Iteratively Reweighted Graph Cut for Multi-label MRFs with Non-convex Priors

Thalaiyasingam Ajanthan, Richard Hartley, Mathieu Salzmann, and Hongdong Li

Australian National University & NICTA
Canberra, Australia

Abstract

While widely acknowledged as highly effective in computer vision, multi-label MRFs with non-convex priors are difficult to optimize. To tackle this, we introduce an algorithm that iteratively approximates the original energy with an appropriately weighted surrogate energy that is easier to minimize. Our algorithm guarantees that the original energy decreases at each iteration. In particular, we consider the scenario where the global minimizer of the weighted surrogate energy can be obtained by a multi-label graph cut algorithm, and show that our algorithm then lets us handle of large variety of non-convex priors. We demonstrate the benefits of our method over state-of-the-art MRF energy minimization techniques on stereo and inpainting problems.

1. Introduction

In this paper, we introduce an algorithm to minimize the energy of multi-label Markov Random Fields (MRFs) with non-convex edge priors. In general, minimizing a multi-label MRF energy function is NP-hard. While in rare cases a globally optimal solution can be obtained in polynomial time, *e.g.*, in the presence of convex priors [10], in most scenarios one has to rely on an approximate algorithm. Even though graph-cut-based algorithms [6] have proven successful for specific problems (*e.g.*, *metric* priors), there does not seem to be a single algorithm that performs well with different non-convex priors such as the truncated quadratic, the Cauchy function and the corrupted Gaussian.

Here, we propose to fill this gap and introduce an iterative graph-cut-based algorithm to minimize multi-label MRF energies with a certain class of non-convex priors. Our algorithm iteratively minimizes a weighted surrogate energy function that is easier to optimize, with weights computed from the solution at the previous iteration. We show that, under suitable conditions on the non-convex priors, and as long as the weighted surrogate energy can be decreased, our approach guarantees that the true energy decreases at each iteration.

More specifically, we consider MRF energies with arbitrary data terms and where the non-convex priors are concave functions of some convex priors over pairs of nodes. In this scenario, and when the label set is linearly ordered, the solution at each iteration of our algorithm can be obtained by applying the multi-label graph cut algorithm of [10]. Since the resulting solution is optimal, our algorithm guarantees that our MRF energy decreases.

In fact, our method is inspired by the Iteratively Reweighted Least Squares (IRLS) algorithm which is well-known for continuous optimization. To the best of our knowledge, this is the first time that such a technique is transposed to the MRF optimization scenario.

We demonstrate the effectiveness of our algorithm on the problems of stereo correspondence estimation and image inpainting. Our experimental evaluation shows that our method consistently outperforms other state-of-the-art graph-cut-based algorithms [6, 24], and, in most scenarios, yields lower energy values than TRW-S [13], which was shown to be one of the best-performing multi-label approximate energy minimization methods [22, 12].

1.1. Related work

Over the years, two different types of approximate MRF energy minimization methods have been proposed. The first class of such methods consists of move-making techniques that were inspired by the success of the graph cut algorithm at solving binary problems in computer vision. These techniques include α-expansion, α-β swap [6] and multi-label moves [24, 23, 11]. The core idea of these methods is to reduce the original multi-label problem to a sequence of binary graph cut problems. Each graph cut problem can then be solved either optimally by the max-flow algorithm [5] if the resulting binary energy is submodular, or approximately via a roof dual technique [4] otherwise. The second type of approximate energy minimization methods consists of message passing algorithms, such as belief propagation (BP) [8], tree-reweighted message passing (TRW) [25, 13] and the dual decomposition-based approach of [14], which TRW is a special case of.

As mentioned above, our algorithm is inspired by the IRLS method. Recently, several methods similarly motivated by the IRLS have been proposed to minimize different objective functions. For instance, in [2], the L_q norm (for $1 \leq q < 2$) was minimized by iteratively minimizing a weighted L_2 cost function. In [16], an iterated L_1 algorithm was introduced to optimize non-convex functions that are the sum of convex data terms and concave smoothness terms. More recently, a general formulation (not restricted to weighted L_2 or L_1 minimization) was studied, together with the conditions under which such iteratively reweighted algorithms ensure the cost to decrease [1]. In the next section, we propose an extension of this formulation that will later allow us to tackle the case of multi-label MRFs.

2. Iteratively reweighted minimization

Given a set $\mathcal{X}$ and functions $f_i : \mathcal{X} \to \mathcal{D}$ and $h_i : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D} \subseteq \mathbb{R}$, let us assume that we want to minimize an objective function of the form

$$C_h(\mathbf{x}) = \sum_{i=1}^{k} h_i \circ f_i(\mathbf{x}) , \qquad (1)$$

and, without loss of generality, that we have a method to minimize a weighted cost function of the form

$$C_w(\mathbf{x}) = \sum_{i=1}^{k} w_i \, f_i(\mathbf{x}) . \qquad (2)$$

For instance, in the IRLS algorithm, $f_i(\mathbf{x})$ is a squared cost.

Our goal is to study the conditions under which C_h can be minimized by iteratively minimizing C_w. To this end, we first give the definition of a supergradient, which we will rely upon in the following discussion.

Definition 2.1. Let $\mathcal{D}$ be a subset of $\mathbb{R}$. A supergradient of a function $h : \mathcal{D} \to \mathbb{R}$ at a point c is a value $h^s(c) \in \mathbb{R}$ such that $h(d) \leq h(c) + (d - c) \, h^s(c)$ for any point d.

A supergradient h^s is called a strict supergradient if the inequality is strict for any point $d \neq c$. If the function is differentiable, then the supergradient at a point is unique and equal to the derivative. A concave function defined on a subset of the real numbers has a supergradient at each interior point.

In [1], the following lemma was provided to study the behavior of iteratively reweighted minimization methods.

