Transformation of General Binary MRF Minimization to the First-Order Case

Hiroshi Ishikawa, *Member, IEEE*

Abstract—We introduce a transformation of general higher-order Markov random field with binary labels into a first-order one that has the same minima as the original. Moreover, we formalize a framework for approximately minimizing higher-order multilabel MRF energies that combines the new reduction with the fusion-move and QPBO algorithms. While many computer vision problems today are formulated as energy minimization problems, they have mostly been limited to using first-order energies, which consist of unary and pairwise clique potentials, with a few exceptions that consider triples. This is because of the lack of efficient algorithms to optimize energies with higher-order interactions. Our algorithm challenges this restriction that limits the representational power of the models so that higher-order energies can be used to capture the rich statistics of natural scenes. We also show that some minimization methods can be considered special cases of the present framework, as well as comparing the new method experimentally with other such techniques.

Index Terms—Energy minimization, pseudo-Boolean function, higher order MRFs, graph cuts.

1 Introduction

Many problems in computer vision, such as segmentation, stereo, and image restoration, are formulated as optimization problems involving inference of the maximum a posteriori (MAP) solution of a Markov Random Field (MRF). Such optimization schemes have become quite popular, largely owing to the success of optimization techniques such as graph cuts [4], [10], [17], belief propagation [6], [25], and tree-reweighted message passing [16]. However, because of the lack of efficient algorithms to optimize energies with higher-order interactions, most are represented in terms of unary and pairwise clique potentials, with a few exceptions that consider triples [5], [17], [36]. This limitation severely restricts the representational power of the models: The rich statistics of natural scenes cannot be captured by such limited potentials [25]. Higher-order cliques can model more complex interactions and reflect the natural statistics better. There are also other reasons, such as enforcing connectivity [35] or histogram [32] in segmentation, for the need of optimizing higher-order energies.

This has long been realized [13], [27], [30], but with the recent success of the new energy optimization methods, there is a renewed emphasis on the effort to find an efficient way to optimize MRFs of higher order. For instance, belief propagation variants [21], [28] have been introduced to do inference based on higher-order clique potentials. In graph cuts, Kolmogorov and Zabih [17] introduced a reduction that can reduce second-order binary-label potentials into pairwise ones, followed by an algebraic simplification by Freedman and Drineas [7]. Kohli et al. [15], [14] extended the class of energies for which the optimal α-expansion and α-β-swap moves can be computed in polynomial time. Komodakis and Paragios [19] employed a master-slave decomposition framework to solve a dual relaxation to the MRF problem. Rother et al. [33] used a soft-pattern-based representation of higher-order functions that may for some energies lead to very compact first-order functions with a small number of nonsubmodular terms, as well as addressing the problem of transforming general multilabel functions into quadratic ones.

In our approach, higher-order energies are approximately minimized by iterative "move-making," in which higher-order binary energies are reduced to first-order ones and minimized.

Two recent advances in graph cuts made it possible. First, there is a generalization of the popular α-expansion algorithm [4] called the fusion move by Lempitsky et al. [22], [23]. Second, a recent innovation allows optimization of first-order nonsubmodular functions of binary variables. This method by Boros, Hammer, and their coworkers [1], [3], [8] is variously called QPBO [18] or roof-duality [31]. This has a crucial impact on the move-making algorithms since the choice of the move in each iteration depends on binary-label optimization. In the context of optimizing higher-order potentials, it means that some limitations that prevented the use of move-making algorithms for higher-order functions can possibly be overcome. As we mentioned, second-order potentials on binary variables can be reduced into pairwise ones [17]. However, the requirement that the result of reduction must be submodular made its actual use quite rare. Thanks to the QPBO technique, now we can think of reducing higher-order potentials into pairwise ones, with a hope that at least part of the solution can be found. Woodford et al. [36] used this strategy very successfully.

So far, the second-order case has remained the only case that could be solved using this group of techniques because

• *The author is with the Department of Computer Science and Engineering, Waseda University, Tokyo 169-8555, Japan. E-mail: hfs@waseda.jp.*

Manuscript received 25 Aug. 2009; revised 28 Dec. 2009; accepted 5 Mar. 2010; published online 31 Mar. 2010.
Recommended for acceptance by N. Paraqios.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-08-0556.
Digital Object Identifier no. 10.1109/TPAMI.2010.91.

the reduction we mention above is only applicable in that case. To be sure, a totally different reduction technique that can reduce binary energies of any order has been known for a long time [29]; however, to our knowledge it has never been used successfully in practice for orders higher than two. This seems to be because, even though it can reduce any function into a pairwise clique potential, the result is always nonsubmodular and hard to optimize in practice. We discuss it in Sections 3.2 and 3.3, as well as experimentally investigating it in Section 8.3.4.

In this paper, we introduce a new reduction technique along the lines of the Kolmogorov-Zabih reduction that can reduce any higher-order minimization problem of Markov random fields with binary labels into an equivalent first-order problem. Then, we combine it with fusion move and QPBO to approximately minimize higher-order multilabel energies. It also turns out that some of the known higher-order energy minimization techniques can be considered special cases of the new framework. We demonstrate its effectiveness by testing it on a third-order potential, an image restoration problem that has been used to test two BP algorithms [21], [28] capable of optimizing higher-order energies.

Part of this paper previously appeared as [11]. This extended version contains a new generalization of the reduction, discussions on the polynomial-time minimizability of the reduced energy and necessary number of auxiliary variables, and an investigation of relations between the new algorithm and other known methods.

Organization of the Paper. In the next section, we provide the notation and definition for higher-order energy minimization. In Section 3, we briefly describe the two known reductions of higher-order binary MRFs into first-order ones and discuss their limitations. In Section 4, we introduce the new reduction in its simplest form. In Section 5, we generalize the new reduction and discuss its various aspects. In Section 6, we describe the higher-order fusion-move algorithm using the new reduction. In Section 7, we investigate the relations of the new method with other known methods. In Section 8, we report the results of our experiments. We conclude in Section 9.

2 Preliminaries

In this section, we provide some notations and definitions pertinent to the areas of Markov random fields and higher-order energy minimization.

2.1 Markov Random Fields and Energy Minimization Problem

We denote the set $\{0,1\}$ of binary labels by $\mathbb{B}$ and the set of real numbers by $\mathbb{R}$. The energy minimization problem considered here is as follows:

Let V be a set of pixels[1] and L a set of labels, both finite sets. Also, let $\mathcal{C}$ be a set of subsets of V. We call an element of $\mathcal{C}$ a clique. The cliques define a form of generalized neighborhood structure. In the case where all cliques consist of one or two pixels, it can be thought of as an undirected graph, where the set of cliques is divided into the set of all singleton subsets and that of pairs, i.e., edges. In the general case, the pair $(V,\mathcal{C})$ can be thought of as a hypergraph.

Let us denote by L^V the set of labelings $X : V \to L$, i.e., assignments of a label $X_v \in L$ to each pixel $v \in V$. In particular, the set of binary labelings is denoted by $\mathbb{B}^V$.

The energy $E(X)$ is a real-valued function defined on this set:

$$E : L^V \to \mathbb{R}.$$

A crucial assumption is that $E(X)$ is decomposable into a sum

$$E(X) = \sum_{C \in \mathcal{C}} f_C(X_C), \tag{1}$$

Here, for a clique $C \in \mathcal{C}$, $f_C(X_C)$ is a function that depends only on labels X_C assigned by X to the pixels in C. We denote the set of labelings on the clique C by L^C; thus $X_C \in L^C$ and $f_C : L^C \to \mathbb{R}$.

Supposing a random variable that takes values in L at each pixel, the labeling X can be thought of as the system of random variables. Such a system, along with the energy, is called a Markov random field (MRF).

The *order* of an MRF and its energy is defined to be the number of pixels in the largest clique minus one. Thus, an MRF of first order is of the form

$$E(X) = \sum_{\{v\} \in \mathcal{C}} f_{\{v\}}(X_{\{v\}}) + \sum_{\{u,v\} \in \mathcal{C}} f_{\{u,v\}}(X_{\{u,v\}}),$$

which is usually written as

$$E(X) = \sum_{v \in V} f_v(X_v) + \sum_{(u,v) \in E} f_{uv}(X_u, X_v). \tag{2}$$

Similarly, a second-order MRF can have cliques containing three pixels and an energy of the form

$$E(X) = \sum_{\{v\} \in \mathcal{C}} f_{\{v\}}(X_{\{v\}}) + \sum_{\{u,v\} \in \mathcal{C}} f_{\{u,v\}}(X_{\{u,v\}}) + \sum_{\{u,v,w\} \in \mathcal{C}} f_{\{u,v,w\}}(X_{\{u,v,w\}}).$$

If we have a labeling $X \in L^V$, we denote its restriction to a subset $U \subset V$ by $X|_U$:

$$X|_U \in L^U, \ (X|_U)_v = X_v, v \in U. \tag{3}$$

2.2 Conditional Random Field

An undirected graphical stochastic model that contains both the hidden and the observed variables is sometimes called a conditional random field (CRF) [20]. When the observed variables are fixed, the rest of the model on the hidden variables is an MRF. The notion recently emerged from the area of machine learning, where the observed data is often not fixed, and is instead used to learn the model itself.

In the vision literature, the term CRF is increasingly used where the term MRF would have been used before, even where such notions as learning are not used. Traditionally, MRFs used in the area tend to have the form where observed data are associated with individual pixels, and the dependence of

1. Sometimes also called sites, they need not necessarily represent actual pixels, though they often do; consider it as just a name for the elements of a finite set.

the variable at a pixel on the observation only comes from the data associated with that pixel. Perhaps because of this, the term CRF is sometimes used to mean that a variable depends on the observed data associated with more than one pixel, even if the problem fixes the data at the beginning, making it effectively an MRF problem.

In the context of energy minimization, the two terms MRF and CRF are used almost interchangeably since, at that level where the optimization problem is already well-separated from the context of stochastic modeling, there is no useful difference. One caveat, though, is that it can be confusing when the order of the random field is discussed. The order of an MRF has long been defined as the number of the pixels in the maximum clique in the energy *minus one* whereas, in the case of the CRF, it is the same as the size of the maximum clique. Thus, a CRF of order k is an MRF of order $k-1$, when the observed variables are fixed.

2.3 Pseudo-Boolean Functions

An MRF energy that has labels in $\mathbb{B}$ is a function of binary variables: $f:\mathbb{B}^n\rightarrow\mathbb{R}$, where n is the number of pixels/variables. Such functions are called *pseudo-Boolean functions* (PBFs).

Any PBF can be uniquely represented as a polynomial of the form

$$f(x_1,\ldots,x_n)=\sum_{S\subset V}c_S\prod_{i\in S}x_i,$$

where $V=\{1,\ldots,n\}$ and $c_S\in\mathbb{R}$. We refer the readers to [2], [9] for proof.

