Kohli, P and Torr, P H
Measuring Uncertainty in Graph Cut Solutions.

Kohli, P and Torr, P H (2008) Measuring Uncertainty in Graph Cut Solutions. *Journal of Computer Vision and Image Understanding, special issue Discrete Optimization in Computer Vision*, 112 (1). pp. 30-38.
DOI: 10.1016/j.cviu.2008.07.002

Measuring Uncertainty in Graph Cut Solutions

Pushmeet Kohli

Microsoft Research,
Cambridge, UK

Philip H. S. Torr

Oxford Brookes University,
Oxford, UK

Abstract

In recent years graph cuts have become a popular tool for performing inference in Markov and Conditional Random Fields. In this context the question arises as to whether it might be possible to compute a measure of uncertainty associated with the graph-cut solutions. In this paper we answer this particular question by showing how the min-marginals associated with the label assignments of a random field can be efficiently computed using a new algorithm based on dynamic graph cuts. The min-marginal energies obtained by our proposed algorithm are exact, as opposed to the ones obtained from other inference algorithms like loopy belief propagation and generalized belief propagation. The paper also shows how min-marginals can be used for parameter learning in conditional random fields.

Key words: Parameter Learning, Inference, Min-marginals, Graph Cuts

1 Introduction

Researchers in computer vision have extensively used graph cuts to compute the Maximum a Posteriori (MAP) solutions for various discrete pixel labelling problems such as image restoration, segmentation and stereo. One of the primary reasons for the growing popularity of graph cuts is their ability to find the globally optimal solutions for an important class of energy functions in polynomial time [20]. Even for problems where graph cuts do not guarantee

Email addresses: pkohli@microsoft.com (Pushmeet Kohli),
philiptorr@brookes.ac.uk (Philip H. S. Torr).

Preprint submitted to Elsevier *2 July 2008*

optimal solutions they can be used to find solutions which are strong local minima of the energy [5]. These solutions for certain problems have been shown to be better than the ones obtained by other methods [4,24].

Graph cuts however do suffer from a big disadvantage. Unlike other inference algorithms, they do not provide any uncertainty measure associated with the solution they produce. This is a serious drawback since researchers do not obtain any information regarding the probability of a particular latent variable assignment in a graph cut solution. Inference algorithms such as Loopy Belief Propagation (LBP), Generalized Belief Propagation (GBP) and the recently introduced Tree Re-weighted message passing (TRW) [19,27] output approximate marginal or min-marginal energies associated with each latent variable. Note that for tree-structured graphs, the simple max-product belief propagation algorithm gives the exact max-marginal probabilities/min-marginal energies [1] for different label assignments in $O(nl^2)$ time where n is the number of latent variables, and l is the number of labels a latent variable can take.

This paper addresses the problem of efficiently computing the min-marginals associated with the label assignments of any latent variable in a Markov Random Field (MRF). Our method can work on all MRFs or CRFs that can be solved using graph cuts. First, we show how in the case of binary variables, the min-marginals associated with the labellings of a latent variable are related to the *flow-potentials* (defined in section 3) of the node representing that latent variable in the graph constructed in the energy minimization procedure. The exact min-marginal energies can be found by computing these *flow-potentials*. We then show how flow potential computation is equivalent to the problem of minimizing a *projection* of the original energy function [2].

Minimizing a *projection* of an energy function is a computationally expensive operation and requires a graph cut to be computed. In order to obtain the min-marginals corresponding to all label assignments of all random variables, we need to solve $O(nl)$ number of st-mincut problems. In this paper, we present an algorithm based on dynamic graph cuts [16] which solves these $O(nl)$ problems extremely quickly. Our experiments show that the running time of this algorithm i.e. the time taken for it to compute the min-marginals corresponding to all latent variable label assignments is of the same order of magnitude as the time taken to compute a single graph cut. The first version of this paper appeared as [17]. This extended version shows how the min-

[1] We will explain the relation between max-marginal probabilities and min-marginal energies later in section 2. To make our notation consistent with recent work in graph cuts, we formulate the problem in terms of min-marginal energies (subsequently referred to as simply min-marginals).

[2] A projection of the function $f(x_1, x_2, ..., x_n)$ can be obtained by fixing the values of some of the variables in the function $f(.)$. For instance $f'(x_2, ..., x_n) = f(0, x_2, ..., x_n)$ is a projection of the function $f(.)$.

marginals obtained using our method can be used for parameter learning in CRFs.

1.1 Overview of Dynamic Graph Cuts

Dynamic computation is a paradigm that prescribes solving a problem by dynamically updating the solution of the previous problem instance. Its hope is to be more efficient than a computation of the solution from scratch after every change in the problem. A considerable speedup in computation time can be achieved using this procedure, especially, when the problem is large and changes are few. Dynamic algorithms are not new to computer vision. They have been extensively used in computational geometry for problems such as range searching, intersections, point location, convex hull, proximity and many others [6].

Boykov and Jolly [3] were the first to use a *partially* dynamic st-mincut algorithm in a vision application. They proposed a technique with which they could update capacities of *certain* graph edges, and recompute the st-mincut dynamically. They used this method for performing interactive image segmentation where the user could improve segmentation results by giving additional segmentation cues (seeds) in an online fashion. However, their scheme was restrictive and did not allow for general changes in the graph. In one of our earlier papers, we proposed a new algorithm overcoming this restriction [16], which is faster and allows for arbitrary changes to be made in the graph. The running time of this new algorithm has been empirically shown to increase linearly with the number of edge weights changed in the graph. In this paper, we will use this algorithm to compute the exact min-marginals efficiently.

1.2 Our Contributions

To summarize, the key contributions of this paper include:

- The discovery of a novel relationship between min-marginal energies and node flow-potentials in the residual graph obtained after the graph cut computation.
- An efficient algorithm based on dynamic graph cuts to compute min-marginals by minimizing energy function projections.
- The use of min-marginals for learning parameters of CRFs used for modelling labelling problems.

1.3 Organization of the Paper

The paper starts by describing the basics of random fields and graph cuts and proceeds to discuss the relationship between min-marginals and node-flow potentials. We then show how max-marginal probabilities can be found by minimizing projections of the energy function defining a MRF or CRF, and how dynamic graph cuts can be used to efficiently compute the minimum values of these projections. Our algorithm can handle all energy functions that can be solved using graph cuts [9,11,14,20].

We discuss random fields and min-marginal energies in section 2. In section 3, we formulate the st-mincut problem, define terms that would be used in the paper, and describe how certain energy functions can be minimized using graph cuts. In section 4, we show how min-marginals can be found by minimizing projections of the original energy function. In the same section we describe a novel algorithm based on dynamic graph cuts to efficiently compute the minima of these energy projections. In section 5, we discuss some applications of our algorithm.

2 Notation and Preliminaries

We will now describe the notation used in the paper. We will formulate our problem in terms of a *pairwise* MRF [3]. Note that the pairwise assumption does not affect the generality of our formulation since any MRF involving higher order interaction terms can be converted to a *pairwise* MRF by addition of auxiliary variables in the MRF [28].

Consider a discrete random field $\mathbf{X}$ defined over a lattice $\mathcal{V} = \{1, 2, \ldots, n\}$ with a neighbourhood system $\mathcal{N}$. Each random variable $X_i \in \mathbf{X}$ is associated with a lattice point $i \in \mathcal{V}$ and takes a value from the label set $\mathcal{X}_v$. The neighborhood system $\mathcal{N}$ of the random field is defined by the sets $\mathcal{N}_i, \forall i \in \mathcal{V}$, where $\mathcal{N}_i$ denotes the set of all neighbours of the variable X_i. Any possible assignment of labels to the random variables is called a *labelling* or *configuration*. It is denoted by the vector $\mathbf{x}$, and takes values from the set $\mathcal{X}$ defined as $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \ \mathcal{X}_n$. Unless noted otherwise, we use symbols u and v to denote values in $\mathcal{V}$, and i and j to denote particular values in $\mathcal{X}_u$ and $\mathcal{X}_v$ respectively.

