HAL
open science

Exact inference in multi-label CRFs with higher order cliques

Srikumar Ramalingam, Pushmeet Kohli, Karteek Alahari, Philip H. S. Torr

To cite this version:

Srikumar Ramalingam, Pushmeet Kohli, Karteek Alahari, Philip H. S. Torr. Exact inference in multi-label CRFs with higher order cliques. CVPR - IEEE Conference on Computer Vision & Pattern Recognition, Jun 2008, Anchorage, United States. 10.1109/CVPR.2008.4587401 . hal-01217304

HAL Id: hal-01217304

https://inria.hal.science/hal-01217304v1

Submitted on 19 Oct 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exact Inference in Multi-label CRFs with Higher Order Cliques

Srikumar Ramalingam [1] Pushmeet Kohli [2] Karteek Alahari [1] Philip H. S. Torr [1]

[1] Oxford Brookes University* [2] Microsoft Research, Cambridge

Abstract

This paper addresses the problem of exactly inferring the maximum a posteriori solutions of discrete multi-label MRF*s or* CRF*s with higher order cliques. We present a framework to transform special classes of multi-label higher order functions to submodular second order boolean functions (referred to as $\mathcal{F}_s^2$), which can be minimized exactly using graph cuts and we characterize those classes. The basic idea is to use two or more boolean variables to encode the states of a single multi-label variable. There are many ways in which this can be done and much interesting research lies in finding ways which are optimal or minimal in some sense. We study the space of possible encodings and find the ones that can transform the most general class of functions to $\mathcal{F}_s^2$. Our main contributions are two-fold. First, we extend the subclass of submodular energy functions that can be minimized exactly using graph cuts. Second, we show how higher order potentials can be used to improve single view 3D reconstruction results. We believe that our work on exact minimization of higher order energy functions will lead to similar improvements in solutions of other labelling problems.*

1. Introduction

The last few years have seen discrete energy minimization emerge as an indispensable tool for computer vision. It enables inference of the maximum a posteriori solutions of Markov and conditional random fields (MRFs and CRFs), which can be used to model labelling problems such as image segmentation, stereo, image restoration and many others [3, 22, 23]. The problem of minimizing a general function is NP-hard. However, due to its immense applicability, a number of approximate energy minimization algorithms have been proposed which produce a locally optimal solution with certain guarantees in polynomial time. A few popular examples of these algorithms are graph cuts [3, 13], tree-reweighted message passing [27] and variants of belief propagation (BP) [17, 29].

It is well known that a special class of functions called *submodular* functions can be minimized globally in polynomial time. These functions are discrete analogues of continuous convex functions. The current best algorithm for general submodular function minimization has complexity O($n^5Q + n^6$), where n is the number of random variables and Q is the time taken to evaluate the function [16]. This makes their use infeasible for problems in computer vision which, in general, involve a large number of variables. However, certain subclasses of submodular functions can be minimized much more efficiently. For example, boolean submodular functions of order[1] at most three can be minimized by solving an st-mincut problem, for which efficient algorithms are known [1, 8, 13]. In CVPR 2005, Freedman and Drineas [6] extended this work and proved that a subclass of submodular boolean functions of order four or more can be minimized. It was also shown that multi-label CRFs with convex energy functions of order two can be minimized in polynomial time [10, 25]. However, it has not been known what the analogue of this is for higher order cliques. We aim to study this in the paper.

Most labelling problems in computer vision involve multi-label MRFs or CRFs [3, 23]. Further, the use of higher order clique structures has proved beneficial [12, 14, 20] for solving certain computer vision problems. However, efficient st-mincut based algorithms used for minimizing submodular second order boolean functions are not directly applicable to these functions. Our work overcomes this restriction by showing how we can transform some submodular multi-label higher order functions to submodular boolean functions, thus enabling their exact minimization. Before proceeding further, we briefly introduce our notation for denoting different classes of energy functions. Let $\mathcal{F}_s^k$ and $\bar{\mathcal{F}}_s^k$ denote the class of submodular and non-submodular boolean energy functions of order k respectively. Similarly, let $\mathcal{M}_s^k$ and $\bar{\mathcal{M}}_s^k$ denote the class of submodular and non-submodular multi-label energy functions of order k respectively.

A generic transformation framework: The basic idea of our framework is to use two or more boolean variables to encode the states of a single multi-label variable. While doing this, we have to ensure that the minimum cost labelling of the boolean problem also encodes the minimum cost labelling of the multi-label energy function. There are several possible ways to encode a multi-label variable using boolean variables. In the rest of the paper, we use the term *encoding* to refer to the mapping between the labellings of the multi-label variable and the corresponding binary variables. The term *transformation* will refer to the conversion of multi-label energy functions to functions of binary variables.

Different transformations are important because the choice of transformation dictates the size of the resulting boolean function and the class of multi-label functions that

*http://cms.brookes.ac.uk/research/visiongroup

[1] Clique size in a CRF corresponds to order of the energy function.

can be transformed to $\mathcal{F}_s^2$. For example, Ishikawa [10] described a transformation that used l boolean variables to encode a single l-label variable. Using this transformation pairwise convex functions of the difference of labels, which is a subclass of $\mathcal{M}_s^2$, can be transformed to $\mathcal{F}_s^2$. Later, Schlesinger and Flach [25] gave a concise definition of submodularity for (ordered) multi-label functions and used $l-1$ encoding boolean variables to transform any function in $\mathcal{M}_s^2$ to $\mathcal{F}_s^2$. In this paper, we study the space of all possible transformations and find the subclasses of multi-label functions that they can transform to $\mathcal{F}_s^2$. In other words, the transformations we develop will lead to submodular boolean functions under some constraints. These constraints will serve to characterize the class of $\mathcal{M}_k$ that can be minimized exactly.

