A Multiscale Variable-grouping Framework for MRF Energy Minimization

Omer Meir Meirav Galun Stav Yagev Ronen Basri
Weizmann Institute of Science
Rehovot, Israel
{omerm, meirav.galun, stav, ronen.basri}@weizmann.ac.il

Irad Yavneh
Technion
Haifa, Israel
irad@cs.technion.ac.il

Abstract

We present a multiscale approach for minimizing the energy associated with Markov Random Fields (MRFs) with energy functions that include arbitrary pairwise potentials. The MRF is represented on a hierarchy of successively coarser scales, where the problem on each scale is itself an MRF with suitably defined potentials. These representations are used to construct an efficient multiscale algorithm that seeks a minimal-energy solution to the original problem. The algorithm is iterative and features a bidirectional crosstalk between fine and coarse representations. We use consistency criteria to guarantee that the energy is nonincreasing throughout the iterative process. The algorithm is evaluated on real-world datasets, achieving competitive performance in relatively short run-times.

1. Introduction

In recent years Markov random fields (MRFs) have become an increasingly popular tool for image modeling, with applications ranging from image denoising, inpainting and segmentation, to stereo matching and optical flow estimation, and many more. An MRF is commonly constructed by modeling the pixels (or regional "superpixels") of an image as variables that take values in a discrete label space, and by formulating an energy function that suits the application.

An expressive model used often is the pairwise model, which is specified by an energy function $E : \mathcal{X} \to \mathbb{R}$,

$$E(\mathbf{x}) = \sum_{v\in\mathcal{V}} \phi_v(x_v) + \sum_{(u,v)\in\mathcal{E}} \phi_{uv}(x_u, x_v), \qquad (1)$$

where $\mathbf{x} \in \mathcal{X}$ is a label assignment of all variables $v \in \mathcal{V}$. The first term in this function is the sum of unary potentials, $\phi_v(x_v)$, which reflect the cost of assigning label x_v to variable v. The second term is the sum of pairwise potentials, $\phi_{uv}(x_u, x_v)$, which model the interaction between pairs of variables by reflecting the cost of assigning labels x_u, x_v to variables u, v, respectively. Here $\mathcal{V}$ denotes the set of variables and $\mathcal{E}$ is the set of pairs of interacting variables. The *inference* task then is to find a label assignment that minimizes the energy.

Considerable research has been reported in the literature on approximating (1) in a coarse-to-fine framework [4, 6, 9, 11, 14, 15, 16, 17, 18]. Coarse-to-fine methods have been shown to be beneficial in terms of running time [6, 9, 16, 18]. Furthermore, it is generally agreed that coarse-to-fine schemes are less sensitive to local minima and can produce higher-quality label assignments [4, 11, 16]. These benefits follow from the fact that, although only local interactions are encoded, the model is global in nature, and by working at multiple scales information is propagated more efficiently [9, 16].

Until recently, coarse-to-fine schemes were confined to geometric structures, *i.e.*, grouping together square patches of variables in a grid [9, 11]. Recent works [4, 14] suggest to group variables which are likely to end up with the same label in the minimum energy solution, rather than by adhering to a geometric structure. Such grouping may lead to over-smoothing [6, 15]. Methods for dealing with the problem of over-smoothing include applying a multi-resolution scheme in areas around boundaries at a segmentation task [15, 18], or pruning the label space of fine scales with a pre-trained classifier [6].

In this paper we present a multiscale framework for solving MRFs with multi-label arbitrary pairwise potentials. The algorithm has been designed with the intention of optimizing "hard" energies which may arise, for example, when the parameters of the model are learned from data or when the model accounts for negative affinities between neighboring variables. Our approach uses existing inference algorithms together with a variable grouping procedure referred to as *coarsening*, which is aimed at producing a hierarchy of successively coarser representations of the MRF problem, in order to efficiently explore relevant subsets of the space of possible label assignments. Our method is it-

This research was supported in part by the Israel Science Foundation Grant No. 1265/14.

erative and monotonic, that is, the energy is guaranteed not to increase at any point during the iterative process. The method can efficiently incorporate any initializeable inference algorithm that can deal with general pairwise potentials, *e.g.*, QPBO-I [20] and LSA-TR [10], yielding significantly lower energy values than those obtained with standard use of these methods.

Unlike existing multiscale methods, which employ only coarse-to-fine strategies, our framework features a bidirectional crosstalk between fine and coarse representations of the optimization problem. Furthermore, we suggest to group variables based on the magnitude of their statistical correlation, regardless of whether the variables are assumed to take the same label at the minimum energy. The method is evaluated on real-world datasets yielding promising results in relatively short run-times.

2. The multiscale framework

An inference algorithm is one which seeks a labeling of the variables that minimizes the energy of Eq. (1). We refer to the set of possible label assignments, $\mathcal{X}$, as a *search space*, and note that this set is of exponential size in the number of variables. In our approach we construct a hierarchy of n additional search sub-spaces of successively lower cardinality by coarsening the graphical model. We denote the complete search space $\mathcal{X}$ by $\mathcal{X}^{(0)}$ and the hierarchy of auxiliary search sub-spaces by $\mathcal{X}^{(1)}, ..., \mathcal{X}^{(n)}$, with $|\mathcal{X}^{(t+1)}| < |\mathcal{X}^{(t)}|$ for all $t = 0, 1, ..., n-1$. We furthermore associate energies with the search spaces, $E^{(0)}, E^{(1)}, ..., E^{(n)}$, with $E^{(t)} : \mathcal{X}^{(t)} \to \mathbb{R}$, and $E^{(0)} = E$. The hierarchy of search spaces is employed to efficiently seek lower energy assignments in $\mathcal{X}$.

