Minimizing Sparse Higher Order Energy Functions of Discrete Variables

Carsten Rother Pushmeet Kohli
Microsoft Research Cambridge
{carrot,pkohli}@microsoft.com

Wei Feng Jiaya Jia
The Chinese University of Hongkong
{wfeng,leojia}@cse.cuhk.edu.hk

Abstract

Higher order energy functions have the ability to encode high level structural dependencies between pixels, which have been shown to be extremely powerful for image labeling problems. Their use, however, is severely hampered in practice by the intractable complexity of representing and minimizing such functions. We observed that higher order functions encountered in computer vision are very often "sparse", i.e. many labelings of a higher order clique are equally unlikely and hence have the same high cost. In this paper, we address the problem of minimizing such sparse higher order energy functions. Our method works by transforming the problem into an equivalent quadratic function minimization problem. The resulting quadratic function can be minimized using popular message passing or graph cut based algorithms for MAP inference. Although this is primarily a theoretical paper, it also shows how higher order functions can be used to obtain impressive results for the binary texture restoration problem.

1. Introduction

Many computer vision problems such as object segmentation, disparity estimation, and 3D reconstruction can be formulated as pixel or voxel labeling problems. The conventional methods for solving these problems use pairwise Conditional and Markov Random Field (CRF/MRF) formulations [20], which allow for the exact or approximate inference of Maximum a Posteriori (MAP) solutions using extremely efficient algorithms such as Belief Propagation (BP) [4, 15, 22], graph cuts [2] and Tree-Reweighted (TRW) [9, 21] message passing. Although pairwise random field models permit efficient inference, they have restricted expressive power as they can only model interactions between pairs of random variables. They are unable to enforce the high level structural dependencies between pixels which have been shown to be extremely powerful for image labeling problems.

The last few years have seen the successful application of higher order CRFs and MRFs to some low level vision problems such as image restoration, disparity estimation and object segmentation [7, 14, 16, 23, 24]. In spite of these encouraging results, the use of such models have not spread to other labeling problems. We believe that this is primarily due to the lack of efficient algorithms for performing inference in such models. This paper proposes a method for minimizing general higher order functions that can be used to perform MAP inference in higher order random fields.

We follow the classical approach for minimizing higher order functions which can be broken down into two essential steps [1]: **(a)** *Transformation of the higher order energy into a quadratic function, and* **(b)** *Minimization of the resulting function using efficient inference algorithms.* The first step in this approach is also the most crucial one. Transformation of a general m-order function to an equivalent quadratic function involves the addition of exponential number of auxiliary variables [1, 5]. Alternatively, the addition of a single random variable with an exponential label space is needed. Both these approaches make the resulting quadratic function minimization problem intractable. Recent work on solving higher order functions in vision have side-stepped the problem of minimizing general higher order functions. Instead they have focused on specific families of potential functions (such as the P^n Potts model [7]) which can be transformed to quadratic ones by the addition of a few auxiliary variables.

In this paper, we address the problem of minimizing general higher order functions. This is intrinsically a computationally expensive problem since even the parametrization of a general m order function of k-state variables requires k^m parameters. However, the higher order functions used in computer vision have certain properties like *sparseness* which makes them easy to handle. A typical example would be the patch based potentials used for image restoration. It is well-known that the set of 5×5 patches of natural images is a small subset of the set of all possible 5×5 patches. Higher order potentials used for image restoration enforce that patches in the restored image come from the set of natural image patches. In other words, these functions assign a low cost (or energy) to only a few label assignments (natural patches). The rest of the labelings (artificial patches) are given a high (almost constant) cost (see section 5 for more details). We show how such sparse higher order functions can be transformed to quadratic ones with the addition of only a small number of auxiliary variables and edges. It should be noted that our method allows for the exact transformation of general higher order functions to quadratic functions albeit with an addition of exponential number of auxiliary variables in the worst case.

Outline of the Paper We provide our notation and review

discrete energy minimization in Section 2. In Section 3, we show how higher order energy functions can be transformed to quadratic ones using a multi-state auxiliary variable. Section 4 explains how higher order pseudo-boolean functions can be transformed to quadratic pseudo-boolean functions by the addition of boolean variables, and specifically presents two types of such transformations. Section 5 describes the experimental evaluation of our transformation schemes on the binary texture restoration problem.

2. Notation and Preliminaries

Consider a random field defined over a set of latent variables $\mathbf{x} = \{x_i | i \in \mathcal{V}\}$ where $\mathcal{V} = \{1, 2, ..., n\}$. Each random variable x_i can take a label from the label set $\mathcal{L} = \{l_1, l_2, ..., l_k\}$. Let $\mathcal{C}$ represent a set of subsets of $\mathcal{V}$ (i.e., cliques), over which the higher order random field is defined. The MAP solution of a random field can be found by minimizing an energy function $E : \mathcal{L}^n \rightarrow \mathbb{R}$. Energy functions corresponding to higher order random fields can be written as a sum of higher order potential functions as: $E(\mathbf{x}) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{x_c})$, where $\mathbf{x_c}$ represents the set of random variables included in any clique $c \in \mathcal{C}$. The higher order potential $\psi_c : \mathcal{L}^{|c|} \rightarrow \mathbb{R}$ is defined over this clique assigns a cost to each possible configurations (or *labelings*) of $\mathbf{x_c}$. Here $|c|$ represents the number of variables included in the clique (also called the clique order) .

