A Principled Approach for Coarse-to-Fine MAP Inference

Christopher Zach
Microsoft Research Cambridge, UK
christopher.m.zach@gmail.com

***Abstract*—In this work we reconsider labeling problems with (virtually) continuous state spaces, which are of relevance in low level computer vision. In order to cope with such huge state spaces multi-scale methods have been proposed to approximately solve such labeling tasks. Although performing well in many cases, these methods do usually not come with any guarantees on the returned solution. A general and principled approach to solve labeling problems is based on the well-known linear programming relaxation, which appears to be prohibitive for large state spaces at the first glance. We demonstrate that a coarse-to-fine exploration strategy in the label space is able to optimize the LP relaxation for non-trivial problem instances with reasonable run-times and moderate memory requirements.**

I. Introduction

In many applications, that use the well-established loopy belief propagation framework (or one of its convergent variants) for maximum a-posterior (MAP) inference, the number of states is relatively small, typically in the order of tens or at most hundreds labels. In this work we address the case when the state space underlying a label assignment problem is a very densely sampled representation of a continuous state space. MAP inference in continuous state spaces is often approximately solved by non-linear optimization, or using some instance of "non-parametric" belief propagation (BP) (which is actually quite parametric), or by maintaining a discrete but relatively small set of particles that are believed to be representative. In most cases there are no guarantees on the obtained solution, and one failure mode for such methods is the presence of very peaked (i.e. non-smooth) potentials. We believe that the linear programming relaxation for discrete MAP inference (and its natural continuous label extension) is a strong and tractable framework to solve labeling problems. Exact inference is possible e.g. using branch and bound, but our intuition is, that exact inference is generally futile in large state spaces.[1]

One disadvantage of the LP relaxation is the high memory cost, that scales with the state space size. We use a natural and simple multi-scale approach to explore the large state space where necessary but without dropping some guarantees (under mild assumptions). The core of our method is already described in the literature [1] (termed "interval convex BP"), but it was discarded quickly as intractable. We reconsider this negative assessment and demonstrate that a principled approach to explore large state spaces is feasible in many cases. Our main observations utilized to allow tractable inference in huge state spaces are (i) that the inner loop in the nested "interval convex BP" is not required to converge, and (ii) that efficient message updates in the original state space imply efficient message passing for "interval states." As byproducts we present a simple derivation for a particular convex BP algorithm, and a novel "dead-end elimination" condition that allows to prune interval states.

We use two classical examples of naturally continuous state spaces for our numerical results: (i) image intensities (represented by 256 states) used in a classical denoising and segmentation energy, and (ii) motion vectors in optical flow computation (with a state space containing $128^2 = 16384$ elements).

II. Related Work

MAP inference is one of the fundamental techniques in machine learning and computer vision. Since MAP inference is generally intractable, a lot of research focuses on approximate inference via suitable convex relaxations. The linear programming (LP) relaxation for MAP inference (see e.g. [2] for the statistical background and [3] for a more "algebraic" review) has received particular attention. The LP relaxation is particularly attractive, since fast algorithms exploiting the particular problem structure are available. These methods are usually convergent variants of the loopy belief propagation/message passing algorithm (e.g. [4], [5], [6]). All these methods become more expensive with increasingly large state spaces, and are not applicable to proper continuous ones. MAP inference over continuous state spaces is often reduced to discrete labels by sampling and filtering particles [7], [8], [9]. In [1] the LP relaxation for discrete inference is naturally extended to the continuous-label setting, leading to an LP with infinite (uncountable) number of constraints. Finitely sized convex relaxations for continuous state spaces are recently proposed in [10], [11] (allowing piecewise convex potentials) and [12] (with potentials that are polynomials in the continuous unknowns). It is straightfordward to show that these finitely sized relaxations are weaker relaxations than the (infinitely sized) continuous one proposed in [1].

[1] One exception seem to be protein folding instances, but the success of dead end elimination means, that the unary and pairwise potentials are at least partially very informative in these cases.

The run-time performance of message passing algorithms depends heavily on whether the crucial message filtering step (essentially a min-convolution) can be performed efficiently. A generic approach has quadratic complexity in terms of the state space size. For certain pairwise potentials, that are important in computer vision applications, this min-convolution step can be done very quickly [13], [14], [15]. Recent approaches for efficient message passing in large state spaces are stochastic belief propagation [16], which only considers a random subset of states for message computation, and the use of a trained classifier to prune the state space [17].

There exists a large literature using a coarse-to-fine framework for state space refinement to make approximate MAP inference more efficient for continuous or huge discrete state spaces, and we mention only a few recent ones: [18] embed a label refinement strategy into a sequence of approximate MAP inference steps for image registration, [19] prunes unpromising disparity values for dense stereo estimation, and [20] aggressively prunes labels (disparities) in a spatial multi-scale framework to achieve an overall constant time and space complexity (again for dense stereo). As local search methods they do not come with any guarantees on the quality of the returned solution.

III. Background

In this work we focus on MAP inference for labeling problems with at most pairwise potentials (although the main results transfer also to higher-order potentials). We assume an underlying graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node set $\mathcal{V}$ and edges $\mathcal{E}$. We denote the set of neighboring nodes of s by $N(s)$, and $\deg(s)$ is the degree of s. The task is to assign optimal labels from a label set $\mathcal{L}$ to nodes and edges,

$$\mathbf{x}^* \stackrel{\text{def}}{=} \arg\min_{\mathbf{x}} \sum_{s\in\mathcal{V}} \hat{\theta}_s(x_s) + \sum_{(s,t)\in\mathcal{E}} \hat{\theta}_{st}(x_s, x_t), \qquad (1)$$

where $x_s \in \mathcal{L}$. The unary potentials $\hat{\theta}_s : \mathcal{L} \to \mathbb{R}$, $s \in \mathcal{V}$, and pairwise potentials $\hat{\theta}_{st} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ are given and task- and instance-specific. Note that finding an optimal $\mathbf{x}^*$ is generally not tractable for arbitrary pairwise potentials $\hat{\theta}_{st}$, and therefore a lot of research is devoted to find approximate, but efficiently computable solutions of the labeling problem.

