

# A Graph Cut Algorithm for Higher-order Markov Random Fields

Alexander Fix  
Cornell University

Aritanan Gruber  
Rutgers University

Endre Boros  
Rutgers University

Ramin Zabih  
Cornell University

## Abstract

*Higher-order Markov Random Fields, which can capture important properties of natural images, have become increasingly important in computer vision. While graph cuts work well for first-order MRF's, until recently they have rarely been effective for higher-order MRF's. Ishikawa's graph cut technique [8, 9] shows great promise for many higher-order MRF's. His method transforms an arbitrary higher-order MRF with binary labels into a first-order one with the same minima. If all the terms are submodular the exact solution can be easily found; otherwise, pseudo-boolean optimization techniques can produce an optimal labeling for a subset of the variables. We present a new transformation with better performance than [8, 9], both theoretically and experimentally. While [8, 9] transforms each higher-order term independently, we transform a group of terms at once. For  $n$  binary variables, each of which appears in terms with  $k$  other variables, at worst we produce  $n$  non-submodular terms, while [8, 9] produces  $O(nk)$ . We identify a local completeness property that makes our method perform even better, and show that under certain assumptions several important vision problems (including common variants of fusion moves) have this property. Running on the same field of experts dataset used in [8, 9] we optimally label significantly more variables (96% versus 80%) and converge more rapidly to a lower energy. Preliminary experiments suggest that some other higher-order MRF's used in stereo [20] and segmentation [1] are also locally complete and would thus benefit from our work.*

## 1. Introduction

Higher-order MRF's have become increasingly important in computer vision, as they can incorporate sophisticated priors such as fields of experts (FoE) [17]. To perform inference in higher-order MRF's, however, requires minimizing significantly more complex energy functions than for standard first-order models. Graph cuts are a popular method for solving first-order MRF's, such as the benchmarks described in [19], but are much more difficult to apply to higher-order MRF's. As a result, until recently this

powerful optimization method has only been used for a few specialized higher-order MRF's, such as [10, 20].

Ishikawa [8, 9] proposed a general-purpose graph cut technique for arbitrary higher-order MRF's that shows great promise. As an application he considered fields of experts for denoising, an important higher-order MRF that is typically solved by belief propagation (BP) [14]. Prior to [8, 9] there was no graph cut method that could successfully handle such an energy function. Ishikawa's graph cut method produced better results than the BP method of [14], and ran an order of magnitude faster [8, 9].

In this paper we propose an alternative construction to Ishikawa's, with improved theoretical and experimental performance. Instead of considering terms in the energy function one at a time, we consider many terms at once. We review existing methods for solving higher-order MRF's with graph cuts in section 2, focusing on methods that directly compete with our work [4, 8, 9, 12, 16, 18]; the comparison is summarized in figure 1. We present our new algorithm in section 3, and analyze its worst case performance in section 4. In section 5 we show that for problems with a specific form of local completeness our method performs even better. Under certain assumptions we prove that some important vision problems are locally complete, including the fields of experts MRF considered by Ishikawa. Experimental results are given in section 7, along with experimental evidence that other vision problems [1, 20] are also locally complete. Some technical details and additional images are provided in the supplementary material.

## 2. Related work

### 2.1. Graph cuts and QPBO

Graph cut methods solve energy minimization problems by constructing a graph and computing the min cut. The most popular graph cut methods, such as the expansion move algorithm of [3], repeatedly solve an optimization problem over binary variables. Such problems have been extensively studied in the operations research community, where they are referred to as pseudoboolean optimization [6] (see [2] for a more detailed survey).

Minimizing an arbitrary binary energy function is NP-hard (see e.g. [12]). Certain binary optimization problems

	Substitution §2.3.1	Negative §2.3.2	Positive §2.3.3	Ours (worst case)	Ours (local completeness $c$ )
New variables	$O(nk)$	$t$	$O(td)$	$n + O(td)$	$n + O(\frac{1}{c}t)$
Non-submodular terms	$O(nk)$	0	$O(nk)$	$n$	$n$
Submodular terms	$O(nk)$	$td$	$O(td^2)$	$O(td^2)$	$O(\frac{1}{c}td)$

Figure 1. Resources required to reduce  $t$  terms of degrees up to  $d$ , for an energy function with  $n$  variables each of which occurs with up to  $k$  other variables. For each existing method we also provide the section in this paper where it is described.

can be solved exactly by min cut; these include computing the optimal expansion move [3] and its generalization to fusion moves [15], as well as a number of other interesting problems such as [10]. The most important class of optimization problems that can be solved exactly with graph cuts are quadratic functions where every term is submodular [2, 5, 12]. Such a function can be written as a polynomial in the binary variables  $(x_1, \dots, x_n) \in \mathbb{B}^n$  in the form

$$\sum_i \alpha_i x_i + \sum_{i,j} \alpha_{i,j} x_i x_j. \quad (1)$$

This function is submodular just in case  $\alpha_{i,j} < 0$  for every  $i, j$ . Note that in practice an individual variable  $x_i$  will appear in terms with several other variables; our algorithm will exploit this property to obtain better performance, even in the worst case.