Combined with the definition of the order, this implies that any binary-labeled MRF of order $d-1$ can be represented as a polynomial of degree d. Thus, the problem of reducing higher-order binary-labeled MRFs to first order is equivalent to that of reducing general pseudo-Boolean functions to quadratic ones.

2.4 Minimization of PBFs

A first-order binary energy (2) is said to be submodular if

$$f_{uv}(0,0)+f_{uv}(1,1)\leq f_{uv}(0,1)+f_{uv}(1,0)$$

holds for every pairwise potential f_{uv} for $(u,v)\in E$. First-order binary energies that are submodular can be minimized exactly by an *s-t*-mincut algorithm [17]. The following well-known proposition gives a convenient criterion for submodularity when the energy is given as a quadratic polynomial in binary variables:

Proposition 1. *A quadratic pseudo-Boolean function*

$$E(x_1,\ldots,x_n)=\sum_{i,j}a_{ij}x_ix_j+\sum_i a_ix_i+a$$

is submodular if and only if $a_{ij}\leq 0$ *for all* i,j.

Proof: The proposition follows directly from Proposition 3.5 in Nemhauser et al. [26]. □

In this paper, we rely on the QPBO algorithm to minimize binary-labeled energies. The algorithm returns a partial solution assigning either 0 or 1 to some of the pixels, leaving the rest unlabeled. Thus, if we run QPBO on a binary energy $\mathcal{E}(x)$ defined on $\mathbb{B}^V$, we obtain a partial labeling $z\in\mathbb{B}^{V'}$, where $V'\subset V$. The algorithm guarantees that the partial labeling is a part of a globally optimal one. That is, there exists a labeling $y'\in\mathbb{B}^V$ that attains the global minimum and such that $y'|_{V'}=z$. If the energy is submodular, QPBO labels all pixels, finding a global minimum.

Crucially for our purpose, QPBO has an "autarky" property [2], [3], [18]: If we take any labeling and "overwrite" it with a partial labeling obtained by QPBO, the energy for the resulting labeling is not higher than that for the original labeling. Let $y\in\mathbb{B}^V$ be an arbitrary binary labeling and $z\in\mathbb{B}^{V'}$ the result of running QPBO on an energy $\mathcal{E}(x)$, with $V'\subset V$. Let us denote by $y\lhd z\in\mathbb{B}^V$ the labeling obtained by overwriting y by z:

$$(y\lhd z)_v=\begin{cases}z_v, & \text{if } v\in V'\\ y_v, & \text{otherwise.}\end{cases}\tag{4}$$

Then, the autarky property means $\mathcal{E}(y\lhd z)\leq\mathcal{E}(y)$.

3 Known Reductions

There are a couple of known methods to reduce a higher-order function of binary variables to first-order one so that the minima of the reduced function can be translated easily to those for the original function. Here, we outline the two known reduction methods and then discuss their limitations.

3.1 Reduction by Minimum Selection

Kolmogorov and Zabih [17] first proposed this reduction in the context of graph-cut optimization. Later, Freedman and Drineas [7] recast it into an algebraic formula.

Consider a cubic pseudo-Boolean function of $x,y,z\in\mathbb{B}$

$$f(x,y,z)=axyz.$$

The reduction is based on the following identity:

$$xyz=\max_{w\in\mathbb{B}}w(x+y+z-2).\tag{5}$$

Let $a<0$ be a real number. Then

$$axyz=\min_{w\in\mathbb{B}}aw(x+y+z-2).$$

Thus, whenever $axyz$ appears in a minimization problem with $a<0$, it can be replaced by $aw(x+y+z-2)$.

If $a>0$, we flip the variables (i.e., replace x by $1-x$, y by $1-y$, and z by $1-z$) of (5) and consider

$$(1-x)(1-y)(1-z)=\max_{w\in\mathbb{B}}w(1-x+1-y+1-z-2).$$

This is simplified to

$$xyz=\min_{w\in\mathbb{B}}w(x+y+z-1)+(xy+yz+zx)-(x+y+z)+1.\tag{6}$$

Therefore, if $axyz$ appears in a minimization problem with $a>0$, it can be replaced by

$$a\{w(x+y+z-1)+(xy+yz+zx)-(x+y+z)+1\}.$$

Thus, in either case, the cubic term can be replaced by quadratic terms. As we mentioned in Section 2.3, any binary MRF of second order can be written as a cubic polynomial.

Then, each cubic monomial in the polynomial can be converted to a quadratic polynomial using one of the formulas above, making the whole energy quadratic.

This reduction only works with cubic terms. For quartic term $axyzt$, the same trick works if $a < 0$:

$$xyzt = \max_{w \in \mathbb{B}} w(x+y+z+t-3),$$
$$axyzt = \min_{w \in \mathbb{B}} aw(x+y+z+t-3).$$

However, if $a > 0$,

$$(1-x)(1-y)(1-z)(1-t) = \max_{w \in \mathbb{B}} w(-x-y-z-t+1)$$

becomes

$$xyzt = \max_{w \in \mathbb{B}} w(-x-y-z-t+1) + (xyz+xyt+xzt+yzt) - (xy+yz+zx+xt+yt+zt) + (x+y+z+t) - 1.$$

Unlike the cubic case, the maximization problem is not turned into a minimization. Similarly, this does not work with any term of even degree. This poses a severe restriction for which function this reduction can be used in the case of degrees higher than three.

In this paper, we remove this limitation in two ways. First, we introduce in Section 4 a new transformation that can be used for higher-degree terms with positive coefficients. In the quartic case, it gives:

$$xyzt = \min_{w \in \mathbb{B}} w(-2(x+y+z+t)+3) + xy+yz+zx+tx+ty+tz,$$

which expresses the quartic term as a *minimum* of a quadratic expression, making it suitable for reduction of terms with positive coefficients.

Also, we introduce in Section 5 a generalization of both the minimum selection and the new transformation that gives even more different ways of converting higher-order energies into first-order ones. This is accomplished by flipping some of the variables before and after transforming the higher-order term to a minimum or maximum. For instance, another reduction for the quartic term can be obtained using the technique: if we define $\bar{x} = 1 - x$, we have $xyzt = (1-\bar{x})yzt = yzt - \bar{x}yzt$. The right-hand side consists of a cubic term and a quartic term with a negative coefficient, which can be reduced using the minimum-selection technique (see equation (19) in Section 5.4.) This generalization gives rise to an exponential number (in terms of occurrences of variables in the function) of possible reductions such that choosing which reduction to use is a whole new nontrivial problem. Although we cannot address the new problem thoroughly in this paper, we discuss some aspects of it in Section 5 and compare some specific reductions experimentally in Section 8.

3.2 Reduction by Substitution

However, it has long since been known that the optimization of pseudo-Boolean function of any degree can always be reduced to an equivalent problem for quadratic pseudo-Boolean function. The method was proposed by Rosenberg [29] and has since been recalled by Boros and Hammer [2].

In this reduction, the product xy of two variables x, y in the function is replaced by a new variable z, which is forced to have the same value as xy at any minimum of the function by adding penalty terms that would have a very large value if they do not have the same value.

More concretely, assume that $x, y, z \in \mathbb{B}$ and define

$$D(x,y,z) = xy - 2xz - 2yz + 3z. \tag{7}$$

Then, it is easy to check, by trying all eight possibilities, that $D(x,y,z) = 0$ if $xy = z$ and $D(x,y,z) > 0$ if $xy \neq z$. Consider an example pseudo-Boolean function

$$f(x,y,w) = xyw + xy + y.$$

The reduction replaces xy by z and add $MD(x,y,z)$:

$$\tilde{f}(x,y,w,z) = zw + z + y + MD(x,y,z), \tag{8}$$

which has one more variable and is of one less degree than the original function f. Here, M is chosen to be a large positive number so that, whenever $xy \neq z$ and thus $D(x,y,z) > 0$, it is impossible for $\tilde{f}$ to take the minimum.

By repeating the above reduction, any higher-order function can be reduced to a quadratic function with additional variables; for any minimum-energy value assignment for the new function, the same assignment of values to the original variables gives the minimum energy to the original function.

3.3 The Problem with Reduction by Substitution

Reduction by substitution has not been used very often in practice because, we suspect, it is difficult to make it work in practice.

Note that, according to (7),

$$MD(x,y,z) = Mxy - 2Mxz - 2Myz + 3Mz$$

in (8). The first term, Mxy, is a quadratic term with a very large positive coefficient, which *in all cases* makes the result of the reduction nonsubmodular, according to Proposition 1. Though in using QPBO submodularity is not the only factor (see Section 5.2), it seems that such an energy cannot be minimized very well even with QPBO: in our experiments (Section 8.3) with this reduction, most variables were not assigned labels, leaving the move-making algorithm almost completely stalled.

4 The New Reduction

In this section, we introduce a new reduction of higher-order PBFs into quadratic ones. It is an extension of the reduction by minimum selection that we described in Section 3.1.

4.1 Quartic Case

Let us look at the quartic case. We would like to generalize (5) and (6) to higher degrees. Upon examination of these formulas, one notices that they are symmetric in the three variables x, y, and z. This suggests that any quadratic polynomial reduced

from $xyzt$ should also be symmetric in the four variables.[2] That is, if a generalized formula exists, it should look like

$$xyzt = \min_w w(\text{linear symmetric polynomial}) + (\text{quadratic symmetric polynomial}).$$

It is known that any symmetric polynomial can be written as a polynomial expression in the *elementary symmetric polynomials*. There is one elementary symmetric polynomial of each degree; the ones we need are:

$$s_1 = x+y+z+t,$$
$$s_2 = xy+yz+zx+tx+ty+tz.$$

Also, when the variables only take values in $\mathbb{B}$, the square of a variable is the same as itself. Thus, we have ${s_1}^2 = s_1 + 2s_2$, implying that any quadratic symmetric polynomial can be written as a linear combination of s_1, s_2, and 1. Thus, the formula should be of the form:

$$xyzt = \min_{w\in\mathbb{B}} w(as_1+b)+cs_2+ds_1+e.$$

An exhaustive search for integers a,b,c,d, and e that makes the right-hand side positive only when $x=y=z=t=1$ and 0 otherwise yields:

$$xyzt = \min_{w\in\mathbb{B}} w(-2s_1+3)+s_2.$$

A similar search fails in finding a quintic formula, but increasing the number of auxiliary variables, we obtain:

$$xyztu = \min_{(v,w)\in\mathbb{B}^2} \{v(-2r_1+3)+w(-r_1+3)\}+r_2,$$

where r_1 and r_2 are the first and second-degree elementary symmetric polynomials in x,y,z,t, and u. In the same way, similar formulas for degrees six and seven can be found, from which the general formula can be conjectured.

4.2 General Case

Now, we introduce similar reductions for general degree. Consider a monomial $ax_1\cdots x_d$ of degree d. We define the elementary symmetric polynomials in these variables as:

$$S_1 = \sum_{i=1}^{d} x_i, \qquad S_2 = \sum_{i=1}^{d-1}\sum_{j=i+1}^{d} x_i x_j = \frac{S_1(S_1-1)}{2}.$$

Case: $a<0$**.** It is simple if $a<0$:

$$ax_1\cdots x_d = \min_{w\in\mathbb{B}} aw\{S_1-(d-1)\}, \tag{9}$$

as given by Freedman and Drineas [7].

