A discriminative view of MRF pre-processing algorithms –supplementary material

Chen Wang[1,2] Charles Herrmann[2] Ramin Zabih[1,2]
[1] Google Research [2] Cornell University
{chenwang, cih, rdz}@cs.cornell.edu

1. Outline of supplementary material

We will give a detailed running time analysis of our proposed algorithm in Section 2. Then we will give the proof to Lemma 4 and Theorem 10 in Section 3 and Section 4 respectively. Generalization of the efficient discriminative criterion check subroutine will be described in Section 5. More implementation details will be given in Section 6. Finally, we will provide more experimental data in Section 7, including visualization results, experimental results on a typical parameter setup, more investigation on parameters sensitivity, the role of worst case bound in practice and preliminary results on multilabel MRFs.

2. Running time analysis

Algorithm 1: MRF inference with pre-processing

```
   Input: Energy function $E(x)$
1  $\hat{x} \leftarrow \emptyset$;    $S \leftarrow \emptyset$;
2  for $t \leftarrow 1$ to $\tau$ do
3      for $i \in V \backslash S, \ell \in \mathcal{L}_i$ do
4          Compute $LB \leq \sum_{z_{\mathcal{N}(i)} \in \hat{\mathcal{L}}_{\mathcal{N}(i)}(x_i = \ell)} q(z_{\mathcal{N}(i)})$;
5          if $LB \geq \kappa$ then
6              $\hat{x} \leftarrow \hat{x} \oplus \{x_i = \ell\}$;
7              $\mathcal{L}_i \leftarrow \{\ell\}$;    $S \leftarrow S \cup \{i\}$;
8          end
9      end
10 end
11 With $\hat{x}_S$ fixed, use one MRF inference algorithm to solve the remaining variables, get $\hat{x}_{V \backslash S}$;
12 return $\hat{x} = \hat{x}_S \oplus \hat{x}_{V \backslash S}$;
```

The pseudo-code of our proposed algorithm is listed in Algorithm 1. It's the same pseudo-code we have in the main paper.

We will give a asymptotic analysis on the running time of our pre-processing algorithm here. Assuming we have an oracle to give us data term $\theta_i(x_i)$ and prior term value $\theta_{ij}(x_i, x_j)$ in $\mathcal{O}(1)$ time. Let $N = |V|, M = |E|$ and $L = \max_i |\mathcal{L}_i|$ to be the number of variables, edges and maximum possible labels, $d = \max_i |\mathcal{N}(i)|$ is the maximum degree of the graph. For a typical vision problem, we usually have a sparse graph like grid, meaning $\mathcal{O}(N) = \mathcal{O}(M)$ and d is also usually a small constant like 4 or 8.

Computation time of the for loop from line 2 to 10 needs some thinking. τ is usually a small constant, so we can omit it in the asymptotic analysis. For the given $x_i = \ell$, a naive implementation of brute force algorithm to compute $\sum_{z_{\mathcal{N}(i)} \in \hat{\mathcal{L}}_{\mathcal{N}(i)}} q(z_{\mathcal{N}(i)})$ needs to enumerate all the possible neighboring configurations $z_{\mathcal{N}(i)}$, and it takes $\mathcal{O}(dL)$ to compute $\min_{y_i \neq x_i} \Delta E(y_i \leftarrow x_i \mid z_{\mathcal{N}(i)})$, so it takes $\mathcal{O}(dL^{d+1})$ time. Therefore, the overall running time is $\mathcal{O}(dNL^{d+2})$ for brute force so it's still feasible when both d and L are small constant.