Minimizing Quadratic Functions Before proceeding further, we review the basics of discrete energy minimization algorithms in computer vision. As we mentioned earlier, the problem of MAP inference in a pairwise random field can be solved by minimizing a quadratic function of discrete variables. Algorithms for MAP inference can be classified into two broad categories: (a) message passing (BP/TRW) and (b) combinatorial algorithms such as graph cuts. The readers should refer to [9, 21, 22] and [2] for more information on message passing and graph cut based algorithms respectively. In their classical form, these algorithms allow for the exact or approximate minimization of quadratic energy functions with certain computation time and solution quality guarantees. We will next look at the minimization of functions of boolean variables.

Quadratic Pseudo-boolean Function Minimization An energy function is called a pseudo-boolean function if the label set $\mathcal{L}$ contains only two labels i.e. $\mathcal{L} = \{0, 1\}$. Formally, the energy is now defined as: $E : \{0, 1\}^n \rightarrow \mathbb{R}$ and can also be written as a set function. The minimization of pseudo-boolean functions is a well studied problem in combinatorial optimization [6] and operations research [1]. It is known that certain classes of pseudo-boolean functions such as *submodular* functions can be minimized exactly in polynomial time. Another important characteristic is that any *Quadratic* Pseudo-boolean Function (QPBF) can be minimized by solving an st minimum cut problem (st-mincut) [1]. Further, if the QPF is *submodular*, all edges in the equivalent st-mincut problem have non-negative weights, which allows it to be solved exactly in polynomial time using maximum flow algorithms [1].

In what follows, we will assume that we have algorithms for approximate minimization of arbitrary multi-label quadratic energy functions, and mainly focus our attention on converting a general higher order energy function to a quadratic one.

3. Transforming Multi-label Functions

We will now describe how to transform arbitrary higher order potential functions to equivalent quadratic ones. We start with a simple example to motivate our transformation. Consider a higher order potential function which assigns a cost θ_0 if the variables $\mathbf{x}_c$ take a particular labeling $\mathbf{X}_0 \in \mathcal{L}^{|c|}$, and θ_1 otherwise. More formally,

$$\psi_c(\mathbf{x}_c) = \begin{cases} \theta_0 & \text{if} \quad \mathbf{x}_c = \mathbf{X}_0 \\ \theta_1 & \text{otherwise.} \end{cases} \tag{1}$$

where $\theta_0 \leq \theta_1$, and $\mathbf{X}_0$ denotes a particular labeling of the variables $\mathbf{x}_c$. The potential is illustrated in figure 1(b). The minimization of this higher order function can be transformed to the minimization of a quadratic function using one additional *switching* variable z as:

$$\min_{\mathbf{x}_c} \psi_c(\mathbf{x}_c) = \min_{\mathbf{x}_c, z \in \{0,1\}} f(z) + \sum_{i \in c} g_i(z, x_i) \tag{2}$$

where the *selection* function f is defined as: $f(0) = \theta_0$ and $f(1) = \theta_1$, while the *consistency* function g_i is defined as:

$$g_i(z, x_i) = \begin{cases} 0 & \text{if} \quad z = 1 \\ 0 & \text{if} \quad z = 0 \text{ and } x_i = \mathbf{X}_0(i) \\ \inf & \text{otherwise.} \end{cases} \tag{3}$$

where $\mathbf{X}_0(i)$ denotes the label of variable x_i in labeling $\mathbf{X}_0$.

3.1. General Higher-order Potentials

The method used to transform the simple potential function (1) can also be used to transform any higher order function into a quadratic one. We observed that higher order potentials for many vision problems assign a low cost (or energy) to only a few label assignments. The rest of the labelings are given a high (almost constant) cost (see figure 1(a)). This motivated to develop a parameterization of higher order potentials which exploits this sparsity. We parameterize higher order potentials by a list of possible labelings $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_t\}$ of the clique variables $\mathbf{x}_c$, and their corresponding costs $\mathbf{\Theta} = \{\theta_1, \theta_2, ..., \theta_t\}$. We also include a high constant cost $\theta_{\max}$ for all other labelings. Formally, the potential functions can be defined as:

$$\psi_c(\mathbf{x}_c) = \begin{cases} \theta_q & \text{if } \mathbf{x}_c = \mathbf{X}_q \in \mathcal{X} \\ \theta_{\max} & \text{otherwise.} \end{cases} \tag{4}$$

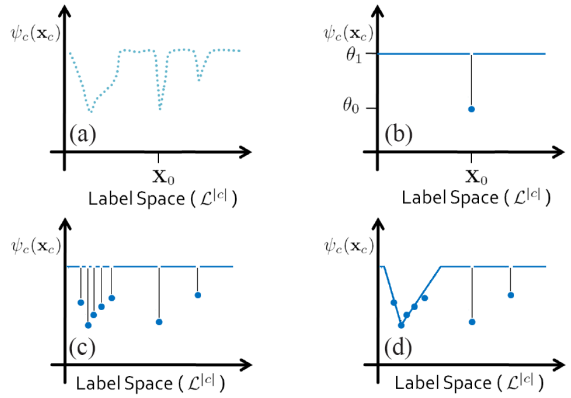

Figure 1. *Different parameterizations of higher order potentials. (a) The original higher order potential function. (b) The higher order basis function defined in equation (1). (c) Approximating function (a) using the functional form defined in equation (4). It can be seen that this representation requires the definition of 7 labelings (t=7), and thus would require the addition of a $t+1=8$-state auxiliary variable for its transformation to a quadratic function (as described in equation 5). (d) The compact representation of the higher function using the functional form defined in equation (7). This representation (7) requires only $t=3$ deviation functions, and thus needs only a $t+1=4$-state label to yield a quadratic transformation.*

where $\theta_q \leq \theta_{\max}, \forall \theta_q \in \mathbf{\Theta}$. The higher order potential is illustrated in Figure 1(c).