A random field is said to be a Markov random field (MRF) with respect to a neighborhood system $\mathcal{N} = \{\mathcal{N}_v | \ v \in \mathcal{V}\}$ if and only if it satisfies the positivity

[3] Pairwise MRFs have cliques of size at most two.

property: $\Pr(\mathbf{x}) > 0 \;\; \forall \mathbf{x} \in \mathcal{X}$, and the Markovian property:

$$\Pr(x_v | \{x_u : u \in \mathcal{V} - \{v\}\}) = \Pr(x_v | \{x_u : u \in \mathcal{N}_v\}) \qquad \forall v \in \mathcal{V}. \tag{1}$$

Here we refer to $\Pr(X = \mathbf{x})$ as $\Pr(\mathbf{x})$ and $\Pr(X_i = x_i)$ as $\Pr(x_i)$. A conditional random field (CRF) may be viewed as an MRF globally conditioned on the data.

The MAP-MRF estimation problem aims to find the configuration $\mathbf{x}$ which has the highest probability. It can be formulated as an energy minimization problem where the energy corresponding to a MRF configuration $\mathbf{x}$ is defined as

$$E(\mathbf{x}|\theta) = -\log \Pr(\mathbf{x}|\mathbf{D}) - \text{const}. \tag{2}$$

Here θ is the energy parameter vector defining the MRF [19]. The energy functions characterizing MRFs used in computer vision can usually be written as a sum of unary and pairwise terms:

$$E(\mathbf{x}|\theta) = \sum_{v \in \mathcal{V}} \left(\phi(x_v) + \sum_{u \in \mathcal{N}_v} \phi(x_u, x_v) \right) + \text{const}. \tag{3}$$

In the paper, $\psi(\theta)$ is used to denote the value of the energy of the MAP configuration of the MRF and is defined as:

$$\psi(\theta) = \min_{\mathbf{x} \in \mathcal{X}} E(\mathbf{x}|\theta). \tag{4}$$

In what follows, the term *optimal solution* will be used to refer to the MAP solution of the random field.

2.1 Min-marginal energies

A min-marginal is a function that provides information about the minimum values of the energy E under different constraints. Following the notation of [19], we define the min-marginal energies $\psi_{v;j}, \psi_{uv;ij}$ as:

$$\psi_{v;j}(\theta) = \min_{\mathbf{x} \in \mathcal{X}, x_v = j} E(\mathbf{x}|\theta), \tag{5}$$

$$\psi_{uv;ij}(\theta) = \min_{\mathbf{x} \in \mathcal{X}, x_u = i, x_v = j} E(\mathbf{x}|\theta). \tag{6}$$

In words, given an energy function E whose value depends on the variables $(X_1, X_2, \ldots, X_n)$, $\psi_{v;j}(\theta)$ represents the minimum energy value obtained if we fix the value of variable X_v to j ($x_v = j$) and minimize over all remaining variables. Similarly, $\psi_{uv;ij}(\theta)$ represents the value of the minimum energy obtained by assigning labels i and j to variables X_u and X_v respectively, and minimizing over all other variables.

Now we show how min-marginals can be used to compute a confidence measure for a particular latent variable label assignment. Given the function $\Pr(\mathbf{x}|\mathbf{D})$, which specifies the probability of a configuration of the MRF, the max-marginal $\mu_{v;j}$ gives us the value of the maximum probability over all possible configurations of the MRF in which $x_v = j$. Formally, it is defined as:

$$\mu_{v;j} = \max_{\mathbf{x}\in\mathcal{X};x_v=j} \Pr(\mathbf{x}|\mathbf{D}) \tag{7}$$

Inference algorithms like max-product belief propagation produce the max-marginals along with the MAP solution. These max-marginals can be normalized to obtain a confidence measure σ for any latent variable labelling as:

$$\sigma_{v;j} = \frac{\max_{\mathbf{x}\in\mathcal{X},x_v=j} \Pr(\mathbf{x}|\mathbf{D})}{\sum_{k\in\mathcal{X}_v} \max_{\mathbf{x}\in\mathcal{X},x_v=k} \Pr(\mathbf{x}|\mathbf{D})} = \frac{\mu_{v;j}}{\sum_{k\in\mathcal{X}_v} \mu_{v;k}} \tag{8}$$

where $\sigma_{v;j}$ is the confidence for the latent variable X_v taking label j. This is the ratio of the max-marginal corresponding to the label assignment $x_v = j$ to the sum of the max-marginals for all possible label assignments of variable X_v.

We now show how max-marginals can be obtained from the min-marginal energies computed by our algorithm. Substituting the value of $\Pr(\mathbf{x}|\mathbf{D})$ from equation (2) in equation (7), we get

$$\mu_{v;j} = \max_{\mathbf{x}\in\mathcal{X};x_v=j} (\exp(-E(\mathbf{x}|\theta) - \text{const})) = \frac{1}{Z}\exp(-\min_{\mathbf{x}\in\mathcal{X};x_v=j} E(\mathbf{x}|\theta)), \tag{9}$$

where Z is the partition function. Combining this with equation (5), we get

$$\mu_{i;j} = \frac{1}{Z}\exp(-\psi_{v;j}(\theta)). \tag{10}$$

As an example consider a binary label object-background image segmentation problem where there are two possible labels i.e. object ('*ob*') and background ('*bg*'). The confidence measure $\sigma_{v;ob}$ associated with the pixel v being labelled as object can be computed as:

$$\sigma_{v;ob} = \frac{\mu_{v;ob}}{\mu_{v;ob} + \mu_{v;bg}} = \frac{\frac{1}{Z}\exp(-\psi_{v;ob}(\theta))}{\frac{1}{Z}\exp(-\psi_{v;ob}(\theta)) + \frac{1}{Z}\exp(-\psi_{v;bg}(\theta))}, \tag{11}$$

$$= \frac{\exp(-\psi_{v;ob}(\theta))}{\exp(-\psi_{v;ob}(\theta)) + \exp(-\psi_{v;bg}(\theta))} \tag{12}$$

Note that the Z's cancel and thus we can compute the confidence measure from the min-marginal energies alone without knowledge of the partition function.

2.3 Computing the M most probable configurations

An important use of min-marginals is to find the M most probable configurations (or labellings) for latent variables in a Bayesian network [29]. Dawid [7] showed how min-marginals on junction trees can be computed, which was later used by [22] to find the M most probable configurations of a probabilistic graphical network. For tree-structured networks, the method of [7] is guaranteed to run in polynomial time. However, its worst case complexity for arbitrary graphs is exponential in the number of the nodes in the graphical model. The method proposed in this paper is able to produce the exact min-marginals for submodular energy functions of the form (3) defined over graphs of arbitrary topology in polynomial time.

3 Energy Minimization using Graph Cuts

In this section we will give a brief overview of graph cuts and show how they can used to minimize energy functions such as the one defined in equation (3).

3.1 The st-Minimum Cut Problem

Consider a weighted graph $G(V, E)$ with two special nodes, namely the source s and the sink t, collectively referred to as the terminals. A cut is a partition of the node set V of a graph G into two parts S and $\overline{S} = V - S$, and is defined by the set of edges (i, j) such that $i \in S$ and $j \in \overline{S}$. The cost of a cut $(S, \overline{S})$ is equal to:

$$C(S, \overline{S}) = \sum_{(i,j)\in E; i\in S; j\in\overline{S}} (c_{ij}) \tag{13}$$

where c_{ij} is the cost associated with the edge (i, j). The st-mincut problem involves finding a cut in the graph with the smallest cost satisfying the properties $s \in S$ and $t \in \overline{S}$.