Lemma 2.1. *Let $h(x)$ be a concave function defined on a subset $\mathcal{D}$ of the real numbers and $h^s(c_i)$ be a supergradient at c_i. If c_i and d_i in $\mathcal{D}$ satisfy*

$$\sum_{i=1}^{k} d_i \, h^s(c_i) \leq \sum_{i=1}^{k} c_i \, h^s(c_i) ,$$

then

$$\sum_{i=1}^{k} h(d_i) \leq \sum_{i=1}^{k} h(c_i) .$$

If the first inequality is strict, so is the second.

Proof. See [1]. □

Note that Lemma 2.1 only considers the case where the function h is the same for all the elements in the sum. This is in contrast with our definition of the cost in Eq. 1, where we want to allow h to be indexed on i. To handle this more general scenario, we introduce the following lemma.

Lemma 2.2. *Given a set $\mathcal{X}$, functions $f_i : \mathcal{X} \to \mathcal{D}$ and concave functions $h_i : \mathcal{D} \to \mathbb{R}$, with $\mathcal{D} \subseteq \mathbb{R}$, such that,*

$$\sum_{i=1}^{k} w_i^t \, f_i(\mathbf{x}^{t+1}) \leq \sum_{i=1}^{k} w_i^t \, f_i(\mathbf{x}^t) ,$$

where $w_i^t = h_i^s(f_i(\mathbf{x}^t))$ and $\mathbf{x}^t$ is the estimate of $\mathbf{x}$ at iteration t, then

$$\sum_{i=1}^{k} h_i \circ f_i(\mathbf{x}^{t+1}) \leq \sum_{i=1}^{k} h_i \circ f_i(\mathbf{x}^t) .$$

If the first inequality is strict, so is the second.

Proof. Let us define $c_i = f_i(\mathbf{x}^t)$ and $d_i = f_i(\mathbf{x}^{t+1})$. Since h_i^s is a supergradient,

$$h_i(d_i) \leq h_i(c_i) + (d_i - c_i) \, h_i^s(c_i) ,$$

for all i. Summing over i gives,

$$\sum_{i=1}^{k} h_i(d_i) \leq \sum_{i=1}^{k} h_i(c_i) + \sum_{i=1}^{k} (d_i - c_i) \, h_i^s(c_i) .$$

The sum $\sum_{i=1}^{k} (d_i - c_i) \, h^s(c_i) = \sum_{i=1}^{k} w_i^t \, f_i(\mathbf{x}^{t+1}) - \sum_{i=1}^{k} w_i^t \, f_i(\mathbf{x}^t)$ is non-positive by hypothesis, which completes the proof. □

It is important to note that this lemma holds for discrete subsets $\mathcal{D}$, as well as continuous ones, and that the functions h_i do not need to be differentiable.

Therefore, for concave functions h_i, by choosing the supergradients of h_i as weights at each iteration, we can minimize the objective function $C_h(\mathbf{x})$ of Eq. 1 by iteratively minimizing the cost $C_w(\mathbf{x})$ of Eq. 2. This general procedure is summarized in Algorithm 1.

Algorithm 1 is applicable to any minimization problem, as long as the objective function takes the form of Eq. 1 with concave functions h_i. Furthermore, to minimize the surrogate cost of Eq. 2, any algorithm (either exact or approximate) can be used, as long as it decreases this cost.

We would like to point out that, at each iteration, we minimize $\sum_{i=1}^{k} h_i^s \left(f_i(\mathbf{x}^t) \right) f_i(\mathbf{x})$, which in general is not an upper bound of $\sum_{i=1}^{k} h_i \circ f_i(\mathbf{x})$. Therefore, our algorithm does not fall into the "majorize-minimize" framework.

Algorithm 1 Iteratively reweighted minimization

$C_h(\mathbf{x}) \leftarrow \sum_{i=1}^k h_i \circ f_i(\mathbf{x})$ ▷ Concave functions h_i
Initialize $\mathbf{x}$
repeat
   $w_i^t \leftarrow h_i^s(f_i(\mathbf{x}^t))$
   $\mathbf{x}^{t+1} \leftarrow \arg\min_{\mathbf{x}} \sum_{i=1}^k w_i^t f_i(\mathbf{x})$
until convergence of $C_h(\mathbf{x})$
return $\mathbf{x}^{t+1}$

3. An iteratively reweighted scheme for MRFs

Recall that our goal is to tackle the problem of MRF energy minimization. Here, we show how this can be achieved by exploiting Algorithm 1.

To this end, let $\mathcal{V}$ be the set of vertices (or nodes) of an MRF, *e.g.*, corresponding to the pixels or superpixels in an image, and $\mathcal{L}$ be a finite set of labels. Furthermore, let $\mathbf{x} = [x_1, \cdots, x_n]^T$, $n = |\mathcal{V}|$, $x_p \in \mathcal{L}$ denote the vector that assigns one label to each node. Finding the best label assignment for each node in an MRF then translates to minimizing the energy of the MRF. In general, the energy of an MRF can be expressed as

$$E(\mathbf{x}) = \sum_{i=1}^{|\mathcal{C}|} \theta_i(\mathbf{x}_i) , \tag{3}$$

where $\mathcal{C}$ is the set of cliques in the graph (*i.e.*, the groups of connected nodes), $\mathbf{x}_i$ represents the set of variables corresponding to the nodes in clique i, and $\theta_i : \mathcal{L}^{|\mathbf{x}_i|} \to \mathbb{R}$ is the energy (or potential) function associated with clique i.