In the following we will usually absorb the unary potentials into the pairwise ones by introducing

$$\theta_{st}(x_s, x_t) \stackrel{\text{def}}{=} \frac{1}{\deg(s)}\hat{\theta}_s(x_s) + \frac{1}{\deg(t)}\hat{\theta}_t(x_t) + \hat{\theta}_{st}(x_s, x_t), \qquad (2)$$

hence Eq. 1 becomes $\mathbf{x}^* \stackrel{\text{def}}{=} \arg\min_{\mathbf{x}} \sum_{(s,t)\in\mathcal{E}} \theta_{st}(x_s, x_t)$. Many algorithms aiming to approximately solve Eq. 1 for discrete state space $\mathcal{L}$ are based on the linear programming relaxation (see e.g. [3]):

$$\begin{aligned} E_{\text{LP-MAP}}(\mu) \stackrel{\text{def}}{=} & \sum_{(s,t)\in\mathcal{E}} \sum_{x_s,x_t} \theta_{st}(x_s, x_t)\mu_{st}(x_s, x_t) \qquad (3)\\ \text{s.t. } \mu_s(x_s) = & \sum_{x_t} \mu_{st}(x_s, x_t) \quad \mu_t(x_t) = \sum_{x_s} \mu_{st}(x_s, x_t)\\ & \sum_{x_s} \mu_s(x_s) = 1 \qquad \mu_{st}(x_s, x_t) \geq 0. \end{aligned}$$

The unknowns $\{\mu_s\}_s$ and $\{\mu_{st}\}_{s,t}$ encode the assigned labels, e.g. in the optimal solution $\mu_s(x_s)$ is ideally 1 iff state x_s is the optimal label at node s, and 0 otherwise. The first set of constraints are usually called *marginalization constraints*, and the unit sum constraint is typically referred as *normalization constraint*. If the labeling problem is defined over bounded but continuous state spaces $\mathcal{L}$ (w.l.o.g. $\mathcal{L} = [0, 1]$), then the sums over states appearing in $E_{\text{LP-MAP}}$ are replaced by respective integrals over $\mathcal{L}$. [1]

There is some freedom in which constraints are part of Eq. 3. In the literature the normalization and non-negativity constraints are often enforced on both μ_s and μ_{st}. Different choices of constraints in the primal program $E_{\text{LP-MAP}}$ lead to different corresponding dual problems. We intentionally chose the constraints as stated in Eq. 3, since these allow a very simple derivation of a block coordinate method to solve the dual program, which reads as

$$\begin{aligned} E^*_{\text{LP-MAP}}(\lambda, \rho) = & \sum_s \rho_s \qquad (4)\\ \text{s.t. } \forall x_s : \rho_s = & \sum_{t\in N(s)} \lambda_{st\to s}(x_s)\\ & \lambda_{st\to s}(x_s) + \lambda_{st\to t}(x_t) \leq \theta_{st}(x_s, x_t). \end{aligned}$$

The main unknowns in the dual objective are the $\lambda_{st\to s}(\cdot)$ and $\lambda_{st\to t}(\cdot)$, which are for historic reasons often called "messages." Since edges are undirected in the underlying graph, we have $\lambda_{st\to s} = \lambda_{ts\to s}$ and $\lambda_{st\to t} = \lambda_{ts\to t}$ by convention. The quantities

$$\begin{aligned} \theta^\lambda_s(x_s) \stackrel{\text{def}}{=} & \sum_{t\in N(s)} \lambda_{st\to s}(x_s), \text{ and}\\ \theta^\lambda_{st}(x_s, x_t) \stackrel{\text{def}}{=} & \theta_{st}(x_s, x_t) - \lambda_{st\to s}(x_s) - \lambda_{st\to t}(x_t) \end{aligned}$$

are often called "reparametrized potentials," since replacing the original potentials θ_{st} by the reparametrized ones does not change the solution of the label assignment problem nor the one of the corresponding LP relaxation.

Complementary slackness: : Complementary slackness allows to (partially) decode the primal optimal solution μ^* from optimal dual variables. Since the reparametrized potentials $\theta^\lambda_s(x_s)$ have always the same value ρ_s for all x_s, one cannot extract $\mu^*_s(x_s)$ directly. We chose the dual program such that the unary reparametrized potentials are least informative about the primal solution. Nevertheless, we

can utilize the following fact: if for some state x_s at node s and optimal dual variables λ^* we have

$$\sum_{t \in N(s)} \min_{x_t} \{\theta_{st}(x_s, x_t) - \lambda^*_{st\to s}(x_s) - \lambda^*_{st\to t}(x_t)\} > 0,$$

then this implies $\mu^*_s(x_s) = 0$ in an optimal primal solution μ^*. This follows immediately from the complementary slackness for pairwise primal variables, $\theta_{st}(x_s, x_t) - \lambda^*_{st\to s}(x_s) - \lambda^*_{st\to t}(x_t) > 0 \implies \mu^*_{st}(x_s, x_t) = 0$, and the marginalization constraints. The quantity

$$\eta^\lambda_s(x_s) \stackrel{\text{def}}{=} \sum_{t \in N(s)} \min_{x_t} \left\{\theta^\lambda_{st}(x_s, x_t)\right\} \tag{5}$$

therefore allows to rank the states x_s according to a reparametrization θ^λ, and this ranking will reveal constraints on the primal solution if λ is optimal.