2.1. The coarsening procedure

The coarsening procedure is a fundamental module in the construction of coarse scales and we now describe it in detail, see Alg. 1 for an overview. We denote the MRF (or its graph) whose energy we aim to minimize, and its corresponding search space by $G^{(0)}(\mathcal{V}^{(0)}, \mathcal{E}^{(0)}, \phi^{(0)})$ and $\mathcal{X}^{(0)}$, respectively, and use a shorthand notation $G^{(0)}$ to refer to these elements. Scales of the hierarchy are denoted by $G^{(t)}$, $t = 0, 1, 2, ..., n$, such that a larger t corresponds to a coarser scale, with a smaller graph and search space.

We next define the construction of a coarser graph $G^{(t+1)}$ from a finer graph $G^{(t)}$, relate between their search spaces, and define the energy on the coarse graph. To simplify notations we use u, v to denote variables (or vertices) at level t, *i.e.*, $u, v \in \mathcal{V}^{(t)}$, and $\tilde{u}, \tilde{v}$ to denote variables at level $t+1$. A label assignment to variable u (or $\tilde{u}$) is denoted x_u (respectively $x_{\tilde{u}}$). An assignment to all variables at levels t, $t+1$ is denoted by $\mathbf{x} \in \mathcal{X}^{(t)}$ and $\tilde{\mathbf{x}} \in \mathcal{X}^{(t+1)}$, respectively.

Algorithm 1 $[G^{(t+1)}, \mathbf{x}^{(t+1)}] = \textsc{Coarsening}(G^{(t)}, \mathbf{x}^{(t)})$

Input: Graphical model $G^{(t)}$, optional initial labels $\mathbf{x}^{(t)}$
Output: Coarse-scale graphical model and labels $G^{(t+1)}$, $\mathbf{x}^{(t+1)}$

```
$\mathbf{x}^{(t+1)} \leftarrow \emptyset$
select a variable-grouping (Sec. 2.4)
set an interpolation rule $f^{(t+1)} : \mathcal{X}^{(t+1)} \to \mathcal{X}^{(t)}$ (Eq. (2),(3))
if $\mathbf{x}^{(t)}$ is initialized then
    modify $f^{(t+1)}$ to ensure monotonicity (Sec. 2.3)
    $\mathbf{x}^{(t+1)} \leftarrow$ inherits[1] $\mathbf{x}^{(t)}$
end if
define the coarse potentials $\phi^{(t+1)}_{\tilde{u}}, \phi^{(t+1)}_{\tilde{u}\tilde{v}}$ (Eq. (4),(5))
return $G^{(t+1)}$, $\mathbf{x}^{(t+1)}$
```

Variable-grouping and graph-coarsening. To derive $G^{(t+1)}$ from $G^{(t)}$, we begin by partitioning the variables of $G^{(t)}$ into a disjoint set of groups. Then, in each such group we select one vertex to be the "seed variable" (or seed vertex) of the group. As explained below, it is necessary that the seed vertex be connected by an edge to each of the other vertices in its group. Next, we eliminate all but the seed vertex in each group and define the coarser graph, $G^{(t+1)}$, whose vertices correspond to the seed vertices of the fine graph $G^{(t)}$. To set the notation, let $\tilde{v} \in \mathcal{V}^{(t+1)}$ represent a variable of $G^{(t+1)}$, and denote by $[\tilde{v}] \subset \mathcal{V}^{(t)}$ the subset of variables of $\mathcal{V}^{(t)}$ which were grouped to form $\tilde{v}$. The collection of subsets $[\tilde{v}] \subset \mathcal{V}^{(t)}$ forms a partitioning of the set $\mathcal{V}^{(t)}$, *i.e.*, $\mathcal{V}^{(t)} = \cup_{\tilde{v} \in \mathcal{V}^{(t+1)}} [\tilde{v}]$ and $[\tilde{u}] \cap [\tilde{v}] = \varnothing$, $\forall \tilde{u} \neq \tilde{v}$. In Subsection 2.4, we shall provide details on how the groupings are selected.

Once the groups have been selected, the coarse-scale graph topology is determined as follows. An edge is introduced between vertices $\tilde{u}$ and $\tilde{v}$ in $G^{(t+1)}$ if there exists at least one pair of fine-scale vertices, one in $[\tilde{u}]$ and the other in $[\tilde{v}]$, that are connected by an edge. See Fig. 1 for an illustration of variable grouping.

Interpolation rule. Next we define a coarse-to-fine interpolation rule, $f^{(t+1)} : \mathcal{X}^{(t+1)} \to \mathcal{X}^{(t)}$, which maps each labeling assignment of the coarser scale $t+1$ to a labeling of the finer scale t. We consider here only a simple interpolation rule, where the label of any fine variable is completely determined by the coarse-scale variable of its group and is independent of all other coarse variables. That is, if $\mathbf{x} = f^{(t+1)}(\tilde{\mathbf{x}})$ and $x_{\tilde{v}}$ denotes the label associated with coarse variable $\tilde{v}$, then for any fine variable $u \in [\tilde{v}]$, x_u depends only on $x_{\tilde{v}}$. More specifically, if $s \in [\tilde{v}]$ is the seed variable of the group $[\tilde{v}]$, then the interpolation rule is defined as follows:

i. The label assigned to the seed variable s is equal to that

[1] For any coarse-scale variable $\tilde{v}$, $x_{\tilde{v}} \leftarrow x_s$, where x_s is the label of the seed variable of the group $[\tilde{v}]$.

