Coarse-to-Fine Lifted MAP Inference in Computer Vision

Haroun Habeeb and **Ankit Anand** and **Mausam** and **Parag Singla**
Indian Institute of Technology Delhi
haroun7@gmail.com and {ankit.anand,mausam,parags}@cse.iitd.ac.in

Abstract

There is a vast body of theoretical research on *lifted inference* in probabilistic graphical models (PGMs). However, few demonstrations exist where lifting is applied in conjunction with top of the line applied algorithms. We pursue the applicability of lifted inference for computer vision (CV), with the insight that a globally optimal (MAP) labeling will likely have the same label for two symmetric pixels. The success of our approach lies in efficiently handling a distinct unary potential on every node (pixel), typical of CV applications. This allows us to lift the large class of algorithms that model a CV problem via PGM inference. We propose a generic template for coarse-to-fine (C2F) inference in CV, which progressively refines an initial coarsely lifted PGM for varying quality-time trade-offs. We demonstrate the performance of C2F inference by developing lifted versions of two near state-of-the-art CV algorithms for stereo vision and interactive image segmentation. We find that, against flat algorithms, the lifted versions have a much superior anytime performance, without any loss in final solution quality.

1 Introduction

Lifted inference in probabilistic graphical models (PGMs) refers to the set of the techniques that carry out inference over groups of random variables (or states) that behave similarly [Jha *et al.*, 2010; Kimmig *et al.*, 2015]. A vast body of theoretical work develops a variety of lifted inference techniques, both exact (e.g., [Poole, 2003; Braz *et al.*, 2005; Singla and Domingos, 2008; Kersting, 2012]) and approximate (e.g., [Singla *et al.*, 2014; Van den Broeck and Niepert, 2015]). Most of these works develop technical ideas applicable to generic subclasses of PGMs, and the accompanying experiments are aimed at providing first proofs of concepts. However, little work exists on transferring these ideas to the top *domain-specific* algorithms for real-world applications.

Algorithms for NLP, computational biology, and computer vision (CV) problems make heavy use of PGM machinery (e.g., [Blei *et al.*, 2003; Friedman, 2004; Szeliski *et al.*, 2008]). But, they also include significant problem-specific insights to get high performance. Barring a handful of exceptions [Jernite *et al.*, 2015; Nath and Domingos, 2016], lifted inference hasn't been applied directly to such algorithms.

We study the potential value of lifting to CV problems such as image denoising, stereo vision, and image segmentation. Most CV problems are structured output prediction tasks, typically assigning a label to each pixel. A large class of solutions are PGM-based: they define a Markov Random Field (MRF) that has each pixel as a node, with unary potential that depends on pixel value, and pairwise neighborhood potentials that favor similar labels to neighboring pixels.

We see three main challenges in applying existing lifted inference literature to these problems. First, most existing algorithms focus on computing marginals [Singla and Domingos, 2008; Kersting *et al.*, 2009; Gogate and Domingos, 2011; Niepert, 2012; Anand *et al.*, 2016; 2017] instead of MAP inference. Second, among the algorithms performing lifted MAP [Noessner *et al.*, 2013; Mladenov *et al.*, 2014; Sarkhel *et al.*, 2014; Mittal *et al.*, 2014], many of the algorithms focus on exact lifting. This breaks the kind of symmetries we need to compute since different pixels may not have exact same neighborhood. Third, the few algorithms that perform approximate lifting for MAP, e.g. [Sarkhel *et al.*, 2015], can't handle a distinct unary potential on every node. This is essential for our application since image pixels take ordinal values in three channels.

In response, we develop an approximate lifted MAP inference algorithm which can effectively handle unary potentials. We initialize our algorithm by merging together pixels having the same order of top-k labels based on the unary potential values. We then adapt an existing symmetry finding algorithm [Kersting *et al.*, 2009] to discover groupings which also have similar neighborhoods. We refer to our groupings as *lifted pixels*. We impose the constraint that all pixels in a lifted pixel must be assigned the same label. Our approximate lifting reduces the model size drastically leading to significant time savings. Unfortunately, such approximate lifting could adversely impact solution quality. However, we vary the degree of approximation in symmetry finding to output a *sequence* of coarse-to-fine models with varying quality-time trade-offs. By switching between such models, we develop a coarse-to-fine (C2F) inference procedure applicable to many CV problems.

We formalize these ideas in a novel template for using

lifted inference in CV. We test C2F lifted inference on two problems: stereo matching and image segmentation. We start with one of the best MRF-based solvers each for both problems – neither of these are vanilla MRF solvers. Mozerov & Weijer [2015] use a *two-way* energy minimization to effectively handle occluded regions in stereo matching. Cooperative cuts [Kohli *et al.*, 2013] for image segmentation use concave functions over a predefined set of pixel pairs to correctly segment images with sharp edges. We implement C2F inference on top of both these algorithms and find that C2F versions have a strong anytime behavior – given any amount of inference time, they output a much higher quality (and are never worse) than their unlifted counterparts, and don't suffer any loss in the final quality. Overall, our contributions are:

1. We present an approximate lifted MAP algorithm that can efficiently handle a large number of distinct unary potentials.
2. We develop a novel template for applying lifted inference in structured prediction tasks in CV. We provide methods that output progressively finer approx. symmetries, leading to a C2F lifted inference procedure.
3. We implement C2F inference over a near state-of-the-art stereo matching algorithm, and one of the best MRF-based image segmentation algorithms. We release our implementation for wider use by the community.[1]
4. We find that C2F has a much superior anytime behavior. For stereo matching it achieves 60% better quality on average in time-constrained settings. For image segmentation C2F reaches convergence in 33% less time.

2 Background

2.1 Computer Vision Problems as MRFs

Most computer vision problems are structured output prediction problems and their PGM-based solutions often follow similar formulations. They cast the tasks into the problem of finding the lowest energy assignment over grid-structured MRFs (denoted by $G = (\mathcal{X}, \gamma)$). The random variables in these MRFs are the set of pixels $\mathcal{X}$ in the input image. Given a set of labels $L : \{1, 2, \ldots, |L|\}$, the task of structured output prediction is to label each pixel X with a label from L. The MRFs have two kinds of potentials (γ) – unary and higher-order. Unary potentials are defined over each individual pixel, and usually incorporate pixel intensity, color, and other pixel features. Higher order potentials operate over cliques (pairs or more) of neighboring pixels and typically express some form of *spatial homophily* – "neighboring pixels are more likely to have similar labels." While the general PGM structure of various tasks are similar, the specific potential tables and label spaces are task-dependent.