Let us now assume that each potential function can be written as

$$\theta_i(\mathbf{x}_i) = h_i \circ f_i(\mathbf{x}_i) , \tag{4}$$

where h_i is a concave function and f_i an arbitrary one. This lets us rewrite the MRF energy of Eq. 3 as

$$E(\mathbf{x}) = \sum_{i=1}^{|\mathcal{C}|} h_i \circ f_i(\mathbf{x}_i) , \tag{5}$$

which has the form of Eq. 1[1]. Therefore, we can employ Algorithm 1 to minimize $E(\mathbf{x})$, and iteratively minimize the surrogate energy

$$\tilde{E}(\mathbf{x}) = \sum_{i=1}^{|\mathcal{C}|} w_i f_i(\mathbf{x}_i) , \tag{6}$$

with weight w_i taken as the supergradient of h_i evaluated using the previous estimate of $\mathbf{x}$.

---

[1]Note that $f_i(\mathbf{x}_i)$ can be equivalently written as $f_i(\mathbf{x})$, where the variables $x_p \notin \mathbf{x}_i$ (*i.e.*, not in clique i) simply have no effect on the function.

It is important to note, however, that for this algorithm to be effective, the minimization of $\tilde{E}(\mathbf{x})$ at each iteration must be relatively easy, and at least guarantee that the surrogate energy decreases. Furthermore, while in practice any existing MRF energy minimization algorithm (either exact or approximate) can be utilized to minimize $\tilde{E}(\mathbf{x})$, the quality of the overall solution found by our algorithm may vary accordingly. In the next section, we discuss a special case of this general MRF energy minimization algorithm, which, as depicted in our experiments, is effective in many scenarios.

4. Iteratively reweighted graph cut

In this section, we introduce an iterative algorithm for the case of multi-label MRFs with pairwise node interactions. In particular, we propose to make use of the multi-label graph cut of [10] at each iteration of our algorithm. The multi-label graph cut yields an optimal solution under the following two conditions: 1) the label set must be ordered; 2) the pairwise potential must be a convex function of the label difference. In practice, such convex priors have limited power due to their poor ability to model noise. In contrast, while still relying on the first condition, our algorithm allows us to generalize to non-convex priors, and in particular to robust norms that have proven effective in computer vision.

4.1. MRF with pairwise interactions

In an MRF with pairwise node interactions, the energy can be expressed as

$$E(\mathbf{x}) = \sum_{p \in \mathcal{V}} \theta_p^u(x_p) + \sum_{(p,q) \in \mathcal{N}} \theta_{pq}^b(x_p, x_q) , \tag{7}$$

where θ^u and θ^b denote the unary potentials (*i.e.*, data cost) and pairwise potential (*i.e.*, interaction cost), respectively, and $\mathcal{N}$ is the set of edges in the graph, *e.g.*, encoding a 4-connected or 8-connected grid over the image pixels.

As discussed in Section 3, to be able to make use of Algorithm 1, we need to have potential functions of the form given in Eq. 4. Under this assumption, we can then rewrite the energy of Eq. 7 as

$$E(\mathbf{x}) = \sum_{p \in \mathcal{V}} h_u \circ f_p(x_p) + \sum_{(p,q) \in \mathcal{N}} h_b \circ f_{pq}(x_p, x_q) , \tag{8}$$

where h_u and h_b are concave functions.

Following Algorithm 1, we minimize this energy by iteratively minimizing a surrogate energy of the form (at iteration $t+1$)

$$\begin{aligned} \tilde{E}(\mathbf{x}) &= \sum_{p \in \mathcal{V}} h_u^s\big(f_p(x_p^t)\big) f_p(x_p) \\ &\quad + \sum_{(p,q) \in \mathcal{N}} h_b^s\big(f_{pq}(x_p^t, x_q^t)\big) f_{pq}(x_p, x_q) , \end{aligned} \tag{9}$$

Algorithm 2 Iteratively Reweighted Graph Cut (IRGC)

$E(\mathbf{x}) \leftarrow \sum_{p\in\mathcal{V}} f_p(x_p) + \sum_{(p,q)\in\mathcal{N}} h_b \circ g\left(|x_p - x_q|\right)$ $\triangleright$ Convex function g and non-decreasing concave function h_b

$w_{pq}^0 \leftarrow 0.5$ $\triangleright$ Initialize the weights

repeat

    if $t \neq 0$ **then** $w_{pq}^t \leftarrow h_b^s\left(g\left(|x_p^t - x_q^t|\right)\right)$ **end if** $\triangleright$ Update the weights except at the first iteration

    $\mathbf{x}^{t+1} \leftarrow \arg\min_{\mathbf{x}} \sum_{p\in\mathcal{V}} f_p(x_p) + \sum_{(p,q)\in\mathcal{N}} w_{pq}^t\, g(|x_p - x_q|)$ $\triangleright$ Minimize using the multi-label graph cut

until $E(\mathbf{x}^{t+1}) = E(\mathbf{x}^t)$ $\triangleright$ Convergence of $E(\mathbf{x})$

return $\mathbf{x}^{t+1}$

---

where h_u^s and h_b^s are the supergradients of h_u and h_b, respectively, and x_p^t denotes the estimate of x_p at the previous iteration.

Since our goal is to employ the multi-label graph cut algorithm [10] to minimize $\tilde{E}(\mathbf{x})$, we need to define the different functions h_u, h_b, f_p and f_{pq} so as to satisfy the requirements of this algorithm. To this end, for the unary potential, we choose h_u to be the identity function. That is,

$$\theta_p^u(x_p) = h_u \circ f_p(x_p) = f_p(x_p)\,. \tag{10}$$

This implies that no reweighting is required for the unary potentials, since the supergradient of h_u is always 1. The multi-label graph cut having no specific requirement on the form of the data term, f_p can be any arbitrary function.