Algorithm 1 Convex (min-sum) belief propagation

Require: Arbitrary λ

while not converged **do**
  loop over $s \in \mathcal{V}$
    $\mu_{st\to s}(x_s) \leftarrow \min_{x_t}\{\theta_{st}(x_s, x_t) - \lambda_{st\to t}(x_t)\}$
    $\rho_s \leftarrow \min_{x_s}\left\{\sum_{t\in N(s)} \mu_{st\to s}(x_s)\right\}$
    $\lambda_{st\to s}(x_s) \leftarrow \mu_{st\to s}(x_s) - \frac{\sum_r \mu_{sr\to s}(x_s) - \rho_s}{\deg(s)}$
  end loop
end while

A dual block coordinate descent method:: If we fix a node s and maximize the dual only with respect to ρ_s and all messages incoming at s, $\lambda_{st\to s}(x_s)$ for $t \in N(s)$ and all x_s, then the restricted dual subproblem becomes

$$\max_{\rho_s, \{\lambda_{st\to s}(x_s)\}} \rho_s \;\text{ s.t. } \rho_s = \sum_t \lambda_{st\to s}(x_s)$$
$$\lambda_{st\to s}(x_s) \le \underbrace{\min_{x_t} \{\theta_{st}(x_s, x_t) - \lambda_{st\to t}(x_t)\}}_{\stackrel{\text{def}}{=}\mu_{st\to s}(x_s)}.$$

Note that we have simple bounds constraints on $\lambda_{st\to s}(x_s)$. Since we want to maximize ρ_s, we need to maximize $\sum_t \lambda_{st\to s}(x_s)$ subject to upper bounds $\mu_{st\to s}(x_s)$ on each $\lambda_{st\to s}(x_s)$. Therefore, ρ_s attains the smallest feasible value of $\sum_t \lambda_{st\to s}(x_s)$, or

$$\rho_s \leftarrow \min_{x_s} \left\{\sum_t \mu_{st\to s}(x_s)\right\}. \tag{6}$$

If for some x_s the upper bounds are tight, i.e. $\rho_s = \sum_t \mu_{st\to s}(x_s)$, then $\lambda_{st\to s}(x_s) \leftarrow \mu_{st\to s}(x_s)$. The update equations for the remaining $\lambda_{st\to s}(x_s)$ with $\rho_s < \sum_t \mu_{st\to s}(x_s)$ allow some freedom. One option, that handles all x_s, is to assign the "slack" $\sum_t \mu_{st\to s}(x_s) - \rho_s$ uniformly over the edges, and the updates are therefore given by

$$\lambda_{st\to s}(x_s) \leftarrow \mu_{st\to s}(x_s) - \frac{\sum_r \mu_{sr\to s}(x_s) - \rho_s}{\deg(s)}. \tag{7}$$

Note that this algorithm is a particular instance of convex max-product belief propagation (e.g. [6]) derived in an extremely simple way. We summarize the updates in Algorithm 1. If the algorithm is provided with an infeasible set of dual variables λ, these will be feasible after all dual messages are updated at least once. Another crucial aspect of the method, which will be important later, is the "warm-restart" capability by providing any (not necessarily feasible) dual variables as input.

IV. Basic Iterative State Space Refinement

In this section we describe a generic approach for approximate MAP inference with either very large or (virtually) continuous state space. Our exposition will focus on 1D state spaces, but extends to product label spaces in a straightforward manner. Let the underlying state space be $\mathcal{L} = \{1, \ldots, L\}$ with $L = |\mathcal{L}| \gg 1$, e.g. L is in the order of 100s or 1000s. In the following we will use lower-case letters to indicate "fine-grained" states from $\mathcal{L}$, e.g. $x_s, x_t \in \mathcal{L}$, and upper-case letters to denote "coarsened" or "group states" such as $X_s, X_t \subseteq \mathcal{L}$. Without loss of generality, but for representational simplicity, we will restrict these group states to form intervals such as

$$X_s = [X_s^-, X_s^+] \subseteq \mathcal{L}.$$

In product label spaces these intervals are replaced by hyper-rectangles. Now assume we are provided with partitions $\mathcal{P}_s$ of $\mathcal{L}$ at each $s \in \mathcal{V}$. Thus, group states X_s are elements from $\mathcal{P}_s$. We introduce lower bound potentials

$$\underline{\theta}_{st}(X_s, X_t) \stackrel{\text{def}}{=} \min_{x_s \in X_s, x_t \in X_t} \theta_{st}(x_s, x_t). \tag{8}$$

It is clear that the following primal program yields a lower bound on the original many-label program,

$$E_{\text{coarse-MAP}}(\mu|\{\mathcal{P}_s\}) = \sum_{(s,t)\in\mathcal{E}} \sum_{X_s, X_t} \underline{\theta}_{st}(X_s, X_t)\mu_{st}(X_s, X_t) \tag{9}$$

subject to

$$\mu_s(X_s) = \sum_{X_t} \mu_{st}(X_s, X_t) \quad \mu_t(X_t) = \sum_{X_s} \mu_{st}(X_s, X_t)$$
$$\sum_{X_s} \mu_s(X_s) = 1 \qquad \mu_{st}(X_s, X_t) \ge 0.$$

By moving to successively finer grained partitions this lower bound $\min_\mu E_{\text{coarse-MAP}}(\mu)$ approaches $\min_\mu E_{\text{LP-MAP}}(\mu)$ optimized over the full state space. In practice we utilize a message passing algorithm to optimize the following dual program,

$$E^*_{\text{coarse-MAP}}(\lambda, \rho|\{\mathcal{P}_s\}) = \sum_s \rho_s \tag{10}$$
$$\text{s.t. } \forall X_s : \rho_s = \sum_{t\in N(s)} \lambda_{st\to s}(X_s)$$
$$\lambda_{st\to s}(X_s) + \lambda_{st\to t}(X_t) \le \underline{\theta}_{st}(X_s, X_t).$$