The goal is to find the MAP assignment over this MRF, which is equivalent to energy minimization (by defining energy as negative log of potentials). We denote the negative log of unary potentials by ϕ, and that of higher-order potentials by ψ.[2] Thus, energy of a complete assignment $\mathbf{x} \in L^{|\mathcal{X}|}$ can be defined as:

$$E(\mathbf{x}) = \sum_{i \in 1..|\mathcal{X}|} \phi(x_i) + \sum_j \psi_j(\hat{x}_j) \tag{1}$$

Here $\hat{x}_j$ denotes the assignment $\mathbf{x}$ restricted to the set of variables in the potential ψ_j. And the output of the algorithm is the assignment $\mathbf{x}_{\text{MAP}}$:

$$\mathbf{x}_{\text{MAP}} = \arg\min_{\mathbf{x} \in L^{|\mathcal{X}|}} E(\mathbf{x}) \tag{2}$$

The problem is in general intractable. Efficient approximations exploit special characteristics of potentials like submodularity [Jegelka and Bilmes, 2011], or use variants of graph cut or loopy belief propagation [Boykov *et al.*, 2001; Freeman *et al.*, 2000].

2.2 Symmetries in Graphical Models

Lifting an algorithm often requires computing a set of symmetries that can be exploited by that algorithm. For PGMs, two popular methods for symmetry computation are color passing for computing symmetries of variables [Kersting *et al.*, 2009], and graph isomorphism for symmetries of states [Niepert, 2012; Bui *et al.*, 2013]. Since our work is based on color passing, we explain it in more detail.

Color passing for an MRF operates over a colored bipartite graph containing nodes for all variables and potentials, and each node is assigned a color. The graph is initialized as follows: all variables nodes get a common color; all potential nodes with exactly same potential tables are assigned a unique color. Now, in an iterative color passing scheme, in each iteration, each variable node passes its color to all neighboring potential nodes. The potential nodes store incoming color signatures in a vector, append their own color to it, and send the vector back to variable nodes. The variable nodes stack these incoming vectors. New colors are assigned to each node based on the set of incoming messages such that two nodes with same messages are assigned the same unique color. This process is repeated until convergence, i.e., no further change in colors.

A coloring of the bipartite graph defines a partition of variable nodes such that all nodes of the same color form a partition element. Each iteration of color passing creates successively finer partitions, since two variable nodes, once assigned different colors, can never get the same color.

3 Lifted Computer Vision Framework

In this section, we will describe our generic template which can be used to lift a large class of vision applications including those in stereo, segmentation etc. Our template can be seen as transforming the original problem space to a reduced problem space over which the original inference algorithm can now be applied much more efficiently. Specifically, our description in this section is entirely *algorithm independent*.

We will focus on MAP inference which is the inference task of choice for most vision applications (refer Section 2).

The key insight in our formulation is based on the realization that pixels which are involved in the same (or similar)

[1] *https://github.com/dair-iitd/c2fi4cv/*

[2] In the interest of readability, we say 'potential' to mean 'negative log of potential' in the rest of the paper.

kinds of unary and higher order potentials, and have the same (or similar) neighborhoods, are likely to have the same MAP value. Therefore, if somehow we could discover such sets of pixels a priori, we could explicitly enforce these pixels to have the same value while searching for the solution, substantially reducing the problem size and still preserving the optimal MAP assignment(s). Since in general doing this exactly may lead to a degenerate network, we do it approximately. Hence trading-off speed for marginal loss in solution quality. The loss in solution quality is offset by resorting to coarse-to-fine inference where we start with a crude approximation, and gradually make it finer, to guarantee optimality at the end while still obtaining significant gains. We next describe the details of our approach.

3.1 Obtaining a Reduced Problem

Consider an energy minimization problem over a PGM $G = (\mathcal{X}, \gamma)$. Let $L = \{1, 2, \cdots, |L|\}$ denote the set of labels over which variables in the set $\mathcal{X}$ can vary. Let $\mathcal{Y}^P = \{Y_1^P, Y_2^P, \cdots, Y_r^P\}$ denote a partition of $\mathcal{X}$ into r disjoint subsets, i.e., $\forall k$, $Y_k^P \subseteq \mathcal{X}$, $Y_{k_1}^P \cap Y_{k_2}^P = \emptyset$ when $k_1 \neq k_2$, and $\cup_k Y_k^P = \mathcal{X}$. We refer to each Y_k^P as a partition element. Correspondingly, let us define $\mathcal{Y} = \{Y_1, Y_2, \cdots, Y_r\}$ as a set of partition variables, where there is a one to one correspondence between partition elements and the partition variables and each partition variable Y_k takes values in the set L. Let $part(X_i)$ denote the partition element to which X_i belongs. Let $\hat{X}_j \subseteq \mathcal{X}$ denote a subset of variables. We say that a partition element Y_k^P is *r*epresented in the set $\hat{X}_j$ if $\exists X_i \in \hat{X}_j$ s.t. $part(X_j) = Y_k$.

Given a subset of variables $\hat{X}_j$, let $\gamma_j(\hat{X}_j)$ be a potential defined over $\hat{X}_j$. Let $\hat{x}_j$ denote an assignment to variables in the set $\hat{X}_j$. Let $\hat{x}_j.elem(i)$ denote the value taken by a variable X_i in $\hat{X}_j$. We say that an assignment $\hat{X}_j = \hat{x}_j$ *r*espects a partition $\mathcal{Y}^P$ if the variables in $\hat{X}_j$ belonging to the same partition element have the same label in $\hat{x}_j$, i.e., $part(X_i) = part(X_{i'}) \Rightarrow \hat{x}_j.elem(i) = \hat{x}_j.elem(i'), \forall X_i, X_{i'} \in \hat{X}_j$. Next, we introduce the notion of a reduced potential.