The minimization of the above defined higher order function can be transformed to a quadratic function using a $(t+1)$-state switching variable as:

$$\min_{\mathbf{x}_c} \psi_c(\mathbf{x}_c) = \min_{\mathbf{x}_c, z \in \{1,2,\ldots,t+1\}} f(z) + \sum_{i \in c} g(z, x_i) \quad (5)$$

$$\text{where} \quad f(z) = \begin{cases} \theta_q & \text{if} \quad z = q \in \{1,..,t\} \\ \theta_{\max} & \text{if} \quad z = t+1, \end{cases} \quad (6)$$

$$\text{and } g_i(z, x_i) = \begin{cases} 0 & \text{if} \quad z = q \in \{1,..,t\} \text{ and } x_i = \mathbf{X}_q(i) \\ 0 & \text{if} \quad z = t+1 \\ \text{inf} & \text{otherwise.} \end{cases}$$

where $\mathbf{X}_q(i)$ denotes the label of variable x_i in labeling $\mathbf{X}_q$. The reader should observe that the last i.e. $(t+1)^{th}$ state of the switching variable z does not penalize any labeling of the clique variables $\mathbf{x}_c$. It should also be noted that the transformation method described above can handle the P^n model potentials proposed in [7]. In fact it can be used to transform any general higher order potential. However, in the worst case, the addition of a switching variable with $|\mathcal{L}|^{|c|}$ states is required, which makes minimization of even moderate order functions infeasible.

3.2. Compact Parameterization

The above defined parametrization significantly reduces the complexity of performing inference in higher order cliques. However, the computation cost is still quite high for potentials which assign a low cost to many labelings. Notice, that the representation defined in equation (5) requires a $t+1$-state auxiliary variable for representing a higher order function where t labelings are assigned a low cost (less than the constant cost $\theta_{\max}$). This would make the use of this representation infeasible for higher order potentials where a large number of labelings of the clique variables are assigned low weights ($< \theta_{\max}$).

We observed that many low cost label assignments tend to be close to each other in term of the difference between labelings of pixels. For instance, consider the case of the two label foreground (f) / background (b) image segmentation problem. It is conceivable that the cost of a segmentation labeling ($fffb$) for 4 adjacent pixels on a line would be close to the cost of the labeling ($ffbb$). We can encode the cost of such groups of *similar* labelings in the higher order potential in such a way that their transformation to quadratic functions does not require increasing the number of states of the switching variable z. The differences of the representations are illustrated in figure 1(c) and (d).

We parameterize the compact higher order potentials by a list of labeling deviation cost functions $\mathcal{D} = \{d_1, d_2, ..., d_t\}$, and a list of associated costs $\theta = \{\theta_1, \theta_2, ..., \theta_t\}$. We also maintain a parameter for the maximum cost $\theta_{\max}$ that the potential can assign to any labeling. The deviation cost functions encode how the cost changes as the labeling moves away from some desired labeling. Formally, the potential functions can be defined as:

$$\psi_c(\mathbf{x}_c) = \min\{\min_{q \in \{1,2,\ldots,t\}} \theta_q + d_q(\mathbf{x}_c), \theta_{\max}\} \quad (7)$$

where deviation functions $d_q : \mathcal{L}^{|c|} \rightarrow \mathbb{R}$ are defined as: $d_q(\mathbf{x}_c) = \sum_{i \in c; l \in \mathcal{L}} w_{il}^q \delta(x_i = l)$, where w_{il}^q is the cost added to the deviation function if variable x_i of the clique c is assigned label l. The function $\delta(x_i = l)$ is the Kronecker delta function that returns value 1 if $x_i = l$ and returns 0 for all assignments of x_i. This higher order potential is illustrated in Figure 1(d). It should be noted that the higher order potential (7) is a generalization of the potential defined in (4). Setting weights w_{il}^q as:

$$w_{il}^q = \begin{cases} 0 & \text{if} \quad \mathbf{X}_q(i) = l \\ \theta_{\max} & \text{otherwise} \end{cases} \quad (8)$$

makes potential (7) equivalent to (4).

