ge \mathbf{y}$; using property (P$^+$) again, we obtain that $\mathbf{y}^+ \le \mathbf{z}$, as desired.

Second statement of part 1(a): $\rho^-(\mathbf{y}) \le \rho^-(\mathbf{x})$ Let us show first that $\mathbf{y}^- \ge \mathbf{x}^-$. We have $\mathbf{x}^- \wedge \mathbf{y}^- \le \mathbf{x}$, therefore $E(\mathbf{x}^- \wedge \mathbf{y}^-) \ge E(\mathbf{x}^-)$. Let us denote $\mathbf{z} = \mathbf{x}^- \vee \mathbf{y}^-$. From submodularity of E we get

$$E(\mathbf{z}) \le E(\mathbf{x}^-) + E(\mathbf{y}^-) - E(\mathbf{x}^- \wedge \mathbf{y}^-) \le E(\mathbf{y}^-)$$

We also have $\mathbf{z} \le \mathbf{y}$. Using property (P$^-$) and definition of $\mathbf{z}$ we obtain $\mathbf{y}^- \ge \mathbf{z} \ge \mathbf{x}^-$.

Now suppose that $\rho^-(\mathbf{y}) \ge d+1$ where $d = \rho^-(\mathbf{x})$. We will show next that this implies $\mathbf{y}^{\text{down}} \ge \mathbf{x}$, so $\mathbf{y}^{\text{down}} \in \mathcal{M}^{\text{up}}(\mathbf{x}, \mathbf{y})$. This leads to a contradiction: by theorem's assumption, $E(\mathbf{y}) \le E(\mathbf{y}^{\text{down}})$, however according to lemma 2 there holds $E(\mathbf{y}^{\text{down}}) < E(\mathbf{y})$.

We need to show that for any node u we have $y_u^{\text{down}} \ge x_u$. If $y_u^{\text{down}} = y_u$ then this clearly holds. Suppose that $y_u^{\text{down}} = y_u^- + \rho^-(\mathbf{y}) - 1$. Then

$$y_u^{\text{down}} \ge y_u^- + d \ge x_u^- + d \ge x_u$$

as desired.

Part 1(b) Suppose that $\rho(\mathbf{y}) \ge d$ where $d = \rho^+(\mathbf{x}) > 0$. We will show next that this implies $\mathbf{y}^{\text{up}} \in \mathcal{M}^{\text{up}}(\mathbf{x})$. This leads to a contradiction: by theorem's assumption, $E(\mathbf{y}) \le E(\mathbf{y}^{\text{up}})$, however according to lemma 2 there holds $E(\mathbf{y}^{\text{up}}) < E(\mathbf{y})$.

We need to show that for any node u we have $y_u^{\text{up}} \le x_u + 1$. If $y_u^{\text{up}} = y_u$ then this clearly holds. Suppose that $y_u^{\text{up}} = y_u^+ - \rho^+(\mathbf{y}) + 1$. Then

$$y_u^{\text{up}} \le y_u^+ - d + 1 \le \max\{x_u^+, y_u\} - d + 1 \le \max\{x_u + d, x_u + 1\} - d + 1 \le x_u + 1$$

as desired. (Note that we used the fact that $\mathbf{y}^+ \le \mathbf{x}^+ \vee \mathbf{y}$ proved earlier).

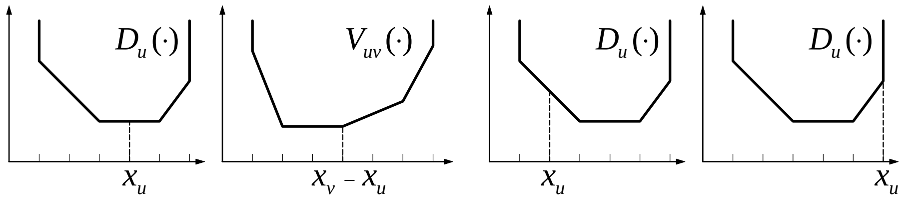
Part 3 Suppose that $\rho^+(\mathbf{x}) = \rho^-(\mathbf{x}) = 0$, so $\mathbf{x}^+ = \mathbf{x}^- = \mathbf{x}$. Let $\mathbf{x}^*$ be a global minimum of E. We have $\mathbf{x} \vee \mathbf{x}^* \ge \mathbf{x}$, $\mathbf{x} \wedge \mathbf{x}^* \le \mathbf{x}$, therefore $E(\mathbf{x} \vee \mathbf{x}^*) \ge E(\mathbf{x})$, $E(\mathbf{x} \wedge \mathbf{x}^*) \ge E(\mathbf{x})$. From submodularity of E we get

$$E(\mathbf{x}^*) \ge E(\mathbf{x} \vee \mathbf{x}^*) + E(\mathbf{x} \wedge \mathbf{x}^*) - E(\mathbf{x}) \ge E(\mathbf{x}) + E(\mathbf{x}) - E(\mathbf{x}) = E(\mathbf{x})$$

Therefore, $\mathbf{x}$ is a global minimum of E.

3 Primal-dual algorithm

In order to describe the second algorithm, we need to review the dual formulation to convex MRF minimization problem. This is done in section 3.1. Based on this, we then present our primal-dual algorithm in section 3.2. It can be viewed as an extension of the primal algorithm. It also uses procedures UP and DOWN; however, during these procedures the algorithm updates not only primal variables $\mathbf{x}$ but also dual variables, namely *flow*. After describing the algorithm we analyze its properties in section 3.3.



(a) (b) (c) (d)

Figure 2: **Complementary slackness conditions.** Any optimal primal-dual pair $(\mathbf{x}, f)$ must satisfy conditions illustrated in (a,b). During the algorithm, complementary slackness condition for edges shown in (b) is always satisfied. However, this condition may be violated for some nodes (possible cases are shown in (a,c,d)).

