SPM-BP: Sped-up PatchMatch Belief Propagation for Continuous MRFs*

Yu Li[1,2] Dongbo Min[3] Michael S. Brown[2] Minh N. Do[4] Jiangbo Lu[1]

[1]Advanced Digital Sciences Center, Singapore [2]National University of Singapore, Singapore
[3]Chungnam National University, Korea [4]University of Illinois at Urbana-Champaign, US

Abstract

Markov random fields are widely used to model many computer vision problems that can be cast in an energy minimization framework composed of unary and pairwise potentials. While computationally tractable discrete optimizers such as Graph Cuts and belief propagation (BP) exist for multi-label discrete problems, they still face prohibitively high computational challenges when the labels reside in a huge or very densely sampled space. Integrating key ideas from PatchMatch of effective particle propagation and resampling, PatchMatch belief propagation (PMBP) has been demonstrated to have good performance in addressing continuous labeling problems and runs orders of magnitude faster than Particle BP (PBP). However, the quality of the PMBP solution is tightly coupled with the local window size, over which the raw data cost is aggregated to mitigate ambiguity in the data constraint. This dependency heavily influences the overall complexity, increasing linearly with the window size. This paper proposes a novel algorithm called sped-up PMBP (SPM-BP) to tackle this critical computational bottleneck and speeds up PMBP by 50-100 times. The crux of SPM-BP is on unifying efficient filter-based cost aggregation and message passing with PatchMatch-based particle generation in a highly effective way. Though simple in its formulation, SPM-BP achieves superior performance for sub-pixel accurate stereo and optical-flow on benchmark datasets when compared with more complex and task-specific approaches.

1. Introduction

Numerous computer vision tasks can be formulated as a global pixel-labeling problem. The goal is to assign each pixel $p = (x_p, y_p)$ in an image a label l_p from the label set $\mathcal{L}$. The labels correspond to quantities that we want to estimate, for instance, disparities in stereo matching or flow vectors in optical flow estimation. These problems are usually defined with the Markov Random Field (MRF) model:

$$E = \sum_{p} E_p(l_p; W) + \sum_{p} \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q) \ , \qquad (1)$$

where $\mathcal{N}_p$ is the pairwise neighborhood set. The pairwise term $E_{pq}(l_p, l_q)$ usually encourages the labels to be piecewise smooth except at object boundaries. The unary term $E_p(l_p)$, also referred to as the data term, computes the local cost for assigning a pixel p the label l_p. This is often computed by performing cost aggregation over a local window W centered at the pixel p. When $|W| = 1$, (1) becomes a special case where no cost aggregation is performed.

Solving for the exact solution that minimizes (1) is well-known to be NP hard. Several discrete optimizers such as Graph Cuts [9] and belief propagation [39, 13] have been proposed for multi-label discrete problems, but they still face prohibitively high computational challenges when the label set $\mathcal{L}$ is huge or densely sampled. This is

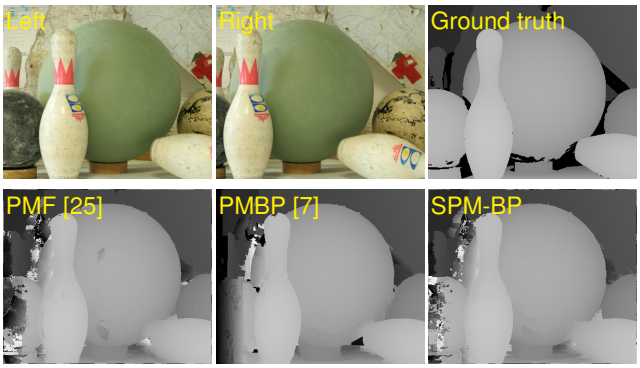

Figure 1. Stereo results on the *Bowling2* dataset. PMF [25], as a local method, is fast ($20s$), but fails at the large textureless region on the ball. PMBP [7], a global minimization method, considers neighborhood smoothness and can overcome this ambiguity to obtain smooth disparities. PMBP, however, has high computation cost ($3100s$). Our SPM-BP produces visually similar results, but runs significantly faster than PMBP ($30s$).