Case: $a>0$**.** This case is our contribution:

$$ax_1\cdots x_d = a\min_{w_1,\ldots,w_{n_d}\in\mathbb{B}} \sum_{i=1}^{n_d} w_i\left(c_{i,d}(-S_1+2i)-1\right)+aS_2, \tag{10}$$

which follows from the following theorem:

2. Actually, the symmetry as a polynomial is not necessary. It has to be symmetric only on $\mathbb{B}^4$. See equation (19) in Section 5.4.

Theorem 1. *For $x_1,\ldots,x_d$ in $\mathbb{B}$,*

$$x_1\cdots x_d = \min_{w_1,\ldots,w_{n_d}\in\mathbb{B}} \sum_{i=1}^{n_d} w_i\left(c_{i,d}(-S_1+2i)-1\right)+S_2, \tag{11}$$

where

$$n_d = \left\lfloor \frac{d-1}{2} \right\rfloor, \qquad c_{i,d} = \begin{cases} 1, & \text{if } d \text{ is odd and } i = n_d, \\ 2, & \text{otherwise.} \end{cases}$$

Proof: Let us suppose that k of the d variables $x_1,\ldots,x_d$ are 1 and the rest are 0. Then, it follows

$$S_1 = k, \qquad S_2 = \frac{k(k-1)}{2}.$$

Let us also define

$$l = \left\lfloor \frac{k}{2} \right\rfloor, \qquad m_d = \left\lfloor \frac{d-2}{2} \right\rfloor, \qquad N = \min(l, m_d),$$
$$A = \min_{w_1,\ldots,w_{n_d}\in\mathbb{B}} \sum_{i=1}^{m_d} w_i(-2S_1+4i-1)+S_2.$$

Since the variables w_i can take values independently,

$$A = \sum_{i=1}^{m_d} \min(0, -2k+4i-1)+\frac{k(k-1)}{2}$$

If k is even ($k=2l$), we have

$$-2k+4i-1<0 \iff 4i<4l+1 \iff i\le l,$$

and if k is odd ($k=2l+1$)

$$-2k+4i-1<0 \iff 4i<4l+3 \iff i\le l.$$

Thus,

$$A = \sum_{i=1}^{N}(-2k+4i-1)+\frac{k(k-1)}{2} = 2N^2-N(2k-1)+\frac{k(k-1)}{2}. \tag{12}$$

We note that $A=0$ if $k\le d-2$. This can be seen by checking the cases $k=2l$ and $k=2l+1$ in (12), noting that $l\le m_d$ and thus $N=l$.

Now, consider the even-degree case of (11). Since, in that case, $n_d = m_d$ and $c_{i,d}=2$, the right-hand side equals A. Thus, both sides are 0 if $k\le d-2$.

If $k=d-1$, $A=0$ similarly follows from $N=l=m_d$.

If $k=d$, then $m_d = l-1$. Thus, substituting $N=l-1$ and $k=2l$ in (12), we have

$$A = 2(l-1)^2-(l-1)(4l-1)+l(2l-1) = 1,$$

which completes the proof in the even-degree case.

When d is odd, we have $m_d = n_d-1$ and the right-hand side of (11) is

$$A+\min(0, -S_1+2n_d-1).$$

Since $d=2n_d+1$,

$$-S_1+2n_d-1\ge 0 \iff k\le d-2.$$

If $k\le d-2$, then $A=0$ and $-S_1+2n_d-1\ge 0$; thus, both sides of (11) are 0.

If $k = d-1$, then

$$-S_1 + 2n_d - 1 = -k + k - 1 = -1.$$

Also, since $m_d = l-1$, using $N = l-1$, $k = 2l$ in (12), it follows that

$$A = 2(l-1)^2 - (l-1)(4l-1) + l(2l-1) = 1,$$

showing that the both sides of (11) are 0.

Finally, if $k = d$, we have

$$-S_1 + 2n_d - 1 = -k + k - 1 - 1 = -2,$$

and from $N = m_d = l-1, k = 2l+1$, and (12),

$$A = 2(l-1)^2 - (l-1)(4l+1) + l(2l+1) = 3,$$

which shows that both sides of (11) are 1. □

Note that the cubic case of (11) is different from (6) and simpler.

As we mentioned above, any pseudo-Boolean function can be written uniquely as a polynomial in binary variables. Since each monomial in the polynomial can be reduced to a quadratic polynomial using (9) or (10), depending on the sign of the coefficient and the degree of the monomial, the whole function can be reduced to a quadratic polynomial that is equivalent to the original function in the sense that, if any assignment of values to the variables in the reduced polynomial achieves its minimum, the assignment restricted to the original variables achieves a minimum of the original function. Note that the reduction is valid whether the function is submodular or not.

5 Generalization and Ramification

In this section, we introduce the "γ-flipped" version of the transformation given in the previous section. By γ-flipping, we obtain many different reductions. So many, in fact, that any serious comparison of different reductions, experimentally or theoretically, will have to be left for future work. Comparing a few different reductions would not offer much information, as it would be akin to comparing the energy of a few labelings in an MRF. Rather, we will need some algorithm or at least a heuristic to guide the selection of which reduction to use. Nevertheless, we discuss some aspects of the transformation in this section, as well as testing the most obvious variants experimentally in Sections 8.3.3 and 8.3.5.

For $b \in \mathbb{B}$, let us denote the negation of b, or $1-b$, by $\bar{b}$. When we allow the negations, the same PBF can be written in many different ways:

$$\begin{aligned} xyzt &= (1-\bar{x})yzt = yzt - \bar{x}yzt \\ &= yzt - \bar{x}zt + \bar{x}\bar{y}zt = zt - \bar{y}zt - \bar{x}zt + \bar{x}\bar{y}zt. \end{aligned}$$

Let us call this manipulation *flipping*. By flipping some of the variables before reducing to quadratic and then flipping them back after, we obtain many more reduction formulas.

An important caveat is that when the energy is minimized, variables x and $\bar{x}$ are not independent. Before using any algorithm, such as QPBO, that assumes independence of all variables, we need to make sure that only one of x or $\bar{x}$ appears in the energy for each variable x.

5.1 γ-Flipping Variables and Formulas

The monomial $ax_1 \cdots x_d$ that we gave the reduction formulas (9) and (10) is a function that has a nonzero value only when all the variables are 1. We generalize this to arbitrary combinations of 0s and 1s.

For an arbitrary binary vector $\gamma = (\gamma_1, \ldots, \gamma_d) \in \mathbb{B}^d$, let us define

$$I_\gamma^0 = \{i \,|\, \gamma_i = 0\}, \qquad I_\gamma^1 = \{i \,|\, \gamma_i = 1\}. \tag{13}$$

Let $x^{(\gamma)} = (x_1^{(\gamma)}, \ldots, x_d^{(\gamma)})$ be a vector of variables defined from γ and another vector $x = (x_1, \ldots, x_d)$ of variables by

$$x_i^{(\gamma)} = \gamma_i x_i + \bar{\gamma}_i \bar{x}_i = \begin{cases} x_i, & \text{if } \gamma_i = 1, \\ \bar{x}_i, & \text{if } \gamma_i = 0. \end{cases}$$

We call $x^{(\gamma)}$ the *γ-flipped* version of the variable vector x. Then, the monomial $ax_1^{(\gamma)} \cdots x_d^{(\gamma)}$ has a nonzero value a if and only if $x = \gamma$:

$$ax_1^{(\gamma)} \cdots x_d^{(\gamma)} = \begin{cases} a & \text{if } x = \gamma, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Let us define the γ-flipped elementary symmetric polynomials of degree one and two:

$$S_1^{(\gamma)} = \sum_{i=1}^{d} x_i^{(\gamma)} = \sum_{i \in I_\gamma^1} x_i + \sum_{i \in I_\gamma^0} (1 - x_i) = -\sum_{i=1}^{d} (-1)^{\gamma_i} x_i + \left| I_\gamma^0 \right|,$$

$$S_2^{(\gamma)} = \sum_{i=1}^{d-1} \sum_{j=i+1}^{d} x_i^{(\gamma)} x_j^{(\gamma)} = \frac{S_1^{(\gamma)}(S_1^{(\gamma)} - 1)}{2}.$$

Then the generalized reduction can be stated as follows:

Theorem 2. *For $x_1, \ldots, x_d$ in $\mathbb{B}$, if $a < 0$,*

$$\begin{aligned} ax_1^{(\gamma)} \cdots x_d^{(\gamma)} &= \min_{w \in \mathbb{B}} aw \left\{ S_1^{(\gamma)} - (d-1) \right\} \\ &= \min_{w \in \mathbb{B}} aw \left\{ -\sum_{i=1}^{d} (-1)^{\gamma_i} x_i - \left| I_\gamma^1 \right| + 1 \right\}, \end{aligned} \tag{15}$$

and if $a > 0$

$$ax_1^{(\gamma)} \cdots x_d^{(\gamma)} = a \min_{w_1, \ldots, w_{n_d} \in \mathbb{B}} \sum_{i=1}^{n_d} w_i \left(c_{i,d}(-S_1^{(\gamma)} + 2i) - 1 \right) + aS_2^{(\gamma)}. \tag{16}$$

Proof: It directly follows from (9) or (10), depending on the sign of a. □

5.2 Polynomial-time Minimizability of the Reduced Function

Looking at the generalized reduction (15) and (16) in view of Proposition 1, we must conclude that there seems to be no obvious and general characterization for submodularity of the resulting quadratic polynomial.

In general, the right-hand side of (15) and (16) is not submodular. However, submodularity is not a necessary condition for the function to be globally minimizable in polynomial time.

For instance, flipping variables can make a nonsubmodular function into a submodular one.

The problem occurs when there is *frustration*, i.e., when it is not possible to consistently flip or not flip each variable so that coefficients of all quadratic terms are negative. For instance, we can make the coefficients of any two of the three quadratic terms negative in

$$\begin{aligned} xy+yz+zx &= y-\bar{x}y+yz+z-z\bar{x} \\ &= x-x\bar{y}+z-\bar{y}z+zx \\ &= xy+y-y\bar{z}+x-\bar{z}x, \end{aligned}$$

but not all three. If we only look at the term xy, we can make it submodular in terms of variables $\bar{x}, y$. But if there are other occurrences of x in the energy as a whole, that may make other terms nonsubmodular, as happens in the above example. If the form of the clique potentials is such that there is a way of flipping variables so that all quadratic monomials become submodular, the energy would be globally minimizable in polynomial time using graph cuts, which happens in minimizing $\mathcal{P}^n$ Potts model with α-β-swap (Section 7.2.1.)

Guaranteed Cases. There are special forms of potentials for which we can guarantee that the reduced polynomial is minimizable in polynomial time using graph cuts.