3.1 Flow and the dual problem

The dual to convex MRF minimization problem is well-known (see [1], for example). It involves *flow* which is a mapping $f : \mathcal{V} \cup \mathcal{A} \to \mathbb{R}$ satisfying

$$\begin{aligned} f_{uv} &= -f_{vu} && \forall\,(u,v) \in \mathcal{E} && \text{(antisymmetry)} \\ f_u &= \sum_{(u,v)\in\mathcal{E}} f_{uv} && \forall\, u \in \mathcal{V} && \text{(flow conservation)} \end{aligned}$$

(Recall that $\mathcal{A}$ denotes the set of *directed* edges in the graph.) Given flow f, we can define energy function $E^f(\mathbf{x})$ as follows:

$$E^f(\mathbf{x}) = \sum_{u\in\mathcal{V}} D_u^f(x_u) + \sum_{(u,v)\in\mathcal{E}} V_{uv}^f(x_v - x_u) \tag{2}$$

where

$$\begin{aligned} D_u^f(z) &= D_u(z) + f_u z && \forall\, z \in \mathbb{Z} \\ V_{uv}^f(z) &= V_{uv}(z) + f_{uv} z && \forall\, z \in \mathbb{Z} \end{aligned}$$

It is not difficult to check that for any flow f functions E^f and E are the same, i.e. $E^f(\mathbf{x}) = E(\mathbf{x})$ for any configuration $\mathbf{x}$. Furthermore, terms $D_u^f(\cdot)$ are $V_{uv}^f(\cdot)$ are convex. Following terminology used in machine learning [25], we say that expression (2) is a *reparameterization* of (1).

For a flow f let us define its *value* as

$$value(f) = \sum_{u\in\mathcal{V}} \min_{z\in\mathbb{Z}} D_u^f(z) + \sum_{(u,v)\in\mathcal{E}} \min_{z\in\mathbb{Z}} V_{uv}^f(z)$$

Clearly, $value(f)$ is a lower bound on energy $E^f(\mathbf{x})$ (and, thus, on $E(\mathbf{x})$):

$$value(f) \le E(\mathbf{x}) \qquad \forall\, f, \mathbf{x} \tag{3}$$

This is a statement of *weak duality.* It turns out that strong duality holds as well, as the following lemma shows.

Lemma 3. *(a) Let f^* be a flow that maximizes $value(f)$, and let $\mathbf{x}^*$ be a global minimum of energy $E(\mathbf{x})$. Then*

$$value(f^*) = E(\mathbf{x}^*)$$

(b) Configuration $\mathbf{x}$ and feasible flow f are optimal primal-dual solutions if and only if the following complementary slackness conditions hold:

$$D_u^f(x_u) = \min_{z\in\mathbb{Z}} D_u^f(z) \qquad \forall\, u \in \mathcal{V} \tag{4a}$$

$$V_{uv}^f(x_v - x_u) = \min_{z\in\mathbb{Z}} V_{uv}^f(z) \qquad \forall\, (u,v) \in \mathcal{E} \tag{4b}$$

Input: initial configuration $\mathbf{x} \in \mathcal{X}$.

1. INITIALIZE-FLOW (updates f)
2. Set SuccessUp := false, SuccessDown := false.
3. Do UP or DOWN in any order until SuccessUp = SuccessDown = true:

   UP (do only if SuccessUp is false):
   - MAXFLOW-UP (updates $\mathbf{x}$ and f)
   - If $\mathcal{V}^+(\mathbf{x}, f) = \emptyset$, set SuccessUp := true
   - Optional: DIJKSTRA-UP (updates $\mathbf{x}$)

   DOWN (do only if SuccessDown is false):
   - MAXFLOW-DOWN (updates $\mathbf{x}$ and f)
   - If $\mathcal{V}^-(\mathbf{x}, f) = \emptyset$, set SuccessUp := true
   - Optional: DIJKSTRA-DOWN (updates $\mathbf{x}$)

4. Optional: DIJKSTRA-DOWN; set $\mathbf{x}^{\min} := \mathbf{x}$.
5. Optional: DIJKSTRA-UP; set $\mathbf{x}^{\max} := \mathbf{x}$.

Figure 3: **Primal-dual algorithm.** *See text for description of different procedures. Upon termination* $\mathbf{x}$ *is a global minimizer of the energy,* $\mathbf{x}^{\min}$ *is the minimal minimizer,* $\mathbf{x}^{\max}$ *is the maximal minimizer, and* f *is an optimal flow.*

A proof of the lemma can be found in [1], for example. Alternatively, it can be deduced from our results. We will prove that our algorithm is guaranteed to find pair $(\mathbf{x}, f)$ satisfying conditions (4). The lemma will then follow from the weak duality (formula (3)).

Conditions (4) are illustrated in Fig. 2(a,b). Note that since terms $D_u^f(\cdot)$ and $V_{uv}^f(\cdot)$ are convex, these conditions are equivalent to

$$\text{grad}\, D_u^f(x_u - 0) \le 0 \qquad \text{grad}\, D_u^f(x_u + 0) \ge 0 \tag{4a$'$}$$

$$\text{grad}\, V_{uv}^f(x_v - x_u - 0) \le 0 \qquad \text{grad}\, V_{uv}^f(x_v - x_u + 0) \ge 0 \tag{4b$'$}$$

where we introduced the following notation for a function $g : \mathbb{Z} \to \mathbb{R} \cup \{+\infty\}$:

$$\text{grad}\, g(z - 0) = g(z) - g(z - 1)\ , \quad \text{grad}\, g(z + 0) = g(z + 1) - g(z)$$

3.2 Algorithm

We are now ready to present our primal-dual algorithm. It maintains configuration $\mathbf{x} \in \mathcal{X}$ and feasible flow f. An important property of the algorithm is that the complementary slackness condition (4b) is always satisfied. However, condition (4a) for nodes may not hold. Therefore, we can have cases shown in Fig. 2(c,d). It is convenient to introduce the following notation for sets of nodes violating the complementary slackness conditions:

$$\mathcal{V}^+(\mathbf{x}, f) = \{u \in \mathcal{V} \mid \text{grad}\, D_u^f(x_u + 0) < 0\}$$
$$\mathcal{V}^-(\mathbf{x}, f) = \{u \in \mathcal{V} \mid \text{grad}\, D_u^f(x_u - 0) > 0\}$$

To simplify notation, for zero flow we denote $\mathcal{V}^+(\mathbf{x}) = \mathcal{V}^+(\mathbf{x}, 0)$ and $\mathcal{V}^-(\mathbf{x}) = \mathcal{V}^-(\mathbf{x}, 0)$. Fig. 2(c) corresponds to a node in $\mathcal{V}^+(\mathbf{x})$, while 2(d) corresponds to node in $\mathcal{V}^-(\mathbf{x})$. Note that since terms $D_u^f(\cdot)$ are convex, a node cannot be in both sets simultaneously. Furthermore, condition (4a) holds if and only if sets $\mathcal{V}^+(\mathbf{x}, f)$ and $\mathcal{V}^-(\mathbf{x}, f)$ are empty.