When we use the approximated way to compute the lower bound using Lemma 5 in the main paper, we need an faster way to compute Eq. 9. We can pre-compute all the terms we may used here in $\mathcal{O}(NL + EL^2)$ time globally and then query it in $\mathcal{O}(d)$ time without solving the min operator each time. Then it takes $\mathcal{O}(d^2L)$ time to compute $\mathcal{A}_j$, $\mathcal{O}(dL)$ time to compute Q_i and $\mathcal{O}(d)$ to compute the sum each iteration. Also note that once we fix a variable, it also takes $\mathcal{O}(L + dL^2)$ to update our pre-computations result. But each variable will only be fixed at most once during the pre-processing, so the amortized running time to update the pre-computations result is $\mathcal{O}(NL + EL^2)$. So in sum, we have the overall running time $\mathcal{O}(d^2NL^2 + EL^2)$ for approximated calculation.

3. Proof of Lemma 4

Lemma 4. *For the same set of decision problems for persistency, we will never increase the number of false positives by increasing κ.*

Proof. This one is trivial. Consider any non-persistent x_S, it will be a false positive with parameter κ_2 if and only if it meets our discriminative criterion, i.e., $\sum_{z_{\mathcal{N}(S)} \in \hat{\mathcal{L}}_{\mathcal{N}(S)}} q(z_{\mathcal{N}(S)}) \geq \kappa_2$. Now for the algorithm using parameter $\kappa_2 > \kappa_1$, our discriminative criterion still holds, hence it's still a false positive for our algorithm with parameter κ_1. □

4. Proof of Theorem 10

Theorem 10. *Suppose we use expansion movess as the inference algorithm, with the β-multiplicative bound, then we will have $E(\hat{x}) \leq \beta \cdot E(x^*) + |S|\epsilon$.*

Proof. Following the proof of the multiplicative bound of expansion moves algorithm [1] (Theorem 6.1), we will see actually the multiplicative factor β will not be applied to unary terms. In other words, $E'(\hat{x}) = \sum_i \theta'_i(\hat{x}_i) + \sum_{ij} \theta'_{ij}(\hat{x}_i, \hat{x}_j) \leq \sum_i \theta'_i(x^*_i) + \beta \sum_{ij} \theta'_{ij}(x^*_i, x^*_j) \leq \beta E'(x^*)$.

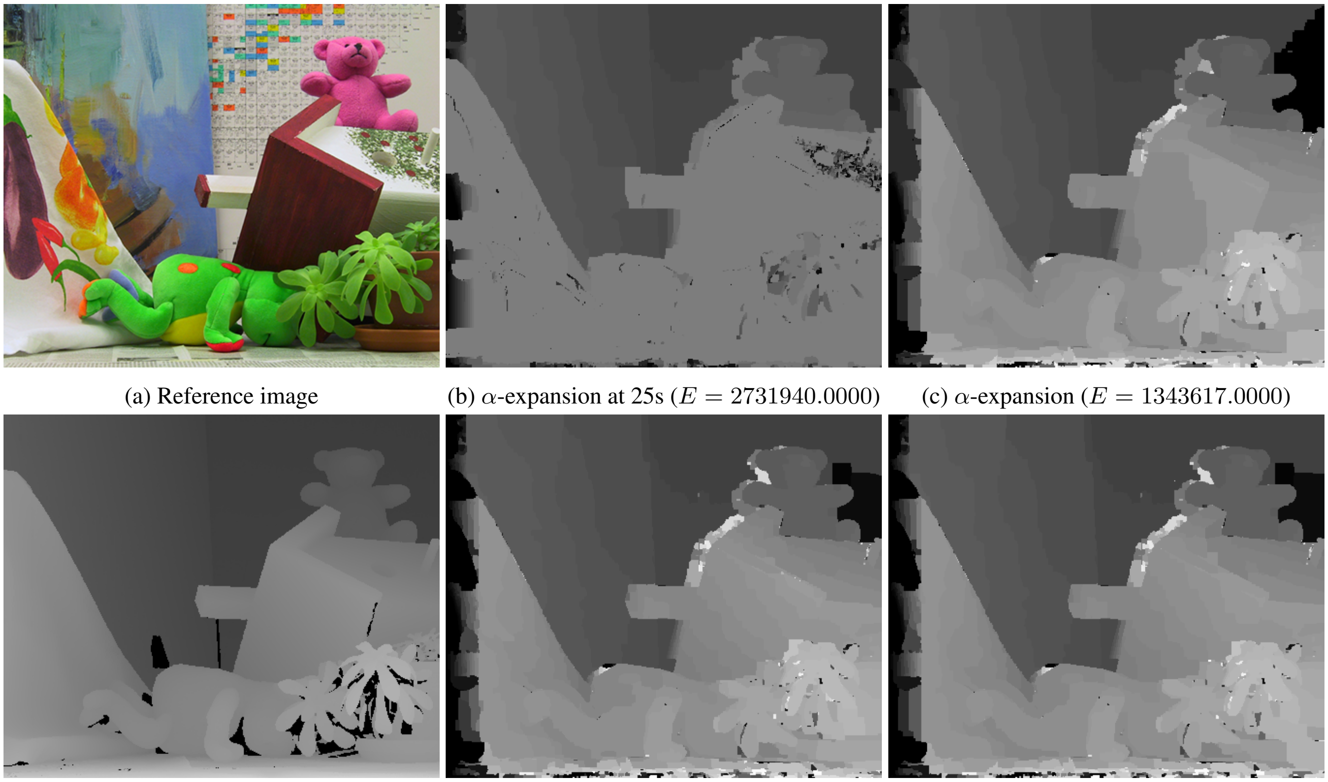
Note that in our algorithm, the energy function $E'(x)$ of expansion moves is induced by fixing $\hat{x}_S$ in $E(x)$, all the pairwise terms θ_{ij} crossing S and $V \backslash S$ could be viewed as the unary terms in $E'(x)$ since one variable will be fixed. Therefore, we will have following.