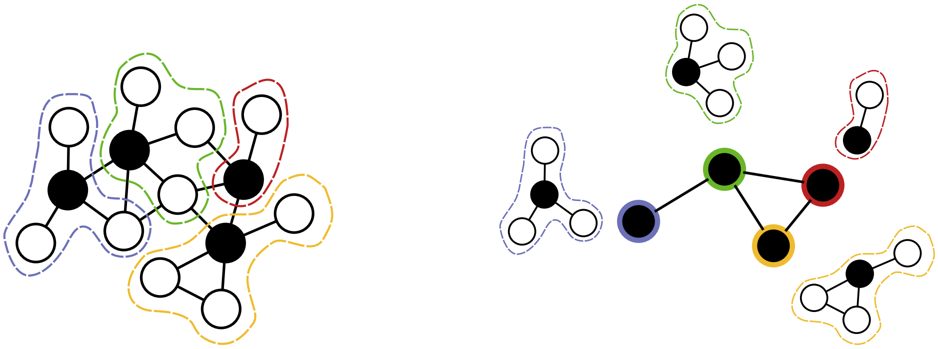

Figure 1. An illustration of variable-grouping, with seed variables denoted by black disks. Note that each seed variable is connected by an edge to each other variable in its group, as required by the interpolation rule. *Right panel:* the coarse graph, whose vertices correspond to the fine-scale seed vertices, and their coarse unary potentials account for all the internal energy potentials in their group. Edges connect pairs of coarse-vertices according to topology at the fine scale.

of the coarse variable,

$$x_s \leftarrow x_{\tilde{v}} \,. \tag{2}$$

ii. Once the seed label has been assigned, the label assigned to any other variable in the group, $u \in [\tilde{v}]$, is that which minimizes the local energy generated by the pair (u, s):

$$x_u \leftarrow \arg\min_x \{\phi_u^{(t)}(x) + \phi_{us}^{(t)}(x, x_s)\}. \tag{3}$$

A single exception to this rule is elaborated in Sec. 2.3. As we produce the coarse graph we store these interpolation assignments (3) in a lookup table so that they can be retrieved when we return to the finer scale. Finally, we henceforth use the shorthand notation $x_u|x_{\tilde{v}}$ to denote that the interpolation rule assigns the label x_u to $u \in [\tilde{v}]$ when the label of $\tilde{v}$ is $x_{\tilde{v}}$.

Coarse-scale energy. The energy associated with $G^{(t+1)}$, denoted $E^{(t+1)}(\tilde{\mathbf{x}})$, depends on the interpolation rule. We require that, for any coarse-scale labeling, the coarse-scale energy be equal to the fine-scale energy obtained after interpolation, that is, $E^{(t+1)}(\tilde{\mathbf{x}}) = E^{(t)}(f^{(t+1)}(\tilde{\mathbf{x}}))$. We call this *consistency*.

To ensure *consistency*, all fine potentials are accounted for exactly once; see Fig. 1. We define the unary potential of a coarse variable $\tilde{v} \in \mathcal{V}^{(t+1)}$ to reflect the internal fine-scale energy of its group $[\tilde{v}]$, according to the interpolation rule:

$$\phi_{\tilde{v}}^{(t+1)}(x_{\tilde{v}}) = \sum_{u \in [\tilde{v}]} \phi_u^{(t)}(x_u|x_{\tilde{v}}) \quad + \tag{4}$$
$$\sum_{u,w \in [\tilde{v}]} \phi_{uw}^{(t)}(x_u|x_{\tilde{v}}, x_w|x_{\tilde{v}}).$$

Note that all the energy potentials of the subgraph induced by $[\tilde{v}]$ are accounted for in Eq. (4). The first term sums up the unary potentials of variables in $[\tilde{v}]$, and the second term takes into account the energy of pairwise potentials of all internal pairs $u, w \in [\tilde{v}]$.

The pairwise potential of a coarse pair $\tilde{u}, \tilde{v} \in \mathcal{V}^{(t+1)}$ accounts for all finer-scale pairwise potentials that have one variable in $[\tilde{u}] \subset \mathcal{V}^{(t)}$ and the other variable in $[\tilde{v}] \subset \mathcal{V}^{(t)}$,

$$\phi_{\tilde{u}\tilde{v}}^{(t+1)}(x_{\tilde{u}}, x_{\tilde{v}}) = \sum_{\substack{u \in [\tilde{u}] \\ v \in [\tilde{v}]}} \phi_{uv}^{(t)}(x_u|x_{\tilde{u}}, x_v|x_{\tilde{v}}). \tag{5}$$

Note that the definition in Eq. (5) is consistent with the topology of the graph, as was previously defined. The coarse scale energy $E^{(t+1)}(\tilde{\mathbf{x}})$ is obtained by summing the unary (4) and pairwise (5) potentials for all coarse variables $\tilde{v} \in \mathcal{V}^{(t+1)}$ and coarse pairs $\tilde{u}, \tilde{v} \in \mathcal{V}^{(t+1)}$.

It is readily seen that consistency is satisfied by the coarsening procedure, by substituting a labeling assignment of $G^{(t+1)}$ into Eqs. (4) and (5) to verify that the energy at scale t of the interpolated labeling is equal to the coarse-scale energy for any interpolation rule. Consistency guarantees that if we reduce the coarse-scale energy, say by applying an inference algorithm to the coarse-scale MRF, then this reduction is translated to an equal reduction in the fine-scale energy via interpolation. Indeed if we minimize the coarse-scale energy and apply interpolation to this solution, then we will have minimized the fine-scale energy over the subset of fine-scale labeling assignments that are in the range of the interpolation, *i.e.*, all label assignments in $f^{(t+1)}(\mathcal{X}^{(t+1)}) \subset \mathcal{X}^{(t)}$.

2.2. The multiscale algorithm

The key ingredient of this paper is the multiscale algorithm which takes after the classical V-cycle employed in multigrid numerical solvers for partial differential equations [5, 21]. We describe it informally first. Our V-cycle is driven by a standard inference algorithm, which is employed at all scales of the hierarchy, beginning with the finest level ($t = 0$), traversing down to the coarsest level ($t = n$), and back up to the finest level. This process comprises a single iteration or cycle. The cycle begins at level $t = 0$ with a given label assignment. One or more iterations of the inference algorithm are applied. Then, a coarsening step is performed: the variables are partitioned into groups, a seed variable is selected for each group, a coarse graph is defined and its variables inherit[1] the labels of the seed variables. This routine of inference iterations followed by coarsening is repeated on the next-coarser level, and so on. Coarsening halts when the number of variables is sufficiently small, say $|\mathcal{V}^{(t)}| < N$, and an exact solution can be easily recovered, *e.g.*, via exhaustive search. The solution of the coarse scale is interpolated to scale $n - 1$, replacing the previous solution of that scale, and some number of inference iterations is performed. This routine of interpolation followed by inference is repeated to the next-finer scale, and so on, until we return to scale 0. As noted above, this completes a single iteration or V-cycle; see illustration in Fig. 2. A formal description in the form of a recursive algorithm appears in Alg. 2. Some remarks follow.