The structure of the algorithm is shown in Fig. 3. We now give details of different procedures. Some of them involve modifying flow f. For simplicity of notation we will make the following convention: after every modification of f expression (1) is replaced with reparameterization (2), and the flow is set to zero[2]. This means that in the beginning of each operation the input flow is zero.

[2]For implementation it is also possible to keep original terms $D(\cdot)$, $V(\cdot)$ and flow f. When calling a particular procedure, we first compute terms $D^f(\cdot)$, $V^f(\cdot)$ and supply them to the procedure. If the output of the operation is flow f' then it is added to f, so $f + f'$ becomes the new flow.

INITIALIZE-FLOW Its goal is to set flow f so that condition (4b′) is satisfied for every edge $(u,v) \in \mathcal{E}$. Since $\text{grad}\, V^f_{uv}(x_v - x_u \pm 0) = \text{grad}\, V_{uv}(x_v - x_u \pm 0) + f_{uv}$, a necessary and sufficient condition for flow f_{uv} is

$$-\text{grad}\, V_{uv}(x_v - x_u + 0) \leq f_{uv} \leq -\text{grad}\, V_{uv}(x_v - x_u - 0)$$

After setting f_{uv} values f_u are computed from the flow conservation constraint.

MAXFLOW-UP This operation is similar to procedure UP of the primal algorithm, except that it modifies not only configuration $\mathbf{x}$ but also flow f. We will show later that it tries to move towards satisfying complementary slackness conditions (4a). In particular, sets $\mathcal{V}^+(\mathbf{x}, f)$ and $\mathcal{V}^-(\mathbf{x}, f)$ will not grow.

The procedure can be summarized as follows. First, we construct directed weighted graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{A}}, \hat{c})$ with nodes $\hat{\mathcal{V}} = \mathcal{V} \cup \{s,t\}$ where s and t are called the *source* and the *sink*, respectively. We then compute maximum flow $\hat{f}$ from the s to t, i.e. solve the problem

$$\begin{array}{ll} \max\limits_{f} & \sum_{(s\to u)\in\hat{\mathcal{A}}} \hat{f}_{su} \\ \text{subject to} & \begin{cases} \hat{f}_{uv} \leq \hat{c}_{uv}\ , \quad \hat{f}_{uv} = -\hat{f}_{vu} & \text{for all } (u \to v) \in \hat{\mathcal{A}} \\ \sum_{(u\to v)\in\hat{\mathcal{A}}} \hat{f}_{uv} = 0 & \text{for all } v \in \hat{\mathcal{V}} - \{s,t\} \end{cases} \end{array}$$

Finally, we update dual variables f based on flow $\hat{f}$, and we also compute new configuration $\mathbf{y}$ based on minimum s-t cut in the graph. As we will show later, the output configuration $\mathbf{y}$ is the same as in the procedure UP of the primal algorithm, i.e. $\mathbf{y} = \arg\min\{E(\mathbf{z}) \mid \mathbf{z} \in \mathcal{M}^{\text{up}}(\mathbf{x})\}$.

The table below shows graph construction as well as correspondence between flows $\hat{f}$ and f. For every arc from the source or to the sink we also add the reverse arc with zero capacity. Note that capacities for arcs $(u \to v)$, $(v \to u)$ are non-negative because of conditions (4b′). It is easy to see that the output flow f satisfies antisymmetry and flow conservation constraints, since the same holds for flow $\hat{f}$.

arc a in $\hat{\mathcal{A}}$	flow $\hat{f}_a$ corresponds to	capacity $\hat{c}_a$	for
$(u \to v)$	f_{uv}	$-\text{grad}\, V_{uv}(x_v - x_u - 0)$	$(u,v) \in \mathcal{E}$
$(v \to u)$	f_{vu}	$\text{grad}\, V_{uv}(x_v - x_u + 0)$	
$(s \to u)$	f_u	$-\text{grad}\, D_u(x_u + 0)$	$u \in \mathcal{V}^+(\mathbf{x})$
$(u \to t)$	$-f_u$	$\text{grad}\, D_u(x_u + 0)$	$u \notin \mathcal{V}^+(\mathbf{x})$

New configuration $\mathbf{y}$ is computed from minimum s-t cut (S,T) as follows: for node $u \in \mathcal{V}$ we set $y_u = x_u + 1$ if $u \in S$, and $y_u = x_u$ otherwise.

MAXFLOW-DOWN This operation is analogous to the previous one. Graph construction and correspondence between flows $\hat{f}$ and f are shown below.

arc a in $\hat{\mathcal{A}}$	flow $\hat{f}_a$ corresponds to	capacity $\hat{c}_a$	for
$(u \to v)$	f_{uv}	$-\text{grad}\, V_{uv}(x_v - x_u - 0)$	$(u,v) \in \mathcal{E}$
$(v \to u)$	f_{vu}	$\text{grad}\, V_{uv}(x_v - x_u + 0)$	
$(s \to u)$	f_u	$-\text{grad}\, D_u(x_u - 0)$	$u \notin \mathcal{V}^-(\mathbf{x})$
$(u \to t)$	$-f_u$	$\text{grad}\, D_u(x_u - 0)$	$u \in \mathcal{V}^-(\mathbf{x})$

New configuration $\mathbf{y}$ is computed from minimum s-t cut (S,T) as follows: for node $u \in \mathcal{V}$ we set $y_u = x_u$ if $u \in S$, and $y_u = x_u - 1$ otherwise.