The minimization of the above defined higher order function can be transformed to that a quadratic function using a $(t+1)$-state switching variable as:

$$\min_{\mathbf{x}_c} \psi_c(\mathbf{x}_c) = \min_{\mathbf{x}_c, z \in \{1,2,\ldots,t+1\}} f(z) + \sum_{i \in c} g(z, x_i) \quad (9)$$

$$\text{where} \quad f(z) = \begin{cases} \theta_q & \text{if} \quad z = q \in \{1,..,t\} \\ \theta_{\max} & \text{if} \quad z = t+1, \end{cases} \quad (10)$$

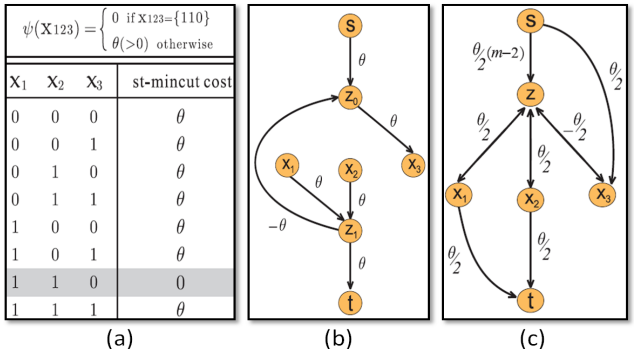

Figure 2. *Transformation of higher order pseudo-boolean functions to equivalent quadratic functions. A higher order pseudo-boolean function (a) represented by its truth table. Type-I graph construction (b) and Type-II graph construction (c) for minimizing its equivalent quadratic function. In graph (c), m denotes the number of variables included in the clique (in this case m = 3).*

$$g_i(z, x_i) = \begin{cases} w_{il}^q & \text{if} \quad z = q \text{ and } x_i = l \in \mathcal{L} \\ 0 & \text{if} \quad z = t + 1. \end{cases} \tag{11}$$

The role of the switching variable in the above mentioned transformation can be seen as that of finding which *deviation* function will assign the lowest cost to any particular labeling. The final higher order function generated using the parametrization (7) is a lower envelop of the *linear* deviation cost functions $\theta_q + d_q(\mathbf{x}_c)$. For instance, the function shown in figure 4(c) is a lower envelop of the higher order functions shown in figure 4(b). This transformation method can be seen as a generalization of the method proposed in [8] for transforming the Robust P^n model potentials.

4. Transforming Pseudo-Boolean Functions

In the previous section, we discussed how to transform multi-label higher order functions to quadratic ones by the addition of a multi-state auxiliary variable. The same method can also be applied to transforming higher order pseudo-boolean functions. However, the resulting quadratic function is not pseudo-boolean as it contains multi-state switching variables. In this section, we discuss two alternative transformation approaches for transforming higher order pseudo-boolean functions which works by adding boolean auxiliary variables. These methods will produce a quadratic pseudo-boolean function (QPBF) which can be minimized using algorithms for quadratic pseudo-boolean optimization (QPBO) [1, 5].

In what follows, we assume binary labels, i.e., $\mathcal{L} = \{0, 1\}$. Consider a higher order potential which assigns a cost $\theta \geq 0$ if the variables $\mathbf{x}_c$ take the labeling $\mathbf{X}_0 \in \{0, 1\}^{|c|}$, and $\theta_{\max} \geq \theta$ otherwise. More formally,

$$\psi_c(\mathbf{x}_c) = \begin{cases} \theta_0 & \text{if} \quad \mathbf{x}_c = \mathbf{X}_0 \\ \theta_{\max} & \text{otherwise,} \end{cases} \tag{12}$$

where $\mathbf{X}_0$ denotes the preferred labeling of the variables $\mathbf{x}_c$.

We will call this higher order potential a δ *basis function* since it assigns a low cost to only one single labelling. A constant θ_0 can be subtracted from this potential to yield:

$$\psi_c(\mathbf{x}_c) = \begin{cases} 0 & \text{if} \quad \mathbf{x}_c = \mathbf{X}_0 \\ \theta & \text{otherwise,} \end{cases} \tag{13}$$

where $\theta = \theta_{\max} - \theta_0 > 0$.

Type-I Transformation The minimization of higher order potential function (13) can be transformed to the minimization of a quadratic function using two additional *switching* variables $z_0, z_1 \in \{0, 1\}$ as: $\min_{\mathbf{x}_c} \psi_c(\mathbf{x}_c) =$

$$\min_{\mathbf{x}_c; z_0, z_1 \in \{0,1\}} \theta z_0 + \theta(1 - z_1) - \theta z_0(1 - z_1) + \theta \sum_{i \in S_0(\mathbf{X}_0)} (1 - z_0) x_i + \theta \sum_{i \in S_1(\mathbf{X}_0)} z_1(1 - x_i). \tag{14}$$

Here, $S_0(\mathbf{X}_0)$ is the set of indices of random variables which were assigned the labels 0 in the assignment $\mathbf{X}_0$. Similarly, $S_1(\mathbf{X}_0)$ represents the set of variables which were assigned the label 1 in $\mathbf{X}_0$. The minimization problem (14) involves a quadratic function with at most one non-submodular term (i.e., $-\theta z_0(1 - z_1)$) [1]. It can be easily verified that the transformed QPBF in (14) is equivalent to (13). The transformation is illustrated for a particular higher order function in Figure 2(b).

Type-II Transformation We now describe an alternative method to transform higher order pseudo-boolean functions which requires the addition of only *one* auxiliary variable, but results in a slightly complex transformation.