By the Ford-Fulkerson theorem [10], the st-mincut problem is equivalent to computing the maximum flow from the source to the sink with the capacity of each edge equal to c_{ij}. While passing flow through the network, a number of edges of the graph become saturated. When the maximum amount of flow is being passed in the network, there remains no path from the source to the sink that does not have a saturated edge. In effect, these saturated edges separate the source from the sink and thus by the Ford-Fulkerson theorem, constitute the minimum cost st-cut (st-mincut).

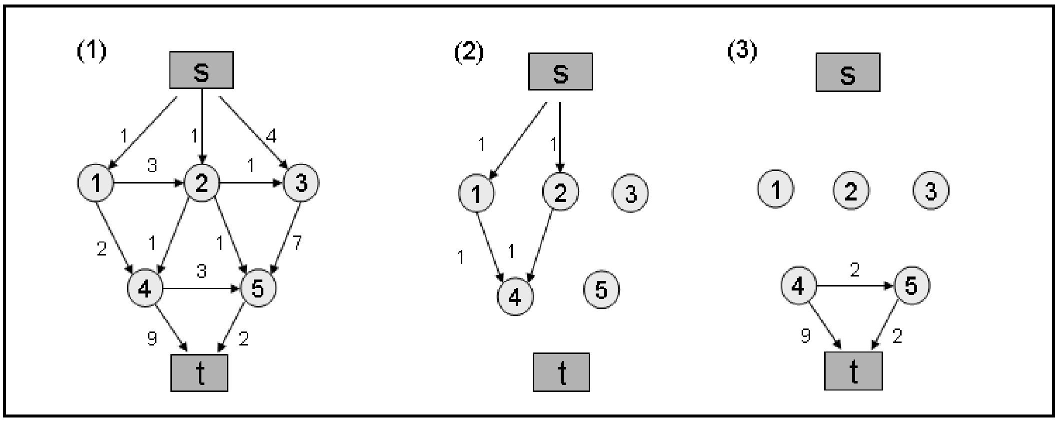


Fig. 1. *Illustrating the flow potentials of graph nodes. (1) The figure shows a directed graph having seven nodes, two of which are the terminal nodes, the source s and the sink t. The number associated with each directed edge in this graph is a capacity which tells us the maximum amount of flow that can be passed through it in the direction of the arrow. The flow potentials for node 4 in this graph when no flow is passing through any of the edges are $f_4^s = 2$ and $f_4^t = 11$ (see graphs (2) and (3)).*

3.1.1 Computing the Maximum Flow

The Max-flow problem for a capacitated network $G(V, E)$ with a non-negative capacity c_{ij} associated with each edge is that of finding the maximum flow f from the source node s to the sink node t subject to the edge capacity and flow balance constraints:

$$0 \leq f_{ij} \leq c_{ij} \quad \forall (i, j) \in E, \quad \text{and} \tag{14}$$

$$\sum_{i \in N(v)} (f_{vi} - f_{iv}) = 0 \quad \forall v \in V - \{s, t\} \tag{15}$$

where f_{ij} is the flow from node i to node j, and $N(v)$ is the neighbourhood of node v.

3.1.2 Augmenting Paths, Residual Graphs and Flow Potentials

Given any flow f_{ij}, the residual capacity r_{ij} of an edge $(i, j) \in E$ is the maximum additional flow that can be sent from node i to node j using the edges (i, j) and (j, i). A residual graph $G(f)$ of the graph G consists of the node set V and the edges with positive residual capacity (with respect to the flow f). An augmenting path is a path from the source to the sink along unsaturated edges of the residual graph.

We define the *source/sink flow potential* of a graph node $v \in V$ as the maximum amount of net flow that can be pumped into/from it without invalidating any edge capacity (14) or mass balance (15) constraints with the exception of the mass balance constraint of the node v itself. Formally, we can define the source flow potential of node v as:

$$f_v^s = \max_{\mathbf{f}} \sum_{i \in N(v)} f_{iv} - f_{vi}$$

subject to:

$$0 \leq f_{ij} \leq c_{ij} \quad \forall (i,j) \in E, \quad \text{and} \tag{16}$$

$$\sum_{i \in \; N(j) \backslash \{s,t\}} (f_{ji} - f_{ij}) = f_{sj} - f_{jt} \quad \forall j \in V \backslash \{s,t,v\} \tag{17}$$

where $\max_{\mathbf{f}}$ represents the maximization over the set of all edge flows

$$\mathbf{f} = \{f_{ij}, \; \forall (i,j) \in E\}. \tag{18}$$

Similarly, the *sink flow potential* f_v^t of a graph node v is defined as:

$$f_v^s = \max_{\mathbf{f}} \sum_{i \in N(v)} f_{vi} - f_{iv} \tag{19}$$

subject to constraints (16) and (17).

The computation of a flow potential of a node is not a trivial process and in essence requires a graph cut to be computed as explained in figure 3. The flow potentials of a particular graph node are shown in figure 1. Note that in a residual graph $G(f_{\max})$ where $f_{\max}$ is the maximum flow, all nodes on the sink side of the st-mincut are disconnected from the source and thus have the source flow potential equal to zero. Similarly, all nodes belonging to the source have the sink flow potential equal to zero. We will later show that the flow-potentials we have just defined are intimately linked to the min-marginal energies of latent variable label assignments.

3.2 Submodular functions and Energy Minimization

Submodular set functions play an important role in energy minimization [2]. The key property that makes them special is the fact that they can be minimized in polynomial time [15]. In fact some submodular functions can be minimized by solving an st-mincut problem. Functions of binary random variables can be seen as set functions. A function $f(x_1, x_2)$ of two binary random variables $\{x_1, x_2\}$ is submodular if and only if:

$$f(0,0) + f(1,1) \leq f(0,1) + f(1,0) \tag{20}$$

A function $f : \mathbb{B}^n \to R$ is submodular if and only if all its projections on 2 variables are submodular [2,20].

The basic procedure for energy minimization using graph cuts comprises of building a graph in which each cut defines a configuration $\mathbf{x}$, and the cost

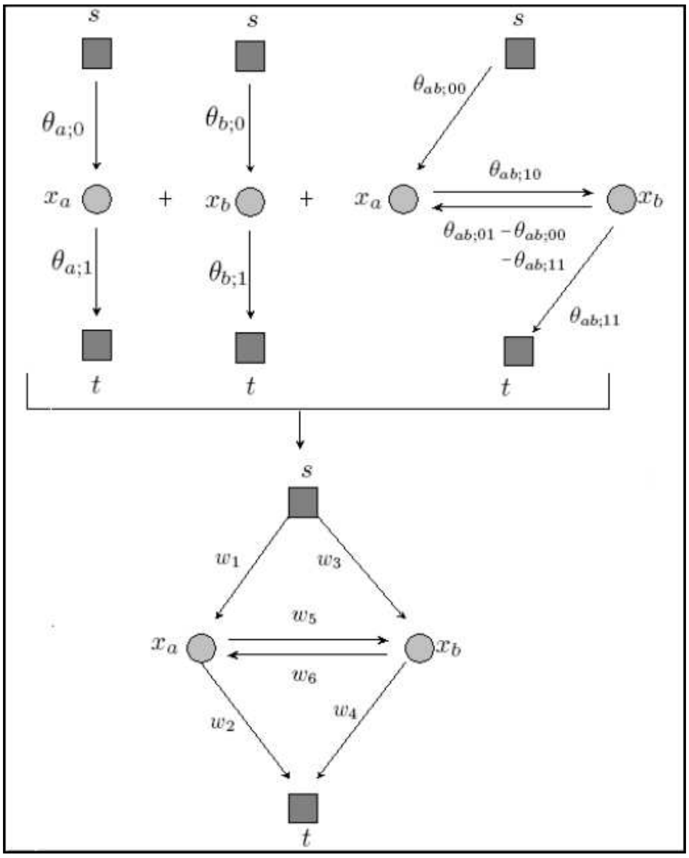


Fig. 2. *Energy minimization using graph cuts. The figure shows how individual unary and pairwise terms of an energy function taking two binary variables are represented and combined in the graph. The cost of a st-cut in the final graph is equal to the energy* $E(\mathbf{x})$ *of the configuration* $\mathbf{x}$ *the cut induces. The minimum cost st-cut induces the least energy configuration* $\mathbf{x}$ *for the energy function.*

of the cut is equal to the energy value associated with $\mathbf{x}$ i.e. $E(\mathbf{x}|\theta)$. The minimum cost configuration can be computed by finding the st-mincut in this graph. The st-mincut can be computed in polynomial time if the costs of all edges in the graph are non-negative. This condition restricts the class of energy functions which can be minimized using polynomial time algorithms for the st-mincut problem.

