Beyond Loose LP-Relaxations: Optimizing MRFs by Repairing Cycles

Nikos Komodakis[1] and Nikos Paragios[2]

[1] University of Crete
komod@csd.uoc.gr
[2] Ecole Centrale de Paris
nikos.paragios@ecp.fr

Abstract. This paper presents a new MRF optimization algorithm, which is derived from Linear Programming and manages to go beyond current state-of-the-art techniques (such as those based on graph-cuts or belief propagation). It does so by relying on a much tighter class of LP-relaxations, called cycle-relaxations. With the help of this class of relaxations, our algorithm tries to deal with a difficulty lying at the heart of MRF optimization: the existence of inconsistent cycles. To this end, it uses an operation called cycle-repairing. The goal of that operation is to fix any inconsistent cycles that may appear during optimization, instead of simply ignoring them as usually done up to now. The more the repaired cycles, the tighter the underlying LP relaxation becomes. As a result of this procedure, our algorithm is capable of providing almost optimal solutions even for very general MRFs with arbitrary potentials. Experimental results verify its effectiveness on difficult MRF problems, as well as its better performance compared to the state of the art.

1 Introduction

Optimization algorithms for discrete MRFs are known to be of fundamental importance for numerous problems from the fields of vision, graphics and pattern recognition. It is also known that many of the most successful of these algorithms are tightly related to Linear Programming (LP) [1][2][3][4][5][6][7]. In particular, they are connected to the following LP relaxation $\mathcal{P}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$:

$$\mathcal{P}(\bar{\mathbf{g}}, \bar{\mathbf{f}}) = \min_{\mathbf{x}} \sum_{p \in \mathcal{V}} \bar{\mathbf{g}}_p \cdot \mathbf{x}_p + \sum_{pp' \in \mathcal{E}} \bar{\mathbf{f}}_{pp'} \cdot \mathbf{x}_{pp'}, \tag{1}$$

$$\text{s.t.} \sum_{l \in \mathcal{L}} x_p(l) = 1, \qquad \forall\, p \in \mathcal{V} \tag{2}$$

$$\sum_{l' \in \mathcal{L}} x_{pp'}(l, l') = x_p(l), \quad \forall\, pp' \in \mathcal{E}, \ \ l \in \mathcal{L} \tag{3}$$

$$x_p(l) \geq 0, \quad x_{pp'}(l, l') \geq 0. \tag{4}$$

The connection between $\mathcal{P}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ and MRFs lies in that if one replaces (4) with the constraints $x_p(\cdot), x_{pp'}(\cdot, \cdot) \in \{0, 1\}$, then the resulting integer program optimizes the energy of an MRF with unary potentials $\bar{\mathbf{g}}_p = \{\bar{g}_p(\cdot)\}$ and pairwise potentials $\bar{\mathbf{f}}_{pp'} = \{\bar{f}_{pp'}(\cdot, \cdot)\}$. This MRF will be denoted by $\mathrm{MRF}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ hereafter and $\mathcal{L}, \mathcal{V}, \mathcal{E}$ (in (2), (3)) represent respectively its labels, graph vertices and graph edges.

Despite their success, LP-based methods are doomed to fail if relaxation $\mathcal{P}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ does not approximate well the actual problem MRF$(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ (*i.e*, it is not tight), which is exactly what happens when one has to deal with very hard MRF problems. We believe this is an issue that has to be addressed, if a further advance is to be made in MRF optimization. Motivated by this observation, the present work makes important practical, as well as theoretical, contributions in this regard. In particular, in this work we attempt to go beyond most existing MRF techniques, by deriving algorithms that are based on much tighter LP relaxations. Towards this goal, we try to strengthen relaxation $\mathcal{P}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. However, instead of doing that in the primal domain, which is inefficient, our strategy consists of applying this procedure in the dual domain. As a result, we create an hierarchy of tighter and tighter dual relaxations that starts from the dual of $\mathcal{P}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ and goes all the way up to a dual relaxation that actually coincides with the original problem MRF$(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ (§2). From this hierarchy, we choose to deal with one particular class of relaxations, which we call *cycle-relaxations*. This is achieved via a dual-based operation called *cycle-repairing* (§3), which helps us to deal with a difficulty that, we believe, lies at the heart of MRF optimization: the existence of *inconsistent* cycles (see definition later). As the name of that operation reveals, its role is to eliminate any inconsistent cycles that may appear during optimization. Furthermore, the more the repaired cycles, the tighter the underlying relaxation becomes. One efficient way of how one can actually do this cycle-repairing is described in §3.1, thus leading to a powerful algorithm that further promotes the use of general MRFs in future vision applications.

Let us note at this point that Sontag and Jaakkola [8] have very recently tried as well to make use of a tighter LP relaxation for MRF optimization. However, their method relies on a weaker relaxation than ours. Furthermore, they use a primal-based cutting plane algorithm that requires solving a large primal LP (of growing size) at each iteration (i.e., after each new violated inequality is found), which makes their algorithm impractical for large MRFs. On the contrary, by working in the dual domain, our method is able to improve the relaxation (cycle repairing) by reusing work done in previous iterations. It is thus much more efficient, while it is also adaptive as it makes the relaxation tighter only when it needs to be. Moreover, being dual-based, it can provide lower bounds to the optimum MRF energy, which can be useful for verifying a solution's optimality. Let us also note two other dual-based methods of similar spirit to ours, that have been proposed concurrently with our work, appear in [9][10]. Finally, it is important to mention that, although one might be easily tempted to try to strengthen other type of MRF relaxations (*e.g*, quadratic or SOCP), these have been recently shown to be actually weaker than LP relaxation (1) [11].

2   Tighter LP-Relaxations for MRFs

Before examining how cycle-repairing works, let us first describe the hierarchy of dual relaxations that we will use. At one end of this hierarchy, of course, there will lie the dual of relaxation $\mathcal{P}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. For the moment, we will simply refer to

it as $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. The dual cost of any feasible solution to that relaxation is a lower bound to the optimum (*i.e*, minimum) energy of $\mathrm{MRF}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. However, it is often the case that even the maximum of these bounds will be much lower than the optimum MRF energy, which is exactly what happens when $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ is not tight.