Initialization. The algorithm can be warm-started with any choice of label assignment, $\mathbf{x}^{(0)}$, and with the modifications described in Sec. 2.3 it is guaranteed to maintain or improve its energy. If an initial guess is unavailable our algorithm readily computes a labeling in a coarse-to-fine manner, similarly to existing works. This is done by skipping the inference module that precedes a coarsening step in the V-cycle, *i.e.*, by skipping Step 6 of Alg. 2.

Computational complexity and choice of inference module. The complexity of the algorithm is governed largely by the complexity of the method used as the inference module employed in Steps 6 and 11. Note that the inference algorithm should not be run until convergence, because its goal is not to find a global optimum of the (restricted) search sub-space; rather, a small number of iterations suffice in order to obtain a label assignment for which the interpolation rule heuristic is useful and for which a coarsening step is therefore efficient. The inference method must satisfy two requirements. The first is that the method can be warm-started, otherwise each scale would be solved from scratch without utilizing information passed from other scales of the hierarchy, so we would lose the ability to improve iteratively. Second, the method must be applicable to models with general potentials. Even when the potentials of an initial problem are of a specific type (*e.g.* submodular, semi-metric), it is not guaranteed that this property is conserved in coarser scales due to the construction of the interpolation rule (3) and to the definition of coarse potentials (5). Subject to these limitations we use QPBO-I [20] and LSA-TR [10] for binary models. For multilabel models we use Swap/Expand-QPBO ($\alpha\beta$-swap/α-expand with a QPBO-I binary step) [20] and Lazy-Flipper with a search depth 2 [2].

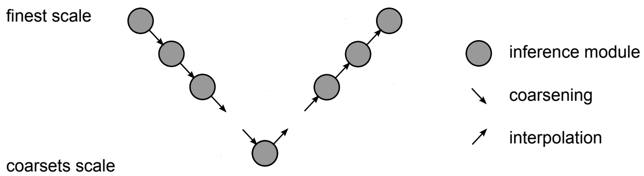

Figure 2. The multiscale V-cycle. Starting at the finest scale $G^{(0)}$ (top left circle), a label assignment is improved by an inference algorithm (Step 6 in Alg. 2) and the graph is coarsened (denoted by an arrow pointing downwards). This repeats until the number of variables is sufficiently small and exact solution can be easily recovered. The labeling is then interpolated to the next finer scale (denoted by an arrow pointing upwards), followed by an application of the inference algorithm. Interpolation and inference repeat until the finest scale is reached. This process, referred to as a V-cycle, constitutes a single iteration.

Algorithm 2 $\mathbf{x} = \text{V-CYCLE}(G^{(t)}, \mathbf{x}^{(t)}, t)$

Input: Graphical model $G^{(t)}$, optional initial labels $\mathbf{x}^{(t)}$, $t \geq 0$
Output: $\mathbf{x}^{(t)}$, a label assignment for all $v \in \mathcal{V}^{(t)}$

```
 1: if $|\mathcal{V}^{(t)}| < N$ then
 2:     compute minimum-energy solution $\mathbf{x}^{(t)}$.
 3:     return $\mathbf{x}^{(t)}$
 4: end if
 5: if $\mathbf{x}^{(t)}$ is initialized then
 6:     $\mathbf{x}^{(t)} \leftarrow$ inference on $G^{(t)}$, $\mathbf{x}^{(t)}$
 7: end if
 8: $G^{(t+1)}$, $\mathbf{x}^{(t+1)} \leftarrow$ COARSENING($G^{(t)}$, $\mathbf{x}^{(t)}$) (Alg. 1)
 9: $\mathbf{x}^{(t+1)} \leftarrow$ V-CYCLE($G^{(t+1)}, \mathbf{x}^{(t+1)}, t+1$) (recursive call)
10: $\mathbf{x}^{(t)} \leftarrow$ interpolate $\mathbf{x}^{(t+1)}$ (Sec. 2.1)
11: $\mathbf{x}^{(t)} \leftarrow$ inference on $G^{(t)}$, $\mathbf{x}^{(t)}$
12: return $\mathbf{x}^{(t)}$
```

Remark. Evidently, the search space of each MRF in the hierarchy corresponds, via the interpolation, to a search subspace of the next finer MRF. When coarsening, we strive to eliminate the less likely label assignments from the search space, whereas more likely solutions should be represented on the coarser scale. The locally minimal energy (3) is chosen with this purpose in mind. It is assumed heuristically that a neighboring variable of the seed variable is more likely to end up with a label that minimizes the energy associated with this pair than with any other choice. The approximation is exact if the subgraph is a star graph.

2.3. Monotonicity

The multiscale framework described so far is not monotonic, due to the fact that the initial state at a coarse level may incur a higher energy than that of the fine state from which it is derived. To see this, let $\mathbf{x}^{(t)}$ denote the state at level t, right before the coarsening stage of a V-cycle. As noted above, coarse-scale variables inherit the current state of seed variables. When we interpolate $\mathbf{x}^{(t+1)}$ back to level t, it may well be the case that the state which we get back to is different from $\mathbf{x}^{(t)}$, *i.e.* $f^{(t+1)}(\mathbf{x}^{(t+1)}) \neq \mathbf{x}^{(t)}$. If the energy associated with $\mathbf{x}^{(t+1)}$ happens to be higher than the energy associated with $\mathbf{x}^{(t)}$ then monotonicity is compromised.