Kolmogorov and Zabih [4] showed that submodular functions of binary variables which are defined over cliques of size 3 or less can be minimized in this manner. They also described how to construct the graph for this particular class of energy functions. Freedman and Drineas [11] added to this result by characterizing a class of functions involving higher order cliques whose minimization can be translated to a st-mincut problem. Some classes of multi label functions which can be minimized exactly by solving a st-mincut problem have also been characterized independently by [9,14,23].

We will now briefly discuss the method for minimizing binary submodular functions. The minimization procedure works by decomposing the MRF energy function (3) into unary and pairwise energy terms. The energy in this form can be written as:

$$E(\mathbf{x}|\theta) = \theta_{\text{const}} + \sum_{v\in V, i\in \mathcal{X}_v} \theta_{v;i}\delta_i(x_v) + \sum_{(s,t)\in E, (j,k)\in(\mathcal{X}_s,\mathcal{X}_t)} \theta_{st;jk}\delta_j(x_s)\delta_k(x_t), \quad (21)$$

where $\theta_{v;i}$ is the penalty for assigning label i to latent variable X_v, $\theta_{st;ij}$ is the penalty for assigning labels i and j to the latent variables X_s and X_t, and each $\delta_j(x_s)$ is an indicator function which is defined as:

$$\delta_j(x_s) = \begin{cases} 1 \text{ if } x_s = j, \text{ where} j \in \mathcal{X}_s \\ 0 \text{ otherwise} \end{cases},$$

These energy terms are represented by weighted edges in the graph. Multiple edges between the same nodes are merged into a single edge by adding their weights. Finally, the st-mincut is found in this graph, which provides us with the MAP solution. The cost of this cut corresponds to the energy of the MAP solution. The labelling of a latent variable depends on the terminal it is disconnected from by the minimum cut. If the node is disconnected from the source, we assign it the value zero and one otherwise. The graph construction for a two node MRF is shown in figure 2.

4 Computing Min-marginals using Graph Cuts

We now explain the procedure for the computation of min-marginal energies using graph cuts. The total flow f_{total} flowing from the source s to the sink t in a graph can be found by computing the difference between the total amount of flow coming in to a terminal node and that going out, or more formally:

$$f_{\text{total}} = \sum_{i\in N(s)} (f_{si} - f_{is}) = \sum_{i\in N(t)} (f_{it} - f_{ti}). \quad (22)$$

The cost of the st-mincut in an energy representing graph is equal to the energy of the optimal configuration. From the Ford-Fulkerson theorem, this is also equal to the maximum amount of flow $f_{\max}$ that can be transferred from the source to the sink. Hence, from the minimum energy (4) and total flow (22) equations for a graph in which maxflow has been achieved i.e. $f_{\text{total}} = f_{\max}$, we obtain:

$$\psi(\theta) = \min_{\mathbf{x}\in\mathcal{X}} E(\mathbf{x}|\theta) = f_{\max} = \sum_{i\in N(s)} (f_{si} - f_{is}). \quad (23)$$

Note that flow cannot be pushed into the source i.e. $f_{is} = 0, \forall i \in V$, thus $\psi(\theta) = \sum_{i \in N(s)} f_{si}$. The MAP configuration $\mathbf{x}^*$ of a MRF is the one having the least energy and is defined as $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} E(\mathbf{x}|\theta)$.

Let a be the label for random variable X_v under the MAP solution and b be any label other than a. Then in the case of the assignment $x_v = a$, the min-marginal energy $\psi_{v;x_v^*}(\theta)$ is equal to the minimum energy i.e. $E(\mathbf{x}|\theta) = \psi(\theta)$. Thus it can be seen that the maximum flow equals the min-marginals for the case when the latent variables take their respective MAP labels.

The min-marginal energy $\psi_{v;b}(\theta)$ corresponding to the non-optimal label b can be computed by finding the minimum value of the energy function projection E' obtained by enforcing the constraint $x_v = b$ as:

$$\psi_{v;b}(\theta) = \min_{\mathbf{x}\in\mathcal{X}, x_v = b} E(\mathbf{x}|\theta) \tag{24}$$

$$\text{or, } \psi_{v;b}(\theta) = \min_{(\mathbf{x}-x_v)\in(\mathcal{X}-\mathcal{X}_v)} E(x_1, .., b, x_{v+1}..x_n|\theta). \tag{25}$$

In the next subsection, we will show that this constraint can be enforced in the original graph construction used for minimizing $E(\mathbf{x}|\theta)$ by modifying certain edge weights. These changes to the graph ensure that the latent variable X_v takes the label b. The exact modifications needed in the graph for the case of binary labels are given first while those required in the case of multi-label functions are discussed later.

4.1 Min-marginals and Flow potentials

We now show how in the case of binary variables, flow-potentials in the residual graph $G(f_{\max})$ are related to the min-marginal energy values. Again, a and b are used to represent the MAP and non-MAP labels respectively.

Theorem 1 *The min-marginal energies of a binary latent variable X_v are equal to the sum of the maximum flow and the source/sink flow potentials of the node representing it in the residual graph $G(f_{\max})$ resulting from the max-flow solution i.e.*

$$\psi_{v;j}(\theta) = \min_{x\in\mathcal{X}, x_v = j} E(\mathbf{x}|\theta) = \psi(\theta) + f_v^{T(j)} = f_{\max} + f_v^{T(j)} \tag{26}$$

where $T(j)$ is the terminal node representing label j, and $f_{\max}$ is the value of the maximum flow in the graph G representing the energy function $E(\mathbf{x}|\theta)$.