To counter that, *i.e*, to raise the maximum lower bound, one has no choice but to introduce extra variables as well as extra constraints to $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, which is exactly what relaxation $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, lying at the other end of our hierarchy, does:

$$\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}}) = \max_{\mathbf{f}} \mathcal{D}(\bar{\mathbf{g}}, \mathbf{f}) \tag{5}$$

$$\text{s.t. } \mathbf{f} \preceq_{\mathcal{E}} \bar{\mathbf{f}}. \tag{6}$$

In (6), we have used an abstract comparison operation $\preceq_{\mathcal{E}}$ between pairwise potential functions. In general, given any subset $\mathcal{C} \subseteq \mathcal{E}$ of the MRF edges and any pairwise potential functions $\mathbf{f}$, $\mathbf{f}'$, the operation $\mathbf{f} \preceq_{\mathcal{C}} \mathbf{f}'$ implies that, for each labeling $\mathbf{l} = \{l_p\}$, it should hold:

$$\sum_{pp' \in \mathcal{C}} f_{pp'}(l_p, l_{p'}) \le \sum_{pp' \in \mathcal{C}} \bar{f}_{pp'}(l_p, l_{p'}), \quad \forall \mathbf{l} = \{l_p\} \tag{7}$$

(This means that, instead of comparing pairwise potentials on individual edges, we compare sums of pairwise potentials over all edges in $\mathcal{C}$).

The reason that we expressed $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ in the above form is to illustrate its relation to $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. As can be seen, one difference between $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ and $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ is that the latter contains an additional set of variables $\mathbf{f} = \{f_{pp'}(\cdot, \cdot)\}$, which can actually be thought of as a new set of pairwise potentials (also called *virtual potentials* hereafter). In relaxation $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, these virtual potentials $\mathbf{f}$ do not appear, as they are essentially kept fixed to the true potentials $\bar{\mathbf{f}}$ (*i.e*, it is as if constraints (6) have been replaced with the trivial constraints $\mathbf{f} = \bar{\mathbf{f}}$). On the contrary, in $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, we can vary these new potentials in order to achieve a higher dual objective value (*i.e*, a higher lower bound to the optimum MRF energy). The restriction, of course, is that we must never allow $\mathbf{f}$ to become larger than the actual potentials $\bar{\mathbf{f}}$. However, the comparison of $\mathbf{f}$ to $\bar{\mathbf{f}}$ is done based not on the standard operator $\le$, but on the generalized operator $\preceq_{\mathcal{E}}$.

As a result, relaxation $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ is much tighter than $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. In fact, as the next theorem certifies, it is so tight that it represents the MRF problem exactly:

Theorem 1 ([12]). $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ *is equivalent to* $\mathrm{MRF}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$.

This should, of course, come as no surprise, since $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ already contains an exponential number of constraints on $\mathbf{f}$.

One may thus argue that relaxations $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ lie at opposite ends: $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ imposes trivial constraints on $\mathbf{f}$ and is thus efficient but not tight, whereas $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ has an exponential number of constraints on $\mathbf{f}$ and is thus tight but not easy to handle. However, one can adjust the amount of constraints on $\mathbf{f}$ by concentrating only on the virtual potentials at a subset $\mathcal{C} \subseteq \mathcal{E}$ of the MRF edges. Indeed, assuming that initially $\mathbf{f} = \bar{\mathbf{f}}$, and that all $f_{pp'}(\cdot, \cdot)$ with $pp' \notin \mathcal{C}$ will be kept fixed during the current step, constraints (6) then reduce to the

easier upper-bounding constraints $\mathbf{f} \preceq_{\mathcal{C}} \bar{\mathbf{f}}$, thus creating a relaxation in between $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ and $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. Contrary to $\mathbf{f} \preceq_{\mathcal{E}} \bar{\mathbf{f}}$, constraints $\mathbf{f} \preceq_{\mathcal{C}} \bar{\mathbf{f}}$ focus only on a subset of the virtual potentials $f_{pp'}(\cdot, \cdot)$, *i.e*, only on those with pp' in subset $\mathcal{C}$, while the rest are left untouched. Not only that, but, as optimization proceeds, one can choose a different local subset $\mathcal{C}_i$ to focus on at each step. Indeed, assuming that $\mathbf{f}'$ are the virtual potentials from the previous step, one can then simply focus on satisfying the constraints $\mathbf{f} \preceq_{\mathcal{C}_i} \mathbf{f}'$, while keeping all $f_{pp'}(\cdot, \cdot)$ with $pp' \notin \mathcal{C}_i$ unchanged (*i.e*, equal to $f'_{pp'}(\cdot, \cdot)$) during the current step (note that, together, these two things imply $\mathbf{f} \preceq_{\mathcal{E}} \mathbf{f}'$ and, since $\mathbf{f}' \preceq_{\mathcal{E}} \bar{\mathbf{f}}$ already holds from the previous iteration, feasibility constraint (6) is thus maintained). In this manner, different constraints are *dynamically* used at each step, implicitly creating a dual relaxation that becomes tighter and tighter as time passes by. This relaxation (denoted by $\mathcal{D}^{\{\mathcal{C}_i\}}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ hereafter) is actually part of an hierarchy of dual relaxations that starts from $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ (*e.g*, if $\mathcal{C}_i$ contains only a single edge) and goes all the way up to $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ (*e.g*, if $\mathcal{C}_i$ contains all MRF edges). Here we will deal with one particular type of relaxations from this hierarchy, where the edges from each set $\mathcal{C}_i$ always form a simple cycle on the MRF graph. We will call these "*cycle-relaxations*" hereafter. Note that these are completely different from the more familiar cycle-inequalities relaxations, which impose constraints not on the dual but on the primal variables.

Let us end this section by describing relaxation $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f})$, which forms the building block of our hierarchy. Some terminology must be introduced first. We will hereafter refer to each MRF vertex $p \in \mathcal{V}$ as an *object*, to each object-label combination (p, l) as a *node* and to each pair $\{(p, l), (p', l')\}$ of nodes (from adjacent objects in $\mathcal{E}$) as a *link*. Relaxation $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f})$ is then defined as follows:

$$\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f}) = \max_{\mathbf{h}, \mathbf{r}} \sum_{p \in \mathcal{V}} \left(\min_{l \in \mathcal{L}} h_p(l) \right) \tag{8}$$

$$\text{s.t. } r_{pp'}(l, l') \geq 0 \quad \forall pp' \in \mathcal{E}, \ l, l' \in \mathcal{L} \tag{9}$$

Here, the dual variables are all the components of vectors $\mathbf{h}, \mathbf{r}$. Each variable $h_p(l)$ will be called the *height* of node (p, l), while variable $r_{pp'}(l, l')$ will be called the *residual* of link $\{(p, l), (p', l')\}$. Furthermore, if a node (p, l) has minimal height at p (*i.e*, $h_p(l) = \min_{l'} h_p(l')$) it will be called *minimal*, while a link $\{(p, l), (p', l')\}$ with zero residual (*i.e*, $r_{pp'}(l, l') = 0$) will be called *tight* (see Fig. 1(a)).