The most widely used technique for minimizing binary energy functions with non-submodular terms relies on the roof duality technique of [7] and the associated graph construction [2]. This method, commonly referred to as QPBO [11], uses min cut to compute the global minimum for any submodular energy function and for certain other functions as well (see [2, 11] for a discussion). Even when QPBO does not compute the global minimum, it provides a partial optimality guarantee called persistency [7]; QPBO computes a partial assignment which, when applied to an arbitrary complete assignment, will never cause the energy to increase. Such a partial assignment gives the variables it labels their values in the global minimum. Efficient techniques for computing persistencies, along with generalizations and implementations were proposed in [2, 11].

## 2.2. Higher-order MRF's

Higher-order MRF's, of course, have a long history in computer vision. They have recently become more popular, especially due to models such as fields of experts [17]. Message passing approaches, such as belief propagation or dual decomposition [14, 13], can be used for higher-order MRF's, but their efficiency is a concern. For first-order MRF's, the performance of graph cut methods is competitive with message passing, and sometimes exceeds it [19]. This has lead to significant interest in applying graph cuts to higher-order MRF's.

QPBO represents the state of the art in graph cuts, but it is restricted to handling energy functions of the form

given in equation (1), i.e. quadratic functions of binary variables. Popular move making algorithms such as expansion moves [3] or fusion moves [15], which are very successful on first-order MRF's, generate a series of binary optimization problems. These moves can also be applied to higher-order MRF's, and the resulting binary optimization problem involves a multilinear polynomial instead of a quadratic one. So instead of equation (1) we also have terms like  $\sum_{i,j,k} \alpha_{i,j,k} x_i x_j x_k$  and so on. In summary, the the main technical challenge with higher-order MRF's is to reduce this multilinear polynomial into quadratic form; if the resulting quadratic is submodular it can be solved exactly, while otherwise QPBO can be used to identify persistencies.

Of course it is preferable to identify moves for higher-order MRF's where the binary optimization problem can be reduced to a submodular quadratic one and thus solved exactly. This approach has proven successful for some cases, notably the  $\mathcal{P}^n$  model of [10]. However, this has only been done for a limited class of higher-order MRF's. For arbitrary higher-order MRF's the natural approach is reduction to a quadratic followed by QPBO.

## 2.3. Reduction techniques

There are a number of methods for reducing an arbitrary multilinear polynomial over binary variables into a quadratic one. Our primary contribution is to introduce a new such transformation; the performance of the different methods is summarized in figure 1.

Until fairly recently there was only one general-purpose reduction, discussed in section 2.3.1, which is widely viewed as impractical. Another transformation, which only works for terms with negative coefficients, is described in section 2.3.2; this transformation was recently used for stereo reconstruction [20]. Ishikawa's method, which we discuss in section 2.3.3, fills the gap by reducing terms with positive coefficients as well, and thus allows arbitrary higher-order MRF's to be handled.

### 2.3.1 Reduction by substitution

The original reduction was introduced by Rosenberg [16]. To reduce higher-order terms, the reduction eliminates all occurrences of some product  $xy$  by introducing a new variable  $z$ , replacing  $xy$  by  $z$  everywhere it occurs, and then adding the following penalty terms to the energy function:

$Mxy - 2Mxz - 2Myz + 3Mz$ , where  $M$  is a suitably large constant (larger than the highest value of the unmodified energy function). This forces  $z$  to take the value of  $xy$  in any optimal solution.

The complexity of the reduction depends on finding pairs of variables to substitute that eventually reduce all terms. If each variable is in terms with at most  $k$  other variables, this can be done with  $O(nk)$  pairs. Overall, we get  $O(nk)$  new variables, non-submodular terms and submodular quadratic terms.

Note that the non-submodular terms have large coefficients. Experimentally it has been reported that QPBO performs very poorly on such energy functions (see, for example, [9, §8.3.4], which states that QPBO finds almost no persistencies). This is consistent with our experience, and as a practical matter no one in the vision literature appears to have obtained good results with this method.

### 2.3.2 Reducing negative-coefficient terms

Kolmogorov and Zabih [12] for  $d = 3$  and Freedman and Drineas [4] for  $d \geq 3$  suggested the following transformation for negative higher degree terms:

$$-x_1 \cdots x_d = \min_{y \in \mathbb{B}} y \left( (d-1) - \sum_{j=1}^d x_j \right)$$

If we have  $t$  negative-coefficient terms of degree  $d$ , this gives  $t$  new variables and  $td$  submodular quadratic terms, but no new non-submodular terms.

Let us note that the above equality remains valid even if we replace some of the  $x_j$  variables with their complements  $\bar{x}_j = (1 - x_j)$ . Rother [18] suggested recently a transformation equivalent with this observation (see type-II transformations in [18]) together with a new transformation (see type-I in [18]). If we have  $t$  negative coefficient terms, the new transformation results in  $2t$  new variables and  $td$  submodular quadratic terms, as well as  $t$  non-submodular terms.