To avoid this undesirable behavior we modify the interpolation rule such that if $\mathbf{x}^{(t+1)}$ was inherited from $\mathbf{x}^{(t)}$ then $\mathbf{x}^{(t+1)}$ will be mapped back to $\mathbf{x}^{(t)}$ by the interpolation. Specifically, assume we are given a labeling $\hat{\mathbf{x}}^{(t)}$ at level t. Consider every seed variable $s \in \mathcal{V}^{(t)}$ and let $\tilde{s}$ denote its corresponding variable in $G^{(t+1)}$. Then, for all $u \in [\tilde{s}]$ we reset the interpolation rule (3),

$$x_u | (x_{\tilde{s}} = \hat{x}_{\tilde{s}}) \leftarrow \hat{x}_u, \tag{6}$$

and coarse energy potentials are updated accordingly to reflect those changes. Now, *consistency* ensures that the initial energy at level $t+1$, $E^{(t+1)}(\hat{\mathbf{x}}^{(t+1)})$, is equal to the energy of $\hat{\mathbf{x}}^{(t)}$ at level t. Consequently, assuming that the inference algorithm which is employed at every level of a V-cycle is monotonic, the energy is non-increasing.

2.4. Variable-grouping by conditional entropy

We next describe our approach for variable-grouping and the selection of a seed variable in each group. Heuristically, we would like v to be a seed variable, whose labeling determines that of u via the interpolation, if we are relatively confident of what the label of u should be, given just the label of v.

Conditional entropy measures the uncertainty in the state of one random variable given the state of another random variable [7]. We use conditional entropy to gauge our confidence in the interpolation rule (3). Exact calculation of conditional entropy,

$$H(u|v) = \sum_{x_u, x_v} P_{uv}(x_u, x_v) \cdot \log \frac{P_v(x_v)}{P_{uv}(x_u, x_v)}, \tag{7}$$

involves having access to the marginal probabilities of the variables and marginalization is in general NP-hard. Instead, we use an approximation of the marginal probabilities for pairs of variables by defining the local energy of two variables u, v,

$$E_{uv}(x_u, x_v) = \phi_u(x_u) + \phi_v(x_v) + \phi_{uv}(x_u, x_v). \tag{8}$$

The local energy is used for the approximation of marginal probabilities by applying the relation $Pr(x_u, x_v) = \frac{1}{Z_{uv}} \cdot \exp\{-E_{uv}(x_u, x_v)\}$, where Z_{uv} is a normalization factor, ensuring that probabilities sum to 1.

Algorithm 3 VARIABLE-GROUPING($G^{(t)}$)

Input: Graphical model $G^{(t)}$ at scale t
Output: A variable-grouping of $G^{(t)}$

```
initialize: SCORES $= \emptyset$, SEEDS $= \emptyset$, VARS $= \mathcal{V}^{(t)}$
for each edge $(u, v) \in \mathcal{E}^{(t)}$ do
    calculate $H(u|v)$ for $(v, u)$, and $H(v|u)$ for $(u, v)$
    store the (directed) pair at the respective bin in SCORES
end for
while VARS $\neq \emptyset$ do
    pop the next edge $(u, v) \in$ SCORES
   // check if we can define $u$ to be $v$'s seed
    if ($v \in$ VARS & $u \in$ VARS $\cup$ SEEDS) then
        set $u$ to be $v's$ seed
        SEEDS $\leftarrow$ SEEDS $\cup \{u\}$
        VARS $\leftarrow$ VARS $\setminus \{u, v\}$
    end if
end while
```

Our algorithm for selecting subgraphs and their respective seed variable is described below, see also Alg. 3. First, the local conditional entropy is calculated for all edges in both directions. A directed edge, whose direction determines that of the interpolation, is binned in a score-list according to its local conditional entropy score. We then proceed with the variable-grouping procedure; for each variable we must determine its status, namely whether it is a seed variable or an interpolated variable whose seed must be determined. This is achieved by examining directed edges one-by-one according to the order by which they are stored in the binned-score list. For a directed edge (u, v) we verify that the intended-to-be interpolated variable v has not been set as a seed variable nor grouped with a seed variable. Similarly, we ensure that the status of the designated seed variable u is either undetermined or that u has already been declared a seed variable. The process terminates when the status of all the variables has been set. As a remark, we point to the fact that the score-list's range is known in advance (it is the range of feasible entropy scores) and that no ordering is maintained within its bins. The motivation to use a binned-score list is twofold: refrain from sorting the score-list and thus maintain a linear complexity in the number of edges, and introduce randomization to the variable-grouping procedure. In our experiments we fixed the number of bins to 20.

3. Evaluation

The algorithm was implemented in the framework of OpenGM [1], a C++ template library that offers several inference algorithms and a collection of datasets to evaluate on. We use QPBO-I [20] and LSA-TR [10] for binary models and Swap/Expand-QPBO ($\alpha\beta$-swap/α-expand with a QPBO-I binary step) and Lazy-Flipper with a search depth of 2 [2] for multilabel models. The recursion was halted when the number of variables reached $N \leq 2$. Unless otherwise indicated, 3 V-cycles were applied on "hard" energy models (Sec. 3.1) and a single V-cycle on Potts models (Sec. 3.2).

Recent benchmark studies indicate that there is no one algorithm that performs best on all types of models [12, 13]. Even when the model type is restricted to multilabel Potts, different algorithms are optimal under different circumstances, see for example Table 5. The algorithm presented in this work has been designed with the intention of optimizing "hard" energies, *e.g.*, when neighboring variables are allowed to have negative affinities (repulsive potentials), or arbitrary pairwise potentials in the general case. These conditions may arise in various applications such as boundary-driven image segmentation [3], or when the parameters of the model are learned from data [19].