Definition 3.1 *Let $\mathcal{X}$ be a set of variables and let $\mathcal{Y}^P$ denote its partition. Given the potential $\gamma_j(\hat{X}_j)$, the reduced potential Γ_j is defined to be the restriction of $\gamma_j(\hat{X}_j)$ to those labeling assignments of $\hat{X}_j$ which respect the partition $\mathcal{Y}^P$. Equivalently, we can define the reduced potential $\Gamma_j(\hat{Y}_j)$ over the set of partition variables $\hat{Y}_j$ which are represented in the set $\hat{X}_j$.*

For example, consider a potential $\gamma(X_1, X_2, X_3)$ defined over three Boolean variables. The table for γ would have 8 entries. Consider the partition $\mathcal{Y}^P = \{Y_1^P, Y_2^P\}$ where $Y_1^P = \{X_1, X_2\}$ and $Y_2^P = \{X_3\}$. Then, the reduced potential Γ is the restriction of γ to those rows in the table where $X_1 = X_2$. Hence Γ has four rows in its table and equivalently can be thought of defining a potential over the 4 possible combinations of Y_1 and Y_2 variables. We are now ready to define a reduced graphical model.

Definition 3.2 *Let $G = (\mathcal{X}, \gamma)$ represent a PGM. Given a partition $\mathcal{Y}^P$ of $\mathcal{X}$, the reduced graphical model $\mathcal{G}(\mathcal{Y}, \Gamma)$ is the graphical model defined over the set of partition variables $\mathcal{Y}$ such that every potential $\gamma_j \in \gamma$ in G is replaced by the corresponding reduced potential $\Gamma_j \in \Gamma$ in $\mathcal{G}$.*

Let $E(\mathbf{x})$ and $\mathcal{E}(\mathbf{y})$ denote the energies of the states $\mathbf{x}$ and $\mathbf{y}$ in G and $\mathcal{G}$, respectively. The following theorem relates the energies of the states in the two graphical models.

Theorem 3.1 *For every assignment $\mathbf{y}$ of $\mathcal{Y}$ in $\mathcal{G}$, there is a corresponding assignment $\mathbf{x}$ of $\mathcal{X}$ such that $\mathcal{E}(\mathbf{y}) = E(\mathbf{x})$.*

The theorem can be proved by noting that each potential $\Gamma_j(\hat{Y}_j)$ in $\mathcal{G}$ was obtained by restricting the original potential $\gamma_j(\hat{X}_j)$ to those assignments where variables in X_j belonging to the same partition took the same label. Since this correspondence is true for every potential in the reduced set, to obtain the desired state $\mathbf{x}$, for every variable $X_i \in \mathcal{X}$ we simply assign it the label of its partition in $\mathbf{y}$.

Corollary 3.1 *Let $\mathbf{x}_{\mathrm{MAP}}$ and $\mathbf{y}_{\mathrm{MAP}}$ be the MAP states (i.e. having the minimum energy) for G and $\mathcal{G}$, respectively. Then, $\mathcal{E}(\mathbf{y}_{\mathrm{MAP}}) \geq E(\mathbf{x}_{\mathrm{MAP}})$.*

The process of reduction can be seen as curtailing the entire search space to those assignments where variables in the same partition take the same label. A reduction in the problem space will lead to computational gains but might result in loss of solution quality, where the solution quality can be captured by the difference between $\mathcal{E}(\mathbf{y}_{\mathrm{MAP}})$ and $E(\mathbf{x}_{\mathrm{MAP}})$. Therefore, we need to trade-off the balance between the two.

Intuitively, a good problem reduction will keep those variables in the same partition which are likely to have the same value in the optimal assignment for the original problem. How do we find such variables without actually solving the inference task? We will describe one such technique in Section 3.3.

There is another perspective. Instead of solving one reduced problem, we can instead work with a series of reduced problems which successively get closer to the optimal solution. The initial reductions are coarser and far from optimal, but can be solved efficiently to quickly reach in the region where the solution lies. The successive iterations can then refine the solution iteratively getting closer to the optimal. This leads us to the coarse-to-fine inference described next.

3.2 Coarse to Fine Inference

We will define a framework for C2F (coarse-to-fine) inference so that we maintain the computational advantage while still preserving optimality. In the following, for ease of notation, we will drop the superscript P in $\mathcal{Y}^p$ to denote the partition of $\mathcal{X}$. Therefore, $\mathcal{Y}$ will refer to both the partition as well as the set of partition variables. Before we describe our algorithm, let us start with some definitions.

Definition 3.3 *Let $\mathcal{Y}$ and $\mathcal{Y}'$ be two partitions of $\mathcal{X}$. We say that $\mathcal{Y}$ is coarser than $\mathcal{Y}'$, denoted as $\mathcal{Y} \preceq \mathcal{Y}'$, if $\forall y' \in \mathcal{Y}' \exists y \in \mathcal{Y}$ such that $y' \subseteq y$. We equivalently say that $\mathcal{Y}'$ is finer than $\mathcal{Y}$.*

It is easy to see that $\mathcal{X}$ defines a partition of itself which is the finest among all partitions, i.e., $\forall \mathcal{Y}$ such that $\mathcal{Y}$ is a partition

of $\mathcal{X}$, $\mathcal{Y} \preceq \mathcal{X}$. We also refer it to as the degenerate partition. For ease of notation, we will denote the finest partition by $\mathcal{Y}^*$ (same as $\mathcal{X}$). We will refer to the corresponding PGM as $\mathcal{G}^*$ (same as G). Next, we state a theorem which relates two partitions with each other.