In contrast, for the pairwise potentials, the multi-label graph cut requires f_{pq} to be a convex function of the label difference. That is, for a convex function g defined on a subset of $\mathbb{R}$,

$$f_{pq}(x_p, x_q) = g(|x_p - x_q|)\,. \tag{11}$$

In addition, because the energy $\tilde{E}(\mathbf{x})$ depends on a *weighted* sum of pairwise terms, we need the weights to satisfy some conditions. More precisely, to be able to use the max-flow algorithm within the multi-label graph cut, the weights need to be non-negative. Since these weights are computed from the supergradient of h_b, this translates into a requirement for h_b to be non-decreasing. Note that, in the context of smoothness potentials in an MRF, this requirement comes at virtually no cost, since we hardly expect the potentials to decrease as the label difference increases.

Under these conditions, the surrogate energy to be minimized by the multi-label graph cut at each iteration of our algorithm can be written as

$$\tilde{E}(\mathbf{x}) = \sum_{p\in\mathcal{V}} f_p(x_p) + \sum_{(p,q)\in\mathcal{N}} w_{pq}^t\, g(|x_p - x_q|)\,, \tag{12}$$

where g is a convex function, and $w_{pq}^t = h_b^s\left(f_{pq}(x_p^t, x_q^t)\right)$, with h_b a concave, non-decreasing function. In the first iteration, we set the weights w_i^0 to some constant value[2] to

---

[2]We set $w_{pq}^0 = \epsilon$, where $0 \leq \epsilon \leq 1$, so that the effect of the edge terms is smaller for the first estimate. In our experiments, we found $\epsilon = 0.5$ to work well and thus always use this value.
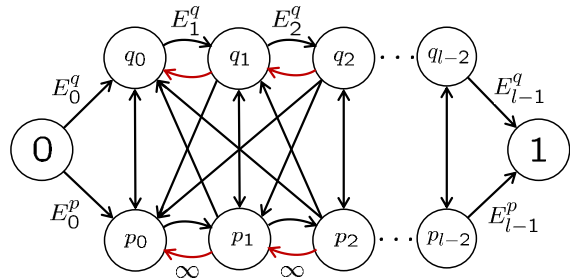

Figure 1: *Multi-label graph construction. Constraint edges are highlighted in red.*

make the algorithm independent of any initial estimate $\mathbf{x}^0$. Our overall Iteratively Reweighted Graph Cut (IRGC) algorithm is summarized in Algorithm 2.

Multi-label graph cut. Here, we briefly describe the multi-label graph cut algorithm employed at each iteration of Algorithm 2. Note that, although equivalent, our graph construction slightly differs from that of [10].

The multi-label graph cut works by constructing a graph such as the one illustrated in Fig. 1 for two neighboring nodes p and q of the original MRF, and applying graph cut to this graph. More specifically, let the label set $\mathcal{L} = \{0, 1, 2, \cdots, l-1\}$. For each node $p \in \mathcal{V}$, the new graph contains $l-1$ nodes denoted by $p_0, p_1, \cdots, p_{l-2}$. Furthermore, two additional nodes, the start and end nodes denoted by 0 and 1, are included in the graph.

For each node p and each $i \in \mathcal{L}$, there is a directed horizontal edge from p_{i-1} to p_i, where p_{-1} and p_{l-1} denote the start and end nodes, respectively. Additionally, the graph contains constraint edges with infinite weights going in the opposite direction. Since the graph cut algorithm partitions the set of nodes into two disjoint subsets $\mathcal{V}_0$ and $\mathcal{V}_1$, where $0 \in \mathcal{V}_0$ and $1 \in \mathcal{V}_1$, the constraint edges ensure that for each $p \in \mathcal{V}$ exactly one horizontal edge is present in the minimum cut. If an edge $p_{i-1} \rightarrow p_i$ is in the minimum cut, then it can be interpreted as node p taking the label i. Therefore, the horizontal edges directly represent the unary potentials and are given a weight $E_i^p = f_p(i)$.

Furthermore, for any $(p, q) \in \mathcal{N}$ and for each $i, j \in \{0, 1, \cdots, l-2\}$ there is a directed edge from p_i to q_j and

vice versa. The weight of the edge $p_i \to q_j$ is defined as

$$E_{ij}^{pq} = \begin{cases} 0 & \text{if } i < j \\ \frac{w_{pq}^t}{2} \, g''(|i-j|) & \text{if } i = j \\ w_{pq}^t \, g''(|i-j|) & \text{if } i > j \, , \end{cases} \qquad (13)$$

where $g''(|z|) = g(|z+1|) + g(|z-1|) - 2\,g(|z|)$, which is positive for a convex function g.

In our scenario, with our condition that w_{pq}^t be non-negative, the multi-label graph contains no negative edges. Therefore, the global minimum of the corresponding energy can be found in polynomial time using the max-flow algorithm. Note that for a sparsely connected graph, *e.g.*, 4 or 8-connected neighborhood, the memory requirement of a general multi-label graph is $\mathcal{O}(|\mathcal{V}| \cdot |\mathcal{L}|^2)$. However if g is linear, then only the vertical edges $p_i \to q_i$ will have non-zero weights and the memory required drops to $\mathcal{O}(|\mathcal{V}| \cdot |\mathcal{L}|)$.

4.2. Choice of functions g and h_b

While, in Section 4.1, we have defined conditions on the functions g and h_b (*i.e.*, g convex and h_b concave, non-decreasing) for our algorithm to be applicable with the multi-label graph cut, these conditions still leave us a lot of freedom in the actual choice of these functions. Here, we discuss several such choices, with special considerations on the memory requirement of the resulting algorithm.

In the context of computer vision problems with ordered label sets, *e.g.*, stereo and inpainting, it is often important to make use of robust estimators as pairwise potentials to better account for discontinuities, or outliers. Many such robust estimators belong to the family of functions with a single inflection point in $\mathbb{R}^+$. In other words, they can be generally defined as non-decreasing functions $\theta_{pq}(|x_p - x_q|)$[3], such that for a given $\lambda \geq 0$, $\theta(z)$ is convex if $z \leq \lambda$, and concave otherwise. Such functions include the truncated linear, the truncated quadratic and the Cauchy function $\theta(z) = \lambda^2/2 \log\left(1 + (z/\lambda)^2\right)$ [9]. Note that any convex or concave function on $\mathbb{R}^+$ also belongs to this family.