$$\begin{aligned}
&E(\hat{x}_S \oplus \hat{x}_{V\backslash S}) \\
=&\sum_{i\in S}\theta_i(\hat{x}_i) + \sum_{i,j\in S,(i,j)\in E}\theta_{ij}(\hat{x}_i,\hat{x}_j) + \sum_{i\in S,j\in V\backslash S,(i,j)\in E}\theta_{ij}(\hat{x}_i,\hat{x}_j) + \sum_{i\in V\backslash S}\theta_i(\hat{x}_i) + \sum_{i,j\in V\backslash S,(i,j)\in E}\theta_{ij}(\hat{x}_i,\hat{x}_j) \\
\leq&\sum_{i\in S}\theta_i(\hat{x}_i) + \sum_{i,j\in S,(i,j)\in E}\theta_{ij}(\hat{x}_i,\hat{x}_j) + \sum_{i\in S,j\in V\backslash S,(i,j)\in E}\theta_{ij}(\hat{x}_i,x^*_j) + \sum_{i\in V\backslash S}\theta_i(x^*_i) + \beta\sum_{i,j\in V\backslash S,(i,j)\in E}\theta_{ij}(x^*_i,x^*_j) \\
\leq&\sum_{i\in S}\theta_i(x^*_i) + \sum_{i,j\in S,(i,j)\in E}\theta_{ij}(x^*_i,x^*_j) + \sum_{i\in S,j\in V\backslash S,(i,j)\in E}\theta_{ij}(x^*_i,x^*_j) + \sum_{i\in V\backslash S}\theta_i(x^*_i) + \beta\sum_{i,j\in V\backslash S,(i,j)\in E}\theta_{ij}(x^*_i,x^*_j) + |S|\epsilon \\
\leq&\beta \cdot E(x^*) + |S|\epsilon.
\end{aligned} \tag{1}$$