Another way to relate approximate inference using coarse states with the one using the full state space, is that $E^*_{\text{coarse-MAP}}(\cdot|\{\mathcal{P}_s\})$ adds the following equality constraints to $E^*_{\text{LP-MAP}}$,

$$\forall x_s, x'_s \in X_s : \lambda_{st\to s}(x_s) = \lambda_{st\to s}(x'_s)$$

and

$$\forall x_t, x'_t \in X_t : \lambda_{st\to t}(x_t) = \lambda_{st\to t}(x'_t).$$

Therefore it becomes clear that $E^*_{\text{coarse-MAP}}(\cdot|\{\mathcal{P}_s\})$ is a lower bound for $E^*_{\text{LP-MAP}}$. We propose the following basic procedure to obtain a sequence of lower bounds for $E_{\text{LP-MAP}}$ and associated dual and primal variables:

Algorithm 2 Iterative refinement of the state space

1: Initialize $\{\mathcal{P}_s\}_{s\in\mathcal{V}}$
2: Optional: apply DEE (Sec. VI-A)
3: **while** target accuracy not reached **do**
4: Improve $E^*_{\text{coarse-MAP}}(\cdot|\{\mathcal{P}_s\})$
5: Refine state space $\{\mathcal{P}_s\}$
6: Extend the dual variables to the refined state space
7: Optional: merge group states (Sec. VI-B)
8: **end while**
9: Extract (approximate) primal solution

This is essentially the same meta-algorithm as also considered in [1], but there the authors quickly discarded the approach as non-practical. We list some challenges making the above algorithm look less appealing, and also state how we address these issues: (i) As a nested iterative algorithm the above method is slow, since it requires message passing to (reasonably) converge in the inner loop. Below we outline that it is not neccesary to find the exact dual optimum in step 4 in each round, and a single iteration of convex BP (with "warm restart") is theoretically sufficient. (ii) Because of the irregular state space structure, fast message passing algorithms in step 4 are not available. We show in Section V that if efficient message passing can be done for the fine-grained problem, it is also available for coarse MAP inference. (iii) Computation of the lower bound $\underline{\theta}_{st}(X_s, X_t)$ can be very expensive. In this work we focus our attention on problem instances where the lower bound is relatively straightforward to obtain. Such instances occur frequently in low-level computer vision problems, which is the main application behind this work. For some unstructured pairwise potentials (such as the ones occurring in the protein design) one potential answer is our extension of a well-known criterion for DEE to grouped states (Section VI-A).

One question is when Algorithm 2 and optimizing the fine-grained problem $E^*_{\text{LP-MAP}}$ lead to the same solution. By using a complementary slackness argument it is relatively easy to see that the following two conditions are sufficient for Algorithm 2 to converge to the dual optimum equivalent to $E^*_{\text{LP-MAP}}$ (for a formal proof we refer to the supplementary material):

1) The iterative maximization method used in line 4 of Algorithm 2 is guaranteed to converge to the dual optimal solution of $E^*_{\text{coarse-MAP}}(\cdot|\{\mathcal{P}_s\})$.
2) Coarse states in $X_s \in \mathcal{P}_s$ which have the smallest value $\eta^\lambda_s(X_s)$ are ultimately refined, where $\eta^\lambda_s(X_s)$ for a group state X_s is defined as

$$\eta^\lambda_s(X_s) \stackrel{\text{def}}{=} \sum_{t\in N(s)} \min_{X_t} \left\{\underline{\theta}^\lambda_{st}(X_s, X_t)\right\}, \qquad (11)$$

with $\underline{\theta}^\lambda_{st}(X_s, X_t) \stackrel{\text{def}}{=} \underline{\theta}_{st}(X_s, X_t) - \lambda_{st\to s}(X_s) - \lambda_{st\to t}(X_t)$.

One of the main questions is how to refine the partitions $\mathcal{P}_s$ after each message passing round. Refining $\mathcal{P}_s$ is one instance of the classical exploration/exploitation dilemma: subdividing fewer group states in $\mathcal{P}_s$ may need many more (outer) rounds, but splitting many group states slows down step 3 in Alg. 2 and leads to higher memory consumption. It is clear that "promising" states with the lowest reparametrized costs seem to be good candidates for refinement. We use the following simple strategy: states X_s, that are in the better half according to the current unary reparametrized cost $\eta^\lambda_s(X_s)$, are refined (until $X_s = \{x_s\}$ is only a singleton).

Another, equally important question, is whether the size of the partitions $\mathcal{P}_s$ can be bounded or its growth slowed down. In Section VI-B we show that some group states can be merged (i.e. the partition $\{\mathcal{P}_s\}$ can be coarsened) without decreasing the dual objective.