### 2.3.3 Reducing positive-coefficient terms

Recently Ishikawa [8, 9] suggested the following transformation for positive monomials: let  $n_d = \lfloor \frac{d-1}{2} \rfloor$  and set  $c_{i,d} = 1$  if  $d = i$  is odd, and  $c_{i,d} = 2$  otherwise. Then,

$$x_1 \cdots x_d = \min_{u_1, \dots, u_d} \sum_{i=1}^{n_d} u_i \left( c_{i,d} \left( -\sum_{j=1}^d x_j + 2i \right) - 1 \right) + \sum_{i < j} x_i x_j$$

Since all the non-submodular terms produced are of products of the original variables (i.e., they don't involve

new variables), their number is  $O(n^2)$ . Furthermore, if each variable occurs in terms with at most  $k$  other variables, then the number of non-submodular terms is  $O(nk)$ . There will be no cancellation of the submodular terms, since they all involve new variables which are distinct for each term. With  $t$  positive-coefficient terms of degree  $d$  this transformation gives  $t \lfloor \frac{d-1}{2} \rfloor$  new variables,  $t \lfloor \frac{d-1}{2} \rfloor d$  submodular terms, and  $O(nk)$  non-submodular terms.

## 3. Reducing groups of higher-order terms

Our main results are two theorems, each suited to dealing with positive or negative coefficient terms respectively. In contrast to earlier methods which reduce term-by-term, our new method reduces a group of terms all at once. The terms of a multilinear polynomial form a hypergraph  $\mathcal{H}$ . The vertices are the polynomial's variables, and there is a hyperedge  $H = \{x_1, \dots, x_d\}$  with weight  $\alpha_H$  whenever the polynomial has a term  $\alpha_H x_1 \cdots x_d$ .

The two theorems are both concerned with reducing all positive or negative terms containing a single variable, or small set of variables; we will write this common subset of variables as  $C$ . The most important special case of our reduction is shown in figure 2, where we consider positive terms containing the single variable  $C = \{x_1\}$ .

**Theorem 3.1** *Consider a set  $\mathcal{H}$  of terms each of which contains a common subset of variables  $C$ , whose hyperedges  $H$  have positive weights  $\alpha_H > 0$ . For any assignment of the boolean variables  $x_1, \dots, x_n$ ,*

$$\sum_{H \in \mathcal{H}} \alpha_H \prod_{j \in H} x_j = \min_{y \in \{0,1\}} \left( \sum_{H \in \mathcal{H}} \alpha_H \right) y \prod_{j \in C} x_j + \sum_{H \in \mathcal{H}} \alpha_H \bar{y} \prod_{j \in H \setminus C} x_j. \quad (2)$$

**PROOF:** Given any assignment of the variables  $x_1, \dots, x_n$ , either (case 1) all the variables in  $C$  are 1, or (case 2) some variable in  $C$  is 0.

Case 1: The LHS is  $\sum_{H \in \mathcal{H}} \alpha_H \prod_{j \in H \setminus C} x_j$  and the RHS is  $\min_y (\sum_{H \in \mathcal{H}} \alpha_H) y + \sum_{H \in \mathcal{H}} \alpha_H \bar{y} \prod_{j \in H \setminus C} x_j$ . If we assign  $y = 1$ , then the RHS becomes  $\sum_{H \in \mathcal{H}} \alpha_H$ , and if we assign  $y = 0$ , then it becomes  $\sum_{H \in \mathcal{H}} \alpha_H \prod_{j \in H \setminus C} x_j$ . This quantity is always less than or equal to  $\sum_{H \in \mathcal{H}} \alpha_H$ , so the minimum is achieved when  $y = 0$ , in which case, the LHS is equal to the RHS.

Case 2: The product  $\prod_{j \in C} x_j$  is 0. Since all the terms on the LHS share the common subset  $C$ , the LHS is 0. Since  $\prod_{j \in C} x_j = 0$ , the RHS is  $\sum_{H \in \mathcal{H}} \alpha_H \bar{y} \prod_{j \in H \setminus C} x_j$ . If we assign  $y = 1$ , then this sum is 0, whereas if we assign  $y = 0$ , then it is positive, since each  $\alpha_H$  is positive. Thus, the minimum is achieved when  $y = 1$ , in which case the RHS is 0, and hence equal to the LHS. ■

For every positive term containing the common subset  $C$ , equation (2) replaces it with a new term  $\alpha_H \bar{y} \prod_{j \in H \setminus C} x_j$ . Since we are working with multilinear polynomials, we substitute  $\bar{y} = 1 - y$  to split this term into two:  $\alpha_H \prod_{j \in H \setminus C} x_j$  and  $-\alpha_H y \prod_{j \in H \setminus C} x_j$ . This gives us the following corollary, which summarizes the performance of our reduction.

**Corollary 3.2** *When we apply equation (2) to a positive term, we obtain a positive term of smaller degree, and a negative term with  $y$  replacing the common subset  $C$ .*

For reducing the negative-coefficient terms all sharing some common subset, we have a similar theorem.

**Theorem 3.3** *Consider  $\mathcal{H}$  and  $C$  as above, where now the coefficients  $\alpha_H$  are negative for all  $H$ . Then for any assignment of the variables  $x_1, \dots, x_n$*

$$\sum_{H \in \mathcal{H}} \alpha_H \prod_{j \in H} x_j = \min_{y \in \{0,1\}} \sum_{H \in \mathcal{H}} -\alpha_H \left( 1 - \prod_{j \in C} x_j - \prod_{j \in H \setminus C} x_j \right) y \quad (3)$$

**PROOF:** The proof is similar to the proof for Theorem 3.1. The minimum is achieved when  $y = \prod_{j \in C} x_j$ . ■

A crucial difference between this reduction and theorem 3.1 is that in the positive case, we could let the common subset  $C$  be a single variable. Doing this here removes the term  $\alpha_H \prod_{j \in H} x_j$  and replaces it with  $\alpha_H y \prod_{j \in H \setminus \{1\}} x_j$ , another negative term of the same degree. Trying to apply this reduction repeatedly will thus never terminate. However, if  $C$  consists of two or more variables, then grouping all terms containing  $C$  and reducing results in smaller degree terms replacing every term that we start with.