For a given such function θ, according to our algorithm, we need to write

$$\theta(z) = h_b \circ g(z) \, , \qquad (14)$$

with a concave, non-decreasing function h_b and a convex function g. Note that the multi-label graph structure is determined by the function g. Therefore, to make the graph as sparse as possible, and thus limit the required memory, we need to choose g such that the second order difference $g''(z)$ is zero for as many values z as possible. Table 1 gives the functions g and h_b such that $g''(z)$ is zero $\forall z \geq \lambda$ while satisfying Eq. 14 and the necessary conditions on h_b and

---

[3]Note that, here, since we only consider pairwise terms, we dropped the superscript b in θ^b for ease of notation.

	$y = g(z)$		$h_b(y)$
$z \leq \lambda$	$\theta(z)$	$y \leq \theta(\lambda)$	y
$z \geq \lambda$	$\theta'(\lambda)(z-\lambda)$ $+\theta(\lambda)$	$y \geq \theta(\lambda)$	$\theta\left(\frac{y+\lambda\,\theta'(\lambda)-\theta(\lambda)}{\theta'(\lambda)}\right)$

Table 1: *Functions g and h_b corresponding to a given $\theta(z)$, such that $\theta(z)$ is convex if $z \leq \lambda$ and concave otherwise. It can easily be verified that $\theta(z) = h_b \circ g(z)$, and that g is convex and h_b is concave, as well as that both functions are non-decreasing, because θ is non-decreasing. Here $\theta'(\lambda)$ is the derivative of θ at λ, or its left derivative $\theta'(\lambda^-)$ if θ is not differentiable at λ. See Fig. 2(a-b) for example plots.*

g. Fig. 2(a-b) provide the plots corresponding to the truncated linear and Cauchy function. For a function g derived according to Table 1, the memory requirement of the multi-label graph is $\mathcal{O}(|\mathcal{V}| \cdot |\mathcal{L}| \cdot \lambda)$.

Note that our method is not limited to the family of functions described above. As an example, we consider the case of another robust estimator, the corrupted Gaussian function $\theta_G(z) = -\log(\alpha \exp(-z^2) + (1-\alpha)\exp(-z^2/\beta^2)/\beta)$ [9], which does not follow the definitions of the functions described before. However, since $\theta_G(\sqrt{\cdot})$ is concave, we can minimize $\theta_G(z)$ by choosing $g(z) = z^2$ and $h_b(y) = \theta_G(\sqrt{y})$. The corresponding plots for the corrupted Gaussian are provided in Fig. 2(c).

4.3. Hybrid strategy

While our algorithm guarantees that the energy value decreases at each iteration, it remains prone to getting trapped in local minima (with respect to the iterative reweighting scheme). Here, we propose a hybrid optimization strategy that combines IRGC with a different minimization technique, and helps us escape from some of the local minima of the energy.

In particular, here we make use of α-expansion [6] as an additional minimization technique. At each iteration of our algorithm, instead of updating $\mathbf{x}^t \to \mathbf{x}^{t+1}$ in one step, our hybrid strategy performs the following steps:

1. Update $\mathbf{x}^t \to \mathbf{x}'$ by minimizing the surrogate energy using the multi-label graph cut.

2. Improve the new estimate $\mathbf{x}' \to \mathbf{x}^{t+1}$ using one pass of α-expansion with the true energy, such that $E(\mathbf{x}^{t+1}) \leq E(\mathbf{x}')$.

For non-metric pairwise potentials, for which regular α-expansion does not apply, we truncate the non-submodular terms as suggested in [19]. Note that this still guarantees that the energy will decrease. We found that the variety in the optimization strategy arising from this additional α-expansion step was effective to overcome local minima. Since both algorithms guarantee to decrease the energy
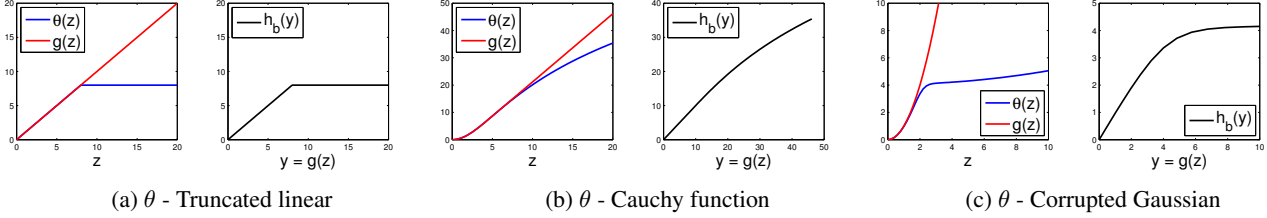
(a) θ - Truncated linear   (b) θ - Cauchy function   (c) θ - Corrupted Gaussian

Figure 2: *Plots of θ, g and h_b with $\theta(z) = h_b \circ g(z)$, when θ is (a) the truncated linear, (b) the Cauchy function and (c) the corrupted Gaussian. Here g is convex and h_b is concave. In (a) and (b), the functions g and h_b are derived from Table 1. In (c), $g(z) = z^2$ and $h_b(y) = \theta(\sqrt{y})$.*

$E(\mathbf{x})$ at each step, our hybrid algorithm also decreases the energy at each iteration. In our experiments, we refer to this hybrid algorithm as IRGC+expansion.

Note that other methods, such as α-β swap, or any algorithm that guarantees to improve the given estimate can be employed. Alternatively, one could exploit a fusion move strategy [15] to combine the estimates obtained by the two different algorithms. However, this would come at an additional computation cost, and, we believe, goes beyond the scope of this paper.