V. Convex BP and Group States

In this section we focus on how to improve $E^*_{\text{coarse-MAP}}(\cdot|\{\mathcal{P}_s\})$ in line 4 of Alg. 2 efficiently. In the following the set of group states $\{\mathcal{P}_s\}$ is fixed. The main computational step of convex BP (recall Alg. 1) applied on $E^*_{\text{coarse-MAP}}(\cdot|\{\mathcal{P}_s\})$ is the determination of

$$\mu_{st\to s}(X_s) \leftarrow \min_{X_t\in\mathcal{P}_t} \left\{\underline{\theta}_{st}(X_s, X_t) - \lambda_{st\to t}(X_t)\right\}.$$

for all $X_s \in \mathcal{P}_s$. We obtain of $\mu_{st\to s}(X_s)$,

$$\begin{aligned}
\mu_{st\to s}(X_s) &= \min_{X_t} \left\{\underline{\theta}_{st}(X_s, X_t) - \lambda_{st\to t}(X_t)\right\} \\
&= \min_{X_t} \left\{\min_{x_s\in X_s, x_t\in X_t} \theta_{st}(x_s, x_t) - \lambda_{st\to t}(X_t)\right\} \\
&= \min_{x_s\in X_s} \min_{x_t} \left\{\theta_{st}(x_s, x_t) - \lambda_{st\to t}(x_t)\right\} \\
&= \min_{x_s\in X_s} \mu_{st\to s}(x_s).
\end{aligned}$$

This means in particular, that if $\mu_{st\to s}(x_s)$ can be efficiently computed in the original many-label state space $\mathcal{L}$, then one can also obtain $\mu_{st\to s}(X_s)$ relatively efficiently: first, expand $\lambda_{st\to t}(X_t)$ to $\mathcal{L}$ by setting $\lambda_{st\to t}(x_t) \leftarrow \lambda_{st\to t}(X_t)$ for $x_t \in$

X_t, and perform subsequent min-convolution on $\mathcal{L}$. Finally, extract

$$\mu_{st\to s}(X_s) \leftarrow \min_{x_s \in X_s} \mu_{st\to s}(x_s).$$

This approach becomes less efficient if the number of fine-grained states, $|\mathcal{L}|$, is very large, but it retains low memory requirements. If we allow to use term-wise lower bounds, i.e. we introduce

$$\begin{aligned}\underline{\theta}_s(X_s) &\stackrel{\text{def}}{=} \frac{1}{\deg(s)} \min_{x_s \in X_s} \hat{\theta}_s(x_s) \\ \underline{\theta}_{st}(X_s, X_t) &\stackrel{\text{def}}{=} \underline{\theta}_s(X_s) + \underline{\theta}_t(X_t) \\ &\quad + \min_{x_s \in X_s, x_t \in X_t} \hat{\theta}_{st}(x_s, x_t)\end{aligned}$$

(which is a weaker lower bound on the potentials than the one defined in Eq. 8), then we can design a much more efficient method to compute $\mu_{st\to s}(X_s)$, whose runtime complexity depends on $|\mathcal{P}_s|$ and $|\mathcal{P}_t|$ rather than $|\mathcal{L}|$. The method relies on efficient computation of the min-convolution for non-uniformely sampled collections of location/function-value pairs. In Alg. 3 we illustrate a slightly more general method than the one proposed in [13] for pairwise potentials $\hat{\theta}_{st}(x_s, x_t)$ that can be written as $\hat{\theta}_{st}(x_s, x_t) = f(x_t - x_s)$ for a convex function f attaining its minimal value at 0.

Algorithm 3 Non-uniform min-convolution

Require: array of input values $h[0 : N-1]$
Require: array of locations $b[0 : N-1]$
Require: Convex function f
  $j \leftarrow 0$, $v[0] \leftarrow 0$, $z[0] \leftarrow -\infty$, $z[1] \leftarrow \infty$
  for $q = 1 : N-1$ **do**
    repeat
      $j \leftarrow j + 1$
      $t_v \leftarrow b[v[j-1]]$, $t_q \leftarrow b[q]$
      $s \leftarrow$ intersection of $t \mapsto h(q) + f(t - t_q)$
        and $t \mapsto h(v[j-1]) + f(t - t_v)$
    until $s > z[j-1]$
    $v[j] \leftarrow q$, $z[j] \leftarrow s$, $z[j+1] \leftarrow \infty$
  end for
  $j \leftarrow 0$
  for $q = 0 : N-1$ **do**
    while $z[j+1] < q$ **do** $j \leftarrow j + 1$
    $D_h[q] \leftarrow h(v[j]) + f(q - b[v[j]])$
  end for

The computation of $\mu_{st\to s}(X_s)$ outlined in Alg. 4 first computes the min-convolution at all interval boundary locations

$$\mathcal{B} \stackrel{\text{def}}{=} \underbrace{\bigcup \{X_s^-, X_s^+\}}_{\stackrel{\text{def}}{=}\mathcal{B}_s} \cup \underbrace{\bigcup \{X_t^-, X_t^+\}}_{\stackrel{\text{def}}{=}\mathcal{B}_t}$$

and subsequently determines $\mu_{st\to s}(X_s)$ as the minimum of $D_h(x)$ over all states $x \in X_s$. The correctness of the algorithm can be seen as follows: first, D_h can be naturally extended from the set of breakpoints $\mathcal{B}$ to all $x_s \in \mathcal{L}$. Let $x_s \in [X^-, X^+]$ with X^-, X^+ be the two nearest breakpoints in $\mathcal{B}$. Then we have

$$\min_{x_s \in [X^-, X^+]} \mu_{st\to s}(x_s) = \min\left\{\mu_{st\to s}(X^-), \mu_{st\to s}(X^+)\right\}$$

since the smoothness cost f is minimal at 0. Therefore

$$\begin{aligned}\mu_{st\to s}(X_s) &= \min_{x_s \in [X_s^-, X_s^+]} \mu_{st\to s}(x_s) \\ &= \min_{X \in X_s \cap \mathcal{B}} \mu_{st\to s}(X) = \min_{j : b[j] \in X_s} \min D_h[j]\end{aligned}$$

as computed in line 7 in Alg. 4. The overall runtime of the algorithm is linear in $|\mathcal{B}|$. The methods immediately generalizes when the pairwise potential is actually the minimum of convex potentials. Further, product label spaces with decomposable pairwise terms can be handled by higher dimensional distance transforms (e.g. [21]).