### 3.1. Our method

Equations (2) and (3) can be used for different reduction strategies. Both depend upon the choice of common variables  $C$ , although as noted above we need  $|C| > 1$  to apply equation (3). Besides choosing  $|C|$ , we can also decide the order to consider different choices of  $C$ ; for example, which single variable to use to apply equation (2), or which pair of variables to use to apply equation (3).

We will focus on the simplest case: we let the common part  $C$  be a single variable in the order  $x_1, \dots, x_n$ , and reducing positive terms containing this variable via equation (2). Negative terms will be reduced using the method described in section 2.3.2. More complicated schemes are also possible, such as picking pairs of variables and reducing both positive and negative terms containing this pair via equations (2) and (3).

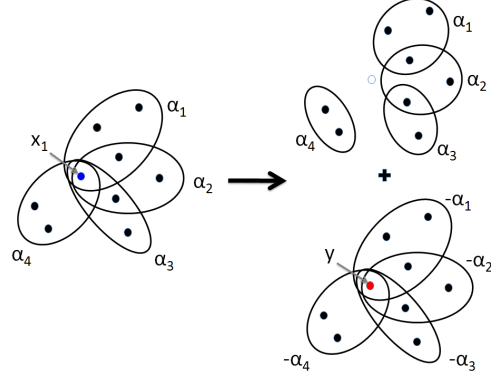


Figure 2. Our main reduction. At left are all the original positive terms containing the common variable  $x_1$  (so  $\alpha_i > 0$ ). At right are all the new terms we obtain from equation (2). The positive terms on top are just the original terms minus  $x_1$ , and the negative terms on bottom are the original terms with  $y$  replacing  $x_1$ .

Our method reduces a multilinear polynomial with higher-order terms, to quadratic form in two steps:

**Step 1.** Eliminate all positive terms by repeated application of Theorem 3.1, with the common subset  $C$  set to a single variable  $x_1$ . Gather all terms containing  $x_1$ , and replace them with the RHS of equation (2). If  $\mathcal{H}$  consists of all positive terms containing  $x_1$ , then

$$\begin{aligned} \sum_{H \in \mathcal{H}} \alpha_H \prod_{j \in H} x_j &= \min_{y \in \{0,1\}} \left( \sum_{H \in \mathcal{H}} \alpha_H \right) x_1 y \\ &+ \sum_{H \in \mathcal{H}} \alpha_H \prod_{j \in H \setminus \{1\}} x_j \\ &- \sum_{H \in \mathcal{H}} \alpha_H y \prod_{j \in H \setminus \{1\}} x_j \end{aligned} \quad (4)$$

The positive terms now form a hypergraph on one fewer variable, so repeat with  $x_2, \dots, x_n$  until all positive terms are reduced.

**Step 2.** All higher-order terms now have negative coefficients. Reduce them term-by-term using the methods in section 2.3.2.

Note that equation (4) is simply the special case of equation (2) for a single variable. This special case, which is illustrated in figure 2, is the one used for our experiments.

## 4. Worst case performance

For every application of equation 4, we get a single new variable and a positive quadratic term. Additionally, for each term we reduce by this rule, we get a positive term on the same variables minus  $x_1$ , and a new negative term with  $y$  replacing  $x_1$ .

The negative term will never cancel or combine with any existing term, since it contains the new variable  $y$ . How-



ever, it is possible that the positive term  $\alpha_H \prod_{j \in H \setminus \{1\}} x_j$  combines with an existing term. To analyze the worst-case performance, we will assume that this never happens: combining terms only reduces the final amount of variables and quadratic terms produced.

If terms produced by equation 4 never combine with existing terms, then the reduction takes all terms containing  $x_1$ , removes the variable  $x_1$  from each, and adds single positive quadratic edge and a negative term with  $y$  replacing  $x_1$  for each original term.

Thus, each positive term of degree  $d$  that we start with will have a single variable removed every time we apply the reduction, so to be fully reduced it must go through  $d - 1$  applications of the rule. These reductions produce negative terms of degrees  $2, 3, \dots, d$ .

Reducing these  $d - 1$  negative terms by section 2.3.2 results in  $d - 1$  new variables and  $2 + 3 + \dots + d = \frac{(d-1)(d+2)}{2}$  submodular quadratic terms.

Overall, to reduce  $t$  positive terms of degree  $d$  on  $n$  variables, in the worst case our method requires  $n + t(d - 1)$  new variables,  $t \frac{(d-1)(d+2)}{2}$  submodular terms and at most  $n$  non-submodular terms. Even in the worst case our algorithm's asymptotic performance is superior to Ishikawa's method from [8, 9]. We use twice as many variables and slightly more submodular terms (specifically,  $\frac{(d-1)(d+2)}{2}$  vs  $\frac{(d-1)d}{2}$ ). Asymptotically, however, both methods require  $O(td)$  new variables and  $O(td^2)$  submodular terms. However, our method produces at most  $n$  non-submodular terms, even in the worst case, compared to  $O(nk)$  for [8, 9].