As can be seen from (8), the goal of the above relaxation is to raise the heights of minimal nodes (*i.e*, to increase the quantity $\min_{l \in \mathcal{L}} h_p(l)$ for each object p). But, due to (9), this must be done without any of the residuals becoming negative. Note, however, that heights $\mathbf{h} = \{h_p(\cdot)\}$ and residuals $\mathbf{r} = \{r_{pp'}(\cdot, \cdot)\}$ are tightly related to each other, since they are both defined in terms of a third set of dual variables $\mathbf{y} = \{y_{pp'}(\cdot)\}$ as follows:

$$h_p(l) = \bar{g}_p(l) + \sum_{p':pp' \in \mathcal{E}} y_{pp'}(l), \tag{10}$$

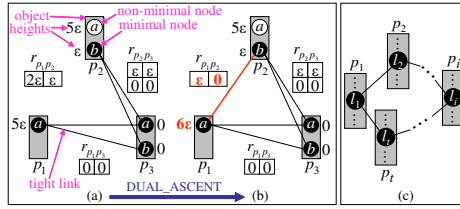$$r_{pp'}(l, l') = f_{pp'}(l, l') - y_{pp'}(l) - y_{p'p}(l'), \tag{11}$$

Fig. 1. We will hereafter visualize each object with a shaded parallelogram, each minimal/non-minimal node with a filled/unfilled circle and each tight link with a line segment (non-tight links are not drawn at all). **(a)** No line segment exists between minimal node (p_1, a) and (the only) minimal node (p_2, b) of object p_2 (this means their link is non-tight, *i.e*, $r_{p_1p_2}(a, b) > 0$). Current dual solution thus does not satisfy Thm. 2. **(b)** One can, therefore, perturb this solution by adding ε to $y_{p_1p_2}(a)$. Then (due to (11)) residuals $r_{p_1p_2}(a, \cdot)$ decrease by ε and thus the link between $(p_1, a), (p_2, b)$ becomes tight (see red segment). Hence, the new solution satisfies Thm. 2. Furthermore, it satisfies Thm. 3, since a cyclic path goes through the black (*i.e*, minimal) nodes $(p_1, a), (p_2, b), (p_3, a)$. Note also that the dual objective has increased, due to the increase by ε of the height of minimal node (p_1, a). **(c)** Visualization of a cycle $p_1 p_2 \ldots p_t p_1$ that is consistent to minimal node (p_1, l_1) (see also text).

Hence, if one adds ε to $y_{pp'}(l)$ in order to raise the height $h_p(l)$ of a minimal node, thus increasing the dual objective (8), residual $r_{pp'}(l, l')$ will then decrease by ε due to (11) and, so, (9) may become violated.

An observation (that will prove to be crucial later on) is that, due to (11), in a relaxation such as $\mathcal{D}^{\{C_i\}}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ or $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, one can alter a residual $r_{pp'}(\cdot, \cdot)$ by changing either variables $\mathbf{y}$, or potentials $\mathbf{f}$. The latter, however, is not possible in weak relaxation $D(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, since $\mathbf{f}$ is assumed to be constant in this case, *i.e*, $\mathbf{f} = \bar{\mathbf{f}}$. As we shall see later, this difference is exactly the reason why one is able to repair cycles while using relaxation $\mathcal{D}^{\{C_i\}}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$.

3   Exploiting Tighter LP-Relaxations Via Cycle-Repairing

In order to see how to make use of a tighter relaxation, let us first examine when relaxation $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ itself is not tight, *i.e*, when even an optimal dual solution of $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ has a lower cost than the minimum cost of MRF$(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. To do that, however, we will first need to characterize the optimal dual solutions of $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$.

Theorem 2 ([6]). *An optimal dual solution of $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ must satisfy the following conditions: if (p, l) is a minimal node of object p, then for any neighboring object p' there must exist a minimal node (p', l') such that the link $\{(p, l), (p', l')\}$ is tight, i.e, $r_{pp'}(l, l') = 0$.*

According to our visualization convention, this means that each minimal node must connect with a solid line to at least one minimal node from each neighboring object. For instance, this property doesn't hold for the dual solution in Fig. 1(a), since (p_1, a) does not have a tight link with any of the minimal nodes in
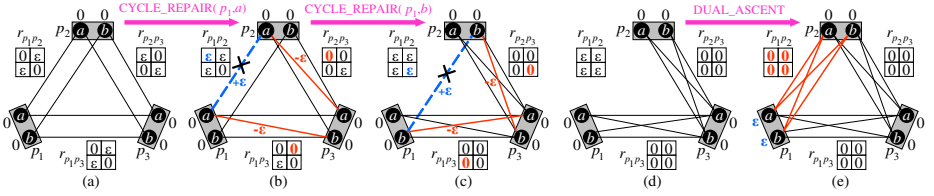
Fig. 2. (a) Current dual solution satisfies Thm. 2, but not Thm. 3. The latter due to that cycle $\mathcal{C} = p_1 p_2 p_3$ is inconsistent, *e.g*, to both nodes (p_1, a), (p_1, b). **(b)** To allow (p_1, a) to later raise its height and possibly become non-minimal, CYCLE_REPAIR(p_1, a) removes the blue tight link (*i.e*, makes it non-tight) by adding ε to potential $f_{p_1 p_2}(a, a)$, thus making residual $r_{p_1 p_2}(a, a) = \varepsilon > 0$. However, to also maintain feasibility constraint $\mathbf{f} \preceq_\mathcal{C} \mathbf{f}'$ (where $\mathbf{f}'$ simply denotes the virtual potentials $\mathbf{f}$ before adding ε), it then has to add $-\varepsilon$ to potentials $f_{p_2 p_3}(a, a)$, $f_{p_1 p_3}(a, b)$, thus creating 2 new tight links (red segments). **(c)** CYCLE_REPAIR(p_1, b) does the same for node (p_1, b) so as to allow him to later raise its height. As a result, a tight link (blue segment) is again removed and 2 new tight links (red segments) are created. **(d)** Resulting dual solution after the 2 calls to CYCLE_REPAIR (same as (c), but reshown for clarity). This solution no longer satisfies Thm. 2, since, due to the cycle-repairs, both $(p_1, a), (p_1, b)$ do not have tight links with any of the minimal nodes of p_2. **(e)** Therefore, DUAL_ASCENT can raise by ε the heights of (p_1, a), (p_1, b) (via adding ε to $y_{p_1 p_2}(a), y_{p_1 p_2}(b)$, thus making all red links tight). Hence, the dual objective increases by ε as well, and now actually coincides with the global MRF optimum.