Proof The proof is trivial for the case when the latent variable takes the optimal label. We already know that the value of the min-marginal $\psi_{v;a}(\theta)$ is equal to the lowest energy $\psi(\theta)$. Further, the flow potential of the node for

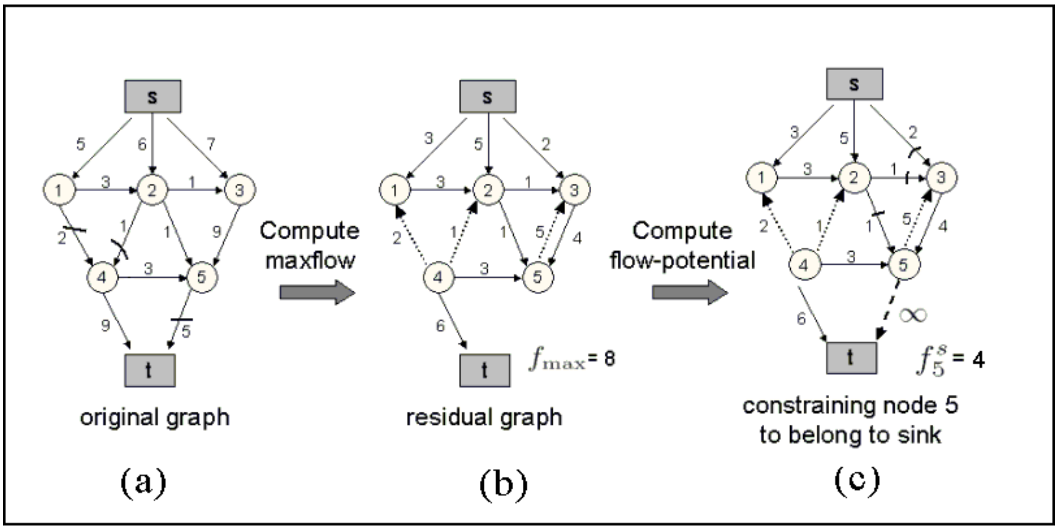


Fig. 3. *Computing min-marginals using graph cuts. In (a) we see the graph representing the original energy function. This is used to compute the minimum value of the energy* $\psi(\theta)$ *which is equal to the max-flow* $f_{\max} = 8$. *The residual graph obtained after the computation of max-flow is shown in (b). In (c) we show how the flow-potential* f_5^s *can be computed in the residual graph by adding an infinite capacity edge between it and the sink and computing the max-flow again. The addition of this new edge constrains node 5 to belong to sink side of the st-cut. A max-flow computation in the graph (c) yields* $f_5^s = 4$. *From theorem 1, we obtain the min–marginal* $\psi_{5;z} = 8 + 4 = 12$ *for variable* 5 *taking label* z. *Label* z *is represented by the source in this construction, i.e. T(z) = source(s). The dotted arrows in (b) and (c) correspond to edges in the residual graph whose residual capacities are due to flow passing through the edges in their opposite direction.*

the terminal corresponding to the label assignment $x_v = a$ is zero since the node is disconnected from the terminal $T(a)$ by the minimum cost st-cut [4].

We now address the case of the label assignment $x_v = b$. We already know from (25) that the min-marginal $\psi_{v;b}(\theta)$ corresponding to the non-optimal label b can be computed by finding the minimum value of the function E under the constraint $x_v = b$. This constraint can be enforced in our original graph (used for minimizing $E(x|\theta)$) by adding an edge with infinite weight between node v and the terminal corresponding to the label a, and then computing the st-mincut in this updated graph [5].

Adding an infinite weight edge between the node and the terminal $T(a)$ is equivalent to putting a hard constraint on the variable X_v to have the label

4 The amount of flow that can be transferred from the node to the terminal $T(a)$ in the residual graph is zero since otherwise it would contradict our assumption that the max-flow solution has been achieved.

5 In section 4.3 we shall explain how to solve the new st-mincut problem efficiently using the dynamic graph cut algorithm proposed in [16].

b[6]. It can be easily seen that the additional amount of flow that would now flow from the source to the sink is equal to the flow potential $f_v^{T(b)}$ of the node. Thus the value of the max-flow now becomes equal to $\psi(\theta) + f_v^{T(b)}$ where $T(b)$ is the terminal corresponding to the label b. The whole process is shown graphically in figure 3.

We have shown how minimizing an energy function with constraints on the value of a latent variable, is equivalent to computing the flow potentials of a node in the residual graph $G(f_{\max})$. Note that a similar procedure can be used to compute the min-marginal $\psi_{uv;ij}(\theta)$ by taking the projection and enforcing hard constraints on pairs of latent variables.

4.2 Extension to Multiple labels

As discussed earlier, graph cuts can also be used to minimize some energy functions where the size of the label set is more than 2 [14,23]. We refer to these functions as multi-label functions. Graphs representing the projections of such energy functions can be obtained by incorporating hard constraints in a fashion analogous to the one used for binary variables.

Ishikawa [14] proposed a graph construction to minimize multi-label functions with convex pairwise terms. In this construction, the MAP label of a variable is found by observing which data edge is cut. The value of a variable can be constrained or 'fixed' in this framework by making sure that the data edge corresponding to a particular label is cut. This can be enforced by adding edges of infinite capacity from the source and the sink to the tail and head node of the edge respectively as shown in figure 4. The cost of the st-mincut in this modified graph will give the exact value of the min-marginal energy associated with that particular labelling. It should be noted here that the method of Ishikawa [14] applies to a restricted class of energy functions. These do not include energies with non-convex priors (such at the Potts model) which are used in many computer vision applications. Measuring uncertainty in solutions of such energies is thus still an open problem.

4.3 Minimizing Energy Function Projections using Dynamic Graph Cuts

Having shown how min-marginals can be computed using graph cuts, we now explain how this can be done efficiently. As explained in the proof of Theo-

[6] The addition of an infinite weight edge can be realized by using an edge whose weight is more than the sum of all other edges incident on the node. This condition would make sure that the edge is not saturated during the max-flow computation.

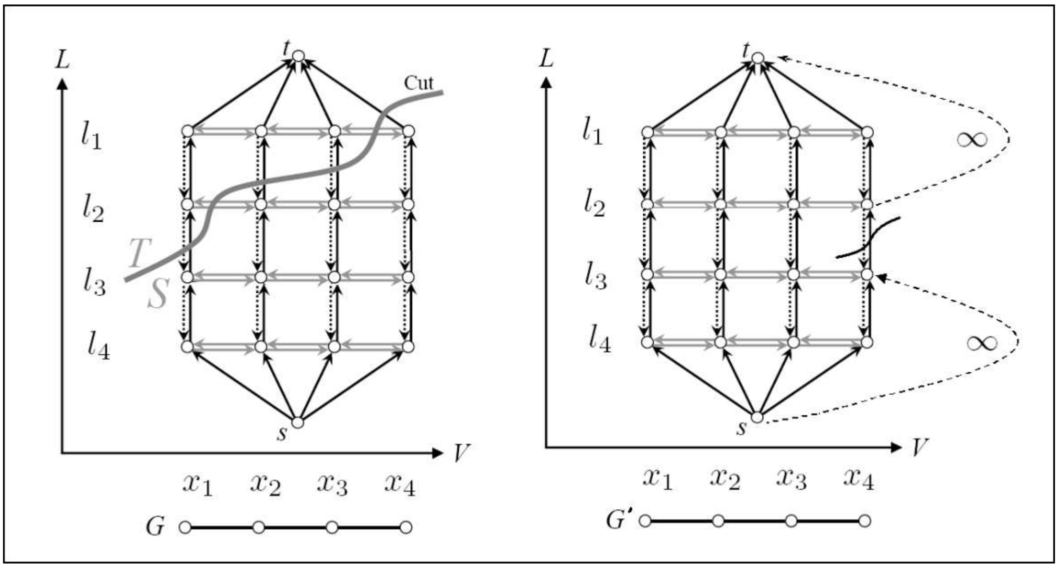


Fig. 4. *Graph construction for projections of energy functions involving multiple labels. The first graph G shows the graph construction proposed by Ishikawa [14] for minimizing energy functions representing* MRF*s involving latent variables which can take more than 2 labels. All the label sets $\mathcal{X}_v$, $v \in V$ consist of 4 labels namely l_1, l_2, l_3 and l_4. The* MAP *configuration of the* MRF *induced by the st-mincut is found by observing which data edges are cut (data edges are depicted as black arrows). Four of them are in the cut here (as seen in graph G), representing the assignments $x_1 = l_2$, $x_2 = l_3$, $x_3 = l_3$, and $x_4 = l_4$. The graph G' representing the projection $E' = E(x_1, x_2, x_3, l_2)$ can be obtained by inserting infinite capacity edges from the source and the sink to the tail and head node respectively of the edge representing the label l_2 for latent variable X_4.*

rem 1, we can compute min-marginals by minimizing projections of the energy function. It might be thought that such a process is extremely computationally expensive as a graph cut has to be computed for every min-marginal computation. We observed that when modifying the graph in order to minimize the projection E' of the energy function, the weights of only a few edges have to be changed[7]. This is illustrated in figure 3, where only one infinite capacity edge had to inserted in the graph. We have shown in [16] how the st-mincut can be recomputed rapidly for such minimal changes in the problem using the dynamic graph cut algorithm. Our proposed algorithm is given in Table 1. It should be noted that in the case of binary variables, the partially dynamic algorithm proposed in [3] can be used for efficient min-marginal computation since only the weights of t-edges in the graph have changed. However, this is not advisable since the method of [3] does not reuse search trees. In problems where only a few changes have been made to the graph, as in the case of min-marginal computation, the reuse of search trees results in much faster computation. This speed-up increases with an increase in the size of the graph.