## 5. Local completeness

While our method is asymptotically superior to [8, 9], there is reason to believe that it performs particularly well on some common vision problems such as [17, 20]. For low-level vision problems an individual variable usually corresponds to a pixel, and there are generally terms involving the spatially adjacent pixels (for example, to impose a spatial smoothness prior). This suggests that  $k$ , the number of other variables that an individual variable shares terms with, is at least 4, and thus that we produce an energy function with  $\frac{1}{4}$  of the number of non-submodular terms. Note that the number of non-submodular terms in an energy function seems to be correlated with their difficulty, as mentioned by [19].

We have identified a property of certain energy functions that we call local completeness, where our algorithm (but not Ishikawa's) has improved asymptotic performance. Consider an energy function on the binary variables  $x_1, \dots, x_n$ , written as a multilinear polynomial, and denote by  $\mathcal{H}$  the hypergraph of its monomials, as before.

**Definition 5.1** *We say that the polynomial is locally complete with completeness  $c$  (or has local completeness  $c$ )*

*if there is a number  $c \in (0, 1]$  such that, considering the larger hypergraph  $\mathcal{H}'$  formed by all subsets of edges in  $\mathcal{H}$  (that is,  $\mathcal{H}' = \bigcup_{H \in \mathcal{H}} 2^H$ ), we have  $|\mathcal{H}| \geq c|\mathcal{H}'|$ .*

Every polynomial is locally complete for some completeness  $c$ , as we can always choose  $c = \frac{|\mathcal{H}|}{|\mathcal{H}'|}$ . However, we are interested in classes of problems which remain complete as the problem size grows, so we define that a family of polynomials is locally complete if there is a fixed  $c$  such that all the polynomials have local completeness  $c$ . For example, a family  $P$  of polynomials arising from a particular vision problem would be locally complete if we always had  $1/2$  of all subsets of terms appearing in all instances of  $P$ .

### 5.1. Performance on locally complete problems

Recall the procedure for reducing positive terms, using equation 4. We would like the extra positive terms we create, those with variables in  $H \setminus \{1\}$ , to combine with existing terms. If it happens that  $H \setminus \{1\}$  is already a term with coefficient  $\beta_{H \setminus \{1\}}$ , then we add  $\alpha_H$  to this coefficient, and do not create a new term. This could either make an existing positive term more positive, or cause a previously negative term to become positive, but we will see shortly that in either case, combining these two terms helps.

This explains the definition of local completeness: the new positive term  $\alpha_H \prod_{j \in H \setminus \{1\}} x_j$  has variables which are a subset of one of our original terms, so if our energy function has local completeness  $c$ , with our method these new positive terms will combine with existing terms some fraction  $c$  of the time. This results in the following asymptotic analysis.

**Theorem 5.2** *If an energy function has local completeness  $c$ , our procedure for reducing positive terms will result in at most  $\frac{1}{c}|\mathcal{H}|$  negative coefficient terms, where  $\mathcal{H}$  is the hypergraph for the original terms (both positive and negative terms).*

**PROOF:** By the definition of local completeness  $|\mathcal{H}'| \leq \frac{1}{c}|\mathcal{H}|$ . For every edge  $H$  which occurs in  $\mathcal{H}'$  and not  $\mathcal{H}$ , as a notational convenience we can write in a new term with variables from  $H$ , and with coefficient 0. All edges of  $\mathcal{H}'$  now occur as terms. Whenever we reduce a term  $\alpha_H \prod_{j \in H} x_j$ , the positive term  $\alpha_H \prod_{j \in H \setminus \{1\}} x_j$  always combines with an existing term, since  $\mathcal{H}'$  is closed under subsets.

This cancellation may either make an existing positive term more positive, or make a previously negative become positive (or a 0 term positive, etc.): in all cases, applying equation 4 removes the term  $\alpha_H \prod_{j \in H} x_j$ , changes the coefficient on the term with variables  $H \setminus \{1\}$ , and adds a new negative term  $\alpha_H y \prod_{j \in H \setminus \{1\}} x_j$ . The total number of terms remains constant.

	Extra variables	Non-submodular terms	Total terms	Percent labeled by QPBO
Ishikawa	224,346	421,897	1,133,811	80.4%
Our method	236,806 ( $\Delta = +6\%$ )	38,343 ( $\Delta = -90\%$ )	677,183 ( $\Delta = -40\%$ )	96.1% ( $\Delta = +20\%$ )

Figure 3. Performance comparison of reductions, on Ishikawa’s benchmarks in [9]. Relative performance of our method is shown as  $\Delta$ .

