Supplementary Material: Efficient and Exact MAP-MRF Inference using Branch and Bound

Min Sun **Murali Telaprolu** **Honglak Lee** **Silvio Savarese**
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109
{sunmin,muralikt}@umich.edu; {honglak,silvio}@eecs.umich.edu

1 Experimental Settings

By default, the edge-consistent LPR is solved using Message Passing (MP) algorithm to initialize β until convergence[1] or for at most 1000 iterations, whichever comes first. If the gap between the upper and lower bounds is not smaller than 10^{-4} (stopping criteria) already, we further apply our BB method or MPLP-CP method [4]. Both methods stop when the same stopping criteria (gap $< 10^{-4}$) is reached. For MPLP-CP method [4], by default, we alternate between adding 20 clusters at a time and running MPLP for 100 more iterations.

In the human pose estimation experiment, since the problems can be solved most of the time without cluster pursuit, we allow the MP algorithm to try harder to solve the edge-consistent LPR. We follow the suggestions from the authors of [4] to allow the MP algorithm to continue running until the difference between two consecutive upper bounds is smaller than 10^{-5} (instead of 10^{-4} by default), and to add one triplet at a time (instead of 20 clusters by default). In this way, we ensure that the MPLP-CP method does not slow down by adding unnecessary clusters.

2 Improved Naive Branch-and-Bound

Recall that the functionals β are updated at each branch while running the improved naive BB method. To make the improved naive BB a competitive baseline, we initialize the memory for storing the functionals once at the beginning of the method and update only a subset of functionals in each branch. In this way, our implementation is not allocating memory for functionals at each branch and the functionals are not re-initialized from scratch at each branch. The same stopping criteria mentioned above is used in all the branches.

[1]The convergence condition is when the upper bound improvement is smaller than 10^{-4}.

3 Synthesizing the Pairwise Potentials

We synthesize the missing pairwise potentials as follows:

$$\theta_{i,j}(x_i, x_j) = sim(x_i, x_i^{gt}) \times sim(x_j, x_j^{gt}) \in [0, 1] \quad (1)$$

where x_i^{gt}, x_j^{gt} denote the ground truth body part states, and $sim(x_1, x_2) \in [0, 1]$ is a similarity function measuring how similar two states x_1, x_2 are.

Recall that a state $x = \{u, v, l, \phi\}$ is parameterized by its anchor location (u, v), length l, and orientation ϕ. Hence, the two end points $(u_1, v_1), (u_2, v_2)$ of a state can be calculated. The similarity function is defined as follows:

$$sim(x, x^{gt}) = \max(1 - \frac{\sqrt{d_1^2 + d_2^2}}{l^{gt}}, 0) \in [0, 1] \quad (2)$$

where $d_1 = \sqrt{(u_1 - u_1^{gt})^2 + (v_1 - v_1^{gt})^2}$ and $d_2 = \sqrt{(u_2 - u_2^{gt})^2 + (v_2 - v_2^{gt})^2}$.

Notice that the average maximum potential across all pairs of parts from the tree model CPS [3] is about 5. Hence, our synthesized potential ranging from 0 to 1 will not dominate other potentials.

4 Percentage of Correct Parts (PCP)

The typical measure of performance on the buffy dataset [2] is a matching criteria based on both endpoints of each part (e.g., matching the elbow and the wrist correctly): the state of a body part is correct if the endpoints corresponding to the state (u, v, l, ϕ) are, on average, within r of the corresponding ground truth segments, where r is a fraction of the ground truth part length. By varying r, a performance curve is produced where the performance is measured in the percentage of correct parts (PCP) matched with respect to r. In our experiment, we set $r = 0.5$ which is commonly used for evaluation.

5 Comparison between Improved Naive BB and Efficient BB

For the synthetic dense problem experiment, we retrieve the problem instances with the largest number of states that can be solved by the improved naive BB method for different numbers of nodes (3,4,5,6) within 1200 secs. The comparison between B and $\hat{B}H$ is shown in Table 1. The results show that when H is large, $\hat{B}H >> B$ is satisfied empirically, which explains why our efficient BB is much faster than the improved naive BB.

Table 1: Empirical comparison between improved naive BB and efficient BB.

#nodes	#states	$\frac{\hat{B}H}{B}$
3	1083	69.871
4	503	28.743
5	83	7.012
6	23	25.158

6 Distant Transform

Distance transform proposed by Felzenszwalb and Huttenlocher [1] can reduce the time for calculating $\max_{x_j} \beta_{ji}(x_j, x_i)$ for all x_i from $O(H^2)$ to $O(H)$ for certain types of potentials. For such potentials, our method would offer no significant gain. However, we note that all problems in our experiments and many real-world problems cannot be benefited from distance transform.

References

[1] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004.

[2] V. Ferrari, M. M. Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

[3] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *ECCV*, 2010.

[4] D. Sontag, T. Meltzer, A. Globerson, Y. Weiss, and T. Jaakkola. Tightening LP relaxations for MAP using message-passing. In *UAI*, 2008.