The first case is when the term is of the form $ax_1 \cdots x_d$ with a negative coefficients a, i.e., when $\gamma = (1, \ldots, 1)$ in (15). This function takes the value a when all of $x_1, \ldots, x_d$ are 1, and 0 otherwise. By the reduction, we have

$$ax_1 \cdots x_d = \min_{w \in \mathbb{B}} aw\{S_1 - (d-1)\},$$

in which all of the quadratic monomials in awS_1 have negative coefficients, making it submodular according to Proposition 1.

Another case is the opposite, i.e., when the term is of the form $a\bar{x}_1 \cdots \bar{x}_d$ with a negative coefficient a, i.e., when $\gamma = (0, \ldots, 0)$ in (15). By the reduction, we have

$$a\bar{x}_1 \cdots \bar{x}_d = \min_{w \in \mathbb{B}} aw\{-S_1 + 1\},$$

in which all of the quadratic monomials in $-awS_1$ have positive coefficients. This makes the potential supermodular. However, we can flip the auxiliary variable w so that the quadratic monomials in

$$-awS_1 = \min_{\bar{w} \in \mathbb{B}} (-a(1-\bar{w})S_1)$$

have negative coefficients. This can be done without influencing other parts of the energy since w appears nowhere else. If we use the QPBO algorithm, this is automatically taken care of and no explicit flipping is necessary.

These two cases are hidden in the minimization of $\mathcal{P}^n$ Potts model with α-expansion (Section 7.2.2).

5.3 γ-Flipped Representation of PBFs

Let $f : \mathbb{B}^d \to \mathbb{R}$ be a pseudo-Boolean function. Then, from (14) it follows that

$$f(x) = \sum_{\gamma \in \mathbb{B}^d} f(\gamma) x_1^{(\gamma)} \cdots x_d^{(\gamma)}.$$

Moreover, for any $\lambda \in \mathbb{R}$,

$$f(x) = \lambda + \sum_{\gamma \in \mathbb{B}^d} (f(\gamma) - \lambda) x_1^{(\gamma)} \cdots x_d^{(\gamma)}. \tag{17}$$

In particular, by taking

$$\lambda = \min_{\gamma \in \mathbb{B}^d} f(\gamma),$$

we can make all of the coefficients in the sum (17) nonnegative. This is an example of the *posiform* representation of the PBF (See [2]).

Similarly, by taking

$$\lambda = \max_{\gamma \in \mathbb{B}^d} f(\gamma), \tag{18}$$

we can make all of the coefficients in (17) nonpositive. Thus, any PBF can be transformed to a quadratic PBF using only (15), by making all nonzero coefficients negative first. This is essentially what is done in the binary reduction by Rother et al. [33]. See Section 7.1 for more discussion.

5.4 The Number of Necessary Auxiliary Variables

In our reduction, the number of additional variables per clique in the worst case is exponential in terms of the degree. This is because there are not only the highest-degree terms but also lower-degree terms, each of which needs its own new variables.

The exponential growth in the number of monomials in the general case is in a sense unavoidable. Any PBF of d variables can be represented as a polynomial of degree d, which has 2^d coefficients. For almost all such functions, all coefficients are nonzero. Since the reduction needs at least one variable for each monomial with three or higher degree, the number of extra variables must also increase exponentially as the order of the energy increases.

As mentioned above, we can make one coefficient in the sum (17) zero and all other coefficients nonpositive by setting λ by (18). Since the transformation (15) in the negative case requires only one new variable for converting each term, a minimization problem of any function of d binary variables can be translated into the quadratic case by adding at most $2^d - 1$ auxiliary variables.

On the other hand, if we expand the function as a polynomial as in the previous section, there are at most $\binom{d}{i}$ nonzero monomials of degree i. In the worst case, where each monomial has a positive coefficient, the number of auxiliary variables needed for using the transformation (10) is

$$N^{\text{all}+}(d) = \sum_{i=3}^{d} \binom{d}{i} \left\lfloor \frac{i-1}{2} \right\rfloor = 2^{d-2}(d-3) + 1,$$

which is less than $2^d - 1$ when $d \leq 6$. If the coefficients are all negative, the number is:

$$N^{\text{all}-}(d) = \sum_{i=3}^{d} \binom{d}{i} = 2^d - 1 - \frac{d(d+1)}{2},$$

which is the best case. In the average case, where one half of the coefficients are negative and the other half positive, we

are better off in this respect using the polynomial expansion when $d \leq 7$ (Table 1).

Note that in a clique potential, which generally would consist of one or more monomials, each monomial can have a different mix of variables x_i and their negations $\bar{x}_i$. Thus, for instance, the monomial $xyzt$ can be flipped and reduced:

$$\begin{aligned}\min_{x,y,z,t\in\mathbb{B}} xyzt &= \min_{x,y,z,t\in\mathbb{B}} \{-\bar{x}yzt - \bar{y}zt + zt\} \\ &= \min_{x,y,z,t,u,w\in\mathbb{B}} \{-u(\bar{x}+y+z+t-3) - w(\bar{y}+z+t-2) + zt\} \\ &= \min_{x,y,z,t,u,w\in\mathbb{B}} \{-u(-x+y+z+t-2) - w(-y+z+t-1) + zt\}.\end{aligned} \tag{19}$$

Although this particular example increases the number of extra variables compared to simply using (10), in general the combination of variables and their negations to minimize the number of auxiliary variables in the whole energy would depend on that particular energy.

Since there is an exponential number—not just in terms of the degree, but in terms of the number of occurrences of the variables in the whole energy—of possible combinations, finding the combination with the minimum number of necessary auxiliary variables seems a nontrivial problem. It may be an interesting direction for future research, especially if we allow an error tolerance for the energy value to deviate from the given exact value in favor of less auxiliary variables.

6 Higher-Order Graph Cuts

Turning our attention to higher-order MRFs with more than two labels, in this section we formalize a framework for approximately minimizing higher-order multilabel MRFs that combines the new reduction with the fusion-move and QPBO algorithms.

6.1 Graph Cuts and Move-Making Algorithms

Currently, one of the most popular optimization techniques in vision is α-expansion [4]. Here, we explain it and other algorithms to which the framework is directly related.

6.1.1 α-Expansion and α-β-Swap

The α-expansion and α-β-swap algorithms keep a current labeling and iteratively make a move, i.e., a change of labeling. In one iteration, the algorithms change the current labeling so that the energy becomes smaller or, at least, stays the same. Such algorithms in general are called the move-making algorithms.

In the α-expansion algorithm, a label $\alpha \in L$ is chosen at each iteration. The move is decided by either keeping the original label or replacing it with α at each pixel. Thus, the "area" labeled α can only expand; hence the name α-expansion.

The choice of whether changing the label or not at each pixel defines a binary labeling. The labeling after the move from $X \in L^V$ according to a binary labeling $y \in \mathbb{B}^V$ is defined by:

$$(X^{y,\alpha})_v = \begin{cases} X_v, & \text{if } y_v = 0, \\ \alpha, & \text{if } y_v = 1, \end{cases}$$

so that $y_v = 0$ means that X_v stays the same and $y_v = 1$ that X_v is changed to α.

An energy $\mathcal{E}(y)$ for the binary labeling y is defined by $\mathcal{E}(y) = E(X^{y,\alpha})$ as the multilabel energy after the move corresponding to binary labeling. By minimizing this binary energy, the move that reduces the energy most is chosen. When the binary problem is submodular, it can be solved globally by an s-t mincut algorithm. By visiting all labels α in some order and repeating it, $E(X)$ is approximately minimized.

In the case of α-β-swap, allowed moves are defined as those that, at each pixel, keep the current label or swap the two fixed labels α and β.

This was all done assuming that the energy $E(X)$ is a first-order MRF. The algorithm has a guarantee on how close it can approach the global minima in that case. Kohli et al. [15] investigated the case when $E(X)$ is of higher order, and characterized the class of higher-order energies for which the globally optimal α-expansion and α-β-swap moves can be computed in polynomial time.

6.1.2 Fusion Moves

The fusion move [22], [23] is a simple generalization of α-expansion: in each iteration, define the binary problem as the pixelwise choice between two arbitrary labelings, instead of between the current label and the fixed label α. Especially important to us is the variation where we maintain a current labeling $X \in L^V$ as in α-expansion and iteratively merge it with a "proposal" labeling $P \in L^V$ by arbitrarily choosing one of the two labelings to take the label from at each pixel. For instance, in the α-expansion algorithm, the proposal is a constant labeling that assigns the label $\alpha \in L$ to all pixels. Here, P can be any labeling.

It seems so simple and elegant that one may wonder why the move-making algorithm was not formulated this way from the beginning. The answer is simple: it is because fusion moves are nonsubmodular in general, whereas in the case of α-expansion and α-β-swap moves, the submodularity can be guaranteed by some simple criteria. It is only because of the emergence of the QPBO/roof-duality optimization that we can now consider the general fusion move.

6.1.3 Combining Fusion Moves and QPBO

The development of fusion moves and QPBO has an important implication in the optimization of higher-order energies. As we mentioned, the reduction in Section 3.1 has been known for some time. However, for the result of the reduction to be minimized with the popular techniques such as graph cuts, it must be submodular. This requirement has kept its actual use quite rare. In combination with the QPBO algorithm, higher-order potentials may be approximately minimized by move-making algorithms. The derived binary energy, which would be of the same order, can be reduced into pairwise one and then at least a part of the solution may be found by QPBO, improving the current labeling iteratively. This is done by changing the labels only at the nodes that were labeled by QPBO, which was proposed by Rother et al. [31].

The combination, which we might call the *higher-order graph cuts*, was first introduced by Woodford et al. [36],

TABLE 1
Number of necessary auxiliary variables.

Degree	d	3	4	5	6	7	8	9
Polynomial expansion (worst case)	$N^{\text{all}+}(d)$	1	5	17	49	129	321	769
Polynomial expansion (average)	$\left(N^{\text{all}+}(d)+N^{\text{all}-}(d)\right)/2$	1	5	16.5	45.5	114	270	617.5
All negative coefficients	2^d-1	7	15	31	63	127	255	511

where they showed that second-order smoothness priors can be used for stereo reconstruction by introducing an interesting framework. It integrates existing stereo algorithms, which may sometimes be rather *ad hoc*, and combines their results in a principled way. In the framework, an energy is defined to represent the global trade-off of various factors. The energy is minimized using the fusion-move algorithm, in which the various existing algorithms are utilized to generate the proposals. In particular, this allows powerful segmentation-based techniques to be combined with the global optimization methods.

6.2 Higher-Order Graph Cuts

Here, we introduce a general framework for approximate minimization of multilabel MRFs of general order, made possible by our transformation of general binary MRFs into first-order ones.

The algorithm maintains the current labeling X. In each iteration, the algorithm fuses X and a proposed labeling $P \in L^V$ by minimizing a binary energy. How P is prepared is problem-specific, as is how X is initialized at the beginning of the algorithm.