p_2, but it does hold for the dual solutions in Figs. 1(b), 2(a). The above theorem essentially provides necessary (but not sufficient) conditions for having dual solutions which are optimal for $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, *i.e*, which have maximum dual cost. This means that if these conditions do not hold, one knows how to increase the dual objective by keep applying perturbations to the current dual solution (*i.e*, the current heights and residuals) until these conditions are satisfied. The iterative routine used for this purpose will be called DUAL_ASCENT hereafter. Its name comes from the fact that these perturbations can only improve (*i.e*, increase) the dual objective, but never decrease it. Just as an illustration, we show one such perturbation in Fig. 1. In this case, the dual solution of Fig. 1(a) does not satisfy Thm 2, since node (p_1, a) does not have tight links with any of the minimal nodes in p_2. We can therefore raise that node's height (by increasing variable $y_{p_1 p_2}(a)$ until one of these links becomes tight), thus increasing the dual objective and getting the dual solution of Fig. 1(b), which now, of course, satisfies Thm. 2 and is actually optimal for $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$.

Let us now return to the issue of tightness of relaxation $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ with respect to problem MRF$(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ and, to this end, let's assume that, by applying DUAL_ASCENT, we have managed to increase the cost of the current feasible dual solution until Thm. 2 holds true. When is the resulting dual cost high enough so as to have reached the minimum cost of MRF$(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ (thus proving that $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ is tight)? The next theorem provides a sufficient condition for this.

Theorem 3 ([12]). *If there is a labeling $\{l_p\}$ such that each node (p, l_p) is minimal and each link $\{(p, l_p), (p', l_{p'})\}$ is tight then $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ is tight. Hence,*

$\{l_p\}$ *optimizes* $\text{MRF}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, *current feasible dual solution optimizes* $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ *and their costs coincide.*

For instance, labeling $\{l_{p_1} = a, l_{p_2} = b, l_{p_3} = a\}$ in Fig. 1(b) is optimal (w.r.t. $\text{MRF}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$), as it satisfies this theorem. Visually, this can be understood by observing that there is a cyclic path in Fig. 1(b), passing only through the minimal nodes $(p_1, l_{p_1}), (p_2, l_{p_2}), (p_3, l_{p_3})$. No similar cyclic path exists, however, in Fig. 2(a), despite the fact that the shown dual solution is actually optimal to $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ and satisfies Thm. 2 (this is a case where relaxation $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ is not tight). Indeed, if we start from any minimal node in Fig. 2(a), say, (p_1, a) and keep traversing tight links until we return to that node, we won't be able to do that without first passing from node (p_1, b). We then say that cycle $p_1 p_2 p_3$ is *inconsistent* to (p_1, a). More generally, cycle $p_1 p_2 \ldots p_t p_{t+1}$ (with $p_{t+1} = p_1$) is said to be inconsistent to minimal node (p_1, l_1) if no labeling $\{l_i\}$ exists such that each node (p_i, l_i) is minimal and each link $\{(p_i, l_i), (p_{i+1}, l_{i+1})\}$ is tight (see Fig. 1(c)). This inconsistency is, in fact, one of the main reasons why relaxation $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ is often not tight (*i.e*, why $\mathcal{P}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ may not have integral optimal solutions).

In our quest for better solutions, it is therefore crucial to figure out how to eliminate these inconsistent cycles (without, of course, reducing the dual objective). Unfortunately, as demonstrated by the example in Fig. 2(a), this may not be possible when using relaxation $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. It is possible, however, if the tighter relaxation $\mathcal{D}^{\{\mathcal{C}_i\}}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ is used. As we shall see, the reason is that, in the latter case, if one wishes to modify the residuals (*e.g*, for changing whether a link is tight or not), he may modify for this purpose not only variables $\mathbf{y} = \{y_{pp'}(\cdot)\}$, but also variables $\mathbf{f} = \{f_{pp'}(\cdot)\}$, *i.e*, the virtual potentials. The process of repairing inconsistent cycles by modifying the virtual potentials $\mathbf{f}$ will be named *cycle-repairing*, and the associated routine will be called CYCLE_REPAIR.

Given such a routine, the following iterative strategy can then be followed (see Fig. 3): Initially, we set $\mathbf{f} = \bar{\mathbf{f}}$. Then, at each iteration, we apply successively routines DUAL_ASCENT, CYCLE_REPAIR. The former drives the current dual solution towards satisfying Thm. 2 for the relaxation $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f})$ at the current iteration, but it may create inconsistent cycles. Therefore, the latter tries to repair these inconsistent cycles (thus tightening our relaxation even further), but, for this, it has to modify some residuals of the current dual solution by changing potentials $\mathbf{f}$ into $\mathbf{f}^{\text{next}}$. Due to this, a new relaxation $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f}^{\text{next}})$ is formed, which, however, violates Thm. 2 with respect to the current dual solution (DUAL_ASCENT must therefore be applied to it again at the next iteration). This process, of course, repeats until convergence, *i.e*, until no more inconsistent cycles exist.

Note that both DUAL_ASCENT and CYCLE_REPAIR never reduce the dual objective. We are therefore using a dual ascent procedure, applied, however, not to weak relaxation $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$, but to tighter relaxation $\mathcal{D}^{\{\mathcal{C}_i\}}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$. Each time CYCLE_REPAIR is applied, it helps DUAL_ASCENT to escape from the fixed point it has previously got stuck in and thus further increase the dual objective function the next time it runs. Hence, this essentially results in having a series of weak

---

$\mathbf{f} \leftarrow \bar{\mathbf{f}}$;
repeat
   apply DUAL_ASCENT to relaxation $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f})$;
   get next cycle $\mathcal{C}_i = p_1 p_2 \ldots p_n$; $\mathbf{f}^{\mathrm{next}} \leftarrow \{\text{apply CYCLE\_REPAIR to cycle } \mathcal{C}_i\}$;
   $\mathbf{f} \leftarrow \mathbf{f}^{\mathrm{next}}$;
until all cycles in $\{\mathcal{C}_i\}$ are consistent

Fig. 3. Algorithm's pseudocode

relaxations $\{\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f}^k)\}$ with $\mathbf{f}^0 = \bar{\mathbf{f}}$ and $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f}^k) \leq \mathcal{D}(\bar{\mathbf{g}}, \mathbf{f}^{k+1})$. We now proceed to describe one way of implementing the CYCLE_REPAIR routine.