Algorithm 4 Computation of $\mu_{st\to s}(X_s)$

1: Run a "merge sort" step on $\mathcal{B}_s$ and $\mathcal{B}_t$ to fill (increasing) locations $b[\cdot]$
2: **for** $j = 0 : \text{len}(b) - 1$ **do**
3: $\quad h[j] \leftarrow \underline{\theta}_t(X_t) - \lambda_{st\to t}(X_t)$ such that $b[j] \in X_t$
4: **end for**
5: Run Algorithm 3 to obtain D_h
6: **for** $X_s = 0 : |\mathcal{P}_s| - 1$ **do**
7: $\quad \mu_{st\to s}(X_s) \leftarrow \min_{j : b[j] \in X_s} D_h[j]$
8: **end for**

VI. Dead-End Elimination and Coarsening

The basic coarse-to-fine approach presented in Section IV consecutively increases the size of the state space in each round. In this section we discuss the possibilities of early pruning of non-optimal group states and "undoing" refinement steps in order to shrink the state space size.

A. Pruning Group States

Some states are known in advance not to be part of any optimal labeling, e.g. due to extremely large unaries rendering these states unfavorable. Detection of such suboptimal states is frequently applied in protein design and called "dead end elimination" (DEE) in the literature [22], [23]). In the MAP inferenece literature the term "partial optimality" is commonly used (we refer to [24] for a recent overview). The (simple) Goldstein DEE condition [23] (which is stronger but also computationally more expensive than the original DEE condition in [22]) for (fine-grained) states can be phrased as "a state x_s can be discarded if there exists a state x'_s such that for all labels assigned to the neighborhood of s the local contribution to the objective is always smaller than the one of x_s." In our framework we are facing group states, i.e. sets of fine-grained labels, and want to establish

respective DEE conditions. A similar setting is addressed in [25] (where the "minDEE" or minimized DEE criterion is proposed) and [26] (establishing the "iMinDEE" criterion). We use the Goldstein DEE criterion as starting point to obtain a DEE condition for group states, which is provably stronger than minDEE and in practice also stronger than iMinDEE. The Goldstein DEE conditions for group states is now: group state X_s can be pruned if there exists another group state X'_s such that

$$\sum_t \min_{X_t} \left\{ \underline{\theta}_{st}(X_s, X_t) - \overline{\theta}_{st}(X'_s, X_t) \right\} > 0, \text{ or} \quad (12)$$

$$\exists X'_s : \forall X_t : \sum_t \underline{\theta}_{st}(X_s, X_t) > \sum_t \overline{\theta}_{st}(X'_s, X_t), \quad (13)$$

i.e. the local "optimistic" contribution of X_s to the objective is larger than the "pessimistic" one for X'_s for all labels assigned to the neighbors of s. $\overline{\theta}_{st}(X_s, X_t)$ denotes an upper bound of

$$\overline{\theta}_{st}(X_s, X_t) \geq \max_{x_s \in X_s, x_t \in X_t} \theta_{st}(x_s, x_t).$$

In the supplementary material we show that the above condition implies the standard Goldstein DEE one, and prunes states conservatively. Intuitively, the condition in Eq. 12 is stronger than minDEE (and usually iMinDEE), since the latter criteria are based on comparing lower and upper bounds of full min-marginals, not on local contributions to the overall energy. In our tests the iMinDEE condition never pruned any state, since in typical computer vision applications the unaries are not strong enough for the iMinDEE criterion to apply. Dead-end elimination is only useful to prune the initial group states at the beginning of Alg. 2, since unpromising states passing the DEE condition will almost never be refined in line 5 of Alg. 2.

B. Coarsening States

Pruning as described in the previous section discards suboptimal (group) states. In many cases it is not possible to directly prove suboptimality of a state (since all DEE criteria are conservative), but an alternative is to show that merging two group states into a coarser one does not lead to a decrease of the dual. This is addressed in this section.

Assume that two group states X'_s and X''_s are candidates for merging at node s, i.e. to be represented by a single state $X_s = X'_s \cup X''_s$. We need to define dual variables $\lambda_{st \to s}(X_s)$ such that they are still feasible for the dual energy, and therefore guarantee that the dual energy will not immediately decrease in the next round. The dual constraints that need to be satisfied are

$$\rho_s = \sum_t \lambda_{st \to s}(X'_s) = \sum_t \lambda_{st \to s}(X''_s) = \sum_t \lambda_{st \to s}(X_s),$$

and

$$\begin{aligned} \lambda_{st \to s}(X_s) + \lambda_{st \to t}(X_t) &\leq \underline{\theta}_{st}(X_s, X_t) \\ &= \min \left\{ \underline{\theta}_{st}(X'_s, X_t), \underline{\theta}_{st}(X''_s, X_t) \right\}. \end{aligned}$$

The "coarser" dual variables $\lambda_{st \to s}(X_s)$ now jointly represent states X'_s and X''_s. If we check whether X'_s and X''_s can be merged by successively traversing nodes s, then the dual variables $\lambda_{st \to t}(X_t)$ are fixed, and merging X'_s and X''_s will not reduce the dual (render the dual solution infeasible) if

$$\begin{aligned} \rho_s &< \sum_t \min_{X_t} \left\{ \underline{\theta}_{st}(X_s, X_t) - \lambda_{st \to t}(X_t) \right\} \\ &= \sum_t \mu_{st \to s}(X_s). \end{aligned}$$

We require ρ_s to be strictly less than the right hand side to rule out the possibility of non-convergent oscillations in Algorithm 2. If the condition is met, then coarsened and feasible dual variables $\lambda_{st \to s}(X_s)$ can be determined by one iteration of message passing, i.e.

$$\lambda_{st \to s}(X_s) \leftarrow \mu_{st \to s}(X_s) - \frac{\sum_r \mu_{sr \to s}(X_s) - \rho_s}{\deg(s)}.$$

In a nutshell, refining promising (group) states in $\mathcal{P}_s$ and subsequent coarsening of non-promising states essentially resamples the representation of the full state space without sacrificing dual optimality.