The main novelties of this paper are as follows:

- A principled framework for transforming $\mathcal{M}_s^k$ functions to $\mathcal{F}_s^2$.
- The identification of constraints that enable the transformation of $\mathcal{M}_s^k$ of any order into $\mathcal{F}_s^2$ in polynomial time.
- The result that there exists no polynomial transformation from submodular multi-label functions of order four or more ($\mathcal{M}_s^{k\geq 4}$) to submodular boolean second order functions ($\mathcal{F}_s^2$).
- The use of higher order functions to improve the performance of single view 3D reconstruction algorithms [9].

Overview of the paper: §2 describes the basic theory of pseudo-boolean optimization and its relation to minimizing multi-label higher order functions using st-mincut algorithms. The problem statement is given in §3. §4 shows how to encode multi-label variables using boolean (or binary) ones. A characterization of multi-label higher order functions that can be transformed to $\mathcal{F}_s^2$ in polynomial time is given in §5. We show improvement in single view 3D reconstruction using higher order priors in §6. Potential applications of our work and directions for future research are discussed in §7.

2. Notation and Preliminaries

Let $\mathbb{B}$ denote the boolean set $\{0,1\}$ and $\mathbb{R}$ the set of reals. Let the vector $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{B}^n$, and $\mathbf{V} = \{1, 2, ..., n\}$, be the set of all boolean variables and their indices respectively. A pseudo-boolean function $f : \mathbb{B}^n \rightarrow \mathbb{R}$ is a function which takes a boolean vector as an argument and returns a real number. These functions can be uniquely represented using a multi-linear polynomial form [2]. An example is $f(x_1, x_2, x_3, x_4) = 2 - 4x_2x_4 + 7x_1x_2x_3$. Another useful representation known as *posiform* involves the *complements* $(\bar{x}_1, ..., \bar{x}_n)$ of variables. Such a representation for the above example is:

$$\phi(x_1, x_2, x_3, x_4) = -2 + 4\bar{x}_4 + 4\bar{x}_2x_4 + 7x_1x_2x_3. \quad (1)$$

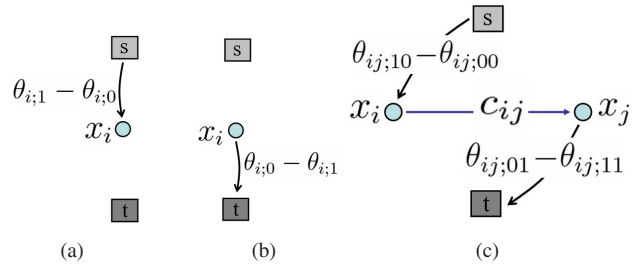

Figure 1. *Converting an energy minimization problem to an st-mincut problem [13]. (a) and (b) show unary potentials are represented using edges in the graph, while (c) shows the same for submodular pairwise potentials.*

An important property of the posiform representation is that all the coefficients, except the constant, are non-negative [2].

2.1. Graph Cuts for Energy Minimization

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph with non-negative edge weights and two special nodes, namely, the source s and the sink t, which represent the labels $\{0, 1\}$. An st-cut partitions the set of vertices in $\mathcal{V}$ into two disjoint sets S and T, such that $s \in S$ and $t \in T$. The cost of this cut is the sum of edge weights (i, j), where $i \in S$ and $j \in T$. The st-mincut problem involves finding the st-cut with the minimum cost. Any $\mathcal{F}_s^2$ function can be minimized exactly by computing the st-mincut in an equivalent graph [13]. The key idea is to design a graph such that cuts in the graph correspond to labellings of the binary variables, with the cost of the cut equal to the cost of the labelling (plus a constant). We call this the *equivalent graph*.We now show how to construct this graph. Consider a second order boolean energy

$$E_b(\mathbf{x}) = \sum_{i\in\mathcal{V}} E_b(x_i) + \sum_{(i,j)\in\mathcal{E}} E_b(x_i, x_j), \quad (2)$$

where $E_b(x_i)$ and $E_b(x_i, x_j)$ represent the first and second order terms of the energy function respectively. Let $\theta_{i;a}$ be the cost of assignment $x_i = a$, and $\theta_{ij;ab}$ be the cost of the assignment $x_i = a, x_j = b$ $(a, b \in \mathbb{B})$.

The graph constructed for minimizing a $\mathcal{F}_s^2$ function has a vertex i for each boolean random variable $x_i \in \mathbb{B}$. There is a mapping between st-cuts in the graph and label assignments. Node $i \in S$ implies $x_i = 0$, while $i \in T$ implies $x_i = 1$. We now show how to create the equivalent graphs for functions belonging to the classes $\mathcal{F}_s^1$ and $\mathcal{F}_s^2$.

The class $\mathcal{F}_s^1$: The unary term $E_b(x_i)$ of the energy can be written as: $E_b(x_i) = \theta_{i;0}\bar{x}_i + \theta_{i;1}x_i$. If $\theta_{i;1} - \theta_{i;0} \geq 0$, we write the energy as: $E_b(x_i) = (\theta_{i;1} - \theta_{i;0})x_i + \theta_{i;0}$. The minimization of this energy is equivalent to finding the st-mincut in the graph shown in figure 1(a). Cutting the edge (s, i) is equivalent to the assignment $x_i = 1$. Similarly, if $\theta_{i;1} - \theta_{i;0} < 0$, we write the energy as $E_b(x_i) = (\theta_{i;0} - \theta_{i;1})\bar{x}_i + \theta_{i;1}$, and the corresponding graph is given in figure 1(b).

The class $\mathcal{F}_s^2$: The pairwise energy $E_b(x_i, x_j) = \theta_{ij;00}\bar{x}_i\bar{x}_j + \theta_{ij;01}\bar{x}_i x_j + \theta_{ij;10}x_i\bar{x}_j + \theta_{ij;11}x_i x_j$ can be written as: $E_b(x_i, x_j) = c_{ij}\bar{x}_i x_j + (\theta_{ij;10} - \theta_{ij;00})x_i + (\theta_{ij;10} - \theta_{ij;11})\bar{x}_j + \theta_{ij;00} + \theta_{ij;11} - \theta_{ij;10}$, where $c_{ij} = (\theta_{ij;01} + \theta_{ij;10} - \theta_{ij;00} - \theta_{ij;11})$. The equivalent graph construction is given in figure 1(c). Since our overall goal is to transform multi-label functions to $\mathcal{F}_s^2$, we do not focus on $\mathcal{F}_s^3$ and higher order functions [6, 13].