Lemma 1 *Let $\mathcal{Y}$ and $\mathcal{Y}'$ be two partitions of $\mathcal{X}$ such that $\mathcal{Y} \preceq \mathcal{Y}'$. Then $\mathcal{Y}'$ can be seen as a partition of the set $\mathcal{Y}$.*

The proof of this lemma is straightforward and is omitted due to lack of space. Consider a set $\mathbf{Y}$ of coarse to fine partitions given as $\mathcal{Y}^0 \preceq \mathcal{Y}^1, \cdots, \preceq, \mathcal{Y}^t, \preceq, \cdots, \mathcal{Y}^*$. Let $\mathcal{G}^t, \mathcal{E}^t, \mathbf{y}^t_{MAP}$ respectively denote the reduced problem, energy function and MAP assignment for the partition $\mathcal{Y}^t$. Using Lemma 1, $\mathcal{Y}^{t+1}$ is a partition of $\mathcal{Y}^t$. Then, using Theorem 3.1, we have for every assignment $\mathbf{y^t}$ to variables in $\mathcal{Y}^t$, there is an assignment $\mathbf{y^{t+1}}$ to variables in $\mathcal{Y}^{t+1}$ such that $\mathcal{E}^t(\mathbf{y^t}) = \mathcal{E}^{t+1}(\mathbf{y^{t+1}})$. Also, using Corollary 3.1, we have $\forall t\, \mathcal{E}^t(\mathbf{y^t_{\text{MAP}}}) \geq \mathcal{E}^{t+1}(\mathbf{y^{t+1}_{\text{MAP}}})$. Together, these two statements imply that starting from the coarsest partition, we can gradually keep on improving the solution as we move to finer partitions.

Our C2F set-up assumes an iterative MAP inference algorithm A which has the anytime property i.e., can produce solutions of increasing quality with time. C2F Function (see Algorithm 1) takes 3 inputs: a set of C2F partitions $\mathbf{Y}$, inference algorithm A, and a stopping criteria $\mathcal{C}$. The algorithm A in turn takes three inputs: PGM $\mathcal{G}^t$, starting_assignment $\mathbf{y}^t$, stopping criteria $\mathcal{C}$. A outputs an approximation to the MAP solution once the stopping criteria $\mathcal{C}$ is met. Starting with the coarsest partition ($t = 0$ in line 2), a start state is picked for the coarsest problem to be solved (line 3). In each iteration (line 4), C2F finds the MAP estimate for the current problem ($\mathcal{G}^t$) using algorithm A (line 5). This solution is then mapped to a same energy solution of the next finer partition (line 6) which becomes the starting state for the next run of A. The solution is thus successively refined in each iteration. The process is repeated until we reach the finest level of partition. In the end, A is run on the finest partition and the resultant solution is output (lines 9,10). Since the last partition in the set is the original problem $\mathcal{G}^*$, optimality with respect to A is guaranteed.

Next, we describe how to use the color passing algorithm (Section 2) to get a series of partitions which get successively finer. Our C2F algorithm can then be applied on this set of partitions to get anytime solutions of high quality while being computationally efficient.

Algorithm 1 Coarse-to-Fine Lifted MAP Algorithm

1: **C2F_Lifted_MAP**(C2F Partitions $\mathbf{Y}$, Algo A,Criteria $\mathcal{C}$)
2: $t = 0$; $T = |\mathbf{Y}|$;
3: $\mathbf{y}^t$ = getInitState($\mathcal{G}^t$);
4: **While** ($t < T$);
5: $\quad \mathbf{y}^t_{\text{MAP}} = A(\mathcal{G}^t, y^t, \mathcal{C})$;
6: $\quad \mathbf{y}^{t+1} = getEquivAssignment(\mathcal{Y}^t, \mathcal{Y}^{t+1}, \mathbf{y}^t_{\text{MAP}})$;
7: $\quad t = t + 1$;
8: **END_While**
9: $\mathbf{y}^T_{\text{MAP}} = A(\mathcal{G}^T, y^T, \mathcal{C})$;
10: **return** y^T_{MAP}

3.3 C2F Partitioning for Computer Vision

We now adapt the general color passing algorithm to MRFs for CV problems. Unfortunately, unary potentials make color passing highly ineffective. Different pixels have different RGB values and intensities, leading to almost every pixel getting a different unary potential. Naive application of color passing splits almost all variables into their own partitions, and lifting offers little value.

A natural approximation is to define a threshold, such that two unary potentials within that threshold be initialized with the same color. Our experiments show limited success with this scheme because because two pixels may have the same label even when their actual unary potentials are very different. What is more important is relative importance given to each label than the actual potential value.

In response, we adapt color passing for CV by initializing it as before, but with one key change: we initialize two unary potential nodes with the same color if their lowest energy labels have the same order for the top N_L labels (we call this unary split threshold). Experiments reveal that this approximation leads to effective partitions for lifted inference.

Finally, we can easily construct a sequence of coarse-to-fine partitions in the natural course of color passing's execution – every iteration of color passing creates a finer partition. Moreover, as an alternative approach, we may also increase N_L. In our implementations, we intersperse the two, i.e., before every next step we pick one of two choices: either, we run another iteration of color passing; or, we increase N_L by one, and split each variable partition based on the N_L^{th} lowest energy labels of its constituent variables.

We parameterize $CP(N_L, N_{iter})$ to denote the partition from the current state of color passing, which has been run till N_{iter} iterations and unary split threshold is N_L. It is easy to prove that another iteration of color passing or splitting by increasing N_L as above leads to a finer partition. I.e., $CP(N_L, N_{iter}) \preceq CP(N_L + 1, N_{iter})$ and $CP(N_L, N_{iter}) \preceq CP(N_L, N_{iter} + 1)$. We refer to each element of a partition of variables as a *lifted pixel*, since it is a subset of pixels.

4 Lifted Inference for Stereo Matching

We first demonstrate the value of lifted inference in the context of stereo matching [Scharstein and Szeliski, 2002]. It aims to find pixel correspondences in a set of images of the same scene, which can be used to further estimate the 3D scene. Formally, two images I^l and I^r corresponding to images of the scene from a left camera and a right camera are taken such that both cameras are at same horizontal level. The goal is to compute a disparity labeling D^l for every pixel $X = (a, b)$ such that $I^l[a][b]$ corresponds to $I^r[a - D^l[a][b]][b]$. We build a lifted version of TSGO [Mozerov and van de Weijer, 2015], as it is MRF-based and ranks 2^{nd} on the Middlebury Stereo Evaluation Version 2 leaderboard.[3]

Background on TSGO: TSGO treats stereo matching as a two-step energy minimization, where the first step is on a