[7] The exact number of edge weights that have to be changed is of the order of the number of variables whose value is being fixed for obtaining the projection.

Initialization

(1) Construct graph G for minimizing the MRF energy E.

(2) Compute the maximum s-t flow in the graph. This induces the residual graph G_r consisting of unsaturated edges.

Computing Min-marginal energies

(3) For computing each min-marginal, perform the following operations:

(a) Obtain the energy projection E' corresponding to the latent variable assignment. Let G' denote the graph used for minimizing E'.

(b) Use dynamic graph cut updates as given in [16] to make G_r consistent with G', thus obtaining the new graph G_r'.

(c) Compute the min-marginal by minimizing E' using the optimized dynamic st-mincut algorithm (which reuses search trees) on G_r'.

Table 1
Algorithm for computing min-marginal energies using dynamic graph cuts.

Please refer to [18] for a detailed discussion on the relative effect of reusing flow and reusing search trees on the max-flow re-computation time.

4.4 Limitations

The method proposed in this paper has the limitation that it can only be used to compute min-marginals for MRFs which are characterized by a graph representable energy function [8]. In their recent work on computing optical flow, Glocker et al.[13] proposed a method which can be used to compute min-marginals in non-submodular MRFs. However, their method was only able to produce approximate min-marginals unlike the exact min-marginals generated by our method.

4.5 Algorithmic Complexity and Experimental Evaluation

We now discuss the computational complexity of our algorithm, and report the time taken by it to compute min-marginals in random fields of different sizes.

[8] All quadratic submodular pseudo-boolean functions are graph representable i.e. they can be minimized by solving a st-mincut problem.

Variables	10^5	2×10^5	4×10^5	8×10^5
4-neighbour	1.8×10^4, 0.70	0.9×10^5, 1.34	3.7×10^5, 3.15	1.7×10^6, 8.21
8-neighbour	4.0×10^4, 1.53	2.8×10^5, 3.59	9.7×10^5, 8.50	4.1×10^6, 15.61

Table 2
Time taken for min-marginal computation. For a sequence of randomly generated MRFs of a particular size and neighbourhood system, a pair of times (in seconds) is given in each cell of the table. On the left is the total time taken to compute the min-marginals corresponding to all latent variable label assignments using the naive method of minimizing each energy function projection from scratch. The corresponding time taken by our dynamic graph cuts based method is on the right. The dynamic algorithm with tree-recycling was used for this experiment [16].

Let $\mathcal{Q}$ denote the set of all label assignments whose corresponding min-marginals have to be computed. Let us assume that the weights of all edges in the graph are integers [9] . In step 3(c) of the algorithm given in Table 1, the amount of flow computed is equal to the difference in the min-marginal $\psi_{v;j}(\theta)$ of the particular label assignment and the minimum energy $\psi(\theta)$. As each augmenting path increases the amount of flow by atleast one unit, the number of augmenting paths that can be found in the graph during the whole algorithm is bounded from above by:

$$U = \psi(\theta) + \sum_{q \in \mathcal{Q}} (\psi_q(\theta) - \psi(\theta)). \tag{27}$$

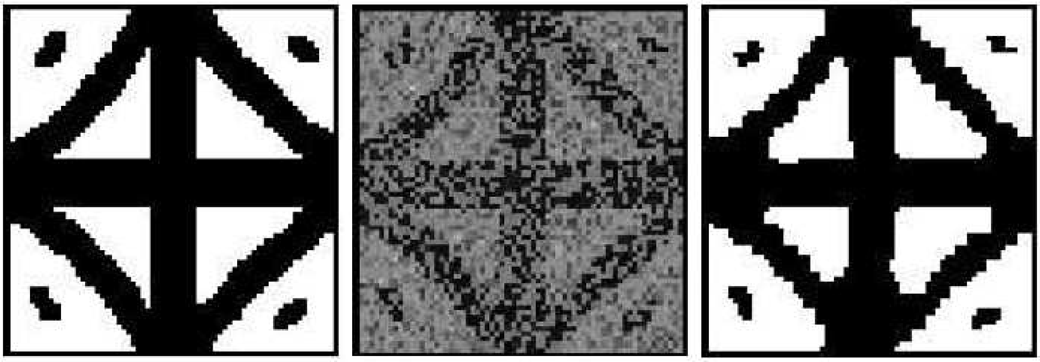
If we want to compute all min-marginals of a MRF involving binary random variables i.e.

$$\mathcal{Q} = \{(u; i) : u \in V, i \in \mathcal{X}_v\}, \text{and} \tag{28}$$

$$q_{max} = \max_{q \in \mathcal{Q}} (\psi_q(\theta) - \psi(\theta)), \tag{29}$$

the complexity of the above algorithm becomes $\mathrm{O}((\psi(\theta) + nq_{max})T(n, m))$, where $T(n, m)$ is the complexity of finding an augmenting path in the graph with n nodes and m edges and pushing flow through it. Although the worst case complexity $T(n, m)$ of the augmentation operation is $\mathrm{O}(m)$, we observe experimentally that using the dual search tree algorithm of [4], we can get a much better amortized time performance. The average running times of our algorithm for computing the min-marginals in some randomly generated MRFs of different sizes are given in Table 2. From the results, it can be seen that the time taken by our dynamic graph cuts based algorithm for computing all the min-marginals is substantially less than taken by the naive method of minimizing each energy function projection from scratch.

[9] If this is not the case, we can scale the weights of edges to make them integers.



(a) (b) (c)

Fig. 5. *Image Denoising. (a) Original noise free image. (b) Noisy image. (c) Result of solving the* CRF *used for modelling the image denoising problem [21].*

5 Applications of Min-marginals

Prior to our work, min-marginals have been rarely used in computer vision. This was primarily due to the fact that it is computationally expensive to compute min-marginals in MRFs having a large number of latent variables. Our new algorithm is able to handle large MRFs which opens up possibilities for many new applications. For instance, the MRF for the image segmentation experiment (see figure 7) had 2×10^5 binary latent variables. The time taken by our algorithm for computing all min-marginals in this MRF was 1.2 seconds. This is roughly four times the time taken for computing the MAP solution in the same MRF by solving a single st-mincut problem.

One of the motivations of our work was to obtain an uncertainty measure associated with the solutions of an energy minimization problem. Such uncertainty measures have been shown to be useful for solving a number of computer vision [12,13,25,26] and machine learning [8] problems. Min-marginals naturally encode the uncertainty of a labeling and their use has been successfully demonstrated in a number of problems[8,12,13]. In the next subsection, we provide another promising application of min-marginals.

5.1 Parameter Learning using Min-Marginals

Learning the parameters of a MRF (or CRF) from labelled training data is an important and challenging problem. The maximum likelihood (ML) estimation of parameters requires computing the partition function of the MRF. This operation is computationally intractable for general MRFs [10] .