Theorem 1. *The minimization of the higher order pseudo-boolean function (13) is equivalent to the following* QPBF *minimization problem:* $\psi_c(\mathbf{x}_c) = \theta + \frac{\theta}{2} \min_{z \in \{0,1\}} f(z, \mathbf{x}_c)$, *where the* QPBF *function f is defined as:*

$$\begin{aligned} f(z, \mathbf{x}_c) &= z(m - 2) \\ &+ z\Big(\sum_{i \in S_1(\mathbf{X}_0)} (1 - x_i) - \sum_{i \in S_0(\mathbf{X}_0)} (1 - x_i)\Big) \\ &+ (1 - z)\Big(\sum_{i \in S_1(\mathbf{X}_0)} x_i - \sum_{i \in S_0(\mathbf{X}_0)} x_i\Big) \\ &+ \sum_{i \in S_0(\mathbf{X}_0)} x_i - \sum_{i \in S_1(\mathbf{X}_0)} x_i \end{aligned} \tag{15}$$

where $m = |c|$ is the order of the clique[2]*. The transformation is illustrated for a particular higher order function in Figure 2(c). Proof in [17].*

[1] This is independent of the clique size $|c|$ or the enforced labeling $\mathbf{X}_0$.

[2] Note that the coefficient of each monomial in (15) exactly corresponds to an edge capacity in Type-II graph construction.

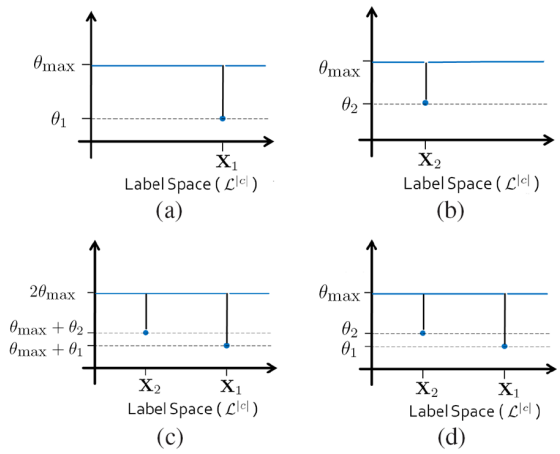

Figure 3. *Composing higher order pseudo-boolean functions by adding basis functions of the form 13.(a) and (b) are two δ basis functions. (c) The potential obtained by summing the basis functions shown in (a) and (b). (d) The potential function obtained after subtracting a constant $\theta_{\max}$ from (c) (this doesn't change the labeling with the minimal energy).*

4.1. Composing General Pseudo-boolean Functions

Multiple instances of the δ *basis* higher order pseudo-boolean potentials (13) can be used to compose general higher order energy functions. The composition method works by summing these pseudo-boolean potential. The equivalent QPBF of the target higher order pseudo-boolean function is obtained by summing the QPBFs corresponding to the individual basis pseudo-boolean potentials of the form (12) (see Figure 3 for illustration). The composition scheme requires the selection of $\theta_{\max}$ which is the highest possible energy that can be assigned to any labeling. A different method for transforming higher order pseudo-boolean functions into QPBFs has been proposed in [5].

The reader should observe that this way of obtaining equivalent quadratic functions for general higher order functions is fundamentally different from the strategy employed in Section 3.1. There, we use a multi-state switching variable to select among different constituent higher order functions; in contrast, here we sum the constituent higher order functions.

4.2. Compact Representation for Higher-order Pseudo-boolean Potentials

The composition method described above would in the worst case require the addition of $2^{|c|+1}$ auxiliary variables for Type-I transformation and $2^{|c|}$ auxiliary variables for Type-II transformation [3]. To reduce the number of auxiliary variables, we use a scheme similar to the one discussed in Section 3.2 to model the cost of multiple labelings using only two auxiliary variables for Type-I transformation and one auxiliary variable for Type-II transformation.

We define the *deviation basis* higher order potential ψ_c^f which assigns the minimum of a deviation cost function $f(\mathbf{x}_c)$ and a constant threshold cost θ. Formally,

$$\psi_c^f(\mathbf{x}_c) = \min\{f(\mathbf{x}_c), \theta\}. \tag{16}$$

where the deviation function $f : \{0,1\}^{|c|} \rightarrow \mathbb{R}$ specifies the penalty for deviating from the favored labeling $\mathbf{X}_0$, and is written as:

$$f(\mathbf{x}_c) = \theta \sum_{i \in c} abs(w_i)(x_i \neq \mathbf{X}_0(i)) \tag{17}$$

where the absolute value of the weights w_i control the cost of different labelings deviating from $\mathbf{X}_0$. The function f can also be seen as assigning a cost equal to a weighted hamming distance of a labelling from $\mathbf{X}_0$.

The function f can alternatively be defined as:

$$f(\mathbf{x}_c) = \theta \sum_{i \in c} w_i x_i + \theta K \tag{18}$$

where the constant $K = \sum_{i|w_i<0} w_i$, and the weight w_i specifies what is the cost of assigning the label 1 to variable x_i. Naturally, the weights would be negative for pixels which have been assigned the label 1 in the favored labeling $\mathbf{X}_0$. Similarly, variables labeled 0 will be assigned a positive weight. On substituting the value of K, f becomes:

$$f(\mathbf{x}_c) = \theta \sum_{i \in c; w_i > 0} w_i x_i + \theta \sum_{i \in c; w_i < 0} (-w_i)(1 - x_i) \tag{19}$$

We now show how the higher order pseudo-boolean function defined in equation (16) can be transformed to a QPBF.