□

5. Generalization of the efficient check of discriminative criterion

When we want to decide if the given partial labeling x_S is persistent or not, we can follow exactly the same idea presented in Section 3.3 of the main paper to compute the lower bound of $\sum_{z_{\mathcal{N}(S)} \in \hat{\mathcal{L}}_{\mathcal{N}(S)}} q(z_{\mathcal{N}(S)})$. The only big difference is that we need a subroutine to efficiently check $\min_{y_S \neq x_S} \Delta E(y_S \leftarrow x_S \mid z_{\mathcal{N}(S)}) > 0$ for $z_{\mathcal{N}(S)} \in \mathcal{L}_{\mathcal{N}(S)}$ with $z_j = \ell$. Persistency relaxation (PR) [11] generalizes dead end elimination (DEE) [2] from checking persistency of a single variable x_i to an independent local minimum (ILM) partial labeling x_S. The subproblem in PR is to decide if $\min_{y_S \neq x_S} \Delta E(y_S \leftarrow x_S \mid z_{\mathcal{N}(S)}) > 0$ for $z_{\mathcal{N}(S)} \in \mathcal{L}_{\mathcal{N}(S)}$, without the additional constraint that $z_j = \ell$, and they proposed a bunch of sufficient conditions to efficiently check it. Actually, it's trivial to enforce the additional constraint $z_j = \ell$ in those approaches. We just need to remove z_j from the free variables and force it takes value ℓ in the subroutine proposed in PR. Note that those subroutines are sound so we can still apply Lemma 5 to partial labeling x_S and get the lower bound of $\sum_{z_{\mathcal{N}(S)} \in \hat{\mathcal{L}}_{\mathcal{N}(S)}} q(z_{\mathcal{N}(S)})$. Once we have our discriminative criterion as the decision subroutine, we can follow the construction algorithm in PR (Algorithm 2) as the generalization of our proposed construction algorithm in the main paper.



(a) Reference image (b) α-expansion at 25s ($E = 2731940.0000$) (c) α-expansion ($E = 1343617.0000$)

(d) Ground truth (e) Ours at 25s ($E = 1418450.0000$) (f) Ours ($E = 1391959.0000$)

Figure 1: Stereo instance `Teddy`

6. More implementation details

Since we applied the proposed method to each induced binary subproblem in the expansion moves algorithm, we only check persistency for $x_i = 0$ (i.e., do not take move in the binary case) after the first epoch of running expansion moves algorithm in order to get the maximum speedup. We observed that after the first epoch, most of the variables won't change its value, hence the extra benefit from checking persistent for $x_i = 1$ is very marginal.

7. Additional experimental results

7.1. Visualization results

We presented the visualization results on the stereo task in Fig. 1. We can see there is no significant visual difference between the expansion moves results and our results, even in the case that our method has slightly higher energy. Therefore, it's appealing to apply our method in practice, since it has almost the same visual quality but makes the inference much faster. When we set up a limited time budget in real applications, see the second column of Fig. 1, our approach can generate much better visual result than regular expansion moves algorithm without pre-processing. In this case, regular expansion moves even doesn't finish its first epoch and has a very poor disparity map.

7.2. Experimental results with a typical parameter setup

Our experiments suggest that the proposed method can achieve good performance with the parameters in a wide range. We report the experimental results in Table 1 with the following fixed parameters to avoid the expense of cross-validation: $\kappa = 0.8, \tau = 3$, using the uniform distribution for $q(x)$ and checking with our efficient subroutine described in Section 3.3 of the main paper.

Even though this is a fairly conservative assumption (we use the exact same parameters for very different energy functions), we still obtain good results. We acheive a 2x-12x speedup on different datasets with the energy increasing 0.1% on the worst

Table 1: Performance of our method on a typical parameter setup

Typical parameter setup (w/o cross validation)							
Dataset	**Stereo**	**Inpainting**	**Denoising-sq**	**Denoising-ts**	**Optical Flow**	**Color-seg-n4**	**Color-seg-n8**
Speedup	2.14x	2.10x	11.71x	10.61x	8.92x	9.31x	8.45x
Energy Change	-0.04%	-0.53%	-0.03%	-0.09%	+0.11%	+0.01%	+0.05%
Labeled Vars	56.77%	47.06%	97.39%	96.64%	93.74%	90.80%	90.43%
Precision	99.71%	99.88%	99.95%	99.95%	99.50%	99.50%	99.76%
Leave one out parameter selection (w/ cross validation)							
Dataset	**Stereo**	**Inpainting**	**Denoising-sq**	**Denoising-ts**	**Optical Flow**	**Color-seg-n4**	**Color-seg-n8**
Speedup	1.78x	3.40x	11.83x	11.91x	4.69x	7.02x	8.33x
Energy Change	-0.06%	-1.71%	-0.02%	0.00%	-0.04%	0.00%	+0.04%
Labeled Vars	44.76%	74.29%	97.91%	98.32%	77.25%	85.74%	90.39%
Precision	99.74%	96.16%	99.95%	99.79%	99.88%	99.79%	99.77%