[10] A feasible alternative is to use the pseudolikelihood method [1] which has been shown to produce decent results.

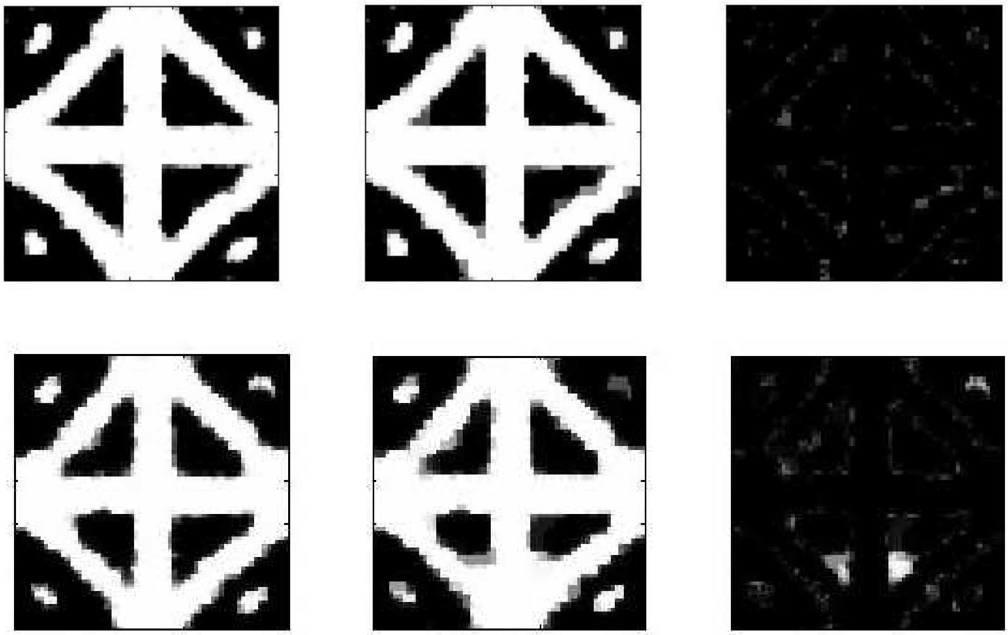


Fig. 6. *Comparing exact normalized max-marginals computed using our method with pseudo marginals obtained from* LBP. *The figure shows the pseudo marginal and normalized max-marginals for the binary image denoising problem under gaussian (row 1) and bimodal (row 2) noise models. The images in the first column show the pseudo marginals for all pixels to be assigned label 'foreground'. Brighter pixels are more likely to be labelled foreground. The images in column 2 show the normalized max-marginal for pixels being assigned label 'foreground'. The images in column 3 show the magnitude of the differences between the values of pseudo-marginals and normalized max-marginals. From the difference images, it can be seen that normalized max-marginals computed by our method are quite close to the pseudo-marginals obtained from LBP.*

Recently Kumar *et al.* [21] proposed an efficient method to learn the maximum likelihood parameter values of random fields. Their method comprised of approximating the gradients of the log likelihood function using an estimate of the marginals. They used their algorithm for learning the parameters of the MRF used for modelling the binary image denoising problem (see figure 5) In this section, we investigate the use of normalized max-marginals (σ) computed using our algorithm, in the parameter learning framework of [21]. First, we will briefly review the approximations used in [21] and then provide the results of using normalized max-marginals for learning the parameters of the CRF used for modelling the binary image denoising problem.

Let $\mathbf{y}^m$ and $\mathbf{x}^m$, $m = 1...M$ denote the observed data and labellings constituting the training set. The maximum likelihood estimates of the CRF parameters θ can be found by maximizing the joint distribution $P(\mathbf{x}, \mathbf{y}; \theta)$. Assuming

Inference Method	MAP	MPM	Learning Time (Sec)
PMA	2.73	2.51	1183
SPA	2.49	7.64	82
MMA	34.34	2.96	636
PMA-MM	2.80	2.48	1054

Table 3
Pixelwise classification errors (%) of parameters learned using various approximations to the gradient of the partition function.

$P(\mathbf{y};\theta)$ to be constant this can be done by maximizing the log likelihood

$$l(\theta) = \sum_{m=1}^{M} \log P(\mathbf{x}^m|\mathbf{y}^m, \theta). \tag{30}$$

using gradient ascent. The derivatives of the log likelihood are a function of the expectations $\langle x_i \rangle_{\theta;\mathbf{y}^m}$. If we have the true marginal distributions $P(x_i|\mathbf{y}, \theta)$ we can compute the exact expectations. For instance, if we know the marginal $P_i(x_i|\mathbf{y}_i, \theta)$ of the random variable X_i, the expectation $\langle x_i \rangle_{\theta;\mathbf{y}^m}$ under $\mathbf{y}$ and θ can be computed as

$$\langle x_i \rangle_{\theta;\mathbf{y}^m} = \sum_{x_i} x_i P_i(x_i|\mathbf{y}_i, \theta). \tag{31}$$

Since computing the exact marginals in a general MRF is infeasible, the exact values of the expectations cannot be computed. Kumar *et al.* [21] proposed a number of approximations for the expectation which we explain below.

(1) **Pseudo Marginal Approximation (PMA):** Pseudo-marginals obtained from loopy belief propagation are used instead of the true marginals for computing the expectations.
(2) **Saddle Point Approximation (SPA):** The label of the random variable in the MAP solution is taken as the expected value. This is equivalent to assuming that all the mass of the distribution $P_i(x_i|\mathbf{y}_i, \theta)$ is on the MAP label.
(3) **Maximum Marginal Approximation (MMA):** In this approximation the label having the maximum value under the marginal distribution is assumed to be the expected value.

We replace the pseudo-marginals in the PMA approximation with the normalized max-marginals obtained from our method. We will refer to this approximation as PMA-MM. The exact normalized max-marginals computed using our method and the pseudo marginals obtained from LBP in the CRF used for modelling the image denoising problem are compared in figure 6. Quantitative results comparing the performance of PMA-MM with that of other approximations are shown in table 3.

The purpose of these experiments was to demonstrate the use of min-marginals

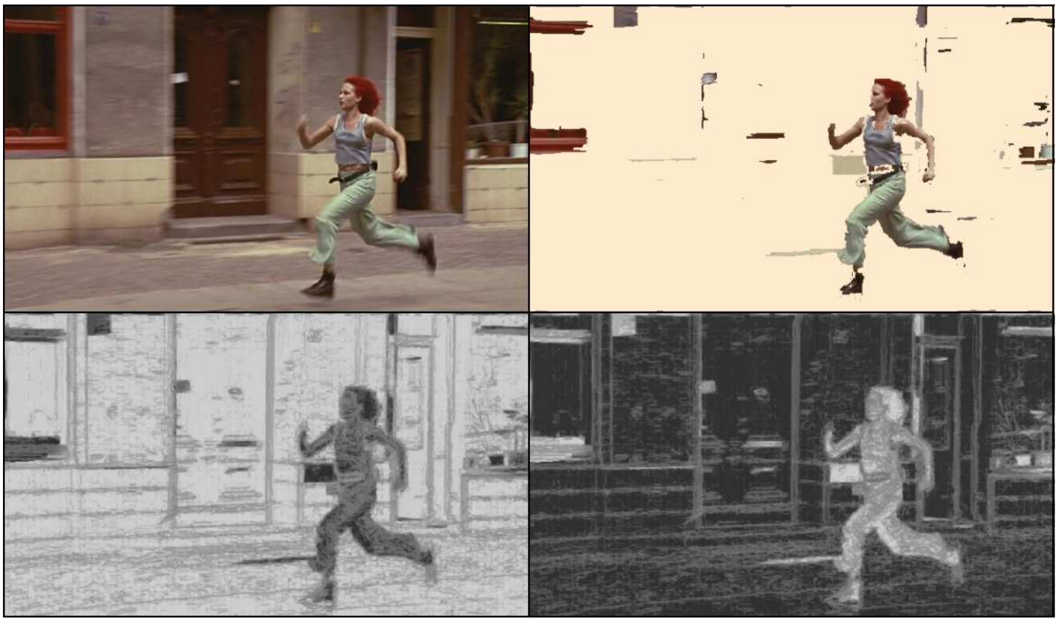


Fig. 7. *Image segmentation with max-marginal probabilities. The first image is a frame of the movie Run Lola Run. The second shows the binary foreground-background segmentation where the aim was to segment out the human. The third and fourth images shows the confidence values obtained by our algorithm for assigning pixels to be foreground and background respectively. In the image, the max-marginal probability is represented in terms of decreasing intensity of the pixel. Our algorithm took 1.2 seconds for computing the max-marginal probabilities for each latent variable label assignment. The time taken to compute the* MAP *solution was 0.3 seconds.*