The result of the merge according to a binary labeling $y \in \mathbb{B}^V$ is defined by

$$\left(X^{y,P}\right)_v = \begin{cases} X_v, & \text{if } y_v = 0, \\ P_v, & \text{if } y_v = 1. \end{cases}$$

The binary energy to be minimized in each iteration is then

$$\mathcal{E}(y) = E\left(X^{y,P}\right).$$

For the energy $E(X)$ of the form (1), the polynomial expression of $\mathcal{E}(y)$ is

$$\mathcal{E}(y) = \sum_{C \in \mathcal{C}} \mathcal{E}_C(y_C) = \sum_{C \in \mathcal{C}} \sum_{\gamma \in \mathbb{B}^C} f_C\left(X_C^{\gamma,P}\right) \theta_C^{\gamma}(y_C), \tag{20}$$

where $X_C^{\gamma,P} \in L^C$ is the labeling $X^{\gamma,P}$ on the clique C and $\theta_C^{\gamma}(y_C)$ is a polynomial of degree $|C|$ defined by

$$\theta_C^{\gamma}(y_C) = \prod_{v \in C} y_v^{(\gamma)},$$

which, similarly to (14), is 1 if $y_C = \gamma$ and 0 otherwise.

The polynomial $\mathcal{E}(y)$ is then reduced into a quadratic polynomial using the technique described in the previous sections. Let $\tilde{\mathcal{E}}(\tilde{y})$ denote the resulting quadratic polynomial in terms of the labeling $\tilde{y} \in \mathbb{B}^W$, where $W \supset V$. We use the QPBO algorithm to minimize $\tilde{\mathcal{E}}(\tilde{y})$ and obtain a labeling $z \in \mathbb{B}^{W'}$ on a subset W' of W. Remember our notation for the restriction of labelings (3) and overwriting labelings (4) in Section 2. We have:

Proposition 2. *Let $y_0 \in \mathbb{B}^V$ be the labeling that assigns 0 to all $v \in V$ and $y_z = y_0 \lhd z|_{V \cap W'}$. Then,*

$$\mathcal{E}(y_z) \le \mathcal{E}(y_0).$$

Proof: Let us define

$$Y_0 = \{\tilde{y} \in \mathbb{B}^W \mid \tilde{y}|_V = y_0\}, \qquad \tilde{y}_0^{\min} = \arg\min_{\tilde{y} \in Y_0} \tilde{\mathcal{E}}(\tilde{y}),$$
$$Y_z = \{\tilde{y} \in \mathbb{B}^W \mid \tilde{y}|_V = y_z\}, \qquad \tilde{y}_z^{\min} = \arg\min_{\tilde{y} \in Y_z} \tilde{\mathcal{E}}(\tilde{y}).$$

Then, by the property of the reduction, we have $\mathcal{E}(y_0) = \tilde{\mathcal{E}}(\tilde{y}_0^{\min})$ and $\mathcal{E}(y_z) = \tilde{\mathcal{E}}(\tilde{y}_z^{\min})$. Also, if we define $\tilde{z} = \tilde{y}_0^{\min} \lhd z$, we have $\tilde{\mathcal{E}}(\tilde{z}) \le \tilde{\mathcal{E}}(\tilde{y}_0^{\min})$ by the autarky property. Since $\tilde{z} \in Y_z$, we also have $\tilde{\mathcal{E}}(\tilde{y}_z^{\min}) \le \tilde{\mathcal{E}}(\tilde{z})$. Thus, $\mathcal{E}(y_z) = \tilde{\mathcal{E}}(\tilde{y}_z^{\min}) \le \tilde{\mathcal{E}}(\tilde{z}) \le \tilde{\mathcal{E}}(\tilde{y}_0^{\min}) = \mathcal{E}(y_0)$. □

It follows that

$$E\left(X^{y_z,P}\right) = \mathcal{E}(y_z) \le \mathcal{E}(y_0) = E(X).$$

Therefore, we update X to $X^{y_z,P}$ and (often) decrease the energy. We iterate the process until some convergence criterion is met.

Proposal Generation. In higher-order graph cuts, it is crucial for the success and efficiency of the optimization to provide proposals that fit the energies being optimized. For higher-order energies, it is even more so because they have a richer class of null potentials. As mentioned above, using the outputs of other algorithms as proposals allows a principled integration of existing algorithms. In [12], a simple technique based on the gradient of the energy is proposed for generating proposal labelings that makes the algorithm much more efficient.

7 Relations to Other Methods

In this section, we discuss other known higher-order energy minimization methods that we found are closely related to the techniques introduced above. Specifically, we first investigate the binary transformation methods by Rother et al. [33] and then the $\mathcal{P}^n$ Potts model by Kohli et al. [15].

Note that other known methods, e.g., the multilabel minimization using the selection functions by Rother et al. [33] and the Robust $\mathcal{P}^n$ Potts model by Kohli et al. [14], do *not* fall in this category as far as we can tell.

7.1 Type-I and Type-II Transformation by Rother et al.

As we mentioned in Section 5.3, by taking

$$\lambda = \max_{\gamma \in \mathbb{B}^d} f(\gamma),$$

we can make all nonzero coefficients in the sum (17) nonpositive. Thus, any PBF can be transformed to a quadratic PBF using only the transformation (15), by making all coefficients nonpositive first. It turns out that is essentially the same as what are called the Type-I and Type-II transformations in Rother et al. [33].

Let $\theta > 0, \gamma \in \mathbb{B}^d$ and define $\psi(x)$ by:

$$\psi(x) = \begin{cases} 0, & \text{if } x = \gamma, \\ \theta, & \text{otherwise.} \end{cases}$$

7.1.1 Type-I Transformation

Using (15), $\psi(x) = \theta - \theta x_1^{(\gamma)} \cdots x_d^{(\gamma)}$ is reduced to

$$\begin{aligned}\psi(x) &= \theta + \theta \min_{w \in \mathbb{B}} w \left\{ \sum_{i=1}^{d} (-1)^{\gamma_i} x_i + \left| I_\gamma^1 \right| - 1 \right\} \\ &= \theta + \theta \min_{w \in \mathbb{B}} w \left\{ \sum_{i \in I_\gamma^0} x_i + \sum_{i \in I_\gamma^1} (1 - x_i) - 1 \right\} \\ &= \theta \min_{w \in \mathbb{B}} \left\{ 1 + vw + (1-v) \sum_{i \in I_\gamma^0} x_i + w \left(\sum_{i \in I_\gamma^1} (1 - x_i) - 1 \right) \right\},\end{aligned}$$

where we define $v = \bar{w} = 1 - w$. Note that $vw = 0$ and also the definition (13).

For $x \in \mathbb{B}^d, v \in \mathbb{B}$, and $w \in \mathbb{B}$, let us define

$$\varphi(x, v, w) = 1 + vw + (1-v) \sum_{i \in I_\gamma^0} x_i + w \left(\sum_{i \in I_\gamma^1} (1 - x_i) - 1 \right).$$

Then, we have

$$\varphi(x, 0, 0) = 1 + \sum_{i \in I_\gamma^0} x_i \geq 1 = \varphi(x, 1, 0)$$

and

$$\varphi(x, 1, 1) = 1 + \sum_{i \in I_\gamma^1} (1 - x_i) \geq 1 = \varphi(x, 1, 0).$$

This implies that the minimum of $\varphi(x, v, w)$ can always be achieved with v and w opposite. That is, having two variables v and w does not improve the minimization; freeing them achieves at best the same minima as in the case when we fix $v = \bar{w}$. Thus, we have

$$\psi(x) = \theta \min_{w \in \mathbb{B}} \varphi(x, \bar{w}, w) = \theta \min_{v, w \in \mathbb{B}} \varphi(x, v, w).$$

The last expression is the same as what is called the Type-I transformation in [33], which is given in the paper as:

$$\psi(x) = \min_{v, w \in \mathbb{B}} \theta \left\{ v + (1-w) - v(1-w) + \sum_{i \in I_\gamma^0} (1 - v) x_i + \sum_{i \in I_\gamma^1} w (1 - x_i) \right\}.$$

7.1.2 Type-II Transformation

In [33], the Type-II transformation is defined as:

$$\begin{aligned}\psi(x) = \theta + \frac{\theta}{2} \min_{w \in \mathbb{B}} \Bigg\{ & w(d-2) + w \left(\sum_{i \in I_\gamma^1} (1 - x_i) - \sum_{i \in I_\gamma^0} (1 - x_i) \right) \\ & + (1-w) \left(\sum_{i \in I_\gamma^1} x_i - \sum_{i \in I_\gamma^0} x_i \right) + \sum_{i \in I_\gamma^0} x_i - \sum_{i \in I_\gamma^1} x_i \Bigg\},\end{aligned}$$

which can be rewritten as

$$\psi(x) = \theta + \frac{\theta}{2} \min_{w \in \mathbb{B}} w \left\{ d - 2 + \sum_{i=1}^{d} (-1)^{\gamma_i} (2 x_i - 1) \right\}. \quad (21)$$

Using (15), $\psi(x) = \theta - \theta x_1^{(\gamma)} \cdots x_d^{(\gamma)}$ can also be written as

$$\begin{aligned}\psi(x) &= \theta + \theta \min_{w \in \mathbb{B}} w \left\{ \sum_{i=1}^{d} (-1)^{\gamma_i} x_i + \left| I_\gamma^1 \right| - 1 \right\} \\ &= \theta + \frac{\theta}{2} \min_{w \in \mathbb{B}} w \left\{ \sum_{i=1}^{d} (-1)^{\gamma_i} \cdot 2 x_i + \left| I_\gamma^1 \right| + \left(d - \left| I_\gamma^0 \right| \right) - 2 \right\} \\ &= \theta + \frac{\theta}{2} \min_{w \in \mathbb{B}} w \left\{ \sum_{i=1}^{d} (-1)^{\gamma_i} (2 x_i - 1) + d - 2 \right\},\end{aligned}$$

which coincides with (21).

To summarize, both Type-I and Type-II transformations in [33] are equivalent to (15), which is the γ-flipped version of the higher-order transformation (9) based on those by Kolmogorov and Zabih [17] and Freedman and Drineas [7].

7.2 $\mathcal{P}^n$ Potts Model by Kohli et al.

In [15], Kohli et al. define the class of the following form of clique potential and call it the $\mathcal{P}^n$ Potts model:

$$f_C(X_C) = \begin{cases} \nu_l, & \text{if } X_v = l \text{ for all } v \in C, \\ \nu_{\max}, & \text{otherwise,} \end{cases} \quad (22)$$

where $\nu_l \in \mathbb{R}$ is defined for each $l \in L$ so that $\nu_{\max} \geq \nu_l$.

We consider minimizing the energy (1) with the higher-order clique potential (22), using the higher-order graph-cut algorithm in Section 6, where the proposals emulate α-β-swap and α-expansion. In the following, we ignore the constant and linear energy terms.