VII. Numerical Results

We illustrate two applications for our method. The first one is a very parametric continuous labeling problem with a one-dimensional state space. The second one uses an arbitrary data term and a 2D product label space. Run-time measurements of our OpenMP enabled C++ implementation are obtained on a standard PC with a 2.5 GHz quad-core processor. For exact details on the algorithmic settings we refer to the supplementary material.

	Lena	Tsukuba	Teddy	Cones	Art
Our	139.38	68.859	94.953	107.64	803.41
Full	2048	864	1318.4	1318.4	12054

Table I
Memory requirements (in MiB) of our implementation and the one for storing all dual variables.

A. Piece-Wise Smooth Denoising

Our "toy" example is a discretized version of the continuous Mumford-Shah functional,

$$E_{\text{MS}}(\mathbf{f}|\mathbf{g}) = \frac{\lambda}{2} \sum_{s \in \mathcal{V}} (f_s - g_s)^2 + \sum_{(s,t) \in \mathcal{E}} \min \left\{ \mu, (f_s - f_t)^2 \right\}, \quad (14)$$

which is a model for image denoising favoring piece-wise smooth solutions $\mathbf{f}$. $\mathcal{V}$ represents the pixel on a regular grid, and $\mathcal{E}$ is the set of 4-connected edges. This is a labeling problem over continuous states, but practical solutions often work with an 8-bit discretized label space. Table I lists the memory requirements of our implementation (as reported by

the operating system) and the minimum memory needed to store the dual variables for a full 256 state space. Our choices of λ and μ are 1 and $1/10$, respectively, leading to substantial smoothing (we refer to the supplementary material for visual results). Figure 1 depicts the increase of the dual energy with respect to clock time. Fig. 2 displays the average number of active group states per pixel over time. At some point this number decreases slightly due to the coarsening step (Section VI-B). The low number of active states (about 16 instead of 256) is reflected in the one order of magnitude memory reduction observed in Table I. In the supplementary material we provide additional graphs for different choices of initial group states $\{\mathcal{P}_s\}$.

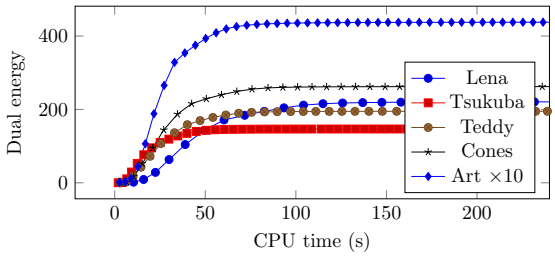

Figure 1. Dual energy evolution with respect to clock time.

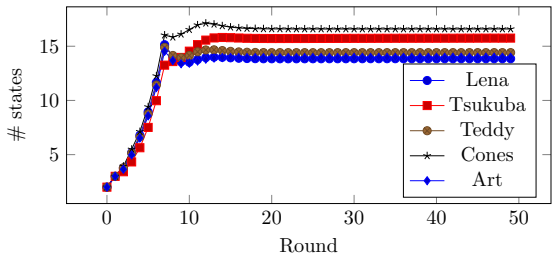

Figure 2. Evolution of the number of active group states.

B. Optical Flow Field Estimation

One aspect of image denoising with non-robust noise models is the strong data term (i.e. very discriminative unary potentials), which makes finding a MAP solution relatively easy. We pick motion estimation as illustrative example for (i) large product label spaces and (ii) more ambiguous unaries.[2] In order to avoid the very expensive 2-dimensional distance transform in each message update, we build on the "decomposed" MRF formulation [27], which maintains two labels (horizontal and vertical disparity) per pixel and encodes the data term as unstructured pairwise potential. This formulation is potentially a weaker relaxation than the one using product labels, but in practice we did not observe differences between them.[3]

[2]Input data from `vision.middlebury.edu`.

[3]We also implementated the proper product label formulation (requiring a much slower 2D min-convolution step).

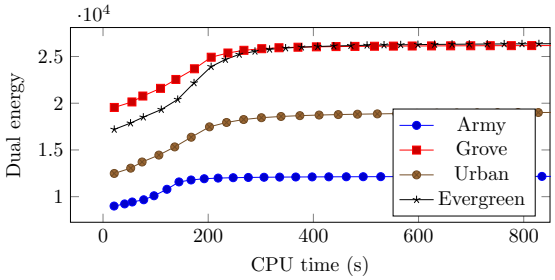

Figure 3. Dual energy evolution with respect to clock time.

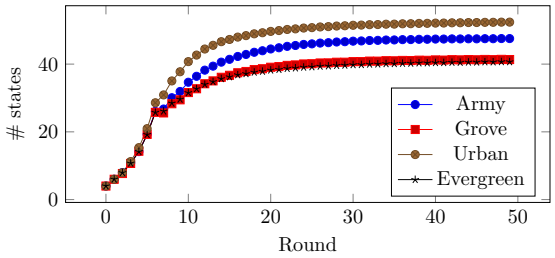

Figure 4. Evolution of the number of active group states.

We use a 128×128 state space to represent 2D flow vectors at quarter-pixel resolution, which allows a ± 16 pixel search range in both image directions. The evolution of the obtained dual energies using a 3×3 SAD data term and an L^1 smoothness prior for the horizontal and vertical components is shown in Fig. 3.