[3] *http://vision.middlebury.edu/stereo/eval/*

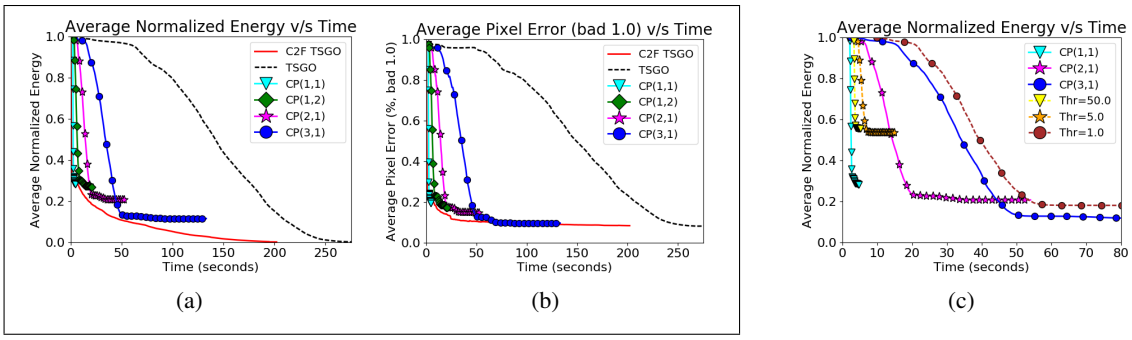
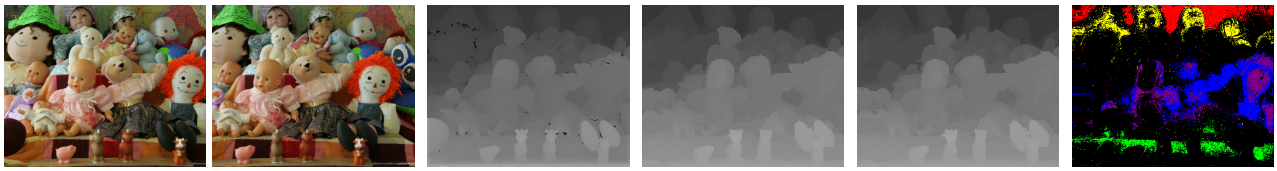


Figure 1: **(a)** Average (normalized) energy vs. inference time **(b)** Average pixel error vs. time. C2F TSGO achieves roughly 60% reduction in time for reaching the optima. It has best anytime performance compared to vanilla TSGO and static lifted versions. **(c)** Average (normalized) energy vs. time for different thresholding values and CP partitions. Plots with the same marker have MRFs of similar sizes.



(a) (b) (c) (d) (e)

Figure 2: Qualitative results for Doll image at convergence. C2F-TSGO is similar to base TSGO.**(a)** Left and Right Images **(b)** Ground Truth **(c)** Disparity Map by TSGO **(d)** Disparity Map by C2F TSGO **(e)** Each colored region (other than black) is one among the 10 largest partition elements from CP(1,1). Each color represents one partition element. Partition elements form non-contiguous regions

fully connected MRF with pairwise potentials and the second is on a conventional locally connected MRF. Lack of space precludes a detailed description of the first step. At the high level, TSGO runs one iteration of message passing on fully connected MRF, computes marginals of each pixel X, which act as unary potentials $\phi(X)$ for the MRF of second step.

The pairwise potential ψ used in step two is $\psi(X, X') = w(X, X')\varphi(X, X')$, where $\varphi(X, X')$ is a truncated linear function of $\|X - X'\|$, and $w(X, X')$ takes one of three distinct values depending on color difference between pixels. The MAP assignment $\mathbf{x}_{\text{MAP}}$ computes the lowest energy assignment of disparities D^l for every pixel for this MRF.

Lifted TSGO: Since step two is costlier, we build its lifted version as discussed in previous section. For color passing, two unary potential nodes are initialized with the same color if their lowest energy labels exactly match ($N_L = 1$). Other initializations are consistent with original color passing for general MRFs. A sequence of coarse-to-fine models is outputted as per Section 3.3. C2F TSGO uses outputs from the sequence $CP(1,1)$, $CP(2,1)$, $CP(3,1)$ and then refines to the original MRF. Model refinement is triggered whenever energy hasn't decreased in the last four iterations of alpha expansion (this becomes the stopping criteria $\mathcal{C}$ in Algorithm 1).

Experiments: Our experiments build on top of the existing TSGO implementation[4], but we change the minimization algorithm in step two to alpha expansion fusion [Lempitsky *et al.*, 2010] from OpenGM2 library [Andres *et al.*, 2010; Kappes *et al.*, 2015], as it improves the speed of the base implementation. We use the benchmark Middlebury Stereo datasets of 2003, 2005 and 2006 [Scharstein and Szeliski, 2003; Hirschmuller and Scharstein, 2007]. For the 2003 dataset, quarter-size images are used and for others, third-size images are used. The label space is of size 85 (85 distinct disparity labels).

We compare our coarse-to-fine TSGO (using $CP(N_L, N_{iter})$ partitions) against vanilla TSGO. Figures 1(a,b) show the aggregate plots of energy (and error) vs. time. We observe that C2F TSGO reaches the same optima as TSGO, but in less than half the time. It has a much superior anytime performance – if inference time is given as a deadline, C2F TSGO obtains 59.59% less error on average over randomly sampled deadlines. We also eyeball the outputs of C2F TSGO and TSGO and find them to be visually similar. Figure 2 shows a sample qualitative comparison. Figure 2(e) shows five of the ten largest partition elements in the partition from $CP(1,1)$. Clearly, the partition elements formed are not contiguous, and seem to capture variables that are likely to get the same assignment. This underscores the value of our lifting framework for CV problems.

We also compare our $CP(N_L, N_{iter})$ partitioning strategy with threshold partitioning discussed in Section 3.3. We merge two pixels in thresholding scheme if the L1-norm distance of their unary potentials is less than a threshold. For each partition induced by our approach, we find a value of threshold that has roughly the same number of lifted pixels. Figure 1(c) shows that partitions based on $CP(1,1)$ and $CP(3,1)$ converges to a much lower energy quickly compared to the corresponding threshold values ($Thr = 50$ and $Thr = 1$ respectively). For $CP(2,1)$, convergence is slower compared to corresponding threshold ($Thr = 5$) but eventually $CP(2,1)$ has significantly better quality.