7.2.1 α-β-Swap

The α-β-swap algorithm can be emulated as follows. Assume that we are in an iteration in the algorithm with the current labeling $X \in L^V$ and let $\alpha, \beta \in L, \alpha \neq \beta$. Let us denote

$$C_l = \{v \in C \,|\, X_v = l\}, \text{ for } l \in L,$$
$$C_\gamma = \{v \in C \,|\, \gamma_v = 1\}, \; C_{\bar{\gamma}} = \{v \in C \,|\, \gamma_v = 0\}.$$

We define the proposal labeling by

$$P_v = \begin{cases} \alpha, & \text{if } v \in C_\beta, \\ \beta, & \text{if } v \in C_\alpha, \\ X_v, & \text{otherwise.} \end{cases}$$

Let us consider the binary clique potential $f_C\left(X_C^{\gamma,P}\right)$ for clique C and binary labeling $\gamma \in \mathbb{B}^C$.

1. If $C_\alpha \cup C_\beta = C$,

$$f_C\left(X_C^{\gamma,P}\right) = \begin{cases} \nu_\alpha, & \text{if } C_{\bar{\gamma}} = C_\alpha \text{ and } C_\gamma = C_\beta, \\ \nu_\beta, & \text{if } C_\gamma = C_\alpha \text{ and } C_{\bar{\gamma}} = C_\beta, \\ \nu_{\max}, & \text{otherwise.} \end{cases}$$

2. If $C = C_l$, where $l \neq \alpha, l \neq \beta$, then

$$f_C\left(X_C^{\gamma,P}\right) = \nu_l,$$

since $P_v = X_v = l$ for all $v \in C$.

3. Otherwise, there is no way the whole clique would have the same label after an α-β-swap. Hence,

$$f_C\left(X_C^{\gamma,P}\right) = \nu_{\max}.$$

Remember the PBF $\mathcal{E}(y)$ in (20). We only have to consider the nonconstant case 1), when

$$\mathcal{E}_C(y_C) = \nu_{\max} + \delta_\alpha \prod_{v \in C_\alpha} \bar{y}_v \prod_{v \in C_\beta} y_v + \delta_\beta \prod_{v \in C_\alpha} y_v \prod_{v \in C_\beta} \bar{y}_v,$$

where $\delta_\alpha = \nu_\alpha - \nu_{\max}$ and $\delta_\beta = \nu_\beta - \nu_{\max}$.

Since $\delta_\alpha \leq 0$ and $\delta_\beta \leq 0$, the energy can be reduced by (15) as

$$\delta_\alpha \prod_{v \in C_\alpha} \bar{y}_v \prod_{v \in C_\beta} y_v = \min_{u_C \in \mathbb{B}} \delta_\alpha u_C \left(1 - |C_\beta| - \sum_{v \in C_\alpha} y_v + \sum_{v \in C_\beta} y_v\right), \tag{23}$$

$$\delta_\beta \prod_{v \in C_\alpha} y_v \prod_{v \in C_\beta} \bar{y}_v = \min_{w_C \in \mathbb{B}} \delta_\beta w_C \left(1 - |C_\alpha| + \sum_{v \in C_\alpha} y_v - \sum_{v \in C_\beta} y_v\right). \tag{24}$$

The transformation adds new variables up to twice the number of cliques. In practice, many cliques would be in cases 2) or 3) above and have the constant ν_l or $\nu_{\max}$ as their binary clique potential, reducing the number of variables in the binary minimization problem.

Unless $C = C_\alpha$ or $C = C_\beta$, neither (23) nor (24) falls in the special cases that allow polynomial-time global minimization, which we discussed in Section 5.2. However, for this particular energy, we can systematically flip variables so that the whole energy becomes globally optimizable. For instance, we can flip all variables in C_β, i.e., consider $\bar{y}_v$ as the variable instead of y_v for all $v \in V$ such that $X_v = \beta$. Then, the same variables are flipped regardless of which clique they appear in. The energy then becomes

$$\mathcal{E}_C(y_C) = \nu_{\max} + \min_{u_C \in \mathbb{B}} \delta_\alpha u_C \left(1 - \sum_{v \in C_\alpha} y_v - \sum_{v \in C_\beta} \bar{y}_v\right) + \min_{w_C \in \mathbb{B}} \delta_\beta w_C \left(1 - |C| + \sum_{v \in C_\alpha} y_v + \sum_{v \in C_\beta} \bar{y}_v\right),$$

and, as we discussed in Section 5.2, this form of energy can be globally minimized with graph cuts.

7.2.2 α-Expansion

Let us next consider using the proposal emulating α-expansion, i.e.,

$$P_v = \alpha \text{ for all } v \in V,$$

where $\alpha \in L$. Let us also denote

$$C_{\bar{\alpha}} = \{v \in C \,|\, X_v \neq \alpha\}.$$

Assume that we are in an iteration in the algorithm, with current labeling $X \in L^V$. Then the clique potential for the binary labeling $\gamma \in \mathbb{B}^C$ would be:

$$f_C\left(X_C^{\gamma,P}\right) = \begin{cases} \nu, & \text{if } \gamma_v = 0 \text{ for all } v \in C, \\ \nu_\alpha, & \text{if } \gamma_v = 1 \text{ for all } v \in C_{\bar{\alpha}}, \\ \nu_{\max}, & \text{otherwise,} \end{cases}$$

where ν depends on the current labeling X_C on the clique C:

$$\nu = \begin{cases} \nu_l, & \text{if } X_v = l \text{ for all } v \in C, \\ \nu_{\max}, & \text{otherwise.} \end{cases}$$

Let us define $\delta = \nu - \nu_{\max}$ and $\delta_\alpha = \nu_\alpha - \nu_{\max}$. The potential function $\mathcal{E}_C(y_C)$ in (20) would be:

$$\mathcal{E}_C(y_C) = \nu_{\max} + \delta \prod_{v \in C} \bar{y}_v + \delta_\alpha \prod_{v \in C_{\bar{\alpha}}} y_v.$$

Since $\delta \leq 0$ and $\delta_\alpha \leq 0$, the energy can be reduced by (15) as

$$\delta \prod_{v \in C} \bar{y}_v = \min_{u_C \in \mathbb{B}} \delta u_C \left(1 - \sum_{v \in C} y_v\right), \tag{25}$$

$$\delta_\alpha \prod_{v \in C} y_v = \min_{w_C \in \mathbb{B}} \delta_\alpha w_C \left(\sum_{v \in C_{\bar{\alpha}}} y_v + 1 - |C_{\bar{\alpha}}|\right). \tag{26}$$

Thus, the transformation adds new variables twice the number of cliques. In practice, the (25) part in $\mathcal{E}(y)$ vanishes for all but those cliques where the current labeling assigns the same label to all of the pixels in the clique.

Note also that (25) and (26) are the two special cases we discussed in Section 5.2 that allow polynomial-time global minimization.

Thus, we have reproduced the results described in Kohli et al. [15], Section 4.3. That is, in optimizing the $\mathcal{P}^n$ Potts model (22) with α-β-swap and α-expansion, the binary energy in each iteration can be solved globally using graph cuts.

8 EXPERIMENTS

As we discussed above, the $\mathcal{P}^n$ Potts model can be minimized using the higher-order graph cuts. Moreover, since the construction in Kohli et al. [15] and our algorithm both minimize the binary energy exactly, the results should be the same, except for the rare cases where there is more than one global minima. Since their experiments in texture-based segmentation uses the $\mathcal{P}^n$ Potts model, we can expect almost exactly the same results if we used the same energy in our algorithm.

Here, we report the results of comparing with other algorithms. The higher-order BP variants by Lan et al. [21] and Potetz [28] were both tested using a particular higher-order image restoration problem. We use the same problem to compare the effectiveness of the higher-order graph-cut algorithm with them.

8.1 Image Denoising with Fields of Experts

The image restoration scheme uses the recent image statistical model called the Fields of Experts (FoE) [30], which captures complex natural image statistics beyond pairwise interactions by providing a way to learn an image model from natural scenes. FoE has been shown to be highly effective, performing well at image denoising and image inpainting using a gradient descent algorithm.

The FoE model represents the prior probability of an image X as the product of several Student's t-distributions:

$$p(X) \propto \prod_{C} \prod_{i=1}^{K} \left(1 + \frac{1}{2}(J_i \cdot X_C)^2\right)^{-\alpha_i}, \tag{27}$$

where C runs over the set of all $n \times n$ patches in the image, and J_i is an $n \times n$ filter. The parameters J_i and α_i are learned from a database of natural images. In both [21] and [28], 2×2 patches were used to show that 2×2 FoE improves over pairwise models significantly.

We are given the noisy image $N \in L^V$ and find the maximum a posteriori solution given the prior model (27). The prior gives rise to a third-order MRF, with clique potentials that depend on up to four pixels. Since the purpose of the experiments is not the image restoration *per se*, but a comparison of the optimization algorithms, we use exactly the same simple model as in the two predecessors.

It is an inference problem with a simple likelihood term: image denoising with a known additive noise. We assume that the images have been contaminated with an i. i. d. Gaussian noise that has a known standard deviation σ. Thus, the likelihood of noisy image N given the true image X is assumed to satisfy

$$p(N|X) \propto \prod_{v \in V} \exp\left(-\frac{(N_v - X_v)^2}{2\sigma^2}\right). \tag{28}$$

To maximize the *a posteriori* probability $p(X|N)$ given the noisy image N, we have to maximize $p(N|X)p(X)$, the product of the likelihood and the prior probability.

8.2 Minimization Algorithm

We use the higher-order graph-cut algorithm described in Section 6 with the following ingredients.

Initialization. We initialize X by N, the given noisy image.

Convergence Criterion. We iterate until the energy decrease over 20 iterations drops below a convergence threshold θ_c. We used the values $\theta_c = 8$ for $\sigma = 10$ and $\theta_c = 100$ for $\sigma = 20$.

Proposal Generation. For proposal P, we use the following two in alternating iterations:

1. A uniform random image created each iteration.
2. A blurred image, which is made every 30 iterations by blurring the current image X with a Gaussian kernel ($\sigma = 0.5625$).

For comparison, we also test the α-expansion proposal, i.e., a constant image that gives the same value α everywhere.

Cliques. We consider the set $\mathcal{C}$ of cliques $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_4$, where $\mathcal{C}_1 = \{\{v\} \mid v \in V\}$ is the set of singleton cliques and $\mathcal{C}_4$ is the set of all 2×2 patches in the image.

Energy. The local energy $f_C(X_C)$ in (1) is defined by

$$f_{\{v\}}(X_{\{v\}}) = \frac{(N_v - X_v)^2}{2\sigma^2}, \qquad (\{v\} \in \mathcal{C}_1)$$

$$f_C(X_C) = \sum_{i=1}^{K} \alpha_i \log\left(1 + \frac{1}{2}(J_i \cdot X_C)^2\right). \qquad (C \in \mathcal{C}_4)$$

These are negative logarithm of the clique potentials in the likelihood (28) and the prior (27), respectively. Minimizing this energy is the same as maximizing the *a posteriori* probability $p(X|N)$ given the data N.