5. Experiments

We evaluated our algorithm on the problems of stereo correspondence estimation and image inpainting. In those cases, the pairwise potentials typically depend on additional constant weights, and can thus be written as

$$\theta^b_{pq}(x_p, x_q) = \gamma_{pq}\,\phi_{pq}(|x_p - x_q|)\,, \qquad (15)$$

where γ_{pq} are the constant weights. As long as $\gamma_{pq} \geq 0$, our algorithm is unaffected by these weights, in the sense that we can simply multiply our weights w^t_{pq} by these additional constant weights. Note that since the main purpose of this paper is to evaluate the performance of our algorithm on different MRF energy functions, we used different smoothing costs $\phi(\cdot)$ for different problem instances without tuning the weights γ_{pq} for the specific smoothing costs.

We compare our results with those of α-expansion, α-β swap [6], multi-label swap [24] and TRW-S [13]. For fairer comparison, we improved the results of TRW-S using α-expansion which is denoted as TRW-S+expansion. For α-expansion, we used the max-flow algorithm when the pairwise potentials were *metric*, and the QPBOP algorithm [4, 18] (denoted as α-expansionQ) otherwise. In the latter case, if a node in the binary problem is unlabeled then the previous label is retained. For our comparison, we used the publicly available implementation of α-expansion, α-β swap, QPBO and TRW-S, and implemented the multi-label swap algorithm as described in [24].

All the algorithms were initialized by assigning the label 0 to all the nodes (note that in [24] multi-label swap was

Problem	γ_{pq}	$\phi(\cdot)$	λ
Teddy	$\begin{cases} 30 & \text{if } \nabla_{pq} \leq 10 \\ 10 & \text{otherwise} \end{cases}$	Truncated linear	8
Map	4		6
Sawtooth	20	Truncated quadratic	3
Venus	50		3
Cones	10	Cauchy function	8
Tsukuba	$\begin{cases} 40 & \text{if } \nabla_{pq} \leq 8 \\ 20 & \text{otherwise} \end{cases}$		2

Table 2: *Pairwise potential $\theta^b_{pq}(x_p, x_q) = \gamma_{pq}\,\phi(|x_p - x_q|)$ used for the stereo problems. Here $\phi(z)$ is convex if $z \leq \lambda$ and concave otherwise, and ∇_{pq} denotes the absolute intensity difference between the pixels p and q in the left image.*

initialized using α-expansion). For multi-label swap the parameter t was fixed to 2 in all our experiments (see [24] for details). The energy values presented in the following sections were obtained at convergence of the different algorithms, except for TRW-S which we ran for 100 iterations and chose the best energy value.[4] All our experiments were conducted on a 3.4GHz i7-4770 CPU with 16 GB RAM, and no effort was made to exploit the multiple cores.

Stereo:
Given a pair of rectified images (one left and one right), stereo correspondence estimation aims to find the disparity map, which specifies the horizontal displacement of each pixel between the two images with respect to the left image. For this task, we employed six instances from the Middlebury dataset [20, 21]: Teddy, Map, Sawtooth, Venus, Cones and Tsukuba. For Tsukuba and Venus, we used the unary potentials of [22], and for the other cases, those of [3]. The pairwise potentials are summarized in Table 2. Note that we do not explicitly model occlusions, which should be handled by our robust potentials.

The final energies and execution times corresponding to the stereo problems are summarized in Table 3. The disparity maps found using our IRGC+expansion algorithm and energy vs time plots of the algorithms for some of the problems are shown in Fig. 3 and Fig. 4(a-c), respectively. Note

---
[4]While the energy of TRW-S decreases slightly by running more iterations, the algorithm becomes very slow.

Algorithm	Teddy		Map		Sawtooth		Venus		Cones		Tsukuba	
	E[10^3]	T[s]	E[10^3]	T[s]	E[10^3]	T[s]	E[10^3]	T[s]	E[10^3]	T[s]	E[10^3]	T[s]
α-β swap	2708.1	35	149.5	2	1079.5	7	3219.7	8	4489.9	135	409.1	6
α-expansion	2664.6	21	144.4	2	1067.5	15	3201.5	16	2480.7	183	403.3	8
Multi-label swap	5502.3	236	470.6	13	1660.6	103	5740.1	163	-	-	-	-
TRW-S	2652.7	318	143.0	34	1038.8	52	3098.6	50	2304.8	311	395.8	19
TRW-S+expansion	**2646.8**	326	**142.9**	35	**1034.4**	56	3083.4	53	2303.7	322	**395.5**	21
IRGC	2687.8	65	144.0	9	1042.1	93	3081.4	49	**2301.4**	397	397.3	20
IRGC+expansion	2650.3	44	143.2	4	1034.9	32	**3078.8**	26	**2301.4**	204	396.1	14

Table 3: *Comparison of the minimum energies (E) and execution times (T) for stereo problems. IRGC+expansion found the lowest energy or virtually the same energy as TRW-S+expansion. For the truncated linear prior (Teddy and Map) IRGC+expansion was 8 times faster than TRW-S+expansion. For the truncated quadratic and the Cauchy prior IRGC outperformed all other graph-cut-based algorithms and found a lower energy than TRW-S for Venus and Cones.*
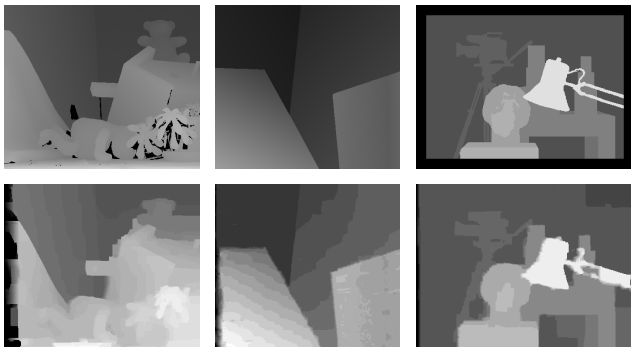
(a) Teddy,
Tr. linear

(b) Venus,
Tr. quad.