3.1   An Algorithm for Repairing Cycles

To get an intuition of how CYCLE_REPAIR might work, let us consider the example in Fig. 2(a). The current dual solution is a fixed point of DUAL_ASCENT (*i.e*, it satisfies Thm. 2), yet it has inconsistent cycles. *E.g*, cycle $\mathcal{C} = p_1 p_2 p_3$ is inconsistent to node (p_1, a). One way to repair for this is by trying to allow (p_1, a) to raise its height, thus becoming a non-minimal node. (if we can manage that, (p_1, a) would then no longer be a problem as one cares only about minimal nodes when checking for inconsistent cycles). To this end, it suffices that we find a neighboring object, say, p_2 so that node (p_1, a) has no tight links with any of the minimal nodes in p_2 (indeed, if (p_1, a) has tight links neither with (p_2, a) nor with (p_2, b), then DUAL_ASCENT can later raise the height of (p_1, a) simply by increasing the value of $y_{p_1 p_2}(a)$ until one of these links becomes tight). Since link $\{(p_1, a), (p_2, b)\}$ is already non-tight in Fig. 2(a), we are thus left with the task of making link $\{(p_1, a), (p_2, a)\}$ non-tight as well (*i.e*, making $r_{p_1 p_2}(a, a)$ positive). But how can we achieve that without decreasing the value of the dual objective function[1], *i.e*, without decreasing the height of any minimal node?

This is impossible if we can manipulate only variables $\mathbf{y}$ (*i.e*, if relaxation $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ is used), but it is easy if we can manipulate variables $\mathbf{f}$ as well (*i.e*, if we use a tighter relaxation). Indeed, in the latter case we can make $r_{p_1 p_2}(a, a)$ equal to $\varepsilon > 0$ by simply adding ε to $f_{p_1 p_2}(a, a)$ (see blue segment in Fig. 2(b)). Due to this update, however, we have violated the feasibility constrain $\mathbf{f} \preceq_{\mathcal{C}} \mathbf{f}'$ (where $\mathcal{C} = p_1 p_2 p_3$), since, for $l_{p_1} = l_{p_2} = a$ and any l_{p_3}, it then obviously holds $\sum_{ij} f_{p_i p_j}(l_{p_i}, l_{p_j}) = \sum_{ij} f'_{p_i p_j}(l_{p_i}, l_{p_j}) + \varepsilon$ (here $\mathbf{f}'$ simply represents the virtual potentials $\mathbf{f}$ before adding ε). To restore that constraint, it thus suffices that we add $-\varepsilon$ to both $f_{p_1 p_3}(a, b)$ and $f_{p_2 p_3}(a, a)$ (see red segments in Fig. 2(b)). In summary, the following update must be applied: $f_{p_1 p_2}(a, a) += \varepsilon, f_{p_1 p_3}(a, b) -= \varepsilon, f_{p_2 p_3}(a, a) -= \varepsilon$, which results into $r_{p_1 p_2}(a, a) = \varepsilon, r_{p_1 p_3}(a, b) = r_{p_2 p_3}(a, a) = 0$. Note that we chose to use the value ε because it was the maximum positive value for which the lowered residuals would remain non-negative after the update (as required by (9)), *i.e*, $\varepsilon = \min(r_{p_1 p_3}(a, b), r_{p_2 p_3}(a, a))$.

---

[1] Making the residual $r_{p_1 p_2}(a, a)$ non-zero is trivial if we are allowed to decrease the dual objective function. But then, when we reapply DUAL_ASCENT, we will obtain the same dual solution that we started with (cycle $\mathcal{C}$ will thus remain inconsistent).

A similar procedure can be applied for also repairing cycle $p_1p_2p_3$ with respect to node (p_1, b) (see Fig. 2(c)). The dual solution resulting after the above 2 cycle-repairs appears in Fig. 2(d). As can be seen, it violates the conditions of Thm. 2 (*e.g*, node (p_1, a) or (p_1, b) has no tight links with any of the minimal nodes in p_2). We can therefore reapply DUAL_ASCENT, thus getting the dual solution in Fig. 2(e), which now has no inconsistent cycles and whose cost actually coincides with the global MRF optimum. This means that, contrary to the initial relaxation, the final relaxation after the 2 cycle repairs is tight. In general, each time we repair a cycle, we effectively make use of additional constraints and we thus tighten our current relaxation.

For the general case, CYCLE_REPAIR may essentially mimic the procedure described above. Let thus $\mathcal{C} = p_1p_2 \ldots p_n$ (with $p_1 = p_n$) be an inconsistent cycle, e.g., with respect to minimal node (p_1, l_1), which will be called the *anchor node* hereafter. One way for CYCLE_REPAIR to repair this inconsistent cycle is to allow the anchor node to later raise its height. For this, as explained in the previous example, it suffices that no tight link exists between the anchor node and any of the minimal nodes in the adjacent object p_2. This, in turn, means that if any such link exists, a positive value ε must be added to its associated virtual potential (the link would then be non-tight, as its residual would be equal to $\varepsilon > 0$). Hence, for all minimal nodes (p_2, l_2) of object p_2 that satisfy $r_{p_1p_2}(l_1, l_2) = 0$, we need to apply the following update:

$$f_{p_1p_2}(l_1, l_2) += \varepsilon \qquad (12)$$

As a result of this, however, feasibility constraint $\mathbf{f} \preceq_{\mathcal{C}} \mathbf{f}'$ is now violated (where $\mathbf{f}'$ simply denotes the virtual potentials before the update). To restore feasibility, we thus have to add $-\varepsilon$ to the virtual potentials of some subset $\mathcal{S}$ of non-tight links (note that the virtual potential of tight links cannot decrease, as this would result into negative residuals, thus violating feasibility). It is not difficult to verify that this set $\mathcal{S}$ suffices to contain all non-tight links $\{(p_k, l_k), (p_{k+1}, l_{k+1})\}$ between an *anchored* node (p_k, l_k) (where $2 \leq k \leq n - 1$) and a *non-anchored* node (p_{k+1}, l_{k+1}) (where $l_{k+1} = l_1$ if $k + 1 = n$). Here we say that a node (p_k, l_k) is *anchored* if only if there exists a path $(p_1, l_1) \rightarrow (p_2, l_2) \rightarrow \ldots \rightarrow (p_k, l_k)$ from the anchor node (p_1, l_1) to node (p_k, l_k) such that all the path links are tight (before applying (12)) and all the path nodes are minimal (*e.g*, in Fig. 2(b), the anchor node is (p_1, a) and the set S consists of the red links). The following transformation must then be applied:

$$\forall \{(p_k, l_k), (p_{k+1}, l_{k+1})\} \in \mathcal{S}, \ f_{p_kp_{k+1}}(l_k, l_{k+1}) -= \varepsilon.$$

A possible choice for ε is to set it equal to any positive value that ensures that none of the decreased residuals of the links in $\mathcal{S}$ will become negative, *e.g*:

$$\varepsilon = \min_{\{(p_k, l_k),(p_{k+1}, l_{k+1})\} \in \mathcal{S}} r_{p_kp_{k+1}}(l_k, l_{k+1}). \qquad (13)$$

Note that, since $\mathcal{C}$ is inconsistent, such a positive ε will always exist.