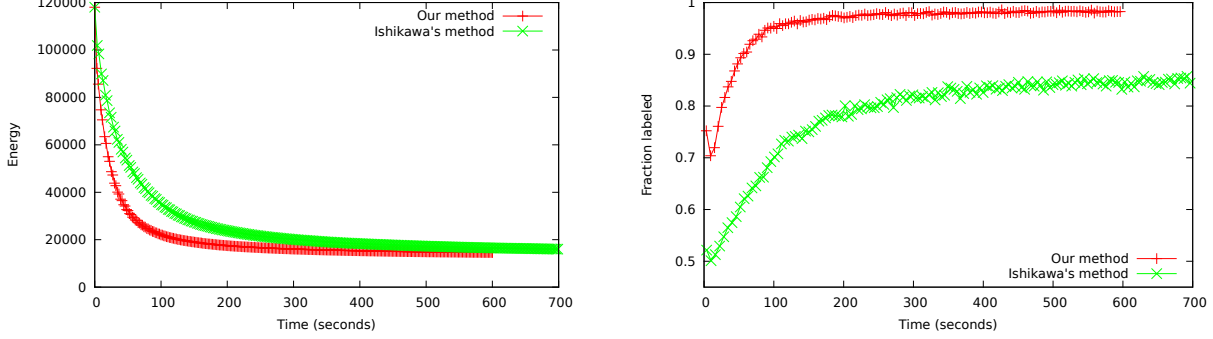


Figure 4. Energy after each fusion move (left), and percentage of pixels labeled by QPBO (right), for the image at top of figure 5. Other images from [9] give very similar curves.

Therefore, when we have finished reducing all positive terms, we are left with only negative terms, and we have exactly as many as we started with, namely  $|\mathcal{H}'| \leq \frac{1}{c}|\mathcal{H}|$ . ■

Note that for a family of polynomials which are locally complete, and with  $|\mathcal{H}| \rightarrow \infty$ , it is straightforward to show an asymptotic bound of  $\Theta(|\mathcal{H}|)$  negative terms required.

As usual, we can reduce all these negative terms term-by-term, as in section 2.3.2. If we started with  $t$  terms of up to degree  $d$  on  $n$  variables, the entire reduction results in at most  $n + \frac{1}{c}t$  new variables,  $\frac{1}{c}td$  submodular terms and  $n$  non-submodular terms.

Finally, it is worth noting that local completeness is actually stronger than strictly necessary to guarantee this performance. If our energy function does not have a good local completeness, it still could be the case that we get a good number of terms combining: note that we don’t actually need all subsets to appear, just that when we reduce all terms containing  $x_1$ , all the terms  $H \setminus \{1\}$  appear. In this case, the number of combining terms depends on the order in which we reduce the variables according to equation 4. Local completeness gives us the stronger property that no matter what order we pick variables to reduce with, we always get a large fraction of terms combining.

## 6. Locally complete energy functions in vision

We believe that many vision problems produce locally complete energy functions, and hence our method will work particularly well on them. In this section we will show that under some reasonable assumptions an important class of vision problems will have locally complete energy functions. Specifically, we consider fusion moves [15] under

an FoE prior [17] with random proposals. Ishikawa used this problem to benchmark his algorithm in [8, 9].

The original (non-binary) energy function can be written as a sum over cliques  $\mathcal{C}$  in the image  $\sum_{\mathcal{C}} f_{\mathcal{C}}(\mathbf{x}_{\mathcal{C}})$ . When we compute a fusion move there is an input image  $I$  and a proposed image  $I'$ , and for every pixel there is binary variable that encodes whether that pixel takes its intensity from  $I$  or  $I'$ . There is a binary energy function on these variables that we wish to solve to compute the optimal fusion move. In a higher-order MRF the cliques  $\mathcal{C}$  contain more than 2 pixels, so the binary energy function contains terms with more than 2 variables.

We can obtain a general theorem about the binary energy functions corresponding to fusion moves, by moving to a continuous framework. We embed the original intensities in  $\mathbb{R}$ , and extend the clique energies  $f_{\mathcal{C}}$  to functions on  $\mathbb{R}^d$ . We need two assumptions: (1)  $f_{\mathcal{C}}$  is  $d - 1$  times continuously differentiable and (2) each of the  $d$  different mixed partials  $\frac{\partial^{d-1} f}{\partial x_1 \dots \partial x_i \dots \partial x_d}$  (where  $\widehat{\partial x_i}$  means to omit the  $i$ -th partial) take their zeros in a set of measure 0.

**Theorem 6.1** *Under these two assumptions the set of proposed-current image pairs  $(I, I')$  for which the fusion move binary energy function does not have local completeness 1 has measure 0 as a subset of  $\mathbb{R}^n \times \mathbb{R}^n$ .*

We defer the proof of this theorem to the supplemental material, but now provide a brief proof sketch. We write the fusion move binary energy function in terms of  $n$  binary variables  $b_i$ . In translating this into a multilinear polynomial in the  $b_i$ , each clique  $\mathcal{C}$  can result in terms  $t_S$  for each subset  $S$  of  $\mathcal{C}$ . We can show that the energy function is locally complete, if the coefficient on  $t_S$  is never (or rarely) zero.