Multi-label functions: Let $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)$ be a directed graph with a set of vertices $\mathcal{V}_m = \{1, 2, ..., m\}$ and edges $\mathcal{E}_m$. Let y_i be a variable taking values in some discrete space $\mathcal{L} = \{1, 2, ..., l\}$, and let $\mathbf{y} = \{y_1, ..., y_m\}$. We use $\mathbf{\Theta}$ to denote the set of higher order potentials whose sum defines the energy function. The unary potential is denoted by $\Theta_{i;a}$, pairwise by $\Theta_{ij;ab}$, where $i, j \in \mathcal{V}_m$ and $a, b \in \mathcal{L}$. Let $\mathbf{i} = i_1 i_2 ... i_k \in \mathcal{V}_m \times \mathcal{V}_m ... \mathcal{V}_m$ (k times)=$\mathcal{V}_m^k$ and $\mathbf{a} = a_1 a_2 ... a_k \in \mathcal{L} \times \mathcal{L} ... \mathcal{L}$ (k times)= $\mathcal{L}^k$. Under this notation, a kth order energy function is written as:

$$E_m(\mathbf{y}) = \sum_{\mathbf{i} \in \mathcal{V}_m^k, \mathbf{a} \in \mathcal{L}^k} \Theta_{\mathbf{i};\mathbf{a}} \prod_{i \in \mathbf{i}, a \in \mathbf{a}} \delta(y_i, a), \tag{3}$$

where

$$\delta(y_i, a) = \begin{cases} 1 & \text{if } y_i = a, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

2.2. Submodular functions

Submodular functions are set functions $f : 2^n \rightarrow \mathbb{R}$ satisfying the following condition:

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y), \tag{5}$$

where X and Y are subsets of the set $\mathbf{V}$, and $\cup$ and $\cap$ denote union and intersection of sets respectively. We briefly describe how the above definition of submodularity maps to functions of boolean variables [13]. A function of one boolean variable is always submodular. A function $\theta : \mathbb{B}^2 \rightarrow \mathbb{R}$ of two boolean variables $\{x_i, x_j\}$ is submodular if and only if:

$$\theta_{ij:00} + \theta_{ij:11} \leq \theta_{ij:01} + \theta_{ij:10}. \tag{6}$$

A function $\theta : \mathbb{B}^n \rightarrow R$ is submodular if and only if all its projections on 2 variables are submodular [2, 13]. The submodularity conditions can be extended to multi-label variables. Let $\mathcal{L}$ be a completely ordered set, where between every pair of states l_1 and l_2, an ordering (above/below) is present. A function $\Theta : \mathcal{L}^2 \rightarrow R$ is submodular if

$$\Theta_{ij:l_1 l_2} + \Theta_{ij:(l_1+1)(l_2+1)} \leq \Theta_{ij:(l_1+1)l_2} + \Theta_{ij:l_1(l_2+1)}, \tag{7}$$

for all l_1, l_2 [25]. Using the work of Schlesinger [24] on permuted submodular functions we can find an ordering (if it exists) for which the functions become submodular. Thus, we can work with a notion of submodularity of multi-label functions which is independent of the ordering of the labels. A function $\Theta : \mathcal{L}^m \rightarrow R$ is submodular if and only if all its projections on 2 variables are submodular [5].

3. Problem Statement

The main goal of this paper is to obtain a boolean second order function $E_b(\mathbf{x})$ equivalent to a given multi-label higher order function $E_m(\mathbf{y})$ in polynomial time. The boolean function also needs to satisfy the following conditions:

- There is an encoding $\mathcal{T} : \mathcal{L}^{|\mathcal{V}_m|} \rightarrow \mathbb{B}^{|\mathcal{V}|}$ which is 1-1 between the feasible labellings of $\mathbf{x}$ and $\mathbf{y}$, and bijective between the set of optimal labellings of the boolean and multi-label variables.
- The minimum value of $E_m(\mathbf{y})$ over $\mathbf{y}$ is equal to the minimum value of $E_b(\mathbf{x})$ over $\mathbf{x}$:
$$\min_{\mathbf{x}} E_b(\mathbf{x}) = \min_{\mathbf{y}} E_m(\mathbf{y}). \tag{8}$$
The energy functions need not be equal at labellings that are not their respective minima.

We also want to answer the following questions:

1. What is the class of multi-label higher order functions for which we will always be able to find an equivalent $\mathcal{F}_s^2$ function? We characterize the class by finding the constraints on the potentials $\mathbf{\Theta}$ of the function.
2. How can the boolean function with the smallest number of variables be obtained?

Before we present our algorithm in detail we mention the three important steps in our algorithm.

1. A second order pseudo-boolean function is constructed which enforces 1-1 mapping between the feasible labellings of $\mathbf{y}$ and $\mathbf{x}$ (See §4).
2. Encoding functions that can replace all occurrences of $\mathbf{y}$ in $E_m(\mathbf{y})$ using $\mathbf{x}$ are computed (See §5).
3. We transform the problem of minimizing the multi-label energy function into that of minimizing a $\mathcal{F}_s^2$ function (See §5).

For simplicity, we demonstrate our method on a specific 4-label energy function. The algorithm is presented as an interplay between graph constructions and transformation of energy functions. As studied in §2.1 both operations are closely related. In what follows we will explain the different steps of our algorithm in detail.