Finally, for implementing the DUAL_ASCENT routine, one of the ways to achieve that is to use the *augmenting DAG algorithm* [6]. This is an iterative

algorithm that increases the dual objective by keep perturbing the current dual solution until it satisfies Thm. 2. To this end, it uses a procedure based on a so called augmenting directed acyclic graph (DAG) (this somewhat resembles the augmenting path algorithm for max-flow/min-cut). In our case, an additional advantage comes from the fact that the change from $\mathbf{f}$ to $\mathbf{f}^{\text{next}}$ after each cycle repair is always small (*i.e*, local), which means that the change from the augmenting DAG of relaxation $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f})$ to that of $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f}^{\text{next}})$ will be small as well. Hence, the convergence of DUAL_ASCENT for $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f}^{\text{next}})$ will be extremely fast, given that this routine has already converged for $\mathcal{D}(\bar{\mathbf{g}}, \mathbf{f})$ (this somewhat resembles the case of path augmenting max flow algorithms when applied to dynamic graphs).

After our algorithm in Fig. 3 converges, we have to compute an MRF labeling based on the final dual solution, *i.e*, the final heights and residuals. To this end, one can traverse all objects in some predefined order, say, $p_1, p_2, \ldots, p_n$, and, then assign to each object p_i a label $\hat{l}_i$ that satisfies the following criteria: it has minimal height (*i.e*, node $(p_i, \hat{l}_i)$ is minimal) and also minimizes the non-negative sum $\sum_{j<i} r_{p_i p_j}(\hat{l}_i, \hat{l}_j)$. The justification for this procedure comes from the next theorem, which generalizes Thm. 3:[2]

Theorem 4 ([12]). *There always exist an optimal labeling* $\hat{\mathbf{l}} = \{\hat{l}_p\}$ *of* $\mathrm{MRF}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ *and an optimal dual solution* $(\mathbf{h}, \mathbf{r})$ *of* $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ *such that each node* $(p, \hat{l}_p)$ *is minimal, i.e.,* $h_p(\hat{l}_p) = \min_l h_p(l)$, *and all links* $\{(p, \hat{l}_p), (p', \hat{l}_{p'})\}$ *are tight, i.e.,* $\sum_{pp'} r_{pp'}(\hat{l}_p, \hat{l}_{p'}) = 0$ *(thus* $\sum_{pp'} r_{pp'}(\hat{l}_p, \hat{l}_{p'})$ *attains its minimum possible value).*

4   Experimental Results

To verify our algorithm's effectiveness, we tested it on various MRF problems. We begun by applying it on a standard set of benchmark MRF problems from vision [13] and Fig. 4 shows some of our results. For instance, Figures 4(a), 4(b) show stereo matching results from the widely used Middlebury stereo data set, Fig. 4(c) contains a result related to image inpainting and denoising, Fig. 4(d) displays an image segmentation result, while Fig. 4(e) contains a result from a panoramic stitching application. Similarly, Fig. 4(f) shows a result from a group photo merging application. In all cases, our method yielded solutions which were either optimal or had energy that was extremely close to the optimum (this can be proved due to that the dual costs always form a lower bound on the minimum MRF energy). For instance, the worst accuracy of the labelings obtained for all MRFs was 0.0071 (where accuracy is defined as $\frac{\text{Energy of labeling} - \text{best lower bound}}{\text{best lower bound}}$).

Next, in order to do a more thorough analysis and to see how our algorithm behaves in much more difficult cases, we went on and tested it on a set of hard synthetic MRF problems. We thus begun by conducting a full set of experiments on binary MRF problems, which are often used as subroutines by many popular MRF algorithms like [14],[15]. The binary MRFs were defined on 4-connected

---

[2] Thm. 3 provides only a sufficient (but not a necessary) condition for optimality, since it refers to a dual solution of $\mathcal{D}(\bar{\mathbf{g}}, \bar{\mathbf{f}})$ and not of $\mathcal{D}^+(\bar{\mathbf{g}}, \bar{\mathbf{f}})$.
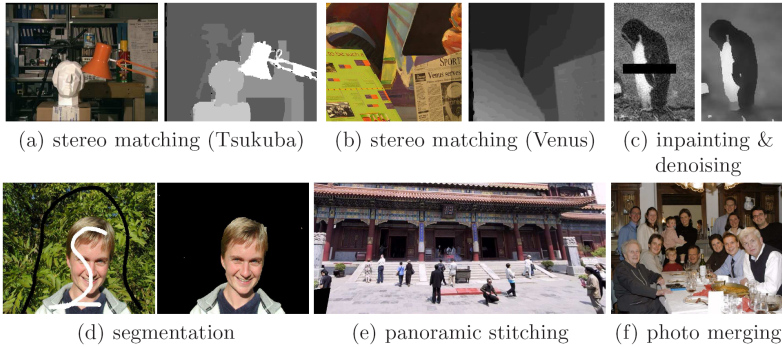
(a) stereo matching (Tsukuba)   (b) stereo matching (Venus)   (c) inpainting &
denoising

(d) segmentation   (e) panoramic stitching   (f) photo merging

Fig. 4. Results from the benchmark MRF problems in [13]

(a) $\rho = 1\%$
non-attractive terms

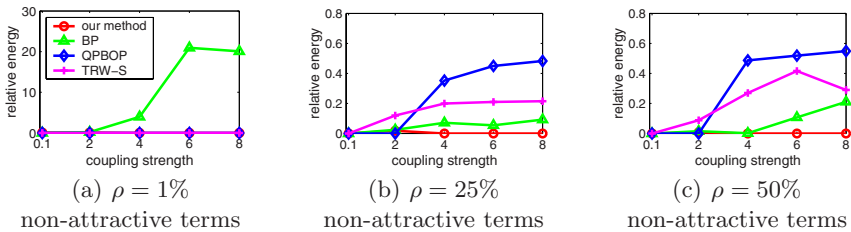(b) $\rho = 25\%$
non-attractive terms

(c) $\rho = 50\%$
non-attractive terms

Fig. 5. Comparison results for binary MRFs with different percentage ρ of non-attractive couplings. For each value of ρ (*i.e*, for each plot), we also vary the coupling strength σ (horizontal axis). The vertical axis represents relative MRF energies (for an algorithm with energy E, its relative energy is $(E - E_{\min})/|E_{\min}|$, where $E_{\min}$ equals the minimum energy among all tested algorithms).