While the memory needed to store the dual messages for 2×128 states are not out of reach (although still high), the message updates between the horizontal and vertical labels have quadratic complexity due to the generally unstructured data term (whose profile depends on the image content). Having a smaller number of (group) states (see Fig. 4 for its evolution) reduces the run-time complexity of these message updates. Additionally, dynamic computation of the matching costs in the message updates is only feasible for the simplest ones, but caching the full 128^2 precomputed scores *per pixel* is not tractable. Consequently we report the number of data terms required to be cached in each round of Alg. 2 in Fig. 5. Much of the actual run-time is spent in (repeatedly) computing the data term. In summary, the fraction of active group states is higher than in Section VII-A due to the more ambiguous data term, and we refer to the supplementary material for visualizations of the returned flow fields.

VIII. Conclusion and Future Work

In this work we demonstrate that a principled coarse-to-fine label space exploration approach allows tractable MAP inference for huge state spaces without losing important properties such as guaranteed lower bounds of the true energy, and obtaining certificates for LP optimality. Our methods allows to obtain a "gold-standard" solution for

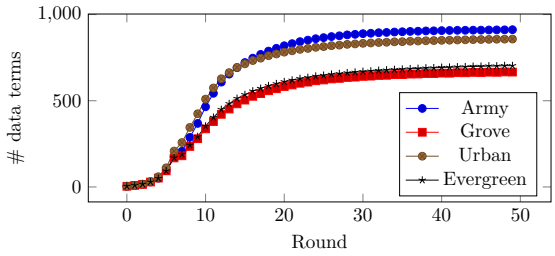

Figure 5. Evolution of the avg. number of cached data terms/pixel.

MAP inference for large problem instances, which can be used to identify failure cases of cheaper and more approximate inference methods. The obtained numerical results give hints of how many "particles" are actually needed in the best case to represent a global labeling solution. One direction of future work addresses the incorporation of computationally expensive potentials, for which more efficient lower and upper bounds are available. This can be seen as generalization of methods used for fast but exact template search to random fields.

References

[1] J. Peng, T. Hazan, D. McAllester, and R. Urtasun, "Convex max-product algorithms for continuous MRFs with applications to protein folding," in *Proc. ICML*, 2011.

[2] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, pp. 1–305, 2008.

[3] T. Werner, "A linear programming approach to max-sum problem: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, 2007.

[4] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1568–1583, 2006.

[5] A. Globerson and T. Jaakkola, "Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations," in *NIPS*, 2007.

[6] T. Hazan and A. Shashua, "Norm-prodcut belief propagtion: Primal-dual message-passing for LP-relaxation and approximate-inference," *IEEE Trans. on Information Theory*, vol. 56, no. 12, pp. 6294–6316, 2010.

[7] J. Coughlan and H. Shen, "Dynamic quantization for belief propagation in sparse spaces," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 47–58, 2007.

[8] A. Ihler and D. McAllester, "Particle belief propagation," in *AISTATS*, 2009, pp. 256–263.

[9] M. Isard, J. MacCormick, and K. Achan, "Continuously-adaptive discretization for message-passing algorithms," in *NIPS*, 2009, pp. 737–744.

[10] C. Zach and P. Kohli, "A convex discrete-continuous approach for Markov random fields," in *Proc. ECCV*, 2012.

[11] C. Zach, "Dual decomposition for joint discrete-continuous optimization," in *Proc. AISTATS*, 2013.

[12] M. Salzmann, "Continuous inference in graphical models with polynomial energies," in *Proc. CVPR*, 2013.

[13] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *IJCV*, vol. 70, no. 1, pp. 41–54, 2006.

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

[15] S. Gu, Y. Zheng, and C. Tomasi, "Extended pairwise potentials," in *CVPR Workshop on Inference in Graphical Models with Structured Potentials*, 2011.

[16] N. Noorshams and M. J. Wainwright, "Stochastic belief propagation: A low-complexity alternative to the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 59, no. 4, pp. 1981–2000, 2013.

[17] M. Guillaumin, L. Van Gool, and V. Ferrari, "Fast energy minimization using learned state filters," in *Proc. CVPR*, 2013.

[18] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios, "Dense image registration through mrfs and efficient linear programming," *Medical Image Analysis*, vol. 12, no. 6, pp. 731–741, 2008.

[19] L. Wang, H. Jin, and R. Yang, "Search space reduction for MRF stereo," in *Proc. ECCV*. Springer, 2008, pp. 576–588.

[20] Q. Yang, L. Wang, and N. Ahuja, "A constant-space belief propagation algorithm for stereo matching," in *Proc. CVPR*. IEEE, 2010, pp. 1458–1465.

[21] B. Potetz, "Efficient belief propagation for vision using linear constraint nodes," in *Proc. CVPR*. IEEE, 2007, pp. 1–8.

[22] J. Desmet, M. D. Maeyer, B. Hazes, and I. Lasters, "The dead-end elimination theorem and its use in protein side-chain positioning," *Nature*, vol. 356, no. 6369, pp. 539–542, 1992.

[23] R. F. Goldstein, "Efficient rotamer elimination applied to protein side-chains and related spin glasses," *Biophysical Journal*, vol. 66, no. 5, pp. 1335–1340, 1994.

[24] A. Shekhovtsov, "Exact and partial energy minimization in computer vision," Czech Technical University, Prague, Tech. Rep., 2013.

[25] I. Georgiev, R. H. Lilien, and B. R. Donald, "The minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm for computing partition functions over molecular ensembles," *Journal of Computational Chemistry*, vol. 29, no. 10, pp. 1527–1542, 2008.

[26] P. Gainza, K. E. Roberts, and B. R. Donald, "Protein design using continuous rotamers," *PLoS computational biology*, vol. 8, no. 1, p. e1002335, 2012.

[27] A. Shekhovtsov, I. Kovtun, and V. Hlaváč, "Efficient MRF deformation model for non-rigid image matching," *CVIU*, vol. 112, no. 1, pp. 91–99, 2008.