(c) Tsukuba,
Cauchy

Figure 3: *Disparity maps obtained with IRGC+expansion. The corresponding ground-truth is shown above each disparity map.*

Algorithm	Tsukuba	
	E[10^3]	T[s]
α-β swap	568.3	10
α-expansionQ	555.2	10
TRW-S	550.0	20
TRW-S+expansion	**548.9**	24
IRGC	614.3	55
IRGC+expansion	549.3	40

Table 4: *Comparison of the minimum energies (E) and execution times (T) on Tsukuba with a corrupted Gaussian prior. While IRGC was trapped in a local minimum, IRGC+expansion found a lower energy than TRW-S.*

that, in most cases, IRGC outperforms the other graph-cut-based algorithms. Note also that IRGC+expansion yields the lowest energy or an energy that is virtually the same as the lowest one.

To illustrate the fact that our algorithm can also exploit priors that are not first convex and then concave, we employed a corrupted Gaussian pairwise potential (with parameters $\alpha = 0.75$ and $\beta = 50$) on the Tsukuba stereo pair, and the results are shown in Table 4.

Algorithm	House		Penguin	
	E[10^3]	T[s]	E[10^3]	T[s]
α-β swap	2488.9	14	4562.8	10
α-expansionQ	2510.0	531	4486.4	18
Multi-label swap	**2399.9**	1457	4520.6	395
TRW-S	2400.1	113	4269.5	138
TRW-S+expansion	2400.0	116	**4230.7**	141
IRGC	**2399.9**	155	4696.9	1045
IRGC+expansion	**2399.9**	108	4238.9	222

Table 5: *Comparison of minimum energies (E) and execution times (T) for the truncated quadratic prior on two inpainting problems. On House, IRGC and multi-label swap also achieved the same lowest energy, but the latter was roughly 15 times slower than IRGC+expansion. On Penguin, while IRGC was trapped in a local minimum, IRGC+expansion was able to find a lower energy than TRW-S and yield an energy similar to TRW-S+expansion.*

Inpainting:
Image inpainting tackles the problem of filling in the missing pixel values of an image, while simultaneously denoising the observed pixel values. In our experiments, we used the Penguin and House images employed in [22]. Due to memory limitation, however, we down-sampled the labels from 256 to 128 for Penguin and from 256 to 64 for House. We used the same unary potential as in [22], *i.e.*, $f_p(x_p) = (I_p - x_p)^2$ if the intensity I_p is observed, and $f_p(x_p) = 0$ otherwise. As pairwise potentials, we employed the truncated quadratic cost $\theta_{pq}^b(x_p, x_q) = \gamma_{pq} \min\left((x_p - x_q)^2, \lambda^2\right)$. For Penguin, $\gamma_{pq} = 20$ and $\lambda = 10$, and for House $\gamma_{pq} = 5$ and $\lambda = 15$. The final energies and execution times are summarized in Table 5, with the inpainted images shown in Fig. 5. Furthermore, in Fig. 4d, we show the energy as a function of time for the different algorithms. Note that, for both images, our IRGC+expansion method outperforms other graph-cut-based algorithms and performs similarly to TRW-S+expansion.
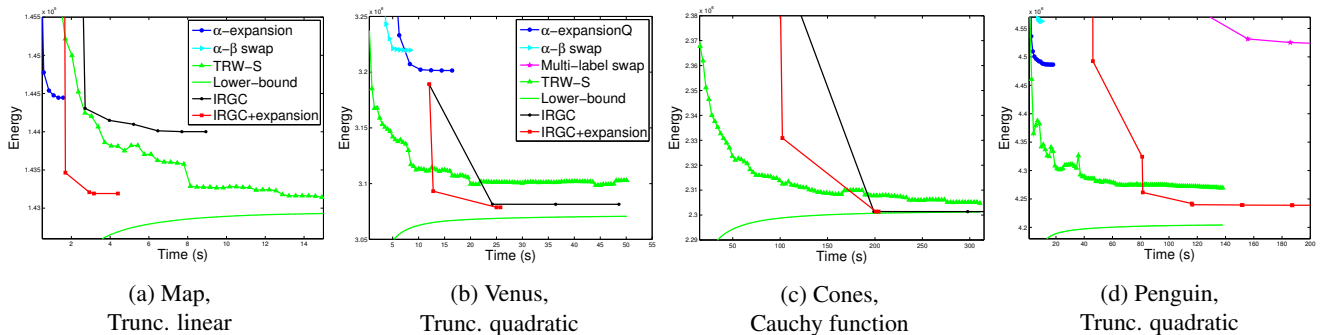
(a) Map,
Trunc. linear

(b) Venus,
Trunc. quadratic

(c) Cones,
Cauchy function

(d) Penguin,
Trunc. quadratic

Figure 4: *Energy vs time (seconds) plots for the algorithms for (a) - (c) some stereo problems and (d) an inpainting problem. The plots are zoomed-in to show the finer details. IRGC+expansion algorithm outperformed all the other algorithms and found the lowest energy within 2–5 iterations. IRGC found a lower energy than α-expansion for Map.*
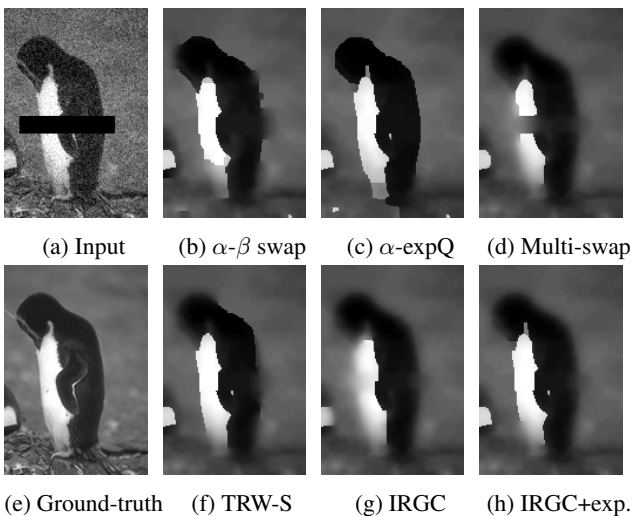

(a) Input (b) α-β swap (c) α-expQ (d) Multi-swap

(e) Ground-truth (f) TRW-S (g) IRGC (h) IRGC+exp.