30×30 grids and had different percentages of non-attractive couplings[3]. Unary potentials $\bar{\mathbf{g}}_p(\cdot)$ were drawn independently from a Gaussian distribution $\mathcal{N}(0, 1)$, while pairwise potentials were set as: $\bar{f}_{pp'}(0, 0) = \bar{f}_{pp'}(1, 1) = 0$, $\bar{f}_{pp'}(0, 1) = \bar{f}_{pp'}(1, 0) = \lambda_{pp'}$, with $\lambda_{pp'}$ drawn from $|\mathcal{N}(0, \sigma^2)|$ for attractive couplings and from $-|\mathcal{N}(0, \sigma^2)|$ for non-attractive couplings. The percentage of non-attractive couplings was controlled by parameter ρ, which was set to 1%, 25% and 50% in 3 different experiments[4]. Also, for each experiment, we allowed parameter σ (that controls the strength of the coupling) to vary and take values from $\{0.1, 2, 4, 6, 8\}$. These two parameters, ρ and σ, affect the difficulty of the resulting MRF.

Along with our algorithm, the BP, TRW-S [4] and QPBOP [16] algorithms have been tested as well[5]. The results appear in Fig. 5, where each point on a plot corresponds to the average of 20 random MRF instances. As can be seen, our algorithm had the best performance in all cases (*i.e*, for all combinations of ρ and σ). Its running times varied depending on the binary MRF's difficulty and

---

[3] For binary (but not for multi-label) MRFs, attractive is equivalent to submodular.
[4] By flipping labels, cases $\rho = 50 + \varepsilon$ and $\rho = 50 - \varepsilon$ reduce to each other.
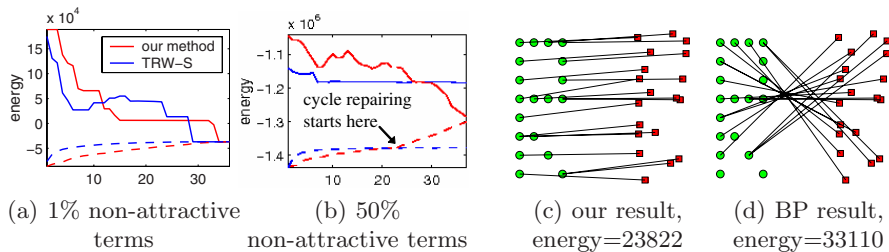[5] We also tested the augmenting DAG method, but it performed similarly to TRW-S.

Fig. 6. (a) Upper (solid) and lower (dashed) bounds after running TRW-S and our method on 2 different binary MRFs. Such bounds correspond to primal costs (*i.e*, MRF energies) and dual costs respectively. For a binary MRF with only 1% non-attractive terms, the upper and lower bounds of TRW-S converge (*i.e*, LP-relaxation (8) is tight). **(b)** This is not the case, however, for a binary MRF with 50% non-attractive terms. But, thanks to cycle repairing, our algorithm uses a much tighter relaxation and so its upper and lower bounds converge even in this case. **(c)** An example of non-rigid point matching. Source points (red) were generated by applying a global rigid motion plus a random non-rigid perturbation to the target points (green). Despite the heavy distortion, our algorithm succeeded to find a meaningful set of point correspondences. **(d)** On the other hand, BP failed as it did not manage to effectively minimize the MRF energy.

were up to 1.5 secs using heavily unoptimized code on a 1.4GHz CPU (contrary, e.g., to 0.53 secs for QPBOP). Note that the QPBOP algorithm did not perform very well in this case (and we speculate that this can be due to a possibly large percentage of unlabeled nodes). Also note that the method from [7] cannot be applied here, due to the large number of ties that would exist.

As can be seen, in the case of binary MRFs, it is only for extremely small ρ (*i.e*, for almost submodular MRFs) that methods such as TRW-S (which was a top performer in [13]) are able to perform equally well with our algorithm. The reason is that, in this case, the standard LP relaxation upon which such methods rely is already a very good approximation to the MRF problem (recall that, in the special case of binary MRFs that are submodular, TRW-S can compute the global optimum, just like our method). This is also shown in more detail in Fig. 6(a), where the lower and upper bounds (generated while applying TRW-S and our algorithm to a binary MRF with $\rho = 1\%$) have been plotted. In this case, the lower/upper bounds of TRW-S correspond to dual/primal costs of the standard LP relaxation. Since, as shown in Fig. 6(a), these bounds converge, this, therefore, implies that the LP relaxation is tight.

However, as can be observed in Fig. 5(c), the situation is completely different when a mixed MRF (with, say, $\rho = 50\%$) is to be optimized. These are really very hard problems to solve, for which TRW-S performs poorly. Fig. 6(b) shows in more detail what is really happening in this case. Contrary to the plot in Fig. 6(a), the upper and lower bounds of TRW-S no longer converge, and the reason, of course, is that the standard LP relaxation is now not tight. However, this is not at all an issue for our algorithm, thanks to the cycle repairing that it uses. This effectively enables him to take advantage of a much tighter relaxation,
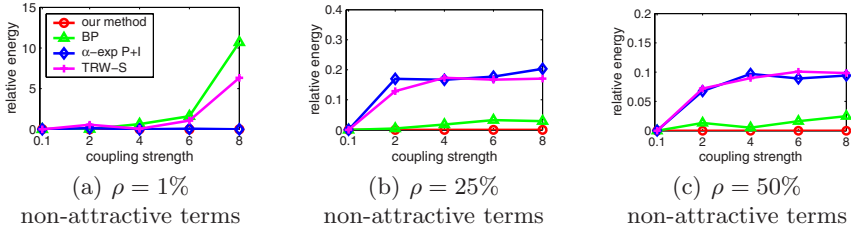
(a) $\rho = 1\%$ non-attractive terms

(b) $\rho = 25\%$ non-attractive terms

(c) $\rho = 50\%$ non-attractive terms

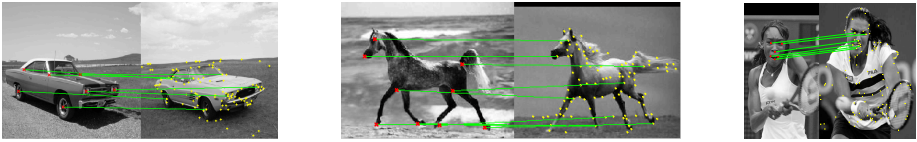**Fig. 7.** Comparison results for multi-label MRFs

Fig. 8. Deformable feature matching via MRF optimization. From user selected features in a source image, we seek correspondences to automatically extracted features in a target image (feature similarity and geometric structure are both taken into account).

which yields upper and lower bounds far closer to each other (*i.e*, which almost converge). As a result, a much better optimum is obtained as well.