in parameter learning by comparing their performance with that obtained using pseudo-marginals obtained by loopy belief propagation. We can see that for maximum posterior marginal (MPM) inference, the PMA-MM approximation yields the parameter estimates with the lowest error. That said, the difference between the accuracy of PMA-MM and PMA is small and there is no clear winner. However, the results do confirm that PMA-MM is a competitive approximation and should be investigated further.

5.2 Min-marginals as a confidence measure

We had shown in section 2 how min-marginals can be used to compute a confidence measure for any latent variable assignment in a MRF. Figure 7 shows the confidence values obtained for the MRF used for modeling the two label (foreground and background) image-segmentation problem as defined in [3]. Such confidence maps can be used for many vision applications. For instance, they could be used in interactive image segmentation to direct user interaction.

The result shown in figure 7 contains segmentation errors with both low and high uncertainty. The pixel labelling errors with low uncertainty (or high confidence) generally occur because the colour of the pixel was present in the appearance model of the incorrect segment label. On the other hand errors with higher uncertainty (or low confidence) appear because either the colour of the pixel was not modeled in the appearance model of any of the segments, or it was modeled in the appearance models of both segments. The confidence maps generated by our algorithm can help users isolate colours which lead to uncertain labellings. This knowledge can be exploited by users to build more accurate colour models for the different segments.

The confidence maps can also be used for many other applications. For instance, they can be used in coarse-to-fine techniques for efficient computation of low level vision problems. Here confidence maps could be used to isolate variables which have low confidence in the optimal label assignment. These variables can be solved at higher resolution to get a better solution. Another interesting application for min-marginals has been recently proposed in [13].

6 Discussion and Conclusions

In this paper we addressed the problem of computing the exact min-marginals for graphs of arbitrary topology in polynomial time. We proposed a novel algorithm based on dynamic graph cuts [16] that computes the min-marginals extremely efficiently. Our algorithm makes it feasible to compute exact min-marginals for MRFs with large number of latent variables. This opens up many new applications for min-marginals which were not feasible earlier. We have presented one such application in the form of parameter learning in MRFs used for modelling labelling problems such as image denoising.

7 Acknowledgements

We would like to thank Sanjiv Kumar for helpful discussions regarding the use of max-marginals for parameter learning. This work was supported by the EPSRC research grant GR/T21790/01(P) and the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

References

[1] P. Besag, Efficiency of pseudolikelihood estimation for simple gaussian fields., Biometrika. 64(3) (1977) 616–618.

[2] E. Boros, P. Hammer, Pseudo-boolean optimization., Discrete Applied Mathematics 123 (1-3) (2002) 155–225.

[3] Y. Boykov, M. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images, in: ICCV, 2001.

[4] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, PAMI 26 (9) (2004) 1124–1137.

[5] Y. Boykov, O. Veksler, R. Zabih, Markov random fields with efficient approximations, in: CVPR, 1998.

[6] Y. Chiang, R. Tamassia, Dynamic algorithms in computational geometry, in: IEEE Special Issue on Computational Geometry, vol. 80, 1992.

[7] P. Dawid, Applications of a general propagation algorithm for probabilistic expert systems, Statistics and Computing. 2 (1992) 25–36.

[8] J. Duchi, D. Tarlow, G. Elidan, D. Koller, Using combinatorial optimization within max-product belief propagation, in: NIPS, 2006.

[9] B. Flach, Strukturelle bilderkennung., Technical report, Fakultat Informatik, Technische Universitat Dresden, Germany. Habilitation thesis, in German, 2002.

[10] L. Ford, D. Fulkerson, Flows in Networks, Princeton University Press, Princeton, 1962.

[11] D. Freedman, P. Drineas, Energy minimization via graph cuts: Settling what is possible., in: CVPR (2), 2005.

[12] B. Glocker, Discrete optimization and uncertainty estimation in non-rigid image registration., Diploma Thesis, Ecole Centrale Paris, 2006.

[13] B. Glocker, N. Paragios, N. Komodakis, G. Tziritas, N. Navab, Optical flow estimation with uncertainties through dynamic mrfs., in: CVPR, 2008.

[14] H. Ishikawa, Exact optimization for markov random fields with convex priors, PAMI 25 (10) (2003) 1333–1336.

[15] S. Iwata, S. McCormick, M. Shigeno, A strongly polynomial cut canceling algorithm for the submodular flow problem, Lecture Notes in Computer Science 1610.
URL `citeseer.ist.psu.edu/iwata99strongly.html`

[16] P. Kohli, P. Torr, Efficiently solving dynamic markov random fields using graph cuts, in: ICCV, vol. II, 2005.

[17] P. Kohli, P. Torr, Measuring uncertainty in graph cut solutions: Efficiently computing min-marginal energies using dynamic graph cuts, in: ECCV, 2006.

[18] P. Kohli, P. H. S. Torr, Dynamic graph cuts for efficient inference in markov random fields, IEEE Trans. Pattern Anal. Mach. Intell. 29 (12) (2007) 2079–2088.

[19] V. Kolmogorov, Convergent tree-reweighted message passing for energy minimization, PAMI (2006).

[20] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts?, in: ECCV02, 2002.

[21] S. Kumar, J. August, M. Hebert, Exploiting inference for approximate parameter learning in discriminative fields: An empirical study., in: EMMCVPR, 2005.

[22] D. Nilsson, An efficient algorithm for finding the m most probable configurations in bayesian networks, Statistics and Computing 8 (2) (1998) 159–173. URL `citeseer.ist.psu.edu/nilsson98efficient.html`

[23] D. Schlesinger, B. Flach, Transforming an arbitrary minsum problem into a binary one, Tech. Rep. TUD-FI06-01, Dresden University of Technology (April 2006).

[24] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, C. Rother, A comparative study of energy minimization methods for markov random fields., in: ECCV (2), 2006.

[25] M. Taron, N. Paragios, M. Jolly, Registration with uncertainties and statistical modeling of shapes with variable metric kernels, IEEE Trans. Pattern Anal. Mach. Intell (2008).

[26] M. Taron, N. Paragios, M.-P. Jolly, Modelling shapes with uncertainties: Higher order polynomials, variable bandwidth kernels and non parametric density estimation, in: ICCV, 2005.

[27] M. J. Wainwright, T. Jaakkola, A. S. Willsky, Tree-based reparameterization for approximate inference on loopy graphs., in: NIPS, 2001.

[28] Y. Weiss, W. Freeman, On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs., IEEE Transactions on Information Theory (2001).

[29] C. Yanover, Y. Weiss, Finding the m most probable configurations in arbitrary graphical models, in: Advances in Neural Information Processing Systems 16, MIT Press, 2004.