4. Boolean encoding for multi-label variables

In this section we propose a method to construct a second order pseudo-boolean function such that the labellings of the boolean variables have a 1-1 mapping with the labellings of the original multi-label variables. For example, in figure 2(a) we show a graph construction[2] to encode a 4-label variable y_1 using three boolean variables $\{x_1, x_2, x_3\}$. The encoding representing the change of variables is:

$$\{y_1 = 1\} \leftrightarrow \{x_1 = 1, x_2 = 1, x_3 = 1\}, \tag{9}$$

$$\{y_1 = 2\} \leftrightarrow \{x_1 = 0, x_2 = 1, x_3 = 1\}, \tag{10}$$

$$\{y_1 = 3\} \leftrightarrow \{x_1 = 0, x_2 = 0, x_3 = 1\}, \tag{11}$$

$$\{y_1 = 4\} \leftrightarrow \{x_1 = 0, x_2 = 0, x_3 = 0\}. \tag{12}$$

[2] This is commonly referred to as the *battleship* construction.

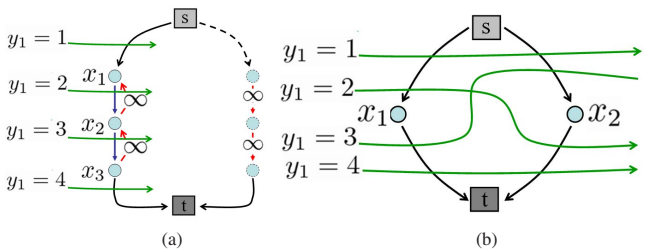

Figure 2. *(a) The battleship transformation [10, 25, 26]: The cuts in the graph are annotated by green arrows. Four possible cuts are shown and each cut corresponds to the assignment of one of the four labels to y_1. For example, if the edge (x_1, x_2) is cut then the labelling for $x_1x_2x_3$ is 011 and the corresponding labelling for y_1 is 2. Overall, the four labels of a multi-label variable $y_1 = \{1, 2, 3, 4\}$ are mapped to the labellings of three binary variables $x_1x_2x_3 = \{111, 011, 001, 000\}$. (b) The log transformation: The four labels of $y_1 = \{1, 2, 3, 4\}$ are mapped to the labellings (cuts) of two binary variables $x_1x_2 = \{11, 10, 01, 00\}$.*

Since three binary variables can take eight (2^3) different labellings, the remaining four labellings ($2^3 - 4$) are not mapped to any labellings of y_1. In order to ensure a bijective encoding between the binary variables and the multi-label variable, these labellings need to be made infeasible. This can be achieved by assigning a very high cost to the unused labellings. In the above encoding the unused labellings are given by $x_1x_2x_3 = \{010, 101, 100, 110\}$. Thus we can use the following penalty term:

$$P(\mathbf{x}) = \lambda(\bar{x}_1x_2\bar{x}_3 + x_1\bar{x}_2x_3 + x_1\bar{x}_2\bar{x}_3 + x_1x_2\bar{x}_3), \quad (13)$$

where $\lambda \rightarrow \infty$. This can also be seen as using the following third order penalty function:

$$\begin{pmatrix} \phi_{123;000} & \phi_{123;001} \\ \phi_{123;010} & \phi_{123;011} \\ \phi_{123;100} & \phi_{123;101} \\ \phi_{123;110} & \phi_{123;111} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ \infty & 0 \\ \infty & \infty \\ \infty & 0 \end{pmatrix}. \quad (14)$$

The above function is submodular and has four ($2^3 - 4$) penalty terms to restrict the infeasible labellings. The penalty function in equation (13) can be simplified to:

$$P(\mathbf{x}) = \lambda x_1\bar{x}_2 + \lambda x_2\bar{x}_3, \quad (15)$$

using simple boolean algebra. The pairwise terms of $P(\mathbf{x})$ correspond to the edges (x_2, x_1) and (x_3, x_2) with infinity costs[3].

A natural question to ask would be whether a different encoding is possible for a 4-label problem. To address this question we consider figure 2(b), where two boolean variables are used to encode a 4-label problem. We refer to this graph construction as the *log* transformation since it uses $\log(l)$ boolean nodes to encode an l-label variable. We present the generalization of this framework for solving any l label problem using k nodes, along with few examples for different transformations, in [18]. For simplicity, we chose a specific transformation to describe our algorithm. *So from this point onwards, we will propose algorithms which are very specific to the battleship transformation shown in figure 2(a).* This transformation can handle the most general class of energy functions. The proof is involved and can be found in [18].

5. Encoding Functions

Our overall goal is to transform a given multi-label higher order energy function into a boolean one. To do this, we need to define a boolean function which maps the labels of the multi-label variable to that of the encoding boolean variables. We refer to these functions as *encoding functions*. They enable us to replace multi-label variables in the energy function by boolean ones. More precisely, an *encoding* function is defined as $f_{y_1;a}(x_1, x_2, x_3) : \mathbb{B}^3 \rightarrow \mathbb{B}$ such that $f_{y_1;a}(x_1, x_2, x_3) = 1$ when $y_1 = a$ and 0 otherwise. We show a general method to compute these functions in [18]. The following example is shown to illustrate the main idea.