Using a procedure similar to the binary case, we also conducted another full set of experiments for multi-label MRFs, which are even more popular in vision. The results are shown in Fig. 7 for various combinations of ρ and σ and when 5 labels were used. Our algorithm had again the best performance among all other methods and did much better than TRW-S (which was a top performer in [13]) in all cases (and even for very small ρ). Note that QPBOP, unlike our method, is applicable only to binary MRFs, and, so, it had to be replaced with the α-exp P+I algorithm proposed in [16] (note that this is essentially a generalization of α-expansion with a QPBOP-based technique being used as the binary optimizer at each step). Note, also, that the set of cycles $\{\mathcal{C}_i\}$ (used by our algorithm for the cycle-repairing) consisted only of 1×1, 1×2 and 2×1 rectangles on the grid (these were the settings used in all the grid-based MRFs of this paper).

We also considered the problem of subgraph matching, where one seeks a mapping $\mathcal{T}$ between a source and a target graph. This can be formulated as a multi-label MRF problem, where source vertices correspond to MRF sites, while each target vertex represents one possible label. The unary potential $\bar{g}_p(\mathcal{T}(p))$ thus measures the "dissimilarity" between source vertex p and target vertex $\mathcal{T}(p)$, while potential $\bar{f}_{pp'}(\mathcal{T}(p), \mathcal{T}(p'))$ measures the "dissimilarity" between source edge pp' and target edge $\mathcal{T}(p)\mathcal{T}(p')$. We used this formulation for the purpose of non-rigid registration between geometric point sets. In this case, the unary potentials are all set to zero (as there is no notion of similarity between mere geometric points), while the pairwise potentials (of the fully connected MRF) measure the geometric distortion for each pair of source points as follows:

$$\bar{f}_{pp'}(\mathcal{T}(p), \mathcal{T}(p')) = |d(p, p') - d(\mathcal{T}(p), \mathcal{T}(p'))|/d(p, p'),$$

where $d(\cdot, \cdot)$ denotes Euclidean distance. Results from applying our method and BP to this problem are shown in Figures 6(c), 6(d). Contrary to BP, which failed to effectively minimize the MRF energy in this case, our algorithm managed to find a good minimum and thus extracted a good point matching despite the heavy geometric distortion (the result of α-exp P+I algorithm is even worse and is thus not shown). Note that, since a fully connected MRF has been used, our algorithm was allowed to use triplets of vertices as the set $\{\mathcal{C}_i\}$ of possible cycles to repair. Similarly, we have applied our registration algorithm to the problem of deformable feature matching between images (see Fig. 8), where the unary potentials (instead of being 0) are now based on comparing *geometric blur* descriptors.

5   Conclusions

We presented a novel LP-based algorithm for MRF optimization, which, contrary to most state-of-the-art techniques, no longer relies on a weak LP-relaxation. To this end, an hierarchy of dual relaxations (that become tighter and tighter) has been introduced, while it was shown how one can take advantage of a very tight class of relaxations from this hierarchy, named cycle-relaxations. This is done through an operation called cycled repairing, where inconsistent cycles get repaired during optimization, instead of simply being ignored as was the case up to now. The more the repaired cycles, the tighter the underlying relaxation becomes. The outcome is an algorithm that can handle difficult MRF problems with arbitrary pairwise potentials. The impact of our work can be both practical and theoretical. On the one hand, the proposed method makes one more step towards the design of optimization algorithms that can effectively handle very general MRFs, thus allowing users to freely exploit such MRFs for a better modeling of future vision problems. On the other hand, for a further advance to be made in MRF optimization, the use of tighter relaxations is a direction that, we believe, has to be followed. To this end, the idea of cycle-repairing, as well as that of a dynamic hierarchy of dual relaxations (of which only the cycle-relaxations have been explored here), provide an extremely flexible and general framework that can be useful in opening new research avenues in this regard.

References

1. Komodakis, N., Tziritas, G.: Approximate labeling via graph-cuts based on linear programming. In: PAMI (2007)
2. Komodakis, N., Paragios, N., Tziritas, G.: MRF Optimization via Dual Decomposition: Message-passing revisited. In: ICCV (2007)
3. Wainwright, M.J., Jaakkola, T.S., Willsky, A.S.: Map estimation via agreement on (hyper)trees: Message-passing and linear programming. IEEE Trans. on Information Theory (2005)
4. Kolmogorov, V.: Convergent Tree-reweighted Message Passing for Energy Minimization. PAMI 28(10) (2006)
5. Komodakis, N., Tziritas, G., Paragios, N.: Fast Approximately Optimal Solutions for Single and Dynamic MRFs. In: CVPR (2007)

6. Werner, T.: A Linear Programming Approach to Max-Sum Problem: A Review. PAMI 29(7) (2007)
7. Yanover, C., Meltzer, T., Weiss, Y.: Linear Programming Relaxations and Belief Propagation – An Empirical Study. JMLR 7 (2006)
8. Sontag, D., Jaakkola, T.: New outer bounds on the marginal polytope. In: NIPS (2008)
9. Werner, T.: High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In: CVPR (2008)
10. Sontag, D., Meltzer, T., Globerson, A., Jaakkola, T., Weiss, Y.: Tightening LP relaxations for MAP using message passing. In: UAI (2008)
11. Kumar, M.P., Kolmogorov, V., Torr, P.H.S.: An analysis of convex relaxations for MAP estimation. In: NIPS (2007)
12. Komodakis, N., Paragios, N.: Beyond loose LP-relaxations for MRF optimization. Technical report, Ecole Centrale Paris (2008)
13. Szeliski, R., Zabih, R., et al.: A Comparative Study of Energy Minimization Methods for Markov Random Fields. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 16–29. Springer, Heidelberg (2006)
14. Boykov, Y., Veksler, O., Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. In: PAMI (November 2001)
15. Raj, A., Singh, G., Zabih, R.: MRF's for MRI's: Bayesian Reconstruction of MR Images via Graph Cuts. In: CVPR (2006)
16. Rother, C., Kolmogorov, V., Lempitsky, V., Szummer, M.: Optimizing binary MRFs via extended roof duality. In: CVPR (2007)