[4] *http://www.cvc.uab.es/∼mozerov/Stereo/*

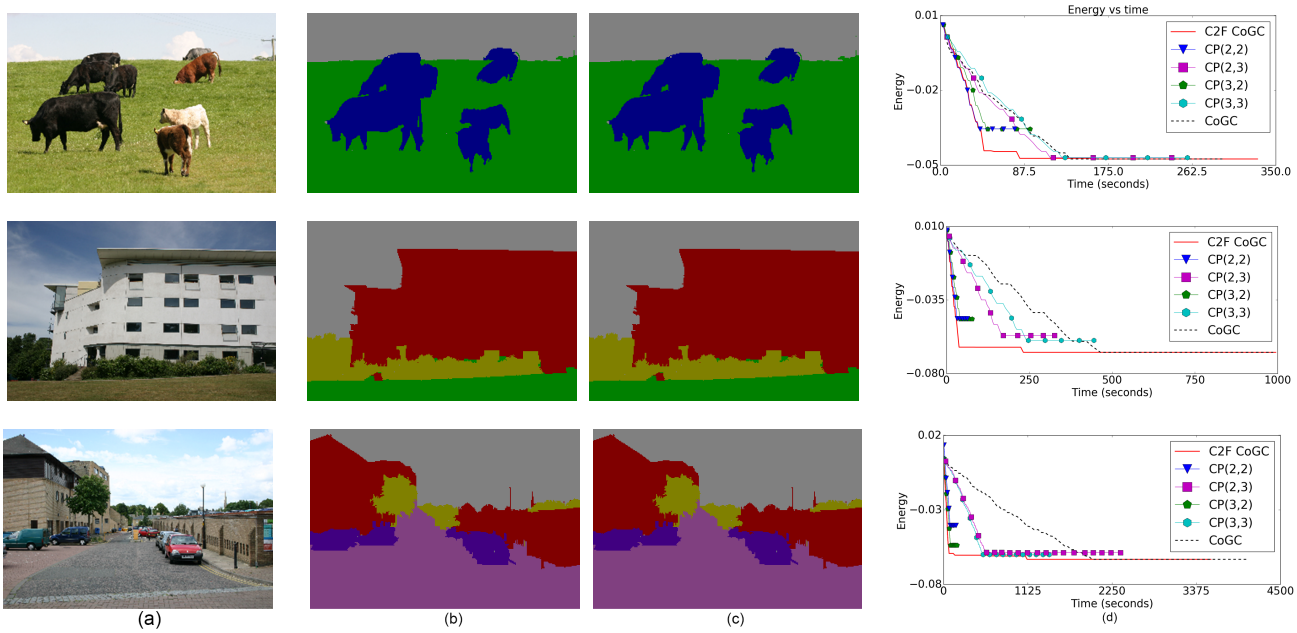


Figure 3: **(a-c)** Qualitative Results for Segmentation. C2F has quality similar to CoGC algorithm **(a)** Original Image **(b)** Segmentation by CoGC **(c)** Segmentation by C2F CoGC **(d)** C2F CoGC has lower energy compared to CoGC and other lifted variant at all times

5 Lifted Inference for Image Segmentation

We now demonstrate the general nature of our lifted CV framework by applying it to a second task. We choose multi-label interactive image segmentation, where the goal is to segment an image I based on a seed labeling (true labels for a few pixels) provided as input. Like many other CV problems, this also has an MRF-based solution, with the best label-assignment generally obtained by MAP inference using graph cuts or loopy belief propagation [Boykov *et al.*, 2001; Szeliski *et al.*, 2008].

However, MRFs with only pairwise potentials are known to suffer from *short-boundary bias* – they prefer segmentations with shorter boundaries, because pairwise potentials penalize every pair of boundary pixels. This leads to incorrect labeling for sharp edge objects. Kohli *et al.* [2013] use CoGC, cooperative graph cuts [Jegelka and Bilmes, 2011], to develop one of the best MRF-based solvers that overcome this bias.

Background on CoGC: Traditional MRFs linearly penalize the number of label discontinuities at edges (boundary pixel pairs), but CoGC penalizes the number of *types* of label discontinuities through the use of a concave energy function over groups of ordered edges. It first clusters all edges on the basis of color differences, and later applies a concave function separately over the number of times a specific discontinuity type is present in each edge group $g \in \mathbb{G}$. Their carefully engineered CoGC energy function is as follows:

$$E(\mathbf{x}) = \sum_{i=1}^{|\mathcal{X}|} \phi_i(x_i) + \sum_{g \in \mathbb{G}} \sum_{l \in L} F\left(\sum_{(x,x') \in g} w(x,x').\mathbb{I}(x = l, x' \neq l) \right)$$

where unary potentials ϕ depend on colors of seed pixels, F is a concave function, $\mathbb{I}$ the indicator function, and $w(x,x')$ depends on the color difference between x, x'. Intuitively, F collects all edges with similar discontinuities and penalizes them sub-linearly, thus reducing the short-boundary bias in the model. The usage of a concave function makes the MRF higher order with cliques over edge groups. However, the model is shown to reduce to a pairwise hierarchical MRF through the addition of auxiliary variables.

Lifted CoGC: CoGC is lifted using the framework of Section 3, with one additional change. We cluster edge groups using color difference *and* the position of the edge. Edge groups that are formed only on the basis of color difference make the error of grouping different segment's boundaries into a single group. For e.g., it erroneously cluster boundaries between white cow and grass, and sky and grass together in the top image in Figure 3.

Coarse-to-fine partitions are obtained by the method described in Section 3.3. C2F CoGC uses outputs from the sequence $CP(\lceil \frac{L}{2} \rceil, 2), CP(\lceil \frac{L}{2} \rceil, 3)$ before refining to the original MRF. Model refinement is triggered if energy has not reduced over the last $|L|$ iterations.

Experiments: Our experiments use the implementation of Cooperative Graph Cuts as provided by [Kohli *et al.*, 2013].[5] Energy minimization is performed using alpha expansion [Boykov *et al.*, 2001]. The implementation of CoGC performs a greedy descent on auxiliary variables while performing alpha expansion on the remaining variables, as described in Kohli et. al. [2013]. The dataset used is provided with the implementation. It is a part of the MSRC V2 dataset.[6].