*This study is supported by the HCCS research grant at the ADSC from Singapore's Agency for Science, Technology and Research (A*STAR). This work was mainly done when Yu and Dongbo were interning and working at ADSC. Corresponding author: dbmin@cnu.ac.kr

true even with the recent particle belief propagation (PBP) algorithm [19, 30]. Integrating the effective particle propagation and resampling from PatchMatch [6], PatchMatch belief propagation (PMBP) [7] has shown good performance while running orders of magnitude faster than PBP.

The solution quality of PMBP, however, still depends on the matching fidelity of E_p. The raw data cost is aggregated over a local window W to mitigate ambiguity in the data constraint, increasing an overall complexity linearly with $|W|$. In general, the cost aggregation for computing $E_p(l_p; W)$ can be written with the raw cost $C_r(l_p)$ as a weighted sum of its neighbour costs:

$$E_p(l_p; W) = \sum_{r \in W} \omega_{pr} C_r(l_p), \tag{2}$$

where ω_{pr} is the normalized adaptive weight of a support pixel r and is defined based on the structure of the input image I (e.g. bilateral weight [36] between p and r). Though PMBP significantly reduces the complexity dependency on the label space size, the brute-force adaptive-weight summation has a linear complexity dependent on the window size $|W| = (2w + 1) \times (2w + 1)$. Generally, to produce better results, the window size should be relatively large, for instance, $w = 15$ in stereo matching, leading to a huge amount of computational overhead as compared with the message passing mechanism itself (Fig. 2).

As an alternative to the global pixel-labeling approaches, local approaches [32, 25, 5] have shown competitive results for certain labeling problems. They compute only the unary term $E_p(l_p; W)$ by employing fast edge-aware filtering (EAF) [16, 24] when repeatedly computing E_p in (2) for each label l_p, thus making the computational complexity independent of the support window size $|W|$. However, such a scheme of efficiently computing E_p with the fast EAF is not directly applicable to the PBP or PMBP kind of global optimizers, which require the fragmented label access and data cost computation.

This paper proposes a novel algorithm called sped-up PMBP (SPM-BP) to tackle this critical computational bottleneck and offer performance improvements of PMBP by 50-100 times typically (Fig. 1). The crux of SPM-BP is to unify efficient filter-based cost aggregation and message passing with PatchMatch-based particle generation in an effective manner. Our key motivation lies on the observation that a labeling solution is often spatially smooth with discontinuities aligned with object edges. This allows for shared label particle generation and aggregated data cost computation for neighboring pixels covered in the same compact superpixel. This superpixel-based label propagation also enables one to further improve the quality of the labeling solution over the original PMBP.

We evaluate our method for sub-pixel accurate stereo and optical flow estimation on various benchmark datasets. As an efficient global optimization algorithm for continuous MRFs, our SPM-BP method outperforms the existing local filter-based methods e.g. PMF [25] in solution quality and global optimization methods e.g. PMBP [7] in runtime. In addition, SPM-BP possesses several distinctive features. First, it has a simple formulation that does not involve complex energy terms or require a separate initialization process (e.g. [31, 38]). Second, it is able to achieve top-tier performance on a few tasks, even compared with the leading task-specific approaches, such as EpicFlow [31], DeepFlow [38] and PPM [43] for the Sintel optical flow, and PM-Huber [17] and PM-PM [41] for sub-pixel accurate stereo. Finally, it works directly on the full pixel grid without a need for coarse-to-fine optimizations [11, 35, 23] avoiding error propagation across scales as argued in [31]. More performance comparisons to this will be given in Sec. 5.

1.1. Related Work

This section reviews related works targeting efficient optimization of MRF-based labeling problems. As motivated earlier, effectively handling continuous labeling problems with huge label spaces has become increasingly important. The particle BP (PBP) techniques [19, 37] have been proposed by applying the Markov chain Monte Carlo (MCMC) sampling to the current belief estimate using a Gaussian proposal distribution. Though not exhaustively evaluating all possible label candidates, PBP is still too slow for continuous label spaces in practice. Inspired by PatchMatch [6], Besse *et al.* [7] unified the two techniques of PBP and PatchMatch, and leveraged the latter to produce particle proposals effectively. This makes PMBP orders of magnitude faster than PBP. However, PMBP still suffers from the heavy computational load consumed by the data cost aggregation.