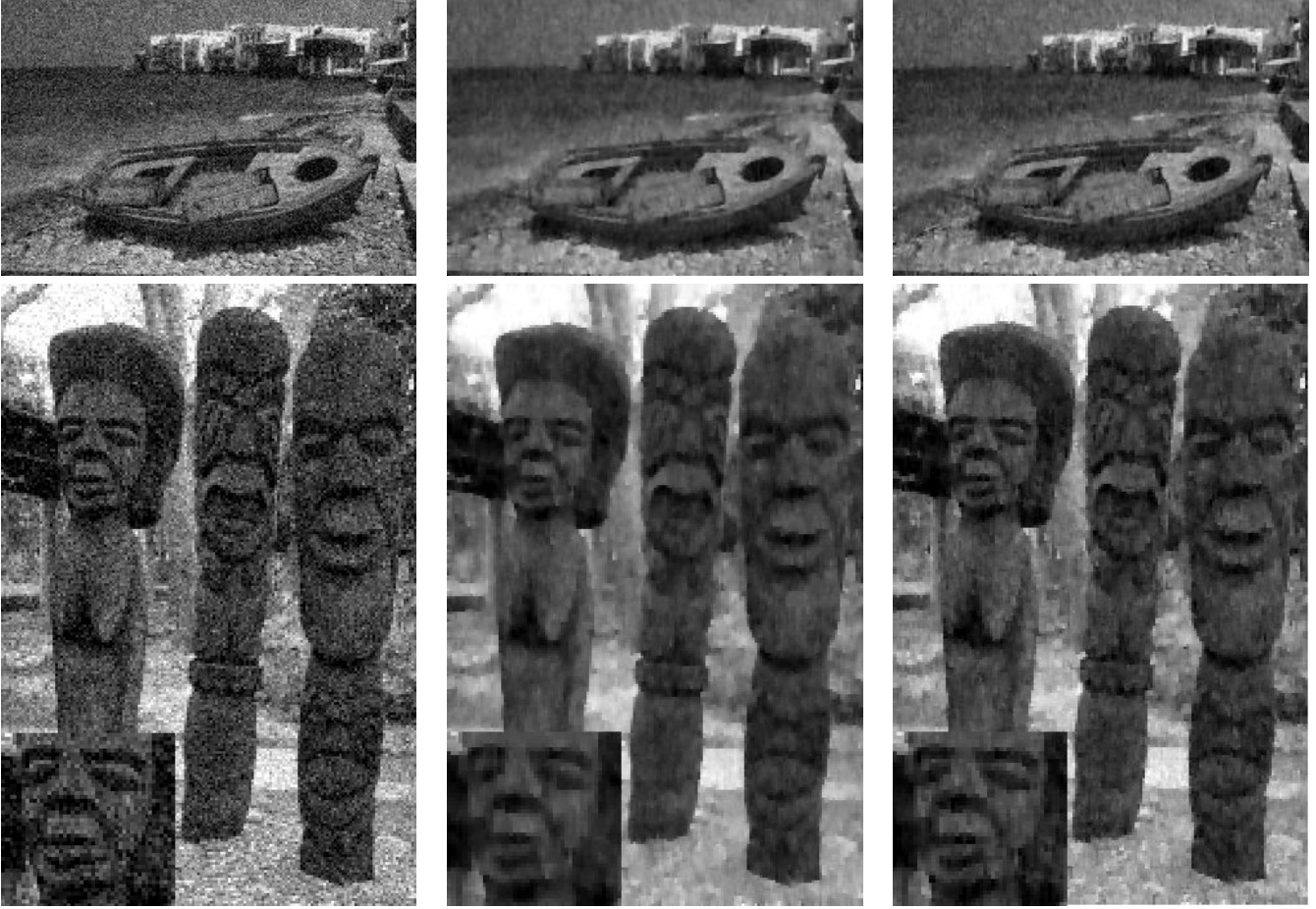


Figure 5. Denoising examples. At left is the noisy input image, with our result in the middle and Ishikawa’s at right. Results are shown after 30 iterations. More images are included in the supplementary material. To compare energy values with visual results, the images on the top row have energies 118,014, 26,103 and 38,304 respectively; those on the bottom have energies 118,391, 25,865 and 38,336.

For example, here is how to calculate the coefficient on the term  $b_1 b_2$  in a clique of size 3. If  $I_1, I_2, I_3$  are the labellings in the current image on  $C$ , and  $I'_1, I'_2, I'_3$  are the proposed labellings, then the coefficient on  $b_1 b_2$  is given by  $f_C(I_0, I_1, I_2) - f_C(I'_0, I_1, I_2) - f_C(I_0, I'_1, I_2) + f_C(I'_0, I'_1, I_2)$

Since our labels embed into  $\mathbb{R}$ , the four 3-pixel images mentioned in this coefficient actually lie on a rectangle in  $\mathbb{R}^3$ . If we give each of these points  $v$  heights of  $f_C(v)$ , then this coefficient is 0 if and only if the four points lie on a plane.

In general, we do not expect 4 arbitrary points to lie on a plane. However, these points depend heavily on  $f_C$ . In the full proof, we show that if there is an open ball of images with zero coefficient on  $b_1 b_2$ , then the energy function is locally flat ( $\frac{\partial^2 f}{\partial x_1 \partial x_2} = 0$ ) in a corresponding ball. We also extend this to larger degree terms, to prove the general case.

**Corollary 6.2** *The energy functions obtained from fusion moves with FoE priors and proposals chosen as random images are locally complete with probability 1.*

**PROOF:** The clique functions  $f_C$  for the FoE model given in [17] are infinitely differentiable, and their mixed partials have their zeros in a set of measure 0. Since the proposed images are chosen from a continuous distribution over  $\mathbb{R}^n$ , events of measure 0 occur with probability 0. ■

## 7. Experimental results

Ishikawa [8, 9] recently provided the first practical algorithm for solving arbitrary higher-order MRF’s with graph cuts, and to our knowledge it is the state of the art. Since our method is a direct competitor to his method, we focused our efforts on experimentally comparing our results with his. Fortunately both his code and data are publicly available, which greatly simplified the comparison.