Figure 5: *Inpainted images for Penguin. Note that multi-label swap was not able to completely inpaint the missing pixels. IRGC+expansion produced smoother results than QPBOP based α-expansion (see the bottom of the penguin) while preserving the finer details compared to TRW-S (see the neck of the penguin). See Table 5 for the energy values.*

Summary:
To evaluate the quality of the minimum energies, we followed the strategy of [22], which makes use of the lower bound found by TRW-S. This quality measure is computed as

$$Q = \frac{E - E_b}{E_b} 100\% , \qquad (16)$$

where E_b is the largest lower bound of TRW-S and E is the minimum energy found by an algorithm. In Table 6, we compare the resulting values of our algorithms with TRW-S, which, from the previous surveys [22, 12] was found to be the best-performing baseline. Note that our IRGC+expansion algorithm yields a better quality measure than TRW-S on average. TRW-S+expansion yields a slightly better average quality measure than IRGC+expansion, namely 0.1873%. Note, how-

Problem	TRW-S	IRGC	IRGC+exp.
Teddy	0.3040%	1.6289%	**0.2102%**
Map	**0.0511%**	0.7387%	0.1728%
Sawtooth	0.6452%	0.9621%	**0.2616%**
Venus	0.9096%	0.3498%	**0.2625%**
Cones	0.1551%	**0.0065%**	0.0074%
Tsukuba	**0.0910%**	0.4678%	0.1679%
Tsu. cor. Gau.	0.3926%	12.1226%	**0.2736%**
House	0.0154%	**0.0058%**	**0.0058%**
Penguin	1.5556%	11.7218%	**0.8259%**
Average	0.4577%	3.1116%	**0.2431%**

Table 6: *Quality of the minimum energies according to Eq. 16. IRGC+expansion clearly yields better quality energies than TRW-S on average.*

ever, that our algorithm was 1.5 – 8 times faster than TRW-S+expansion on stereo problems, except for the corrupted Gaussian prior.

6. Conclusion

We have introduced an Iteratively Reweighted Graph Cut algorithm that can minimize multi-label MRF energies with arbitrary data terms and non-convex priors. We have shown that, while the basic algorithm sometimes gets trapped in local minima, our hybrid version consistently outperforms (or performs virtually as well as) state-of-the-art MRF energy minimization techniques. Due to our use of the multi-label graph cut at each iteration of our algorithm, memory requirement is the current bottleneck of our approach. Note, however, that this can be overcome by replacing the multi-label graph cut with the convex formulation of α-expansion [7], or the approach of [17]. In the latter case, this would further extend the applicability of IRGC to continuous label spaces. Finally, and as discussed in the paper, IRGC really is a special case of an iteratively reweighted approach to MRF, and even continuous, energy minimization. In the future, we therefore intend to study the applicability of such an approach to other types of problems.

7. Acknowledgements

References

[1] K. Aftab and R. Hartley. Convergence of iteratively reweighted least squares to robust m-estimators. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 480 – 487. IEEE, 2015. 2

[2] K. Aftab, R. Hartley, and J. Trumpf. Generalized weiszfeld algorithms for lq optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(4):728 – 745, 2015. 2

[3] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(4):401–406, 1998. 6

[4] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete applied mathematics*, 123(1):155–225, 2002. 1, 6

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004. 1

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001. 1, 5, 6

[7] P. Carr and R. Hartley. Solving multilabel graph cut problems with multilabel swap. In *Digital Image Computing: Techniques and Applications, 2009. DICTA'09.*, pages 532–539. IEEE, 2009. 8

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54, 2006. 1

[9] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5

[10] H. Ishikawa. Exact optimization for markov random fields with convex priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1333–1336, 2003. 1, 3, 4

[11] A. Jezierska, H. Talbot, O. Veksler, and D. Wesierski. A fast solver for truncated-convex priors: quantized-convex split moves. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 45–58. Springer, 2011. 1

[12] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnorr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, J. Lellmann, N. Komodakis, et al. A comparative study of modern inference techniques for discrete energy minimization problems. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1328–1335. IEEE, 2013. 1, 8

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1568–1583, 2006. 1, 6

[14] N. Komodakis, N. Paragios, and G. Tziritas. Mrf energy minimization and beyond via dual decomposition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):531–552, 2011. 1

[15] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1392–1405, 2010. 6

[16] P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock. An iterated l1 algorithm for non-smooth non-convex optimization in computer vision. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1759–1766. IEEE, 2013. 2

[17] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *Computer Vision–ECCV 2008*, pages 792–805. Springer, 2008. 8

[18] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 6

[19] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry [automatic image synthesis]. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 589–596. IEEE, 2005. 5

[20] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. 6

[21] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–195. IEEE, 2003. 6

[22] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):1068–1080, 2008. 1, 6, 7, 8

[23] P. Torr and M. Kumar. Improved moves for truncated convex models. In *Advances in neural information processing systems*, pages 889–896, 2009. 1

[24] O. Veksler. Multi-label moves for mrfs with truncated convex priors. *International journal of computer vision*, 98(1):1–14, 2012. 1, 6

[25] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Map estimation via agreement on trees: message-passing and linear programming. *Information Theory, IEEE Transactions on*, 51(11):3697–3717, 2005. 1