DIJKSTRA-UP This operation is optional. It does not affect the worst-case complexity of the algorithm, but may improve empirical performance. In this procedure we fix flow and compute maximal configuration $\mathbf{y} \geq \mathbf{x}$ such that functions $D_u(\cdot)$ are non-increasing on $[x_u, y_u]$ and complementary slackness conditions (4b) hold. If we denote $d_u = y_u - x_u \geq 0$, then these constraints are equivalent to

$$\begin{aligned} d_u &\leq d_u^{\max} = \max\{d \in \mathbb{Z} \mid D_u(x_u + d) \leq \ldots \leq D_u(x_u + 1) \leq D_u(x_u)\} \\ d_v - d_u &\leq d_{uv}^{\max} = \max\{d \in \mathbb{Z} \mid V_{uv}(x_v - x_u + d) = V_{uv}(x_v - x_u)\} \\ d_u - d_v &\leq d_{vu}^{\max} = \max\{d \in \mathbb{Z} \mid V_{uv}(x_v - x_u - d) = V_{uv}(x_v - x_u)\} \end{aligned}$$

It is well-known [3] that maximal labeling $\mathbf{d}$ satisfying these constraints can be computed efficiently using Dijkstra's algorithm.

DIJKSTRA-DOWN This operation is similar to the previous one: we compute minimal configuration $\mathbf{y} \leq \mathbf{x}$ such that functions $D_u(\cdot)$ are non-decreasing on $[y_u, x_u]$ and complementary slackness conditions (4b) hold. If we denote $d_u = x_u - y_u \geq 0$, then these constraints are equivalent to

$$\begin{aligned} d_u &\leq d_u^{\max} = \max\{d \in \mathbb{Z} \mid D_u(x_u - d) \leq \ldots \leq D_u(x_u - 1) \leq D_u(x_u)\} \\ d_v - d_u &\leq d_{uv}^{\max} = \max\{d \in \mathbb{Z} \mid V_{uv}(x_v - x_u - d) = V_{uv}(x_v - x_u)\} \\ d_u - d_v &\leq d_{vu}^{\max} = \max\{d \in \mathbb{Z} \mid V_{uv}(x_v - x_u + d) = V_{uv}(x_v - x_u)\} \end{aligned}$$

As before, maximal labeling $\mathbf{d}$ (corresponding to minimal labeling $\mathbf{y}$) can be computed using Dijkstra's algorithm.

3.3 Analysis of the algorithm

First we analyze the behaviour of the algorithm without procedures DIJKSTRA-UP and DIJKSTRA-DOWN. In the theorem below we assume that input pair $(\mathbf{x}, 0)$ satisfies complementary slackness conditions (4b).

Theorem 4. *1. Let $(\mathbf{y}, f)$ be the output of* MAXFLOW-UP *applied to $(\mathbf{x}, 0)$. Then*

(a) Complementary slackness conditions (4b) hold for $(\mathbf{y}, f)$.

(b) $\mathbf{y} = \arg\min\{E(\mathbf{z}) \mid \mathbf{z} \in \mathcal{M}^{\mathrm{up}}(\mathbf{x})\}$.

(c) There holds $\mathcal{V}^+(\mathbf{y}, f) \subseteq \mathcal{V}^+(\mathbf{x})$ and $\mathcal{V}^-(\mathbf{y}, f) \subseteq \mathcal{V}^-(\mathbf{x})$.

(d) If $\rho^+(\mathbf{x}) = 0$, then $\mathcal{V}^+(\mathbf{y}, f) = \emptyset$.

2. Let $(\mathbf{y}, f)$ be the output of MAXFLOW-DOWN *applied to $(\mathbf{x}, 0)$. Then*

(a) Complementary slackness conditions (4b) hold for $(\mathbf{y}, f)$.

(b) $\mathbf{y} = \arg\min\{E(\mathbf{z}) \mid \mathbf{z} \in \mathcal{M}^{\mathrm{down}}(\mathbf{x})\}$.

(c) There holds $\mathcal{V}^+(\mathbf{y}, f) \subseteq \mathcal{V}^+(\mathbf{x})$ and $\mathcal{V}^-(\mathbf{y}, f) \subseteq \mathcal{V}^-(\mathbf{x})$.

(d) If $\rho^-(\mathbf{x}) = 0$, then $\mathcal{V}^-(\mathbf{y}, f) = \emptyset$.

Combining theorem 1, theorem 4 and the weak duality result (formula (3)) we can show that the algorithm terminates in at most $2K$ steps and yields optimal primal-dual pair $(\mathbf{x}, f)$ upon termination. Indeed, parts 1(a) and 2(a) imply that conditions (4b) always hold. After at most $K - 1$ steps of procedure UP quantity $\rho^+(\mathbf{x})$ becomes zero, therefore after at most K steps set $\mathcal{V}^+(\mathbf{x}, f)$ becomes empty. At this point flag SuccessUp is set to true, and set $\mathcal{V}^+(\mathbf{x}, f)$ will remain empty. Similar argumentation holds for procedure DOWN. When the algorithm terminates, sets $\mathcal{V}^+(\mathbf{x}, f)$ and $\mathcal{V}^-(\mathbf{x}, f)$ are empty, so complementary slackness conditions (4) are satisfied. The weak duality then implies that we have an optimal primal-dual pair $(\mathbf{x}, f)$.

This analysis remains valid even with procedures DIJKSTRA-UP or DIJKSTRA-DOWN, as follows from the theorem below.

Theorem 5. *Let* $\mathbf{y}$ *be the output of* DIJKSTRA-UP *or* DIJKSTRA-DOWN *applied to* $(\mathbf{x}, 0)$. *Then*

(a) Complementary slackness conditions (4b) hold for $(\mathbf{y}, 0)$.

(b) There holds $\mathcal{V}^+(\mathbf{y}) \subseteq \mathcal{V}^+(\mathbf{x})$ *and* $\mathcal{V}^-(\mathbf{y}) \subseteq \mathcal{V}^-(\mathbf{x})$.

(c) There holds $\rho^+(\mathbf{y}) \leq \rho^+(\mathbf{x})$ *and* $\rho^-(\mathbf{y}) \leq \rho^-(\mathbf{x})$.