To address the reduction of the data aggregation complexity, cost volume filtering techniques [44, 32] have recently emerged as an alternative to solving MRFs. The key emphasis for these approaches is placed on efficiently performing raw data cost aggregation over a local region (often over 35×35 [8]). Constant-time edge-aware filters [28, 16, 14] are adopted to remove the complexity dependency on the local window size. Though simple to implement, this kind of filter-based methods generally do not optimize for a global energy that enforces globally coherent labeling results. Although global labeling quality can be improved by filter-based inference for fully connected CRFs [20], all these mentioned methods are still too slow to solve continuous labeling problems, where the label space is huge or even infinite.

Recently, PatchMatch Filter (PMF) [25] was proposed to address the complexity curse of a huge labeling space faced by filter-based inference methods. The method cleverly exploits the complementary advantages of approximate NNF

search through PatchMatch and efficient edge-aware filtering. This allows a minimal complexity dependency on both the label size and the filter window size. PMF reports over 10-times speedup over the PatchMatch stereo method [8], which computes the local data aggregation cost in a brute-force manner. However, PMF does not model a global objective function, resulting in estimation errors in large low-textured regions.

2. Background

We start by introducing the basic algorithmic steps of belief propagation (BP) and PatchMatch BP (PMBP).

2.1. Belief Propagation and Notations

BP [39] minimizes (1) by iteratively passing messages on a loopy graph defined with a 2D image grid. Each pixel in the image is a node in the graph and neighboring pixels are linked with edges. The message $m_{qp}(l_p)$ represents the neighboring node q's opinion of the (negative log) possibility that a label $l_p \in \mathcal{L}$ is assigned to a node p, which is defined as:

$$m_{qp}^{(t)}(l_p) = \min_{l_q \in \mathcal{L}} \Big(E_{pq}(l_p, l_q) + E_q(l_q) + \sum_{s \in \mathcal{N}_q \backslash p} m_{sq}^{(t-1)}(l_q) \Big), \tag{3}$$

where $\mathcal{N}_q \backslash p$ denotes the neighbors of q other than p. The superscript (t) is a time stamp. For brevity, we drop the aggregation window W from the unary term's notation wherever appropriate. After a certain number of iterations T, a dis-belief is computed at each node as

$$B_p(l_p) = E_p(l_p) + \sum_{q \in \mathcal{N}_p} m_{qp}^{(T)}(l_p). \tag{4}$$

The final label l_p^* that minimizes the dis-belief is selected: $l_p^* = \text{argmin}_{l_p \in \mathcal{L}} B_p(l_p)$.

2.2. PBP and PMBP

The message passing in (3) is performed for all labels. This is a serious challenge to the BP-based discrete optimization as the computational complexity becomes prohibitively high when the label space $|\mathcal{L}|$ is large. To address this challenge, Kothapa *et al*. [19] proposed to associate each node p with a particle set $\mathcal{R}_p$ which stores only a few number of labels as candidate particles. The message passing in (3) becomes

$$m_{qp}^{(t)}(l_p) = \min_{l_q \in \mathcal{R}_q} \Big(E_{pq}(l_p, l_q) + E_q(l_q) + \sum_{s \in \mathcal{N}_q \backslash p} m_{sq}^{(t-1)}(l_q) \Big) \tag{5}$$

for $l_p \in \mathcal{R}_p$. Note that the minimization is executed only over the particles set $\mathcal{R}_q$, and a new set of particles $\mathcal{R}_p$ at node p is chosen and updated at each iteration. Kothapa *et al*. proposed to use MCMC sampling from the current belief estimation with a Gaussian proposal distribution. We denote their method as PBP.