QPBO. For the QPBO algorithm, we used the C++ code made publicly available by Vladimir Kolmogorov. We also used the variations of QPBO included in the code.

8.3 Results

8.3.1 Qualitative Difference

Fig. 1 shows the qualitative difference between the denoising results using first-order and third-order energies. Images in Figs. 1a and 1b show the original image and the image with noise added at $\sigma = 20$, respectively. All images in this figure are 160×240-pixel 8-bit grayscale images. Images in Figs. 1c and 1d show the result of denoising using α-expansion by first-order energy

$$E(X) = \sum_{v \in V} (N_v - X_v)^2 + \lambda \sum_{(u,v)} \mathrm{Potts}(X_u, X_v), \tag{29}$$

where the pairwise Potts potential is defined for vertical and horizontal neighbors $u, v \in V$ so that $\mathrm{Potts}(X_u, X_v)$ is 1 if $X_u = X_v$ and 0 otherwise; and $\lambda = 250$ for Fig. 1c and $\lambda = 400$ for Fig. 1d. The image in Fig. 1e shows the result of denoising by first-order total-variation energy:

$$E(X) = \sum_{v \in V} (N_v - X_v)^2 + \lambda \sum_{(u,v)} |X_u - X_v|. \tag{30}$$

The energy was globally minimized using the technique in [10] with $\lambda = 10$. Finally, the image in Fig. 1f shows the result of denoising by the FoE energy described above.

The Potts model favors piecewise-constant images, which can be clearly seen in the results. If λ is small, much noise remains, as in Fig. 1c. If we make λ larger, the result becomes too flat, as in Fig. 1d. The total variation does a better job (as in Fig. 1e), but the same dilemma as with the Potts model still remains. The result (as in Fig. 1f) by the third-order energy clearly is the best qualitatively.

Some more denoising examples are shown in Fig. 2.

8.3.2 Quantitative Comparison

For quantitative comparison, we measured the mean energy and the mean peak signal-to-noise ratio (PSNR) for denoising results over the same set of 10 images that were also used in both [21] and [28]. The images are from the Berkeley segmentation database [24], grayscaled and reduced in size, as well as with the added Gaussian noise with $\sigma = 10, 20$. Here, PSNR = $20\log_{10}(255/\sqrt{\text{MSE}})$, where MSE is the mean squared error. The test images and the FoE model were kindly provided by Stefan Roth, one of the authors of [21], [30]. Brian Potetz also obliged by providing us with his results.

The PSNR and energy numbers are listed in Table 2. For comparison, we also included the result by simple gradient descent by [30]. The energy results show that the higher-order graph cuts using the third-order energy outperforms the both BP variants optimizing the same energy. The PSNR is also comparable to [28] and better than [21]. Our algorithm takes 6-10 minutes ($250-280$ iterations) to converge on a 2.93 GHz Xeon E5570 processor. By comparison, according to [28], it took 30-60 minutes on a 2.2 GHz Opteron 275, while the algorithm in [21] takes 8 hours on a 3 GHz Xeon. Thus, our algorithm outperforms the two predecessors in quality and speed, though BP is considered to be substantially more parallelizable than graph cuts.

8.3.3 Comparison by Reductions, Proposals, and QPBO Variations

Fig. 3 shows the behavior of some numbers during the optimization when $\sigma = 20$. The three graphs show energy, PSNR, and the ratio of pixels that is labeled by QPBO.

First, compare the solid dark-red diamonds and the solid purple squares. They represent the two ways of handling positive-coefficient terms in binary energy reduction, both using the blur and random proposal and the QPBO algorithm. Shown as the dark-red diamonds, using (11) in Theorem 1 is a little slower in this particular case. Flipping one variable in a positive-coefficient term before reducing it by the Kolmogorov-Zabih technique and then flipping the variable back (see, for example, (19)) yields the purple square plot. It is a little faster, due to the higher ratio of pixels labeled by the QPBO algorithm, although it uses more memory because it needs more (about 4.4 percent in this case) auxiliary variables.

Next, compare the two dark-red plots marked by hollow and solid diamonds in each graph. These compare the blur and random proposal (solid diamond) and α-expansion proposal, using the QPBO algorithm. The result suggests that for higher-order energies, α-expansion does not work very well. The energy never reached the same level as the result using the blur and random proposal. In the case of $\sigma = 10$, α-expansion did go down to about the same energy level, but took significantly longer. In the experiments using the blur and random proposal, the average percentage of the pixels labeled by the QPBO algorithm over two consecutive steps (blur step and random step) starts around 50 percent and almost steadily goes up to about 80 percent when $\sigma = 20$, and from 80 percent to almost 100 percent when $\sigma = 10$. The plot shows the zigzag shape because in blur steps the ratio is consistently lower than the random step. Using the α-expansion proposal, the ratio of the labeled pixels is always less than 20 percent, resulting in a slow decrease of energy that does not reach the same level as the blur and random proposal.

Finally, disregard the α-expansion plot. The rest of the plots (with solid marks) all use the same blur and random proposal. They use different variations of the QPBO algorithms introduced by Rother et al. [31]. Besides the plain QPBO, the extended algorithms we tried are QPBOP and QPBOI. QPBOP tries to label more pixels while preserving the global optimality. QPBOI goes further trying to approximate the solution. The result shows that, while QPBOP tends to decrease energy the most in a single step, in overall performance, QPBOI works best because QPBOP takes more time at each step.

8.3.4 Comparison with Reduction by Substitution

When we use the "reduction by substitution" method explained in Section 3.2, the percentage of labeled pixels stays at almost zero percent, with averages 0.00018 percent and 0.091 percent over 100 iterations for $\sigma = 20$ and 10, respectively, with almost no energy reduction.

8.3.5 Comparison with Type-I and -II Transformations

We also tried the reduction we showed equivalent to the Type-I and -II transformations by Rother et al. The percentage of labeled pixels was very low (at most 2 percent), and most of those labeled were labeled 0, leading to almost no energy reduction. Perhaps this is not too surprising since, at degree four, 15 extra variables are needed (compared to five in the case of the new reduction) for each 4-pixel clique, as we listed in Table 1.

9 Conclusions

In this paper, we have introduced a new transformation of minimization problem of general Markov random fields with binary labels into an equivalent first-order problem. In combination with the fusion-move and QPBO algorithms, it can be used to approximately minimize higher-order energies even with more than two labels.

We have validated the technique by minimizing a third-order potential for image denoising. The results show that the algorithm exceeds the preceding BP algorithms in both optimization capability and speed.

We have also investigated the relations between our method and two similar methods. The binary energy reduction by

TABLE 2
PSNR and Energy, averaged over 10 images using the same FoE model and different optimizations

Noise level	Roth&Black [30]	Lan et al. [21]	Potetz [28]	Our result
$\sigma = 10$	30.53 / 49068	30.36 / 40236	31.54 / 36765	31.44 / 35896
$\sigma = 20$	26.09 / 61284	27.05 / 33053	27.25 / 31801	27.43 / 30858

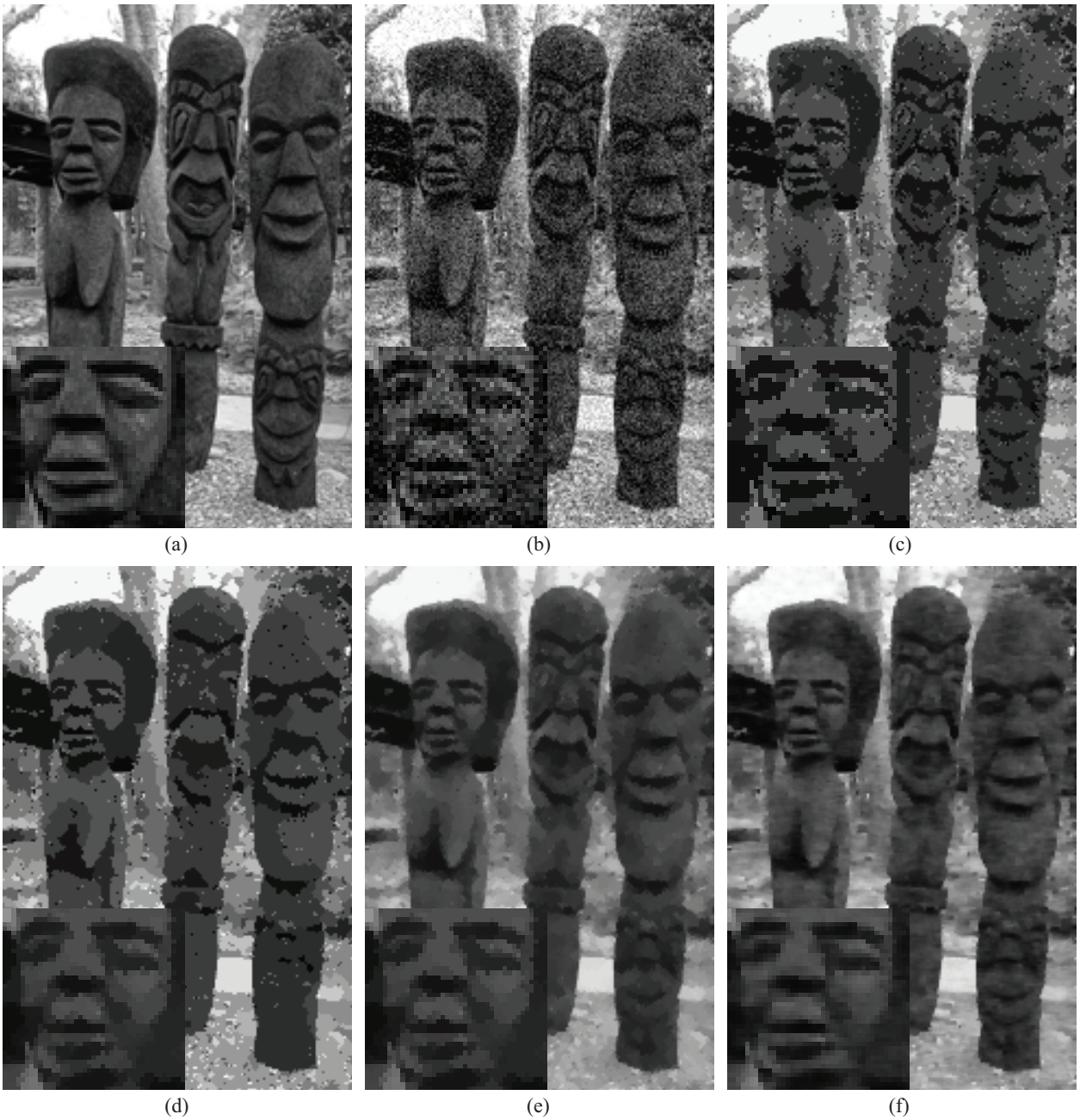


Fig. 1. Qualitative difference of denoising using cliques of different orders. (a) Original image. 160×240 pixel 8-bit grayscale. (b) Noise-added image (Gaussian i. i. d., $\sigma = 20$, PSNR $= 22.31$). (c)(d) Denoised using first-order Potts model (29) and α-expansion with two smoothing factors (c) $\lambda = 250$ (PSNR $= 23.55$), (d) $\lambda = 400$ (PSNR $= 23.66$). (e) Denoised using first-order Total Variation model (30) (PSNR $= 25.67$). (f) Denoised using third-order FoE model (PSNR $= 26.32$).