Figure 3 shows three individual energy vs. time plots. Re-

[5] Available at *https://github.com/aosokin/coopCuts_CVPR2013*

[6] Available at *https://www.microsoft.com/en-us/research/project/image-understanding/?from=http%3A%2F%2Fresearch.microsoft.com%2Fvision%2Fcambridge%2Frecognition%2F*

sults on other images are similar. We find that C2F CoGC algorithm converges to the same energy as CoGC in about two-thirds the time on average. Overall, C2F CoGC achieves a much better anytime performance than other lifted and unlifted CoGC.

Similar to Section 4, refined partitions attain better quality than coarser ones at the expense of time. Since the implementation performs a greedy descent over auxiliary variables, refinement of current partition also resets the auxiliary variables to the last value that produced a change. Notice that energy minimization on output of $CP(2,3)$ attains a lower energy than on $CP(3,2)$. This observation drives our decision to refine by increasing N_{iter}. Qualitatively, C2F CoGC produces the same labeling as CoGC. Finally, similar to stereo matching, partitions based on thresholding scheme perform significantly worse compared to $CP(N_L, N_{iter})$ for image segmentation as well.

6 Related Work

There is a large body of work on exact lifting, both marginal [Kersting *et al.*, 2009; Gogate and Domingos, 2011; Niepert, 2012; Mittal *et al.*, 2015] and MAP [Kersting *et al.*, 2009; Gogate and Domingos, 2011; Niepert, 2012; Sarkhel *et al.*, 2014; Mittal *et al.*, 2014], which is not directly applicable to our setting. There is some recent work on approximate lifting [Van den Broeck and Darwiche, 2013; Venugopal and Gogate, 2014; Singla *et al.*, 2014; Sarkhel *et al.*, 2015; Van den Broeck and Niepert, 2015] but it's focus is on marginal inference whereas we are interested in lifted MAP. Further, this work can't handle a distinct unary potential on every node. An exception is work by Bui et al. [2012] which explicitly deals with lifting in presence of distinct unary potentials. Unfortunately, they make a very strong assumption of exchangeability in the absence of unaries which does not hold true in our setting since each pixel has its own unique neighborhood.

Work by Sarkhel et al. [2015] is probably the closest to our work. They design a C2F hierarchy to cluster constants for approximate lifted MAP inference in Markov logic. In contrast, we partition ground atoms in a PGM. Like other work on approximate lifting, they can't handle distinct unary potentials. Furthermore, they assume that their theory is provided in a normal form, i.e., without evidence, which can be a severe restriction for most practical applications. Kiddon & Domingos [2011] also propose C2F inference for an underlying Markov logic theory. They use a hierarchy of partitions based on a pre-specified ontology. CV does not have any such ontology available, and needs to discover partitions using the PGM directly.

Nath & Domingos [2010] exploit (approximate) lifted inference for video segmentation. They experiment on a specific video problem (different from ours), and they only compare against vanilla BP. Their initial partitioning scheme is similar to our thresholding approach, which does not work well in our experiments.

In computer vision, a popular approach to reduce the complexity of inference is to use superpixels [Achanta *et al.*, 2012; Van den Bergh *et al.*, 2012]. Superpixels are obtained by merging neighboring nodes that have similar characteristics. All pixel nodes in the same superpixel are assigned the same value during MAP inference. SLIC [Achanta *et al.*, 2012] is one of the most popular algorithms for discovering superpixels. Our approach differs from SLIC in some significant ways. First, their superpixels are local in nature whereas our algorithm can merge pixels that are far apart. This can help in merging two disconnected regions of the same object in a single lifted pixel. Second, they obtain superpixels independent of the inference algorithm, whereas we tightly integrate our lifting with the underlying inference algorithm. This can potentially lead to discovery of better partitions; indeed, this helped us tremendously in image segmentation. Third, they do not provide a C2F version of their algorithm and we did not find it straightforward to extend their approach to discover successively finer partitions. There is some recent work [Wei *et al.*, 2016] which addresses last two of these challenges by introducing a hierarchy of superpixels. In our preliminary experiments, we found that SLIC and superpixel hierarchy perform worse than our lifting approach. Performing more rigorous comparisons is a direction for future work.

7 Conclusion and Future Work

We develop a generic template for applying lifted inference to structured output prediction tasks in computer vision. We show that MRF-based CV algorithms can be lifted at different levels of abstraction, leading to methods for coarse to fine inference over a sequence of lifted models. We test our ideas on two different CV tasks of stereo matching and interactive image segmentation. We find that C2F lifting is vastly more efficient than unlifted algorithms on both tasks obtaining a superior anytime performance, and without any loss in final solution quality. To the best of our knowledge, this is the first demonstration of lifted inference in conjunction with top of the line task-specific algorithms. Although we restrict to CV in this work, we believe that our ideas are general and can be adapted to other domains such as NLP, and computational biology. We plan to explore this in the future.

Acknowledgements

We thank anonymous reviewers for their comments and suggestions. Ankit Anand is being supported by the TCS Research Scholars Program. Mausam is being supported by grants from Google and Bloomberg. Parag Singla is being supported by a DARPA grant funded under the Explainable AI (XAI) program. Both Mausam and Parag Singla are being supported by the Visvesvaraya Young Faculty Fellowships by Govt. of India. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of the funding agencies.

References

[Achanta *et al.*, 2012] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *In PAMI*, Nov 2012.

[Anand *et al.*, 2016] A. Anand, A. Grover, Mausam, and P. Singla. Contextual Symmetries in Probabilistic Graphical Models. In *IJCAI*, 2016.

[Anand *et al.*, 2017] A. Anand, R. Noothigattu, P. Singla, and Mausam. Non-Count Symmetries in Boolean & Multi-Valued Prob. Graphical Models. In *AISTATS*, 2017.

[Andres *et al.*, 2010] B. Andres, J. H. Kappes, U. Köthe, C. Schnörr, and F. A. Hamprecht. An Empirical Comparison of Inference Algorithms for Graphical Models with Higher Order Factors Using OpenGM. In *Pattern Recognition*. 2010.

[Blei *et al.*, 2003] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *JMLR*, 3, March 2003.

[Boykov *et al.*, 2001] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *In PAMI*, 23(11), November 2001.

[Braz *et al.*, 2005] R. Braz, E. Amir, and D. Roth. Lifted First-Order Probabilistic Inference. In *IJCAI*, 2005.