Table 2: Parameters chosen from the leave-one-out procedure

Dataset	κ	Choice of q	criterion check	Exception
Stereo	0.8	uniform	approximate	none
Inpainting	0.7	uniform	approximate	1 instance with unary distribution
Denoise-sq	0.8	uniform	approximate	1 instance with $\kappa = 0.9$
Denoise-ts	0.7	uniform	approximate	1 instance with $\kappa = 0.8$
Optical Flow	0.9	unary	exact	1 instance with $\kappa = 0.8$, approximate check
Color-seg-n4	0.9	unary	exact	1 instance with $\kappa = 0.8$, uniform distribution, approximate check
Color-seg-n8	0.9	unary	exact	1 instance with $\kappa = 0.8$

case. In addition, we still get lower energy on 4 of the 5 challenging dataset.

We also listed the performance of our method with the parameters selected with the leave-one-out cross validation procedure as a reference (shown in Table 2 in the main paper). We see that the performance of our method is very similar no matter whether we use fixed parameters or use cross validation to choose the parameters. The key observation of the main paper still holds even with this fixed typical parameter setup, i.e., our method achieves significant speedup against baseline methods with very minor compromise on the accuracy of the partial optical labelings (usually lose $< 0.5\%$ precision). We also achieve comparable or smaller energy even though we compromise the accuracy of the partial optical labelings in the pre-processing step.

Therefore, these experiments demonstrate that it's sufficient to use the typical parameter setup of our method in practice. We can achieve very good performance without using the expensive cross validation parameter selection procedure.

7.3. Investigation on parameter sensitivity

We claimed in the main paper that the parameters chosen by the leave-one-out procedure are very similar for the same dataset. We summarized the parameters chosen by cross validation in Table 2. The exception column shows that, out of the 7 datasets we tested, the leave-one-out procedure only results in 5 cases where the parameters are different from the majority of the dataset. We also observed that the exception instance achieves good performance when applied to the majority parameter setup of the whole dataset. Therefore, we conclude the best parameter suit for one dataset is quite stable, and the parameters chosen from a set of energy can still be applicable to other energy functions derived from the same vision task.

In addition, we also observed that the proposed method achieves good performance across all the datasets we tested when the parameters are chosen from a wide range, including the typical parameter setup we reported in Section 7.2. Therefore, the proposed method is robust to its parameters.

7.4. Experimental results for worst case bounds

In the main paper, we set $\epsilon = \infty$ to investigate how our algorithm performs without the worst case bound. We demonstrated that our algorithm can achieve very good performance in practice without it. Now we will study the role of ϵ in practice.

We conducted experiments on Color-seg-n4 dataset as an example. The experimental results are summarized in Table 3.