Let us assume that the function $f_{y_1;a}(x_1, x_2, x_3)$ is linear[4]. We assume the following representation for the linear function using four unknown parameters c_0, c_1, c_2, and c_3:

$$f_{y_1;a} = c_0 + c_1x_1 + c_2x_2 + c_3x_3. \quad (16)$$

Returning to our example, the possible solutions for the triplet $x_1x_2x_3$ are $(111, 011, 001, 000)$. When $y_1 = 1$, $x_1x_2x_3 = 111$. This can be written as $f_{y_1;1}(x_1 = 1, x_2 = 1, x_3 = 1) = 1$ and $f_{y_1;1}(x_1, x_2, x_3) = 0$ for other values of x_1, x_2 and x_3. Since there are only four possible solutions for the boolean variables $x_1x_2x_3$, we obtain the following conditions:

$$\begin{aligned} f_{y_1;1}(x_1 = 1, x_2 = 1, x_3 = 1) &= c_0 + c_1 + c_2 + c_3 = 1, \\ f_{y_1;1}(x_1 = 0, x_2 = 1, x_3 = 1) &= c_0 + c_2 + c_3 = 0, \\ f_{y_1;1}(x_1 = 0, x_2 = 0, x_3 = 1) &= c_0 + c_3 = 0, \\ f_{y_1;1}(x_1 = 0, x_2 = 0, x_3 = 0) &= c_0 = 0. \end{aligned}$$

On solving the above linear system, we get $f_{y_1;1} = x_1$. Using the same approach we solve for $f_{y_1;2}$, $f_{y_1;3}$ and $f_{y_1;4}$.

$$\begin{pmatrix} \delta(y_1, 1) \\ \delta(y_1, 2) \\ \delta(y_1, 3) \\ \delta(y_1, 4) \end{pmatrix} = \begin{pmatrix} f_{y_1;1} \\ f_{y_1;2} \\ f_{y_1;3} \\ f_{y_1;4} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 - x_1 \\ x_3 - x_2 \\ 1 - x_3 \end{pmatrix}. \quad (17)$$

With the encoding functions in place, we can finally address the energy transformation problem. The main idea is straightforward; as mentioned earlier, the encoding functions are used to replace all occurrences of the multi-label variable in the energy function by boolean variables. This

[3] In practice, we do not need an edge with infinite cost, but some edge having a cost greater than the sum of all edge costs.

[4] The function $f_{y_1;a}$ need not always be linear (See the log construction in [18]).

substitution produces a pseudo-boolean higher order function. We study this reduction and give a characterization of the class of multi-label higher order energy functions that can be transformed to $\mathcal{F}_s^2$, and thus be minimized exactly using graph cuts.

We first show that it is possible to transform all functions in class $\mathcal{M}_s^k$ to functions in $\mathcal{F}_s^2$ if $k \leq 2$. This is not a new result and follows from [24, 25]. We then go on to show that it is not possible to transform all functions in $\mathcal{M}_s^k$ to $\mathcal{F}_s^2$ in polynomial time when $k \geq 4$. In what follows, we present the transformation for different classes of multi-label functions.

The class $\mathcal{M}_s^1$: We now show how to transform a first order energy function $E_m(\mathbf{y})$ involving a single 4-label variable y_1 to a first order boolean energy function $E_b(\mathbf{x})$ involving three boolean variables $\mathbf{x} = \{x_1, x_2, x_3\}$. Let $\mathcal{L} = \{1, 2, 3, 4\}$. $E_m(\mathbf{y})$ can be written as:

$$E_m(y) = \sum_{a \in \mathcal{L}} \Theta_{1;a} \delta(y_1, a). \tag{18}$$

We replace all occurrences of $\delta(y_1, a)$ using the corresponding boolean functions $f_{y_1;a}(x_1, x_2, x_3)$ given in equation (17). This results in an energy function that depends only on $\mathbf{x}$:

$$E_b(\mathbf{x}) = \Theta_{1;1} x_1 + \Theta_{1;2}(x_2 - x_1) + \Theta_{1;3}(x_3 - x_2) + \Theta_{1;4}(1 - x_3). \tag{19}$$

Since the above energy function belongs to $\mathcal{F}_s^1$, all multi-label first order functions can be minimized exactly.

The class $\mathcal{M}_s^2$: Let y_1 and y_2 be two 4-label random variables in the following second order energy function:

$$E_m(\mathbf{y}) = \sum_{a,b \in \mathcal{L}} \Theta_{12;ab} \delta(y_1, a) \delta(y_2, b). \tag{20}$$

We transform this energy into a boolean energy function $E_b(\mathbf{x})$ involving triplets (x_1, x_2, x_3) and (x_1', x_2', x_3') replacing y_1 and y_2 respectively. The encoding function $f_{i;a}$ (given by equation (17)) is used to replace $\delta(y_i, a)$ resulting in the boolean energy:

$$E_b(\mathbf{x}) = \sum_{i,j \in \{1,2,3\}} \alpha_{ij} x_i x_j' + L_1, \tag{21}$$

where $\alpha_{ij} = (\Theta_{12;ij} - \Theta_{12;(i+1)j} - \Theta_{12;i(j+1)} + \Theta_{12;(i+1)(j+1)})$, and L_1 stands for some first order terms. If the coefficients of all quadratic terms in a boolean second order energy function are non-positive, then the energy function is submodular [6, 8]. Thus, for the above energy function in equation (21) to be submodular, we need to ensure that $\alpha_{ij} \leq 0$, *i.e.*

$$\Theta_{12;ij} - \Theta_{12;(i+1)j} - \Theta_{12;i(j+1)} + \Theta_{12;(i+1)(j+1)} \leq 0. \tag{22}$$

Note that the above condition is nothing but the submodularity condition for second order multi-label functions (See equation (7)). Thus we prove that all submodular multi-label second order functions $\mathcal{M}_s^2$ can be transformed to $\mathcal{F}_s^2$.