It can be seen that if procedure DIJKSTRA-UP is applied to optimal pair $(\mathbf{x}, f)$ then the output configuration $\mathbf{y}$ is the maximal optimal configuration. Indeed, according to lemma 3 configuration $\mathbf{y}$ is optimal if and only if it satisfies

$$\begin{aligned} D_u(y_u) &= D_u(x_u) && \forall\, u \in \mathcal{V} \\ V_{uv}(y_v - y_u) &= V_{uv}(x_v - x_u) && \forall\, (u,v) \in \mathcal{E} \end{aligned}$$

For configurations $\mathbf{y} \geq \mathbf{x}$ this is equivalent to saying that functions $D_u(\cdot)$ are non-increasing on $[x_u, y_u]$ and complementary slackness conditions (4b) hold. By construction, DIJKSTRA-UP finds the maximal configuration satisfying these conditions. Similarly, we can show that applying DIJKSTRA-DOWN to optimal pair $(\mathbf{x}, f)$ yields the minimal optimal configuration.

We now turn to the proof of theorems 4 and 5. We omit a proof of part 2 of theorem 4 since it is very similar to that of part 1.

Theorem 4, part 1(a) From capacity constraints we get $f_{uv} \leq -\text{grad}\, V_{uv}(x_v - x_u + 0)$, $-f_{uv} \leq \text{grad}\, V_{uv}(x_v - x_u - 0)$, therefore

$$\text{grad}\, V^f_{uv}(x_v - x_u + 0) = f_{uv} + \text{grad}\, V_{uv}(x_v - x_u + 0) \leq 0$$

$$\text{grad}\, V^f_{uv}(x_v - x_u - 0) = f_{uv} + \text{grad}\, V_{uv}(x_v - x_u - 0) \geq 0$$

which implies that $V^f_{uv}(x_v - x_u) = \min_{z \in \mathbb{Z}} V^f_{uv}(z)$. Thus, if $y_v - y_u = x_v - x_u$ then the complementary slackness condition (4b) holds for edge (u, v). Let us consider the case $y_v - y_u = x_v - x_u + 1$. This can only happen when $u \in T$, $v \in S$, which means that edge $(v \to u)$ must be saturated. Therefore $-f_{uv} = \hat{f}_{vu} = \hat{c}_{vu} = \text{grad}\, V_{uv}(x_v - x_u + 0)$, so

$$\text{grad}\, V^f_{uv}(x_v - x_u + 0) = f_{uv} + \text{grad}\, V_{uv}(x_v - x_u + 0) = 0$$

$$V^f_{uv}(y_v - y_u) = V^f_{uv}(x_v - x_u + 1) = V^f_{uv}(x_v - x_u) = \min_{z \in \mathbb{Z}} V^f_{uv}(z)$$

The case $y_v - y_u = x_v - x_u - 1$ can be considered similarly.

Theorem 4, part 1(b) The rule described in procedure MAXFLOW-UP defines a one-to-one mapping between s-t cuts in graph $\hat{\mathcal{G}}$ and configurations $\mathbf{y} \in \mathcal{M}^{\text{up}}(\mathbf{x})$. The following sequence of equations shows that the cost of any cut (S, T) is equal to the energy of the corresponding configuration $\mathbf{y}$ plus a constant, thus establishing our claim. (Below we use indicator function $[\cdot]$ which is one if its argument is true, and zero otherwise.)

$$\begin{aligned} \mathit{cost}(S,T) &= \sum_{u \in \mathcal{V}^+(\mathbf{x})} \hat{c}_{su} \cdot [u \in T] + \sum_{u \notin \mathcal{V}^+(\mathbf{x})} \hat{c}_{ut} \cdot [u \in S] + \\ &+ \sum_{(u,v) \in \mathcal{E}} \{\hat{c}_{uv} \cdot [u \in S, v \in T] + \hat{c}_{vu} \cdot [u \in T, v \in S]\} = \\ &= \mathit{const} + \sum_{u \in \mathcal{V}} \text{grad}\, D_u(x_u + 0) \cdot [y_u = x_u + 1] + \\ &+ \sum_{(u,v) \in \mathcal{E}} \{-\text{grad}\, V_{uv}(x_v - x_u - 0) \cdot [y_v - y_u = x_v - x_u - 1] + \\ &\qquad + \text{grad}\, V_{uv}(x_v - x_u + 0) \cdot [y_v - y_u = x_v - x_u + 1]\} = \\ &= \mathit{const}' + \sum_{u \in \mathcal{V}} D_u(y_u) + \sum_{(u,v) \in \mathcal{E}} V_{uv}(y_v - y_u) = \mathit{const}' + E(\mathbf{y}) \end{aligned}$$

where we used the fact that

$$\text{grad}\, g(c + 0) \cdot [z = c + 1] = -g(c) + g(z)$$

for any $z \in \{c, c+1\}$, and

$$-\text{grad}\, g(c-0) \cdot [z = c-1] + \text{grad}\, g(c+0) \cdot [z = c+1] = -g(c) + g(z)$$

for any $z \in \{c-1, c, c+1\}$.

Theorem 4, part 1(c) We consider two possible cases.

- $u \in \mathcal{V}^+(\mathbf{x})$, i.e. $\text{grad}\, D_u(x_u + 0) < 0$. We need to show that $u \notin \mathcal{V}^-(\mathbf{y}, f)$. This holds since

$$\begin{aligned} \text{grad}\, D_u^f(y_u - 0) \quad &\leq \text{grad}\, D_u^f((x_u + 1) - 0) = \\ &= \text{grad}\, D_u^f(x_u + 0) = f_u + \text{grad}\, D_u(x_u + 0) \leq 0 \end{aligned}$$

  (The first inequality follows from $y_u \leq x_u + 1$ and convexity of $D^f(\cdot)$, and the last inequality follows from the capacity constraint: $f_u \leq \hat{c}_{su} = -\text{grad}\, D_u(x_u + 0)$.)