Fig. 2. More image restoration results. (Left column) Original images. The size is 240×160 pixels. (Middle column) Noisy images ($\sigma = 20$). PSNR=22.63 and 22.09 from top to bottom. (Right column) Restored images. PSNR=30.92 and 28.29 from top to bottom.

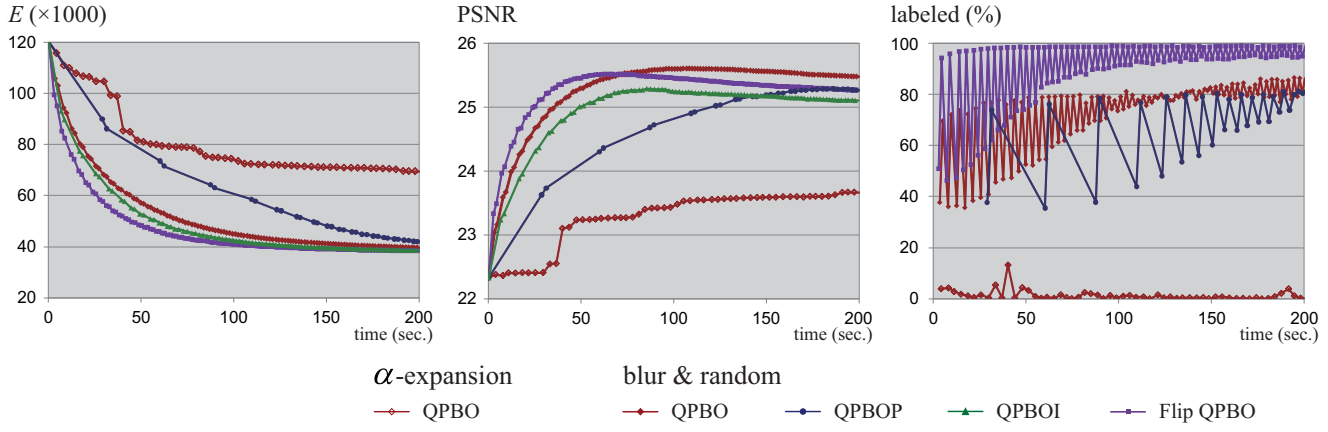


Fig. 3. Comparison of reduction and proposal methods and QPBO variations. Plots during the restoration of the example image in Fig. 1.

Rother et al. [33] turned out to be the same as a special case of our generalized binary reduction. The $\mathcal{P}^n$ Potts model by Kohli et al. [15] can also be approximately minimized within our framework by finding exactly the optimal move in each iteration.

Although we presented the transformation of binary energy mainly in relation with the iterative algorithm that we call the higher-order graph cuts, the transformation may be used in combination with other algorithms. For instance, Ramalingam et al. [34] give a transformation that can convert higher-order multilabel MRF into a higher-order binary-label one. Then, they use what we call the reduction by substitution to further reduce the energy to first order. If the result of our experiments with that reduction is any indication, our reduction should also work better with their algorithm.

Acknowledgments

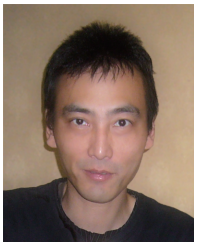
The author thanks Stefan Roth for providing the test images and the FoE models, Brian Potetz for providing his results, and Vladimir Kolmogorov for making his QPBO code publicly available. This work was partially supported by the Kayamori Foundation and the Grant-in-Aid for Scientific Research 19650065 from the Japan Society for the Promotion of Science.

References

[1] E. Boros, P. L. Hammer, R. Sun, and G. Tavares, "A Max-Flow Approach to Improved Lower Bounds for Quadratic Unconstrained Binary Optimization (QUBO)," *Discrete Optimization*, vol. 5, no. 2, pp. 501-529, 2008.

[2] E. Boros and P. L. Hammer, "Pseudo-Boolean Optimization," *Discrete Appl. Math.*, vol. 123, nos. 1-3, pp. 155-225, 2002.

[3] E. Boros, P. L. Hammer, and G. Tavares, "Preprocessing of Unconstrained Quadratic Binary Optimization," RUTCOR Research Report 10-2006, Rutgers Center for Operations Research, Rutgers Univ., Apr. 2006.
[4] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222-1239, Nov. 2001.
[5] D. Cremers and L. Grady, "Statistical Priors for Efficient Combinatorial Optimization via Graph Cuts," *Proc. European Conf. Computer Vision*, vol. 3, pp. 263-274, 2006.
[6] P. Felzenszwalb and D. Huttenlocher, "Efficient Belief Propagation for Early Vision," *Int'l. J. Computer Vision,* vol. 70, pp.41-54, 2006.
[7] D. Freedman and P. Drineas, "Energy minimization via Graph Cuts: Settling What Is Possible," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 939-946, 2005.
[8] P. L. Hammer, P. Hansen, and B. Simeone, "Roof Duality, Complementation and Persistency in Quadratic 0-1 Optimization," *Math. Programming*, vol. 28, pp. 121-155, 1984.
[9] P. L. Hammer and S. Rudeanu, *Boolean Methods in Operations Research and Related Areas*, Springer-Verlag, 1968.
[10] H. Ishikawa, "Exact Optimization for Markov Random Fields with Convex Priors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1333-1336, Oct. 2003.
[11] H. Ishikawa, "Higher-Order Clique Reduction in Binary Graph Cut," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 2993-3000, 2009.
[12] H. Ishikawa, "Higher-Order Gradient Descent by Fusion-Move Graph Cut," *Proc. IEEE Int'l Conf. Computer Vision, pp. 568-574, 2009.*
[13] H. Ishikawa and D. Geiger, "Rethinking the Prior Model for Stereo," *Proc. European Conf. Computer Vision*, vol. 3, pp. 526-537, 2006.
[14] P. Kohli, L. Ladicky, and P. H. S. Torr, "Robust Higher Order Potentials for Enforcing Label Consistency," *Int'l. J. Computer Vision,* vol. 82, no. 3, pp. 302-324, 2009.
[15] P. Kohli, M. P. Kumar, and P. H. S. Torr, "$\mathcal{P}^3$ & Beyond: Move Making Algorithms for Solving Higher Order Functions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1645-1656, 2009.
[16] V. Kolmogorov, "Convergent Tree-Reweighted Message Passing for Energy Minimization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, Oct. 2006.
[17] V. Kolmogorov and R. Zabih, "What Energy Functions Can Be Minimized via Graph Cuts?" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147-159, Feb. 2004.
[18] V. Kolmogorov and C. Rother, "Minimizing Non-submodular Functions with Graph Cuts—A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1274–1279, Jul. 2007.
[19] N. Komodakis and N. Paragios, "Beyond Pairwise Energies: Efficient Optimization for Higher-Order MRFs," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2009.
[20] J. D. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proc. Int'l Conf. Machine Learning*, pp. 282–289, 2001.
[21] X. Lan, S. Roth, D. P. Huttenlocher, and M. J. Black, "Efficient Belief Propagation with Learned Higher-Order Markov Random Fields," *Proc. European Conf. Computer Vision*, vol. 2, pp. 269-282, 2006.
[22] V. Lempitsky, C. Rother, and A. Blake, "LogCut—Efficient Graph Cut Optimization for Markov Random Fields," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.
[23] V. Lempitsky, S. Roth, and C. Rother, "FusionFlow: Discrete-Continuous Optimization for Optical Flow Estimation," in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2008.
[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 416-423, 2001.
[25] T. Meltzer, C. Yanover, and Y. Weiss, "Globally Optimal Solutions for Energy Minimization in Stereo Vision Using Reweighted Belief Propagation," *Proc. IEEE Int'l Conf. Comp. Vision,* pp. 428-435, 2005.
[26] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An Analysis of Approximations for Maximizing Submodular Set Functions," *Math. Programming,* vol. 14, no. 1, pp. 265-294, 1978.
[27] R. Paget and I. D. Longstaff, "Texture Synthesis via a Noncausal Nonparametric Multiscale Markov Random Field," *IEEE Trans. Image Processing*, vol. 7, no. 6, pp. 925-931, 1998.
[28] B. Potetz, "Efficient Belief Propagation for Vision Using Linear Constraint Nodes," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2007.
[29] I. G. Rosenberg, "Reduction of Bivalent Maximization to the Quadratic Case," *Cahiers du Centre d'Etudes de Recherche Operationnelle,* vol. 17, pp. 71-74, 1975.
[30] S. Roth and M. J. Black, "Fields of Experts: A Framework for Learning Image Priors," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 860-867, 2005.
[31] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing Binary MRFs via Extended Roof Duality," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2007.
[32] C. Rother, V. Kolmogorov, T. Minka, and A. Blake, "Cosegmentation of Image Pairs by Histogram Matching—Incorporating a Global Constraint into MRFs," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 993-1000, 2006.
[33] C. Rother, P. Kohli, W. Feng, and J. Jia, "Minimizing Sparse Higher Order Energy Functions of Discrete Variables," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2009.
[34] S. Ramalingam, P. Kohli, K. Alahari, and P. H. S. Torr, "Exact Inference in Multi-label CRFs with Higher Order Cliques," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2008.
[35] S. Vicente, V. Kolmogorov, and C. Rother, "Graph Cut Based Image Segmentation with Connectivity Priors," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2008.
[36] O. J. Woodford, P. H. S. Torr, I. D. Reid, and A. W. Fitzgibbon, "Global Stereo Reconstruction under Second Order Smoothness Priors," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2008.



Hiroshi Ishikawa received the BS and MS degrees in mathematics from Kyoto University in 1991 and 1993, respectively. In 2000, he received the PhD degree in computer science as well as the Harold Grad Memorial Prize from the Courant Institute of Mathematical Sciences, New York University. He worked as an associate research scientist at the Courant Institute until 2001, after which he cofounded Nhega, LLC, in New York City. In 2004, he joined the Department of Information and Biological Sciences, Nagoya City University, as an assistant professor. In 2010, he moved to the Department of Computer Science and Engineering, Waseda University, Tokyo, as a professor. Currently, he is also a Sakigake Researcher in the JST Mathematics Program at the Japan Science and Technology Agency. Dr. Ishikawa has also received the IEEE Computer Society Japan Chapter Young Author Award in 2006 and the MIRU Nagao Award in 2009. He is a member of the IEEE.