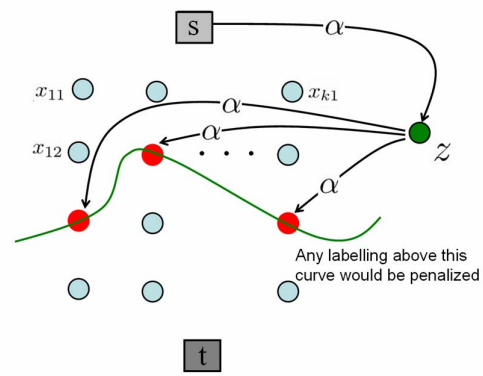

Figure 3. *The graph construction for characterizing a general k^{th} order multi-label energy function. The variable z is an auxiliary node that is connected to k boolean nodes and the source with the same edge cost α. As a result if all the k boolean nodes take the label 0 then the cost of the cut is 0. In all other cases there is a uniform cost of α. This can be seen as a generalization of the graph construction given in [13].*

The class $\mathcal{M}_s^3$: Here we focus on transforming energy functions involving cliques of size three. Let y_1, y_2 and y_3 be the three multi-label variables in the third order energy function $E_m(\mathbf{y})$:

$$E_m(\mathbf{y}) = \sum_{a,b,c} \Theta_{123;abc} \delta(y_1, a) \delta(y_2, b) \delta(y_3, c). \tag{23}$$

We use three boolean triplets (x_1, x_2, x_3), (x_1', x_2', x_3') and (x_1'', x_2'', x_3'') to encode y_1, y_2 and y_3 respectively. After replacing $\delta(y_i, a)$ with $f_{i;a}$ and applying algebraic transformations we can rewrite the energy function using boolean variables as:

$$E_b(\mathbf{x}) = \sum_{i,j,k \in \{1,2,3\}} \alpha_{ijk} x_i x_j' x_k'' + L_2, \tag{24}$$

where L_2 refers to second and first order terms. Similar to previous cases, the coefficients (α_{ijk}'s) are functions of Θ. Next, we reduce the above third order function to a second order one. The underlying idea is to substitute a product of variables by a new one[5]. In addition, appropriate penalty terms are added to make sure that at any point of minimum, the new variable takes the value of the product of the substituted variables. The basic idea of this reduction is similar to the one shown in [6, 13]. We show additional constraints under which it is possible to transform $\mathcal{M}_s^3$ to $\mathcal{F}_s^2$ in polynomial time (Proof in [18]).

The class $\mathcal{M}_s^k$: We now consider the problem of transforming fourth or higher order functions. We will show that not all functions in $\mathcal{M}_s^k$, $k \geq 4$, can be transformed to the class $\mathcal{F}_s^2$ in polynomial time. To prove this we need the following lemma.

Lemma 1 *The recognition of submodularity in quartic (degree 4) posiforms is co-NP-complete*[6] *[7].*

[5]Note that this idea is different from [2, 19]

[6]A problem $\mathcal{X}$ is co-NP if and only if its complement $\bar{\mathcal{X}}$ is in NP.

In simple words, this lemma says that it is a hard problem to say whether a general posiform, involving quartic or higher order terms, defines a submodular function or not.

We say that a transformation $\mathcal{T}_p : \mathcal{F}^k \rightarrow \mathcal{F}^2$ is a *preserving* transformation if it satisfies the following conditions.

- If $f \in \mathcal{F}_s^k$ then $\mathcal{T}(f) \in \mathcal{F}_s^2$.
- If $f \notin \mathcal{F}_s^k$ then $\mathcal{T}(f) \notin \mathcal{F}_s^2$.

A boolean function of order k is the one that can be expressed by a multi-linear polynomial expression of degree k in boolean variables. We say that a transformation works in polynomial time when we can compute a second order multi-linear polynomial expression for $\mathcal{T}_s(f)$ in $O(n^k)$ time, where n is the number of variables.

Theorem 1 *There is no preserving transformation with respect to $\mathcal{F}^k$ ($\bar{\mathcal{F}}_s^k \cup \mathcal{F}_s^k$) for $k \geq 4$, which works in polynomial time unless $P = NP$.*

Proof If such a transformation exists, we can transform any function in $\mathcal{F}^k$ to $\mathcal{F}^2$. Since submodularity can be checked in $\mathcal{F}^2$ in polynomial time[7], this gives a way to check whether any function in $\mathcal{F}^k$ is submodular or not in polynomial time, which is in contradiction with the Lemma 1. ∎

The above theorem states that it is not possible to transform all functions in $\mathcal{M}_s^k$ to $\mathcal{F}_s^2$ in polynomial time. Next, we show that it is possible to transform some functions in $\mathcal{M}_s^k$ to $\mathcal{F}_s^2$ in polynomial time.

Characterizing $\mathcal{F}_s^2$-transformable $\mathcal{M}^k$ functions: We will now characterize some $\mathcal{M}^k$ functions that can be transformed to $\mathcal{F}_s^2$ function in polynomial time. The characterization will be specified by a set of constraints on the potentials of the multi-label higher order functions. We will refer to these constraints as $\xi \leq 0$. In order to specify these constraints, we need to explain the notion of the derivatives of a discrete multi-label function [2, 5]. The derivative of a kth order function $\Theta_{\mathbf{i};\mathbf{a}}$ (See equation (3)) with respect to a variable y_j is given below:

$$\Delta_j \Theta_{\mathbf{i};\mathbf{a}} = \begin{cases} \Theta_{\mathbf{i};\mathbf{a}} - \Theta_{\mathbf{i};a_1 \ldots a_{j-1}(a_j-1)a_{j+1}\ldots a_k} & a_j > 1, \\ 0 & a_j = 1. \end{cases}$$

Recall that a_j specifies the label taken by variable y_j. The derivatives can also be obtained with respect to several variables as shown below:

$$\Delta_{\mathbf{j}} \Theta_{\mathbf{i};\mathbf{a}} = \Delta_{j_1} \ldots \Delta_{j_k} \Theta_{\mathbf{i};\mathbf{a}}, \quad j_1, \ldots, j_k \in \mathbf{j}. \tag{25}$$

The additional constraints $\xi \leq 0$ that will enable us to transform $\mathcal{M}_s^k$ functions to $\mathcal{F}_s^2$ functions are:

$$\Delta_{\mathbf{j}} \Theta_{\mathbf{i};\mathbf{a}} \leq 0, \quad \mathbf{j} \subseteq \mathbf{i}, \ |\mathbf{j}| \geq 3. \tag{26}$$

[7]The recognition of submodularity in $\mathcal{F}^2$ can be done in polynomial time by checking the coefficients of the quadratic terms [8].