As mentioned, Ishikawa’s did his benchmarking on fields of experts priors for image denoising with fusion moves, using a dataset of 200 images. We used the same MRF, and as similar energy functions as possible. His fusion moves alternated between a randomly generated uniform image and



a blurred image, and the energy function has clique size 4. Since he uses randomly generated proposals it is not possible to run the exact experiments reported in [8, 9]; however, we re-ran his code at our institution and obtained very similar results to what he published. We do multiple fusion moves on multiple images, so the effects of randomness are probably minimal.

The overall performance comparison across all images and all fusion moves is summarized in figure 3. It is instructive to compare the experimental data in figure 3 with the asymptotic analysis in figure 1. Overall, our method does better in practice than the asymptotic analysis suggests. As predicted we produce many fewer non-submodular terms, but we also produce fewer submodular terms (a relative improvement of 10%). The improved performance of QPBO is consistent with the observation of [19] that functions with more non-submodular terms are harder to optimize.

On the [9] benchmarks our algorithm always computes a lower energy solution, as shown in figure 4. When run to convergence on all 200 images we obtain an energy that is 8% lower, and converge 20% faster. Before convergence we often do substantially better, especially at the beginning. For example on the images shown in figure 5, where we stopped both methods after 30 iterations (about a minute), we have 33% lower energy. In addition, with our method QPBO labels substantially more pixels at each iteration.<sup>1</sup>

Denoising results are shown in figure 5. In the boat image our results appear more accurate in smooth areas like the water, and the face image (shown magnified at bottom left) is also noticeable smoother. These results are after 30 fusion moves. The images after convergence (shown, along with more examples, in the supplementary material) appear fairly similar, though we still obtain lower energy.

We plan to investigate several difficult higher-order MRF's. As a first step, we experimentally computed the local completeness of two early vision problems that are quite far from denoising, namely stereo [20] (clique size 3) and segmentation [1] (clique size 4). We analyzed the binary energy functions produced from 60 iterations of [20]. These energy functions have a very high local completeness; on average the energy functions are  $c$ -complete for  $c = .98$ , and their least locally complete energy function had  $c = .96$ . We also discovered that the higher-order segmentation energy function of [1] is completely locally complete ( $c = 1$ ). These results suggest that our method may be particularly well suited to a number of important vision problems.

**Acknowledgements:** This work was supported by NSF grant IIS-0803705. We thank Joyce Chen and Josh

Schwartz for helpful comments.

## References

- [1] B. Andres, J. Kappes, U. Köthe, C. Schnörr and F. Hamprecht. An empirical comparison of inference algorithms for graphical models with higher order factors using OpenGM. DAGM-Symposium, 2010. 1, 8
- [2] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Disc. App. Math.*, 123(1-3), 2002. 1, 2
- [3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI* 23, 2001. 1, 2
- [4] D. Freedman and P. Drineas. Energy minimization via graph cuts: Settling what is possible. In *CVPR*, 2005. 1, 3
- [5] P. Hammer. Some network flow problems solved with pseudo-boolean programming. *Op. Res.* 13, 1965. 2
- [6] P. Hammer and S. Rudeanu. *Boolean Methods in Operations Research and Related Areas*. Springer, 1968. 1
- [7] P. L. Hammer, P. Hansen, and B. Simeone. Roof duality, complementation and persistency in quadratic 0-1 optimization. *Math. Prog.* 28, 1984. 2
- [8] H. Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, 2009. 1, 3, 5, 6, 7, 8
- [9] H. Ishikawa. Transformation of general binary MRF minimization to the first order case. *TPAMI*, 33(6), June 2011. 1, 3, 5, 6, 7, 8
- [10] P. Kohli, M. P. Kumar, and P. H. Torr. P3 and beyond: Move making algorithms for solving higher order functions. *TPAMI* 31, 2008. 1, 2
- [11] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *TPAMI* 29, 2007. 2
- [12] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *TPAMI* 26, 2004. 1, 2, 3
- [13] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *CVPR*, 2009. 2
- [14] X. Lan, S. Roth, D. Huttenlocher, and M. J. Black. Efficient belief propagation with learned higher-order MRF's. In *ECCV*, 2006. 1, 2
- [15] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for MRF optimization. *TPAMI* 32, Aug 2010. 2, 6
- [16] I. Rosenberg. Reduction of bivalent maximization to the quadratic case. Technical report, Centre d'Etudes de Recherche Operationnelle, 1975. 1, 2
- [17] S. Roth and M. Black. Fields of experts. *IJCV* 82, 2009. 1, 2, 5, 6, 7
- [18] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, 2009. 1, 3
- [19] 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 MRF's. *TPAMI* 30, 2008. 1, 2, 5, 8
- [20] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *TPAMI* 31, 2009. 1, 2, 5, 8

<sup>1</sup>We averaged together two consecutive fusion moves in the graph shown at right in figure 4. This avoids the distracting sawtooth pattern visible in [8, 9], due to his alternation between random fusion moves and blurred fusion moves.