[Bui *et al.*, 2012] H. Bui, T. Huynh, and R. De Salvo Braz. Exact Lifted Inference with Distinct Soft Evidence on Every Object. In *AAAI*, 2012.

[Bui *et al.*, 2013] H. Bui, T. Huynh, and S. Riedel. Automorphism Groups of Graphical Models and Lifted Variational Inference. In *UAI*, 2013.

[Freeman *et al.*, 2000] W. Freeman, E. Pasztor, and O. Carmichael. Learning Low-Level Vision. *In IJCV*, 40, 2000.

[Friedman, 2004] N. Friedman. Inferring Cellular Networks using Probabilistic Graphical Models. *Science*, 303, 2004.

[Gogate and Domingos, 2011] V. Gogate and P. Domingos. Probabilisitic Theorem Proving. In *UAI*, 2011.

[Hirschmuller and Scharstein, 2007] H. Hirschmuller and D. Scharstein. Evaluation of Cost Functions for Stereo Matching. In *CVPR*, 2007.

[Jegelka and Bilmes, 2011] S. Jegelka and J. Bilmes. Submodularity Beyond Submodular Energies: Coupling Edges in Graph Cuts. In *CVPR*, 2011.

[Jernite *et al.*, 2015] Y. Jernite, A. Rush, and D. Sontag. A Fast Variational Approach for Learning Markov Random Field Language Models. In *ICML*, 2015.

[Jha *et al.*, 2010] A. Jha, V. Gogate, A. Meliou, and D. Suciu. Lifted Inference Seen from the Other Side : The Tractable Features. In *NIPS*, 2010.

[Kappes *et al.*, 2015] J. Kappes, B. Andres, A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems. *In IJCV*, 2015.

[Kersting *et al.*, 2009] K. Kersting, B. Ahmadi, and S. Natarajan. Counting Belief Propagation. In *UAI*, 2009.

[Kersting, 2012] K. Kersting. Lifted Probabilistic Inference. In *ECAI*, 2012.

[Kiddon and Domingos, 2011] C. Kiddon and P. Domingos. Coarse-to-Fine Inference and Learning for First-Order Probabilistic Models. In *AAAI*, 2011.

[Kimmig *et al.*, 2015] A. Kimmig, L. Mihalkova, and L. Getoor. Lifted Graphical Models: A Survey. *Machine Learning*, 2015.

[Kohli *et al.*, 2013] P. Kohli, A. Osokin, and S. Jegelka. A Principled Deep Random Field Model for Image Segmentation. In *CVPR*, 2013.

[Lempitsky *et al.*, 2010] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion Moves for Markov Random Field Optimization. *In PAMI*, Aug 2010.

[Mittal *et al.*, 2014] H. Mittal, P. Goyal, V. Gogate, and P. Singla. New Rules for Domain Independent Lifted MAP Inference. In *NIPS*, 2014.

[Mittal *et al.*, 2015] H. Mittal, A. Mahajan, V. Gogate, and P. Singla. Lifted Inference Rules With Constraints. In *NIPS*, 2015.

[Mladenov *et al.*, 2014] M. Mladenov, K. Kersting, and A. Globerson. Efficient Lifting of MAP LP Relaxations Using k-Locality. In *AISTATS*, 2014.

[Mozerov and van de Weijer, 2015] M. G. Mozerov and J. van de Weijer. Accurate Stereo Matching by Two-Step Energy Minimization. *IEEE Transactions on Image Processing*, March 2015.

[Nath and Domingos, 2010] A. Nath and P. Domingos. Efficient Lifting for Online Probabilistic Inference. In *AAAIWS*, 2010.

[Nath and Domingos, 2016] A. Nath and P. Domingos. Learning Tractable Probabilistic Models for Fault Localization. In *AAAI*, 2016.

[Niepert, 2012] M. Niepert. Markov Chains on Orbits of Permutation Groups. In *UAI*, 2012.

[Noessner *et al.*, 2013] J. Noessner, M. Niepert, and H. Stuckenschmidt. RockIt: Exploiting Parallelism and Symmetry for MAP Inference in Statistical Relational Models. In *AAAI*, 2013.

[Poole, 2003] D. Poole. First-Order Probabilistic Inference. In *IJCAI*, 2003.

[Sarkhel *et al.*, 2014] S. Sarkhel, D. Venugopal, P. Singla, and V. Gogate. Lifted MAP inference for Markov logic networks. In *AISTATS*, 2014.

[Sarkhel *et al.*, 2015] S. Sarkhel, P. Singla, and V. Gogate. Fast Lifted MAP Inference via Partitioning. In *NIPS*, 2015.

[Scharstein and Szeliski, 2002] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *In IJCV*, 2002.

[Scharstein and Szeliski, 2003] D. Scharstein and R. Szeliski. High-accuracy Stereo Depth Maps Using Structured Light. In *CVPR*, 2003.

[Singla and Domingos, 2008] P. Singla and P. Domingos. Lifted First-Order Belief Propagation. In *AAAI*, 2008.

[Singla *et al.*, 2014] P. Singla, A. Nath, and P. Domingos. Approximate Lifting Techniques for Belief Propagation. In *AAAI*, 2014.

[Szeliski *et al.*, 2008] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *In PAMI*, June 2008.

[Van den Bergh *et al.*, 2012] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool. SEEDS: Superpixels Extracted via Energy-Driven Sampling. In *ECCV*, 2012.

[Van den Broeck and Darwiche, 2013] G. Van den Broeck and A. Darwiche. On the Complexity and Approximation of Binary Evidence in Lifted Inference. In *NIPS*, 2013.

[Van den Broeck and Niepert, 2015] G. Van den Broeck and M. Niepert. Lifted Probabilistic Inference for Asymmetric Graphical Models. In *AAAI*, 2015.

[Venugopal and Gogate, 2014] D. Venugopal and V. Gogate. Evidence-Based Clustering for Scalable Inference in Markov Logic. In *Joint ECML-KDD*, 2014.

[Wei *et al.*, 2016] X. Wei, Q. Yang, Y. Gong, M. Yang, and N. Ahuja. Superpixel Hierarchy. *CoRR*, abs/1605.06325, 2016.