Our goal is to establish that applying an inference method within the proposed multiscale framework is superior to applying it in a single-scale fashion. Ideally, this point would be demonstrated on any inference method. As discussed earlier, not all algorithms are immediately applicable in our framework. Hence, we resort to comparing multiscale to single-scale inference for algorithms which can be applied in our framework without modifications. For each dataset we report also the "Ace" inference method for that dataset, where algorithms are ranked according to the percentage of instances on which they achieve the best energy and by their run-time.

3.1. Hard energies

The notion of hardness of optimization in MRFs is informally discussed in [12]. A problem is considered hard if it involves optimizing over many variables, if the graph is highly-connected, or if it includes frustrated cycles. Concretely, the datasets are split into 3 categories: those for which (all/some/none) of the instances are solved to optimality. We follow these notions when we refer to hard models, with special attention to the type of pairwise interaction.

The collection of hard datasets that we tested includes 10 datasets. Results are aggregated and presented in Table 1. Note how different models have different algorithms that are best suit for them. Our multiscale algorithm improves over single scale on the vast majority of these datasets and achieves state of the art results on many of the instances.

The *Chinese Character Inpainting* dataset includes 100 instances of 64-connected grids with approximately 10^4 binary variables, and the parameters of the model are learned from data [19]. One instance from the dataset is displayed in Fig. 3. We further compare our approach to a coarse-to-fine method that uses dynamic variable grouping [4], and to a unified algorithm for message-passing in cluster graphs [22]. Detailed results are presented in Table 2.

		Multiscale	Single scale	"Ace"
scribble	method	QPBO	QPBO	ogm-LBP-LF2
[10 instances]	value	-192130.49	0	-184011.16
	time	100.05sec	0.01sec	152.94sec
	best	100%	0%	20%
chinese	method	LSA-TR	LSA-TR	MCBC-pct
[100 instances]	value	-49549.89	-49548.1	-49550.1
	time	23.7sec	0.05sec	2053sec
	best	53%	22%	66% (56%)
pic-grids	method	QPBO	QPBO	ogm-ILP-pct
[21 instances]	value	-19883.12	-10.33	-20108.61
	time	1.9sec	0.01sec	3130sec
	best	65%	0%	86% (19%)
pic-obj-det	method	Lazy-Flipper	Lazy-Flipper	ogm-TRBP-0.95
[37 instances]	value	-18009.94	-15333.18	-18776.75
	time	8.84sec	7.6sec	58.16sec
	best	30%	10%	76%
pic-DBN	method	LSA-TR	LSA-TR	ogm-LBP-0.95
[6 instances]	value	-2625.41	-1631.2	-2811.27
	time	2.52sec	0.1sec	269sec
	best	0%	0%	83%
matching	method	Lazy-Flipper	Lazy-Flipper	ogm-ATSAR
[4 instances]	value	31.33	40.79	21.22
	time	0.8sec	0.4sec	0.8sec
	best	25%	0%	100% (100%)
modularity	method	Swap	Swap	MCR-CCFDB-OWC
[6 instances]	value	-0.4383	-0.327	-0.4652
	time	14.1sec	3.6sec	602sec
	best	17%	0%	83% (83%)
knott-3d-150	method	Swap	Swap	MCI-CCIFD
[8 instances]	value	-4421.15	-3718.94	-4571.69
	time	634	317	0.6
	best	0%	0%	100% (100%)
stereo	method	Expand	Expand	ogm-CombiLP
[3 instances]	value	1618904	1618904	1587560.67
	time	138sec	35sec	835sec
	best	0%	0%	66% (66%)
photomontage	method	Swap	Swap	mrf-a-Exp-TAB
[2 instances]	value	190538	190538	168457
	time	42.1sec	11.5sec	7.4sec
	best	0%	0%	100%

Table 1. Results for hard energy models. The first column indicates the name of a dataset and number of instances. What follows is a comparison between multiscale and single-scale inference, as well as a comparison to the "Ace" method for each dataset. We report the average energy (*value*), run-time (*time*) and the percentage of instances on which the algorithm reported the best energy (*best*), which sums to more than 100% in case of ties. Enclosed in brackets at the *best* field is the percentage of instances in which the "Ace" method had provided a certificate of global optimality.

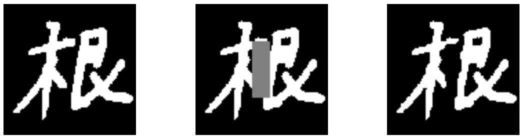

Figure 3. An instance from the *Chinese Character Inpainting* dataset. *Left panel:* ground truth image. *Center:* masked image *Right panel:* result of applying LSA-TR within our framework.

algorithm	mean runtime	mean value	best
MCBC-pct	2053.89 sec	-49550.1	66%
ogm-ILP-pct	3553.71 sec	-49547.41	49%
ogm-LF-3	637.92 sec	-49535.37	13%
SA		-49533.02	12%
BPS-TAB	62.69 sec	-49537.08	10%
Bagon&Galun [4]	41.8 sec	-49547.65	10%
ogm-TRWS-LF	83.78 sec	-49519.42	6%
ogm-LBP-0.5	482.00 sec	-49509.81	3%
Wang&Koller [22]	3600 sec	-49526.49	3%
LSA-TR (euc.)	0.05 sec	-49548.1	22%
LSA-TR (multiscale), 5 V-cycles	12.4 sec	-49549.68	40%
LSA-TR (multiscale), 10 V-cycles	23.7 sec	-49549.89	53%
QPBO	0.14 sec	49501.95	0%
QPBO (multiscale)	8.85 sec	-49542.78	9%
Lazy-Flipper	13 sec	-49531.11	5%
Lazy-Flipper (multiscale)	41.9 sec	-49544.82	11%

Table 2. Performance on the *Chinese Character Inpainting* dataset. Five (*ten*) V-cycles of multiscale inference with LSA-TR reported the best energy on 40% (*53%*) of the dataset with outstanding run-times. Energy and run-times are as reported in [12]. The *best* value can sum to more than 100% in case of ties.