- $u \notin \mathcal{V}^+(\mathbf{x})$, i.e. $\text{grad}\, D_u(x_u + 0) \geq 0$. We have $0 \leq \hat{f}_u \leq \text{grad}\, D_u(x_u + 0)$, which implies $0 \geq f_u \geq -\text{grad}\, D_u(x_u + 0)$. The fact that $u \notin \mathcal{V}^+(\mathbf{y}, f)$ then follows from

$$\text{grad}\, D_u^f(y_u + 0) \geq \text{grad}\, D_u^f(x_u + 0) = f_u + \text{grad}\, D_u(x_u + 0) \geq 0$$

  Now suppose that $u \notin \mathcal{V}^-(\mathbf{x})$, i.e. $\text{grad}\, D_u(x_u - 0) \leq 0$. We need to show that $u \notin \mathcal{V}^-(\mathbf{y}, f)$. If $y_u = x_u$ then this follows from

$$\text{grad}\, D_u^f(y_u - 0) = f_u + \text{grad}\, D_u(x_u - 0) \leq 0$$

  If $y_u = x_u + 1$ then $u \in S$, so edge $(u \to t)$ must be saturated: $-f_u = \hat{f}_u = \text{grad}\, D_u(x_u + 0)$. Then

$$\text{grad}\, D_u^f(y_u - 0) = \text{grad}\, D_u^f(x_u + 0) = f_u + \text{grad}\, D_u(x_u + 0) = 0$$

Theorem 4, part 1(d) For nodes $u \notin \mathcal{V}^+(\mathbf{x})$ this follows from part (c). Let us consider node $u \in \mathcal{V}^+(\mathbf{x})$. Condition $\rho^+(\mathbf{x}) = 0$ means that $\mathbf{x} = \arg\min\{E(\mathbf{z}) \mid \mathbf{z} \in \mathcal{M}^{up}(\mathbf{x})\}$. Therefore, according to part 1(b) cut $(\{s\}, \mathcal{V} \cup \{t\})$ corresponding to configuration $\mathbf{x}$ is a cut with the smallest cost. Thus, edge $(s \to u)$ must be saturated: $f_u = \hat{f}_u = \hat{c}_{su} = -\text{grad}\, D_u(x_u + 0)$. This implies that

$$\text{grad}\, D_u^f(y_u + 0) \geq \text{grad}\, D_u^f(x_u + 0) = f_u + \text{grad}\, D_u(x_u + 0) = 0$$

so $u \notin \mathcal{V}^+(\mathbf{y}, f)$, as desired.

Theorem 5 We consider only procedure DIJKSTRA-UP; operation DIJKSTRA-DOWN is completely analogous. Parts (a) and (b) follow directly from the definition of the procedure. Let us prove part (c).

For $d \in \mathbb{Z}_+$ define configuration $\mathbf{x}^d$ as follows:

$$x_u^d = \min\{x_u + d, y_u\}$$

We have $\mathbf{x}^0 = \mathbf{x}$ and $\mathbf{x}^d = \mathbf{y}$ for sufficiently large d. Furthermore, for any $d \geq 0$ there holds $\mathbf{x}^{d+1} \in \mathcal{M}^{\text{up}}(\mathbf{x}^d)$.

Consider configuration $\mathbf{z} \in \mathcal{M}^{\text{up}}(\mathbf{x}^d, \mathbf{x}^{d+1})$. Since $x_u \leq z_u \leq x_u^{d+1} \leq y_u \leq x_u + d_u^{\max}$, it follows from the construction that $D_u(x_u^{d+1}) \leq D_u(z_u)$. Similarly, for every edge (u, v) there holds

$$(x_v^{d+1} - x_v) - (x_u^{d+1} - x_u) \leq (y_v^{d+1} - x_v) - (y_u^{d+1} - x_u) \leq d_{uv}^{\max}$$

so from the construction $V_{uv}(x_v^{d+1} - x_u^{d+1}) = V_{uv}(x_v - x_u) \leq V_{uv}(z_v - z_u)$. Since every term in the sum for $E(\mathbf{x}^{d+1})$ is the same or smaller than the corresponding term for $E(\mathbf{z})$, we obtain $E(\mathbf{x}^{d+1}) \leq E(\mathbf{z})$.

We can therefore apply part 1(a) of theorem 1. It implies that $\rho^+(\mathbf{x}^{d+1}) \leq \rho^+(\mathbf{x}^d)$ and $\rho^-(\mathbf{x}^{d+1}) \leq \rho^-(\mathbf{x}^d)$. This holds for any $d \geq 0$, which proves the desired result.

Figure 4: **Results of panoramic stitching.** First two columns: input images (courtesy of A. Zomet). Rectangles show the area of overlap. Last three columns: results corresponding to $\mathbf{x}^{\min}$, $\mathbf{x}^{\max}$ and $\mathbf{x}^{\text{av}}$, respectively (note that images are cropped). The additive constant is chosen as described in the text.

4 Experiments

We tested the speed of several algorithms on the panoramic image stitching application. Given two input images I^1 and I^2 defined on overlapping domains Ω^1 and Ω^2, the goal is two compute an output image without a visible seam. Levin et al. [20, 26] proposed several techniques for this problem. One of them, GIST1 algorithm under l_1 norm, was shown to outperform many other stitching methods. It involves minimizing the following function for each color channel:

$$E(\mathbf{x}) = \sum_{(u,v)\in\mathcal{E}} w^1_{uv}|(x_v - x_u) - (I^1_v - I^1_u)| + w^2_{uv}|(x_v - x_u) - (I^2_v - I^2_u)|$$

In other words, we want the gradient of image $\mathbf{x}$ to match gradients of images I^1 and I^2. Weights w^1_{uv} and w^2_{uv} were determined as follows. For edges in $\Omega^1 - \Omega^2$ we set $w^1_{uv} = 2, w^2_{uv} = 0$. For edges in $\Omega^2 - \Omega^1$ we set $w^1_{uv} = 0, w^2_{uv} = 2$. For all other edges we set $w^1_{uv} = w^2_{uv} = 1$. We used a 4-neighborhood system $\mathcal{E}$.