The proof for this claim is presented in [18]. For illustrative purposes we now present the graph construction for functions belonging to a subclass of the $\mathcal{M}_s^k$ family. The functions belonging to this subclass have the form:

$$\Theta_{\mathbf{k};\mathbf{i}} = \begin{cases} \alpha & \exists i \in \mathbf{i} : y_i < l_i, \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

The corresponding graph construction is shown in figure 3. This graph contains a number of auxiliary nodes z. We connect the nodes n_1,n_2,...,n_k representing the boolean encoding variables x_1,x_2,...,x_k to an auxiliary node z, and connect z to the source node s. For a group of variables $y_j, j \in \mathbf{i}$, if any variable y_i takes a label less than l_i there is a penalty of α. Our method can automatically find the required auxiliary nodes and various edge costs for the graph needed to minimize any $\mathcal{M}_s^k$-function that satisfy constraints (26). Note that by having different edge weights [11, 30] that connect the auxiliary node to other nodes we can relax the above constraints (26) and obtain a less restrictive set of constraints. This is an interesting future direction.

5.1. Non-submodular problems

The above transformations succeed when the original multi-label problem satisfies the necessary conditions shown in the previous section. For problems that do not satisfy these constraints we use an alternative two-step transformation procedure. First we use the encoding functions to transform the multi-label higher order function to a boolean higher-order one. We then use the reduction techniques [19] to reduce higher order functions to second order ones[8]. Note that these reduction techniques do not preserve submodularity. The resulting class of non-submodular functions $\bar{\mathcal{F}}_s^2$ can be minimized using QPBO [21] or truncation [22], which are approximate and are not guaranteed to give the global optimum.

6. Application: Single View Reconstruction

We now show how the higher order functions characterized in the previous section can be used to improve single view reconstruction results. Given a 2D image of a scene, the goal is to recover a theatre stage representation containing major surfaces and their geometrical relationships to each other. Hoiem *et al.* [9] formulated this as a classification problem where every pixel in the image is assigned one of the three labels, namely, support (surfaces that lie parallel to the ground plane), vertical (surfaces that *rise* from the ground plane), and sky. They obtained impressive results by learning appearance based models of the three classes. Their method works as follows. The given image is first segmented into superpixels [4] (see second column of figure 5), which provide spatial support for computing features like

[8]The minimization of a pseudo-boolean function of any degree can always be reduced in polynomial time to the minimization of a quadratic pseudo-boolean function (second order) [19].

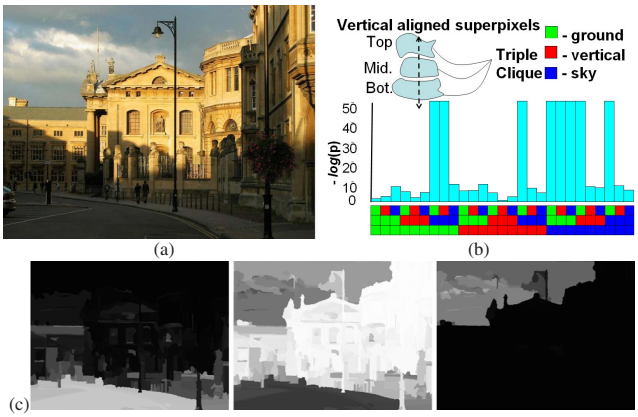

Figure 4. *(a) Original image. (b) Triplets of vertically aligned superpixels are chosen from the superpixel segmented images. The labellings for individual triplet combinations are studied from several ground truth images. Negative log-likelihoods are computed for each of these triplets and used as third order priors in the labelling problem, formulated as an energy minimization task. (c) The three columns, from left to right, show the unary likelihood images of ground, vertical and sky respectively.*

texture filter responses and vanishing points. Using boosted decision tree classifiers, geometrical likelihoods are computed for individual superpixels (cf. figure 4). The final geometrical labelling is achieved using these likelihoods along with pairwise smoothness priors in an energy minimization framework.

In this work we focus on improving the results in [9] using priors obtained from natural statistics. Such priors can only be imposed through CRFs with higher order cliques [14, 20]. We formulate the single view 3D reconstruction problem as a MAP estimation problem in a Bayesian framework. The superpixels extracted from the image act as nodes (variables) in a higher order CRF. The most probable labelling of the superpixels is found by minimizing an equivalent energy function. We minimize a third order three-label energy function, where the three labels for each superpixel correspond to ground, vertical and sky.

The unary likelihoods $\theta_{i;a}$ of the energy function are computed using boosted decision tree classifiers[9]. Motivated by the work of [28], we compute the second and third order energy terms using natural statistics. Yang and Purves [28] study the distribution of geometrical features like size, shape and depth of planar surfaces, from a large training database. Using a similar approach, the second order terms are computed by learning the statistics of all neighbouring superpixel pairs in the training dataset.

As the images are generally taken by people standing on the ground, with the optical axis approximately parallel to the ground, there is a natural ordering of the superpixels labels in the vertical direction. To capture this ordering, we study the distribution of the labelling of vertically-aligned superpixel triplets from several groundtruth labelled images. These statistics, in the form of negative log likelihoods, are shown in figure 4(b). The likelihoods are directly used as the higher order potential $\theta_{ijk;abc}$ in the energy function. As an example, to see the effectiveness of natural statistics, consider the cost of the triplet labelling [Top:Ground, Middle:Vertical, Bottom:Sky] from the figure. Given the label ordering, this configuration is unlikely to occur naturally, and thus has a high cost. We use our algorithm explained in §5 to construct the equivalent boolean graph. A simple truncation method is used to remove the negative edges in the graph [22].

[9] http://www.cs.cmu.edu/∼dhoiem/projects/software.html

Image	Results of [9]	Our method
street	20.78	5.82
highway	19.47	7.32
buildings	31.94	13.36
road	18.52	10.82
college	29.47	13.26

Table 1. *Percentage misclassifications.*

We observed significant improvement over the results of [9], as shown in figure 5. The labelling accuracy is summarized in table 1. The accuracy is reported in terms of the misclassification of individual pixels in the image. In figure 5 we show the original image, superpixel segmentation, results using only pairwise clique potentials, and our results using higher order clique potentials.