Theorem 2. *Using Type-I transformation, the minimization of higher order pseudo-boolean potential function (16) can be transformed to the following* QPBF *minimization problem:* $\min_{\mathbf{x}_c} \psi_f(\mathbf{x}_c) =$

$$\min_{\mathbf{x}_c; z_0, z_1 \in \{0,1\}} \theta z_0 + \theta(1 - z_1) - \theta z_0(1 - z_1) + \theta \sum_{i|w_i \geq 0} w_i(1 - z_0)x_i + \theta \sum_{i|w_i<0} (-w_i) z_1(1 - x_i) \tag{20}$$

Theorem 3. *Using Type-II transformation, the minimization of higher order function (16) can be written as the result of the following* QPBF *minimization problem:* $\psi_c(\mathbf{x}_c) = \theta + \frac{\theta}{2} \min_{z \in \{0,1\}} F(z, \mathbf{x}_c))$*, where the*

[3] There are $2^{|c|}$ possible labelings of the variables in the clique c.

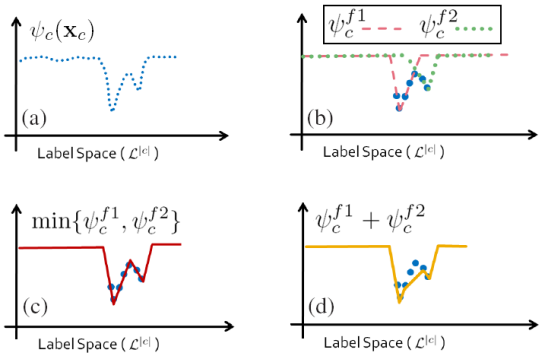

Figure 4. *Difference between the min and sum composition schemes.(a) Original function to be represented. (b) Two deviation basis functions (16). (c) The composition of the functions shown in (b) by taking their lower envelope (by minimizing over a multi-state variable as explained in section 3.2). (d) The composition of the functions shown in (b) by summing them (causes misrepresentations in regions where the basis functions overlap).*

QPBF *function F is defined as:*

$$\begin{aligned} F(z, \mathbf{x}_c) &= z(\sum_{i \in c} abs(w_i) - 2) \\ &+ z(\sum_{i|w_i<0} (-w_i)(1-x_i) - \sum_{i|w_i \geq 0} w_i(1-x_i)) \\ &+ (1-z)(\sum_{i|w_i<0} (-w_i)x_i - \sum_{i|w_i \geq 0} w_i x_i) \\ &+ \sum_{i|w_i \geq 0} w_i x_i - \sum_{i|w_i<0} (-w_i)x_i \end{aligned} \quad (21)$$

where $abs(w_i)$ is the absolute value of the weight w_i. Proof in [17].

Behavior of the Summation Composition Scheme The method of composing higher order potentials using the summation scheme described in the previous sub-sections suffers from a problem when using compact representations of higher order potentials. This occurs when there is significant overlap in the subset of labelings which are assigned a non-threshold cost θ by multiple higher order potentials. This problem is illustrated in figure 4.

5. Experiments

We evaluated our theoretical contributions on the problem of binary texture restoration, which is a popular test-bed for energy minimization methods [3, 11, 18]. We first introduce our formulation of the texture restoration problem. We will then proceed to compare the performance of the three transformation schemes proposed in sections 3 and 4. Our experiments show that the multi-label construction (sec. 3) is empirically superior for this problem. Thus, in the rest of the experiments, we will use it to demonstrate the power of

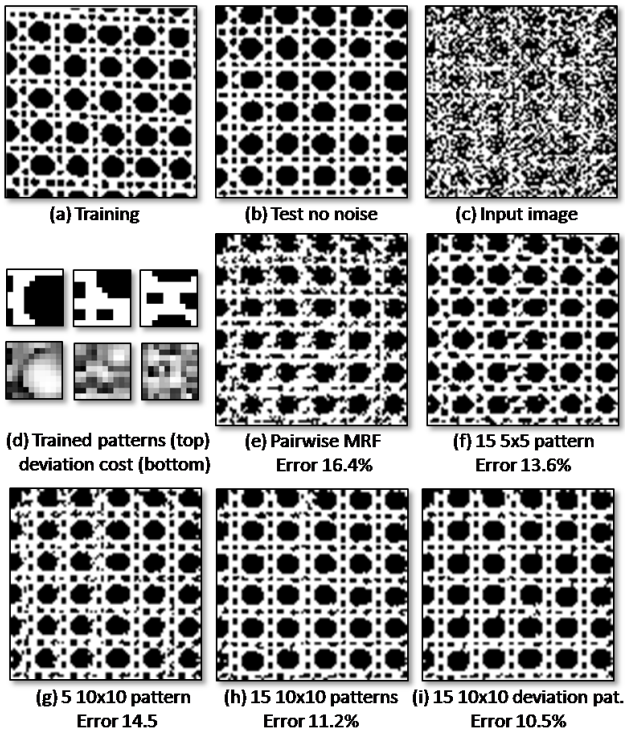

Figure 5. *Binary Texture restoration for Brodatz texture D101. (a) Training image (86×86 pixels). (b) Test image. (c) Test image with 60% noise used as input. (d) Three (out of 10) patterns of size 10×10pixels (top row) with their deviation cost (bottom row). (e-i) Results of various different models.*

the higher-order texture model. In this work we only compare to results of the non-submodular pairwise model, as reported in [11]. A direct comparison to other higher-order models, such as the submodular triple-clique model of [3] or the FRAME model [25] would be interesting future work.