Since there are no unary terms, a global minimum is determined only up to an additive constant. Similar to [20, 26], we computed this constant so that median intensity of I^1 in Ω^1 matches that of the output image. This does not uniquely determines the solution, however - there are multiple optimal configurations $\mathbf{x}$ satisfying this requirement. Levin et al. do not discuss how to choose between them. We propose the following technique. We put constraints $x_u \in [0, K-1]$ on the variables where K is sufficiently large (e.g., 512). We then compute minimal global solution $\mathbf{x}^{\min}$, maximal global solution and $\mathbf{x}^{\max}$ and their average $\mathbf{x}^{\text{av}} = \left\lfloor \frac{\mathbf{x}^{\min}+\mathbf{x}^{\max}}{2} \right\rfloor$. It can be shown that $\mathbf{x}^{\text{av}}$ is an optimal configuration as well (it follows from the fact that function E is $L^\natural$-convex). Furthermore, these configurations have the minimum possible range (defined as $\max_u\{x_u\} - \min_u\{x_u\} + 1$). In our experiments it was very close to 256. Having a small range may be advantageous since intensities must be mapped to interval [0,255], so if the range is too large then some regions may become too dark or saturated.

Fig. 4 shows panoramas corresponding to configurations $\mathbf{x}^{\min}$, $\mathbf{x}^{\max}$ and $\mathbf{x}^{\text{av}}$. It can be seen that the last one looks significantly better than the other two (the overlap area is too dark in $\mathbf{x}^{\min}$ and too bright in $\mathbf{x}^{\max}$).

We compared the speed of three different algorithms. The first two are primal and primal-dual methods described in sections 2 and 3. An important issue is the choice of initial configuration $\mathbf{x}$. If, for example, we start with an optimal configuration, then algorithms will terminate in two steps. We tried two schemes. In the first one we initialized the algorithms with the following configuration: $x_u = I^1_u$ in region $\Omega^1 - \Omega^2$, $x_u = I^2_u$ in $\Omega^2 - \Omega^1$ and $x_u = \left\lfloor \frac{I^1_u + I^2_u}{2} \right\rfloor$ in $\Omega^1 \cap \Omega^2$. In the second scheme we used a two-stage process. First, we solve the problem in region Ω' which is two pixels larger than the overlap $\Omega^1 \cap \Omega^2$. (The initialization for this step is the same as before). The solution is then used for initializing the final stage. Pixels outside Ω' are initialized in the same way as before - either with image I^1 or with I^2. In our experiments the second stage for the primal-dual algorithm always terminated in three steps, which shows that the first stage computes a very good initialization. For both schemes we used maxflow algorithm of Boykov et al. [6] available at http://www.cs.cornell.edu/People/vnk/software.html (version 2.2).

The third technique that we tried is as follows. We converted the original problem to a minimum cost network flow problem. (Note that we did not enforce constraints $x_u \in [0, K-1]$). We then applied MCNF algorithm of Goldberg [12] available at http://www.avglab.com/andrew/soft.html (version 4.0). It has one free parameter, namely scaling factor; we set it to 32 (results for other factors were faster by at most one percent). Convex MRF problem can be converted to MCNF in many different ways. We used a transformation with the following property: if the initial configuration satisfied complementary slackness conditions, then so did the resulting MCNF problem. We initialized the algorithm as in the first scheme above; we found, however, that the performance was affected by the initial configuration less significantly. (If, for example, we initialized with an optimal configuration $\mathbf{x}^*$, then the algorithm was at most 30% faster). In all codes we used 32-bit integers.

Note that we did not test the cost scaling technique of Ahuja et al. [1]. It uses ideas similar to [12], but it works with a smaller graph. Therefore, [1] could potentially be faster than converting the problem to MCNF and then applying the method in [12]. However, in our application graph sizes would differ only slightly, and we argue that direct implementation of the technique in [1] is unlikely to beat the implementation in [12]. Indeed, the latter is highly optimized and includes many heuristics which significantly improve the empirical performance.

The table below shows running times in seconds on Pentium IV 3.2GHz processor (we measure the total time for 3 color channels). We used three datasets D0, D1 and D2 shown in Fig. 4. Their dimensions are 449×193 for D0 and 577×257 for D1 and D2. The percentages of overlap area are 4.9%, 10.0% and 6.9%, respectively. We also used scaled-down datasets D0-s, D1-s and D2-s (both X and Y dimensions are reduced by 2 times).

	D0-s	D1-s	D2-s	D0	D1	D2
primal, 1 stage	4.47	27.39	25.66	70.31	162.35	122.00
primal-dual, 1 stage	0.61	4.09	5.23	9.53	16.41	29.71
primal-dual, 2 stages	0.36	0.71	0.63	1.08	2.87	3.03
MCNF	0.90	1.79	1.50	5.13	14.56	13.09

In the two-stage primal-dual method maxflow algorithm took approximately 5-10% of total time, Dijkstra's algorithm - 30%, and the rest was spent on building graph for computing maxflow and reading flow back.

Not surprisingly, the primal-dual method significantly outperforms the primal algorithm. As for a relative performance of primal-dual and MCNF techniques, the situation depends on initial configuration. In the panoramic stitching problem we can obtain quickly a very good initialization, which makes the primal-dual algorithm a clear winner. In applications when it is not the case MCNF algorithm may be faster.

Appendix: proof of lemma 2

Our proof is based on the fact that convex MRF functions are $L^\natural$-convex. Two equivalent definitions of $L^\natural$-convexity for functions $E : \mathbb{Z}^n \to \mathbb{R} \cup \{+\infty\}$ are given below (see [22]).

Definition 6. *E is called $L^\natural$-convex if for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^n$ there holds*

$$E(\mathbf{x}) + E(\mathbf{y}) \geq E\left(\left\lceil \frac{\mathbf{x}+\mathbf{y}}{2} \right\rceil\right) + E\left(\left\lfloor \frac{\mathbf{x}+\mathbf{y}}{2} \right\rfloor\right)$$

where $\lceil \frac{\mathbf{x}+\mathbf{y}}{2} \rceil$ and $\lfloor \frac{\mathbf{x}+\mathbf{y}}{2} \rfloor$ denote, respectively, the integer vectors obtained from $\frac{\mathbf{x}+\mathbf{y}}{2}$ by component-wise round-up and round-down to the nearest integers.