Higher order clique potentials are useful for the single view reconstruction problem for several reasons. In the street image shown in the first row of figure 5, the ground between the two buildings is incorrectly labelled as vertical, when only pairwise smoothness prior is used. On the other hand, the usage of higher order priors results in the correct labelling. The major advantage comes from the ability to impose priors based on natural statistics. For example, in the second row of figure 5, unary potentials favour the labelling sky for the van due to its high similarity to the 'sky' region. However, our method using priors learned from natural statistics obtains the correct labelling.

7. Discussion

Submodular functions: Development of global minimization algorithms is very important because researchers very often use heuristic methods for optimization, even for problems where exact ones exist. Our transformations can be used for vision problems such as stereo [10], panoramic stitching [15], and image restoration. Recently, the transformation proposed by [10] was used to develop a new move algorithm [26]. Similar techniques can be produced for the transformations proposed in this paper.

Non-submodular functions: Most vision problems are non-submodular in practice. Given a non-submodular second order boolean energy function, recent works using QPBO techniques [2, 21] compute global optimum for a par-

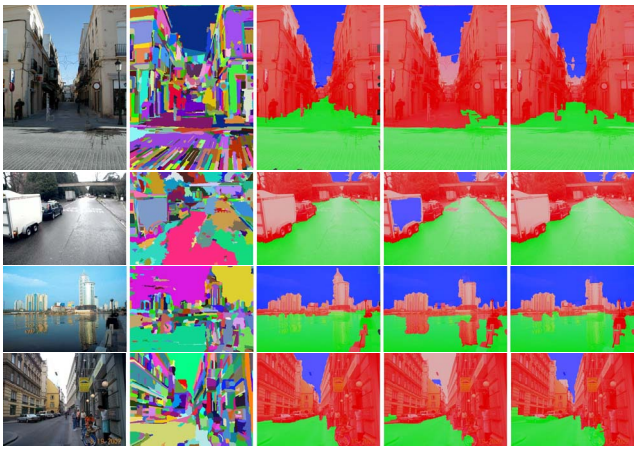

Figure 5. *Original image, superpixel segmentation, ground truth labelling, results from [9] and our results are shown (left to right). Street, highway, buildings and road are the images in rows 1 to 4 respectively. (Best viewed in colour)*

tial set of variables, and use an approximation algorithm for the rest. Our framework can transform any higher order multi-label function, which can also be learned using fields of experts model [20], to a boolean second order function, and if the resulting second order boolean energy function is non-submodular we can use QPBO techniques. We believe that our algorithm can be used to obtain better solutions for vision problems such as stereo, image restoration and image inpainting where the higher order priors are learned through models like fields of experts [20].

Acknowledgments This work was supported by the EPSRC research grants EP/C006631/1(P) and GR/T21790/01(P), the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. We thank Prof. Endre Boros and Christopher Russell for valuable discussions.

References

[1] A. Billionnet and M. Minoux. Maximizing a supermodular pseudo-boolean function: a polynomial algorithm for supermodular cubic functions. *Discrete Appl. Math.*, 1985.

[2] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Appl. Math.*, 123(1-3):155–225, 2002.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), 2001.

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

[5] B. Flach and D. Schlesinger. Best labelling search for a class of higher order Gibbs models. *Pattern Recognition and Image Analysis*, 2004.

[6] D. Freedman and P. Drineas. Energy minimization via graph cuts: Settling what is possible. In *CVPR*, volume 2, pages 939–946, 2005.

[7] G. Gallo and B. Simeone. On the supermodular knapsack problem. *Mathematical Programming: Series A and B*, 45(2):295–309, 1989.

[8] P. L. Hammer. Some network flow problems solved with pseudo-boolean programming. *Operations Research*, 13:388–399, 1965.

[9] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1):151–172, 2007.

[10] H. Ishikawa. Exact optimization for Markov random fields with convex priors. *PAMI*, 25:1333–1336, 2003.

[11] P. Kohli, L. Ladicky, and P. H. S. Torr. Graph cuts for minimizing robust higher order potentials. Technical report, Oxford Brookes University, 2008.

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

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *PAMI*, 26(2), 2004.

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

[15] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stiching in the gradient domain. In *ECCV*, 2004.

[16] J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. In *IPCO*, 2007.

[17] B. Potetz. Efficient belief propagation for vision using linear constraint nodes. In *CVPR*, 2007.

[18] S. Ramalingam, P. Kohli, K. Alahari, and P. H. S. Torr. Transforming multi-label higher order crfs to binary pairwise ones. Technical report, Oxford Brookes University, 2008.

[19] I. G. Rosenberg. Reduction of bivalent maximization to the quadratic case. *Cahiers du Centre d'Etudes de Recherche Operationnelle*, 17:71–74, 1975.

[20] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, 2005.

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

[22] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry. In *CVPR*, 2005.

[23] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithm. *IJCV*, 47:7–42, 2002.

[24] D. Schlesinger. Exact solution of permuted submodular min-sum problems. In *EMMCVPR*, pages 28–38, 2007.

[25] D. Schlesinger and B. Flach. Transforming an arbitrary min-sum problem into a binary one. Technical Report TUD-FI06-01, Dresden University of Technology, 2006.

[26] O. Veksler. Graph cuts based optimization for Mrfs with truncated convex priors. In *CVPR*, 2007.

[27] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-based reparameterization for approximate inference on loopy graphs. In *NIPS*, pages 1001–1008, 2001.

[28] Z. Yang and D. Purves. Image/source statistics of surfaces in natural scenes. *Network: Comput. Neural Syst.*, 2003.

[29] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000.

[30] B. Zalesky. Efficient determination of gibbs estimators with submodular energy functions, http://arxiv.org/abs/math/0304041v1, 2003.