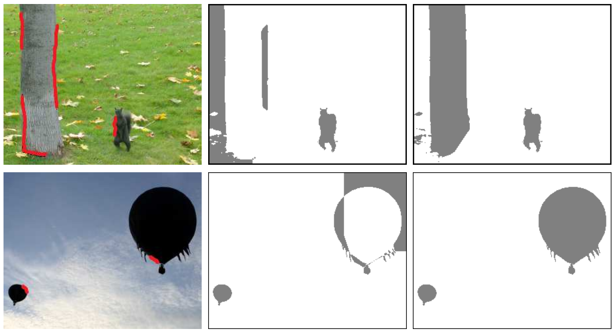

Figure 4. Two instances from the *Scribble* dataset. *Left panel:* user-annotated image. *Center:* segmentation results of single-scale inference using LSA-TR. *Right panel:* segmentation attained with 1 V-cycle of our algorithm using LSA-TR.

The *Scribble* dataset [3] is an image segmentation task with a user-interactive interface, in which the user is asked to mark boundaries of objects in the scene (see Fig. 4). These boundaries are used for determining negative affinities between pixels on different sides of the boundary. The dataset contains 10 instances of approximately $7 \cdot 10^4$ variables and $1.5 \cdot 10^6$ edges each. Detailed results for this dataset are reported in Table 3.

Datasets denoted with a *pic-* prefix were adapted from the *Probabilistic Inference Challenge* (PIC2011) [8]. Two of these involve a binary label space (*grids, DBN*), and the third (*obj-det*) is a multilabel model with up to 21 labels. Detailed results for *obj-det* are presented in Table 4.

algorithm	mean runtime	mean value	best
ogm-LBP-LF2	152.94 sec	-184011.16	20%
ogm-ADSAL	476.17 sec	-183790.07	0%
TRWS-pct	3.62 sec	-170092.79	0%
ogm-ILP-pct	3260.65 sec	-3147	0%
Lazy-Flipper	151.99 sec	-172753.21	0%
Lazy-Flipper (multiscale)	227.96 sec	-191389.34	70%
LSA-TR	196.73 sec	-188332.05	0%
LSA-TR (multiscale)	87.10 sec	-191356.5	90%
QPBO	0.01 sec	0	0%
QPBO (long)	645.04 sec	-190614.31	80%
QPBO (multiscale)	100.05 sec	-192130.49	100%

Table 3. Results for the *Scribble* dataset. We compare single-scale and multiscale inference and report results for a selection of competitive methods that were not incorporated in our framework. On this challenging large-scale dataset, multiscale inference was repeatedly superior to single-scale. *QPBO (long)* denotes a single-scale, iterative application of QPBO-I; this highlights the advantage of multiscale inference, as even when QPBO-I was run exhaustively it came short compared with multiscale inference.

algorithm	mean runtime	mean value	best
ogm-TRBP-0.95	58.16 sec	-18776.75	75.68%
ogm-TRBP-0.5	56.79 sec	-18909.78	67.57%
ogm-LBP-LF2	10.56 sec	-18909.47	64.86%
BPS-TAB	2.43 sec	-18910.84	64.86%
ogm-LBP-0.5	6.17 sec	-18907.58	64.86%
ogm-LBP-0.95	2.25 sec	-18310.95	59.46%
ogm-ILP	3598.39 sec	-11728.67	0%
Lazy-Flipper	7.58 sec	-15333.18	10.81%
Lazy-Flipper (multiscale)	8.84 sec	-18009.94	29.73%
Swap-QPBO	0.15 sec	-10832.74	0%
Swap-QPBO (multiscale)	0.17 sec	-14375.89	0%

Table 4. Results for the *pic-obj-det* dataset. We compare single-scale and multiscale inference and report results for a selection of competitive methods that were not incorporated in our framework. Multiscale inference reached significantly better energies than single-scale with a slight overhead in run-time.

3.2. Multilabel Potts model

Multilabel Potts datasets constitute a large portion of the OpenGM benchmark but with only a small number of instances for each dataset. In terms of the "hardness" notions mentioned earlier, these models can be generally considered easy.

Indeed, as can be seen in Table 5, nearly all of the instances are solved to optimality by existing methods. Even so, there is no one inference method that is consistently ranked as "Ace". While our algorithm does not qualify as the top method for multilabel Potts model, it still improves significantly over single-scale inference.

4. Discussion

We have presented a multiscale framework for MRF energy minimization that uses variable grouping to form coarser levels of the problem. Our first contribution is the advancement from a coarse-to-fine algorithm to a monotonic, full multiscale algorithm. The advantage here is twofold: the algorithm can be applied iteratively, and it can be warm-started by an initial guess. Furthermore, we offer to group variables regardless of whether they are assumed to take the same label at the minimum energy or not, but rather based on the magnitude of their statistical correlation.

We demonstrated these concepts with an algorithm that groups variables based on a local approximation of their conditional entropy, namely based on an estimate of their statistical correlation. The algorithm was evaluated on a collection of datasets and results indicate that it is beneficial to apply existing single-scale methods within the presented multiscale algorithm. Furthermore, on some of the datasets results of our algorithm compete with state of the art.

A possible drawback of our framework is that it is restricted to inference methods that can be warm-started. Specifically, not every algorithm can be applied within the framework. Adapting existing inference methods to render them applicable in the presented framework is a direction for future work.