Training and Test procedure. In this work we have exactly followed the procedure outlined in [11][4]. Our model is based on that of [3] but with additional higher-order terms. We constrain our higher-order cliques to be patches of a fixed size. In brief, given a training image (e.g. fig. 5(a)), we first compute the joint histogram of all pixel pairs with the same shift (s_x, s_y), where we constrained the maximum shift length, i.e. $\max\{|s_x|, |s_y|\} \leq 30$. The pairwise potentials are then defined as $\theta_{i,j}(x_i, x_j) = -\log Pr(x_i, x_j)$, with x_i, x_j being the output labeling at the two pixels i, j. The unary potential is given as $\sum_{i \in \mathcal{V}} -\lambda/(1 + |T_i - x_i|)$, where T_i is the value of the pixel i in the noisy test image (e.g. fig. 5(c)). To obtain good result, it is important to select those pairwise potentials which best describe the texture, and to learn the optimal value for λ, i.e. the trade-off between unary and pairwise terms. As in [11], we used the discriminative learning procedure on a validation dataset, which resulted in e.g. 9 pairwise terms (7 sub- and 2 non-

[4] We also used their data, which is available online.

	Binary Type-I			Binary Type-II			Multi-label	
	BP	TRW	QPBO(P)	BP	TRW	QPBO(P)	BP	TRW
Toy Text.	86.1; 38.5%	100(-179); 60.8%	100%(100%)	86.0; 38.5%	100(-179); 60.8%	100%(100%)	**0; 0%**	**0(0), 0%**
Real Text.	91.8; 19.9%	100(-624); 31.7%	100%(100%)	91.8; 19.9%	100(-624); 31.7%	100%(100%)	**91.2; 19.6%**	99.9(0); 30.9%

Table 1. *Comparing different transformation schemes for two different textures and various optimization methods. We report for BP: (Energy; error), for TRW Energy(Lower bound); error, and for QPBO(QPBOP) the number of unlabeled nodes. Note, the energy values are shifted and scaled so that highest energy is* 100 *and the best lower bound is* 0*. Best performing methods are shown in bold.*

submodular) for the texture in fig. 5. For a test image (fig. 5(b)) with 60% noise (fig. 5(c)) such a pairwise model, optimized with QPBO [11], gives a reasonable result shown in fig. 5(e) with 16.4% error (misclassified pixels).

For higher-order cliques of a fixed patch size we aim to train a few patterns, which are enforced for *each* patch in the image. These patterns should occur frequently in the training set and are as different as possible in terms of their hamming distance. We achieve this by k-means clustering over all training patches. Fig. 5(d) depicts three (out of k=10) such patterns (top row). To compute the deviation functions of a particular pattern $\mathbf{X}_p$ we consider all patterns which belong to the same cluster as $\mathbf{X}_p$. For each position within the patch, we record the frequency of having the same value as $\mathbf{X}_p$. Fig. 5(d - bottom row) shows the associate deviation costs, where a bright value means high frequency (i.e. high cost). As expected, lower costs are at the edge of the pattern. As in the case of the pairwise model, the weights of our higher-order deviation potentials are computed using the frequency of co-occurring patterns, and the global weight of the higher-order potentials is learned discriminatively on the validation set.

How sparse is a higher-order function of a binary texture? To answer this question, we recorded the number of unique patterns which occur in the training image of the binary Brodatz texture D101 (i.e. similar to fig. 5(a) but of size 256×85). We observed that the percentage of unique patterns present in the training image reduces considerably for larger patch sizes. For instance, roughly 38% (194 out of $2^9 = 512$) of possible 3×3 pixel patterns were present in the training image. This number goes down to just 0.014% (4764 out of 2^{25}) for 5×5 pixel patterns. If we cluster these patterns under the constraint that all patterns are at most 4 hamming distance away from a cluster center, we find that roughly only 0.0005% (187 out of 2^{25}) of 5×5 pixel patterns as cluster means would be enough. Obviously, in practice even with a deviation cost functions, modelling all 187 patterns leads to a computationally expensive optimization problem. However, our results show that even modelling only the 25 most frequently occurring patterns already produces impressive results.

To demonstrate the weakness of the pairwise model, we measured the frequency of training patterns occurring in the reconstruction result obtained using the pairwise model (i.e. similar to fig. 5(e) but of size 256×86). As expected these patterns do not occur frequently: Only 10.9% of training patterns of size 5×5 pixel were found in the pairwise result; and no training patterns of size 10×10 are present in the pairwise result. This is the result of the pairwise model not being able to capture the large-scale characteristics of the texture. We will now see that a higher-order model is able to fix this.

Comparing our Transformation Schemes. We compared the performance of our three transformations: Multi-label (sec. 3), binary Type-I and Type-II (sec. 4) on various real and toy textures. For optimization, we used the top performing publicly available methods for minimizing non-submodular binary and multi-label energies: TRW, BP from [20], and the roof-dual relaxation based approaches (QPBO, QPBOP) from [1, 18].