Table 3: Experimental results with different ϵ on Color-seg-n4 dataset

	$\kappa = 0.8$				$\kappa = 0.6$			
ϵ	Speedup	Energy Change	Labeled Vars	Precision	Speedup	Energy Change	Labeled Vars	Precision
0	4.16x	0.00%	37.82%	100.00%	4.16x	0.00%	37.82%	100.00%
0.01	4.31	0.00%	67.73%	99.99%	4.46x	0.00%	68.25%	99.97%
0.1	6.05x	0.00%	72.07%	99.93%	6.47x	+0.01%	73.69%	99.70%
0.2	6.68x	0.00%	74.67%	99.86%	7.93x	+0.20%	78.32%	99.34%
0.3	6.97x	0.00%	76.32%	99.81%	8.38x	+0.34%	81.43%	99.12%
0.4	7.07x	0.00%	77.86%	99.80%	9.62x	+1.29%	85.91%	98.68%
0.5	7.59x	0.00%	81.16%	99.74%	11.73x	+3.01%	88.41%	97.59%
1.0	7.92x	+0.01%	88.48%	99.69%	12.38x	+6.88%	96.25%	96.51%
10.0	8.12x	+0.01%	90.80%	99.50%	15.02x	+7.83%	98.52%	94.77%

We firstly applied the typical parameter setup we used in Section 7.2. The results are reported on the left part of Table 3. Note that $\epsilon = 0$ is the special case where our method only uses the sound condition to check the partial optimal labeling, hence the proposed algorithm degenerates to the DEE algorithm. Therefore, in this special case, we have a 100% precision and label around 38% variables, hence we get a moderate speedup without affecting the energy. We also know that $\epsilon = \infty$ is another special case where we don't try to bound the worst case. These results is reported in the main paper and Table 1. We already know that the fixed parameters we choose here are reasonable, so even in this extreme case, we still get good performance without the theoretical guarantee. As ϵ decreases, we know that the criterion used becomes more strict. Therefore we will have higher precision and less labeled variables. Due to that, we label fewer variables, and the speedup we acheive decreases. In this setup, since we always maintain the precision value at a extremely high level, ϵ's impact on energy change is not that obvious.

To test this, we conducted the experiments under another set of purposely bad parameters, i.e., changed $\kappa = 0.6$. We summarized our results on the right part of Table 3. We see that with $\kappa = 0.6$ and large ϵ value (e.g., $\epsilon = 10$), our criterion is loose enough to hurt the precision and result in 8% higher energy than before. In our experiments, we observed that as ϵ decreases from 10 down to 0, the precision increases dramatically and the energy increment becomes smaller. Therefore, ϵ values not only give us the theoretical worst case guarantee, but also make real impact in practice (make the criterion we used close to the sound condition and makes the energy smaller).

7.5. Comparison to other MRF inference algorithm

The main focus of this paper is to demonstrate that the proposed decision criterion is efficient and effective in finding a partial optimal labeling of MRFs. We achieve a very good tradeoff between the running time and the final energy by employing our proposed method as the pre-processing for the expansion moves algorithm.

Demonstrating that expansion moves is a state-of-the-art MRF inference algorithm is not the main goal of this paper. The comparison among different inference algorithms are provided in survey papers [4, 10]. However, for the completeness of the paper, we still perform the experiments comparing against other widely used MRF inference algorithms besides expansion movess, including loopy belief propagation (LBP) [7, 12], dual decomposition (DD) [5], TRWS [6] and MPLP [3, 8, 9].