There are many possible directions for further developments, beginning with the interpolation rule. The rule that we present in this work is nothing but a local approximation that relies on the pairwise energy. A simple improvement may be to take into account the pairwise potentials between non-seed variables in the group. In fact, seed variables that coincide with coarse variables is not an essential feature; only the correspondence between a group and its coarse variable is essential. Indeed, even the set of labels can be expanded on a coarse scale to enrich the coarse search sub-space. Consider for example the following interpolation rule for a coarse-variable v: evaluate different labeling assignments to the variables of the induced sub-graph $[v] \subset \mathcal{V}^{(t)}$, and sort the evaluated assignments according to their energy value. Now, select the k assignments of lowest energy values; produce a set of k labels for v, and set the interpolation rule to map each of the k labels to one of the selected assignments. In this way potentially high-energy labeling assignments are discarded from coarse search sub-spaces because they are no longer represented at coarse scales. Generally, it may well be the case that different interpolation rules will be better-suited for different models, just as different inference methods are good for different models.

		Multiscale	Single scale	“Ace”
inpainting-n4				mrf-a-Exp-TL
[2 instances]	value	472.81	3453.29	454.35
	time	0.9sec	0.2sec	0.02sec
	best	0%	0%	100%
inpainting-n8				FastPD-pct
[2 instances]	value	469.11	3451.36	464.76
	time	2.56sec	0.9sec	0.26sec
	best	0%	0%	100%
color-seg-n4				ogm-CombiLP
[9 instances]	value	20208.47	23775.38	20012.14
	time	35.6sec	11.7sec	42.11sec
	best	0%	0%	100% (100%)
color-seg-n8				ogm-CombiLP
[9 instances]	value	20190.71	21129.07	19991.21
	time	84sec	39.8sec	128.77sec
	best	0%	0%	100% (100%)
object-seg				TRWS-pct
[5 instances]	value	31745.58	64937.24	31317.23
	time	10.3sec	2.6sec	1.15sec
	best	0%	0%	100% (100%)
color-seg				MCR-pct
[3 instances]	value	308482942.67	309850181	308472274.33
	time	45sec	18.7sec	0.8sec
	best	0%	0%	100% (100%)
brain-3mm				MCI-pct
[4 instances]	value	25214041	25238328.25	25162493
	time	737sec	123sec	27.3sec
	best	0%	0%	100% (100%)
brain-5mm				MCI-pct
[4 instances]	value	19124827.5	19140692.25	19087612.5
	time	498sec	81sec	25.6sec
	best	0%	0%	100% (100%)
brain-9mm				MCI-pct
[4 instances]	value	9201092	9207084	9185280.75
	time	233sec	38sec	8.32sec
	best	0%	0%	100% (100%)

Table 5. Results for multilabel Potts datasets. The first column indicates the name of a dataset and the number of instances. What follows is a comparison between multiscale and single-scale inference of Lazy-Flipper, as well as a comparison to the “Ace” method for each dataset. We report the average energy (*value*) and run-time (*time*), the percentage of instances on which an algorithm reported the best energy (*best*) and provided a certificate of global optimality (enclosed in brackets). Note that nearly all of the instances are solved to optimality, and that different inference methods perform well on different datasets.

References

[1] B. Andres, T. Beier, and J. H. Kappes. OpenGM: A C++ library for discrete graphical models. *arXiv preprint arXiv:1206.0111*, 2012. 5

[2] B. Andres, J. H. Kappes, T. Beier, U. Köthe, and F. A. Hamprecht. The Lazy Flipper: Efficient depth-limited exhaustive search in discrete graphical models. In *Computer Vision–ECCV 2012*, pages 154–166. Springer, 2012. 4, 6

[3] S. Bagon. Boundary driven interactive segmentation. In *Information Science and Applications (ICISA), 2012 International Conference on*, pages 1–5. IEEE, 2012. 6, 7

[4] S. Bagon and M. Galun. A multiscale framework for chal-

lenging discrete optimization. *NIPS Workshop on Optimization for Machine Learning*, 2012. 1, 6, 7

[5] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977. 3

[6] B. Conejo, N. Komodakis, S. Leprince, and J. P. Avouac. Inference by learning: Speeding-up graphical model optimization via a coarse-to-fine cascade of pruning classifiers. In *Advances in Neural Information Processing Systems*, pages 2105–2113, 2014. 1

[7] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 5

[8] G. Elidan, A. Globerson, and U. Heinemann. The Probabilistic Inference Challenge (PIC2011), 2011. 7

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54, 2006. 1

[10] L. Gorelick, Y. Boykov, O. Veksler, I. B. Ayed, and A. Delong. Submodularization for binary pairwise energies. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1154–1161. IEEE, 2014. 2, 4, 5

[11] F. Heitz, P. Perez, and P. Bouthemy. Multiscale minimization of global energy functions in some visual recovery problems. *CVGIP: image understanding*, 59(1):125–134, 1994. 1

[12] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. 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. *International Journal of Computer Vision*, pages 1–30, 2015. 6, 7

[13] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, J. Lellmann, N. Komodakis, and C. Rother. A comparative study of modern inference techniques for discrete energy minimization problems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 6

[14] T. Kim, S. Nowozin, P. Kohli, and C. D. Yoo. Variable grouping for energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1913–1920. IEEE, 2011. 1

[15] P. Kohli, V. Lempitsky, and C. Rother. Uncertainty driven multi-scale optimization. In *Proceedings of the 32nd DAGM conference on Pattern recognition*, pages 242–251. Springer-Verlag, 2010. 1

[16] N. Komodakis. Towards more efficient and effective LP-based algorithms for MRF optimization. In *Computer Vision–ECCV 2010*, pages 520–534. Springer, 2010. 1

[17] L. J. Latecki, C. Lu, M. Sobel, and X. Bai. Multiscale random fields with application to contour grouping. In *Advances in Neural Information Processing Systems*, pages 913–920, 2009. 1

[18] H. Lombaert, Y. Sun, L. Grady, and C. Xu. A multilevel banded graph cuts method for fast image segmentation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 259–265. IEEE, 2005. 1

[19] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli. Decision tree fields. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1668–1675. IEEE, 2011. 6

[20] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 2, 4, 5

[21] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Academic press, 2000. 3

[22] H. Wang and K. Daphne. Subproblem-tree calibration: A unified approach to max-product message passing. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 190–198, 2013. 6, 7