Table 1 shows results on two example textures (see [17]) which capture the main conclusions we have drawn from this experiment. The first toy texture (top row) is a perfectly repeated texture (30×30pixels) which has only four distinctively different patterns of size 3×3. We encoded these patterns as higher-order cliques, and did not use any pairwise terms. The second texture (bottom row) is a 30×30 crop of the D103 texture (see [17]) where we trained 25 5×5 patterns without deviation cost function.

The first conclusion is that the two binary constructions (Type-I and -II) perform very similarly. Secondly, the binary constructions are considerably inferior to the multi-label construction, for both toy and real texture. For example, QPBO and QPBOP could not label any node for the simple toy texture. (Note, in these binary cases, it is not surprising that TRW has also a bad lower bound since it is solving the same LP relaxation as QPBO.) In contrast to this, for the toy texture the multi-label construction finds the global minimum with TRW. Unfortunately, the real texture is more challenging and non of the methods find a global minimizer. a. The third conclusion is that BP is clearly the best performing method and hence we used it for all later experiments. We believe that this is because our graphs are very densely connected. A similar observation can be found in empirical studies of e.g. [10, 18].

Given the above results, one might question the usefulness of the binary constructions. Two arguments in support of the binary constructions are **(a)** we found a few instances where the binary construction slightly outperformed the multi-label construction using BP, and **(b)** a lot of active research in the optimization community is on pseudo-boolean optimization which hopefully will yield superior

methods in the future. Finally, we see that low energies correlate with low reconstruction error which confirms the quality of our model. In terms of runtime of BP, we see that multi-label and binary type-II construction have about the same runtime, e.g. 2.1sec for the real texture, while binary type-I has slightly higher runtime (2.5sec) since more auxiliary nodes are used.

Comparing higher-order texture models Given the superiority of the multi-label construction and its better modeling power for compact representations (fig. 4), we use it for the remaining experiments. BP is used for inference.

We tested our method on four different Brodatz textures D101, D103, D22, and D20, with a noise level of 60%. The results on D101 are shown in fig.5 (other results can be seen in [17]). The key conclusion is that higher-order cliques always manage to capture visually the main characterizes of a texture on a large-scale, see e.g. fig. 5(i). For most textures this is reflected in an error rate improvement, for D101 even by a big margin (improvement of 36% - fig. 5(e,i)). Furthermore, as expected modelling more patterns helps (compare fig. 5(g,h)). Also, large cliques give typically better results (compare fig. 5(f,h)). Finally, the deviation functions help to improve results further, especially visually (see fig. 5(h,i)). In terms of runtime, using more and large higher-order cliques is obviously more computationally expensive. The example in fig. 5(i), which is an image of size 86×86, with 88935 patterns, each of size 100, takes 32sec (14 rounds of BP) on a 1.8Ghz CPU. Note that deviation functions do not increase the runtime.

6. Conclusion and Future Work

This paper provides a method for minimizing sparse energy functions. We studied the behavior of our methods in dealing with different energy functions, and showed how it could be used to obtain impressive results on binary texture restoration problem. We believe that our sparse higher-order models will find wide applicability to many research problems within computer vision.

The transformation methods presented in the paper result in difficult optimization problems which are NP-hard to solve. The use of sophisticated optimization algorithms such the ones proposed recently in [12, 13, 19, 23] in solving these problems is a interesting direction for future work.

References

[1] E. Boros and P. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 2002.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001.

[3] D. Cremers and L. Grady. Learning statistical priors for efficient combinatorial optimization via graph cuts. In *ECCV*, 2006.

[4] P. Felzenszwalb and D. Huttenlocher. Efficient Belief Propagation for Early Vision. In *CVPR*, 2004.

[5] H. Ishikawa. Higher-order clique reduction in binary graph cut. In *CVPR*, 2009.

[6] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM*, 48(4):761–777, 2001.

[7] P. Kohli, M. Kumar, and P. Torr. P^3 and beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[8] P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[9] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 2006.

[10] V. Kolmogorov and C. Rother. Comparison of energy minimization algorithms for highly connected graphs. In *ECCV (2)*, pages 1–15, 2006.

[11] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts – a review. *PAMI*, 2007.

[12] N. Komodakis and N. Paragios. Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles. In *ECCV*, 2008.

[13] M. Kumar and P. Torr. Efficiently solving convex relaxations for MAP estimation. In *ICML*, 2008.

[14] X. Lan, S. Roth, D. Huttenlocher, and M. Black. Efficient belief propagation with learned higher-order markov random fields. In *ECCV (2)*, pages 269–282, 2006.

[15] J. Pearl. Fusion, propagation, and structuring in belief networks. *Artif. Intell.*, 29(3):241–288, 1986.

[16] S. Roth and M. Black. Fields of experts: A framework for learning image priors. In *CVPR*, pages 860–867, 2005.

[17] C. Rother and P. Kohli. Transforming sparse higher order functions. In *Microsoft Research Technical Report*, 2008.

[18] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR*, 2007.

[19] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. . Weiss. Tightening LP relaxations for MAP using message passing. In *UAI*, 2008.

[20] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *ECCV (2)*, pages 16–29, 2006.

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

[22] Y. Weiss and W. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *Transactions on Information Theory*, 2001.

[23] T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *CVPR*, 2008.

[24] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008.

[25] S.-C. Zhu, Y. Wu, and D. Mumford. FRAME: Filters, Random fields And Maximum Entropy— towards a unified theory for texture modeling. *IJCV*, 27, 1998.