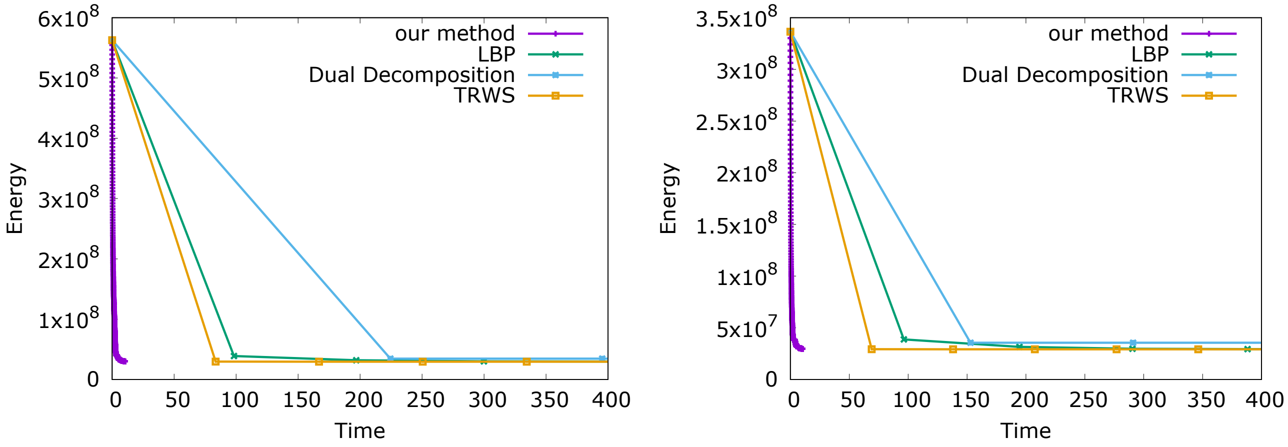
The experimental results are reported in Table 4. We set the time budget for the baseline methods as the 10x of the running time used by expansion moves. In our experiments, expansion movess are usually significantly faster than other methods, and results in comparable or even better energy. This observation is consistent with the survey papers [4, 10]. We can see that LBP, DD, and MPLP usually will get higher energy compared to expansion moves even with 10x of time budget. TRWS is promising since it can provide (slightly) lower energy than expansion moves, although it's much slower. On the datasets we tested, TRWS will spend 3-10x longer time to get energy comparable to our proposed method, through its final energy might be slightly smaller. Typical energy-time curves are presented in Fig. 2. We can see that LBP, DD, TRWS are usually much slower than our method with comparable converging energy.

7.6. Experimental results for multilabel MRFs

In the main paper, we mainly focused on applying the proposed pre-processing technique to each induced binary subproblem from the expansion moves algorithm. We can also apply the proposed pre-processing technique to the multilabel

Table 4: Additional experimental results (TO: time out, MEM: out of memory)

	Dataset	Measurement	Ours	DEE	PR	IRI	LBP	DD	TRWS	MPLP
Challenging Datasets (non- Potts energy, large $\lvert\mathcal{L}\rvert$)	**Stereo**	Speedup	1.78x	1.06x	1.13x	0.51x	0.17x	0.10x	0.10x	MEM
	12–20 labels	Energy Change	-0.06%	0.00%	0.00%	-0.15%	+86.55%	+92.25%	-0.63%	MEM
	Trunc. L1/L2	Labeled Vars	44.76%	10.07%	18.06%	56.45%	-	-	-	MEM
	Inpainting	Speedup	3.40x	1.28x	1.32x	0.12x	0.10x	0.10x	0.10x	MEM
	256 labels	Energy Change	-1.71%	0.00%	0.00%	0.00%	+25.94%	+51.39%	-9.71%	MEM
	Trunc. L2	Labeled Vars	74.29%	21.05%	23.75%	0.36%	-	-	-	MEM
	Denoising-sq	Speedup	12.76x	1.15x	1.33x	0.29x	0.10x	0.09x	0.10x	MEM
	256 labels	Energy Change	-0.02%	0.00%	0.00%	0.00%	-0.65%	+17.14%	-0.65%	MEM
	L2	Labeled Vars	97.93%	17.42%	33.71%	0.39%	-	-	-	MEM
	Denoising-ts	Speedup	13.08x	11.97x	11.86x	0.18x	0.10x	0.09x	0.10x	MEM
	256 labels	Energy Change	0.00%	0.00%	0.00%	-0.03%	-0.78%	+13.29%	-0.99%	MEM
	Trunc. L2	Labeled Vars	98.22%	95.54%	97.71%	5.85%	-	-	-	MEM
	Optical Flow	Speedup	4.69x	2.63	3.40x	TO	0.10x	0.09x	0.10x	MEM
	225 labels	Energy Change	-0.04%	0.00%	0.00%	TO	+9.63%	+16.07%	-0.58%	MEM
	L1	Labeled Vars	77.25%	54.34%	65.51%	TO	-	-	-	MEM
Easy Datasets (Potts, small $\lvert\mathcal{L}\rvert$)	**Color-seg-n4**	Speedup	7.02x	4.55x	6.34x	3.67x	0.14x	0.10x	0.36x	0.10x
	4–12 labels	Energy Change	0.00%	0.00%	0.00%	-0.12%	+1.72%	+3.17%	-0.13%	+0.25%
	Potts	Labeled Vars	85.74%	65.38%	77.50%	98.44%	-	-	-	-
	Color-seg-n8	Speedup	8.33x	5.61x	6.37x	1.45x	0.10x	0.10x	0.12x	0.10x
	4–12 labels	Energy Change	+0.04%	0.00%	0.00%	-0.10%	+0.39%	+4.49%	-0.11%	+0.22%
	Potts	Labeled Vars	90.39%	71.62%	82.05%	99.35%	-	-	-	-