Definition 6′. *E is called $L^\natural$-convex if it satisfies* translation submodularity *property for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^n$, $\alpha \in \mathbb{Z}_+$:*

$$E(\mathbf{x}) + E(\mathbf{y}) \geq E((\mathbf{x} - \alpha\mathbf{1}) \vee \mathbf{y}) + E((\mathbf{y} + \alpha\mathbf{1}) \wedge \mathbf{x})$$

(In both definitions it is understood that the inequality is satisfied if one of the terms $E(\mathbf{x})$, $E(\mathbf{y})$ is $+\infty$.)

We can also apply definitions above to functions $E : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ where $\mathcal{X} \subset \mathbb{Z}^n$. Then we need to extend E as follows: we set $E(\mathbf{x}) = +\infty$ if $\mathbf{x} \in \mathbb{Z}^n - \mathcal{X}$.

We now turn to the proof of the lemma. We consider only the first part; the second part is completely analogous. We need to show that $E(\mathbf{x}^{\text{up}}) < E(\mathbf{x})$. We can assume without loss of generality that configuration $\mathbf{x}$ is minimal in set $\mathcal{X}$. (If not, we can consider the restriction of E onto set $\mathcal{X}^+ = \{\mathbf{y} \in \mathcal{X} \,|\, \mathbf{y} \geq \mathbf{x}\}$, which is an $L^\natural$-convex function. Definition of $\mathbf{x}^{\text{up}}$ stays the same).

Let $d = \rho^+(\mathbf{x}) - 1 \geq 0$, and let $\mathcal{V}_0 = \{u \in \mathcal{V} \,|\, x^+_u - x_u = d + 1\}$. (From construction, $\mathcal{V}_0 \neq \emptyset$ and $0 \leq x^+_u - x_u \leq d$ for all $u \in \mathcal{V} - \mathcal{V}_0$.) It is easy to see that configuration $\mathbf{x}^{\text{up}}$ can be defined as follows: $\mathbf{x}^{\text{up}} = (\mathbf{x}^+ - d\,\mathbf{1}) \vee \mathbf{x}$.

Consider configuration $\mathbf{y} = (\mathbf{x} + d\,\mathbf{1}) \wedge \mathbf{x}^+$. It can be seen that $\mathbf{x} \leq \mathbf{y} \leq \mathbf{x}^+$ and $\mathbf{y} \neq \mathbf{x}^+$. (In particular, $y_u < x^+_u$ for $u \in \mathcal{V}_0$.) Since $\mathbf{x}^+$ is the minimal optimal configuration, there holds $E(\mathbf{x}^+) < E(\mathbf{y})$. Using translation submodularity property, we obtain

$$E(\mathbf{x}) + E(\mathbf{x}^+) \geq E(\mathbf{y}) + E(\mathbf{x}^{\text{up}})$$

Therefore, $E(\mathbf{x}^{\text{up}}) - E(\mathbf{x}) \leq E(\mathbf{x}^+) - E(\mathbf{y}) < 0$. This proves the first part of lemma 2.

References

[1] R. K. Ahuja, D. S. Hochbaum, and J. B. Orlin. Solving the convex cost integer dual network flow problem. *Management Science*, 49:7:950–964, 2003.

[2] R. K. Ahuja, D. S. Hochbaum, and J. B. Orlin. A cut based algorithm for the convex dual of the minimum cost network flow problem. *Algorithmica*, 39:3:189–208, April 2004.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[4] J. Bioucas-Dias and G. Valadão. Phase unwrapping via graph cuts. In *Pattern Recognition and Image Analysis: 2nd Iberian Conference (IbPRIA)*, LCNS. Springer, June 2005.

[5] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155 – 225, November 2002.

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9), September 2004.

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), November 2001.

[8] J. Darbon and M. Sigelle. A fast and exact algorithm for total variation minimization. In *Pattern Recognition and Image Analysis: 2nd Iberian Conference (IbPRIA)*, LCNS. Springer, June 2005.

[9] L. R. Ford and D. R. Fulkerson. A primal-dual algorithm for the capacitated hitchcock problem. *Naval Research Logistics Quarterly*, 4:47–54, 1957.

[10] L. R. Ford and D. R. Fulkerson. *Flows in networks.* Princeton Univ. Press, 1962.

[11] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *PAMI*, 6:721–741, 1984.

[12] A. V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *J. Algorithms*, 22:1–29, 1997.

[13] D. S. Hochbaum. An efficient algorithm for image segmentation, Markov Random Fields and related problems. *J. ACM*, 48:2:686–701, July 2001.

[14] D. S. Hochbaum and J. G. Shanthikumar. Convex separable optimization is not much harder than linear optimization. *J. ACM*, 37:843–862, 1990.

[15] H. Ishikawa. Exact optimization for Markov Random Fields with convex priors. *PAMI*, 25(10):1333–1336, October 2003.

[16] A. B. Karzanov and S. T. McCormick. Polynomial methods for separable convex optimization in unimorudular linear spaces with applications. *SIAM J. Computing*, 4:1245–1275, 1997.

[17] J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov Random Fields. *FOCS*, pages 14–23, 1999.

[18] N. Komodakis and G. Tziritas. A new framework for approximate labeling via graph cuts. In *ICCV*, October 2005.

[19] I. V. Kovtun. *Image segmentation based on sufficient conditions of optimality in NP-complete classes of structural labelling problems.* PhD thesis, IRTC ITS National Academy of Sciences, Ukraine, 2004. (In Ukranian).

[20] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *ECCV*, May 2004.

[21] K. Murota. Algorithms in discrete convex analysis. *IEICE Transactions on Systems and Information*, E83-D:344–352, 2000.

[22] K. Murota. *Discrete Convex Analysis.* SIAM Monographs on Discrete Mathematics and Applications, Vol. 10, 2003.

[23] K. Murota. On steepest descent algorithms for discrete convex functions. *SIAM J. Optimization*, 14(3):699–707, 2003.

[24] O. Veksler. *Efficient graph-based energy minimization methods in computer vision.* PhD thesis, Cornell University, Dept. of Computer Science, Ithaca, NY, 1999.

[25] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. *Statistics and Computing*, 14(2):143–166, April 2004.

[26] A. Zomet, A. Levin, S. Peleg, and Y. Weiss. Seamless image stitching by minimizing false edges. *IEEE Trans. Image Processing*, 2005.