(a) Instance `21077`

(b) Instance `86000`

Figure 2: More speed-energy curve for denoise-sq dataset

MRFs directly. We give preliminary results in Table 5. Since the multilabel MRFs are NP-hard, it's very challenging to get the ground truth persistent labeling for each variable. Therefore, we don't report the precision/recall values. We just use the energy change as an indirect measurement to evaluate the quality of the persistent labeling we found. We also reported the percentage of labeled variables. Both metrics are computed in the per-dataset fashion. We set our distribution $q_i(x_i) = e^{-\theta_i(x_i)}$, which is only from the unary terms and used the fast approximation subroutine to check our discriminative criterion. Then we vary the κ value in $\{0.6, 0.7, 0.8, 0.9\}$. We can see from Table 5 that the proposed method labels significantly more variables than the baseline method DEE, while increases the energy by a couple of percents. It's still the case that the proposed method can achieve a better tradeoff between the number of labeled variables and the energy we can get for the multilabel MRFs. In practice, it's more effective to apply our proposed method to each induced binary subproblem.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2001. 2

[2] J. Desmet, M. D. Maeyer, B. Hazes, and I. Lasters. The dead-end elimination theorem and its use in protein side-chain positioning.

Table 5: Preliminary experimental results on multilabel MRFs

Algorithm	Color-seg-n4		Color-seg-n8	
	% of labeled variables	Energy Change	% of labeled variables	Energy Change
α-EXP	0.00%	0.0000%	0.00%	0.0000%
DEE	15.62%	0.0000%	19.67%	0.0000%
Ours with $\kappa = 0.9$	25.55%	+2.3589%	29.80%	+0.1049%
Ours with $\kappa = 0.8$	30.77%	+3.3480%	30.62%	+0.1161%
Ours with $\kappa = 0.7$	34.01%	+4.4572%	31.05%	+0.1243%
Ours with $\kappa = 0.6$	44.62%	+5.7841%	31.27%	+0.1297%

Nature, 356(6369):539–542, 1992. 2

[3] A. Globerson and T. S. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *NIPS*, pages 553–560, 2008. 5

[4] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, T. Kroeger, B. X. Kausler, J. Lellmann, B. Savchynskyy, N. Komodakis, and C. Rother. A comparative study of modern inference techniques for discrete energy minimization problems. *IJCV*, 115(2):155–184, 2015. 5

[5] J. H. Kappes, B. Savchynskyy, and C. Schnorr. A bundle approach to efficient MAP-inference by lagrangian relaxation. In *CVPR*, pages 1688–1695, 2012. 5

[6] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *TPAMI*, 28(10):1568–1583, 2006. 5

[7] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. 5

[8] D. Sontag, D. K. Choe, and Y. Li. Efficiently searching for frustrated cycles in MAP inference. In *UAI*, pages 795–804, 2012. 5

[9] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. In *UAI*, pages 503–510, 2008. 5

[10] 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. *TPAMI*, 30(6):1068–1080, 2008. 5

[11] C. Wang and R. Zabih. Relaxation-based preprocessing techniques for Markov Random Field inference. In *CVPR*, pages 5830–5838, 2016. 2

[12] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):736–744, 2001. 5