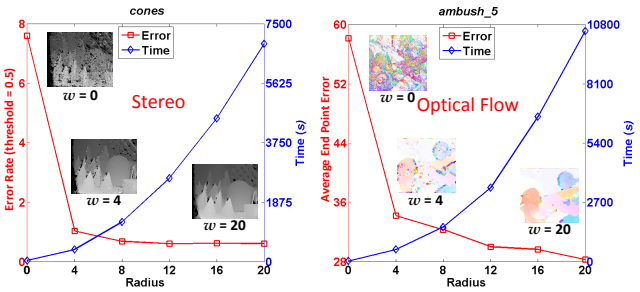

Figure 2. An example illustrating the importance of the data cost E_p in a global optimization framework. The errors of the stereo matching and optical flow results were measured by varying the radius w of the local aggregation window W in PMBP [7]. The disparity was modeled by using a 3D plane parameter $[a_p, b_p, c_p]^\top$ (see Sec. 4.1). The flow vectors are with subpixel-accuracy (see Sec. 4.2). PMBP was applied on the full pixel grid without using a coarse-to-fine framework like hierarchical BP (HBP) [13].

Recently, Besse *et al*. [7] augmented PBP by leveraging the PatchMatch concept [6]. At each iteration, the particle proposals $\mathcal{R}_p$ are re-sampled from the subset of labels from neighboring pixels, and then a random search is performed to avoid local minima. The new optimization method, named PMBP, runs orders of magnitude faster than the original PBP [19], and shows more accurate results than other local labeling methods (e.g. PatchMatch stereo [8]).

3. Sped-up PMBP Algorithm: SPM-BP

Now we present our SPM-BP algorithm: a superpixel-based computational framework to minimize (1).

3.1. Motivation

As mentioned in Sec. 1, global labeling algorithms usually encourage the solution to be piecewise smooth except at object boundaries. Nevertheless, their solution quality is also similarly affected by the matching fidelity of the data cost E_p in (1). Fig. 2 shows how important E_p is even in the global optimization framework. The test was done using PMBP [7] for the '*Cone*' image in the Middleburry stereo dataset. All parameters were fixed except the radius w that controls the aggregation window size $|W|$. It should be noted that PMBP was applied on the full pixel grid without using a coarse-to-fine framework like hierarchical BP (HBP) [13]. As expected, using a larger window size improves the quality of PMBP's result significantly at the cost of a longer runtime. This runtime penalty primarily stems from the fact that the data cost E_p should be computed for each pixel *independently* due to the operation structure of the PMBP using fragmented label propagation. Recently, Xu *et al.* [41] proposed a convex formulation of the multi-

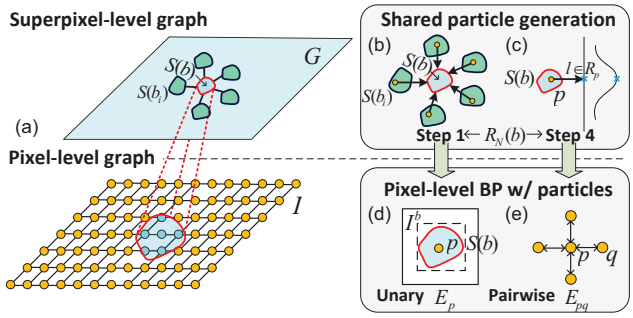

Figure 3. Two-layer graph structure used in SPM-BP: (b)(c) A superpixel-level graph generates new particle proposals $\mathcal{R}_N(b)$ to be tested on the pixel-level graph (see **Step 1** and **Step 4** of Sec. 3.3). (d) For the reference superpixel $S(b)$, the EAF is applied to obtain the data cost E_p for $p \in S(b)$. (e) The message passing algorithm proceeds in the inner loop of $S(b)$ using (5), while outgoing messages on the boundary of $S(b)$ are fixed.

label Potts Model with the PatchMatch stereo approach [8], demonstrating very competitive results in subpixel accurate stereo matching. They discussed the importance of the data cost, and reserved its fast computation (for a window of $|W| = 41 \times 41$) as a future work.

3.2. Two-Layer Graph Structures in SPM-BP

The key idea of SPM-BP stems from the observation that the labeling solution for a natural image is often spatially smooth except for discontinuities aligned with object boundaries. Such an assumption allows for the shared label and data cost computation for similar pixels within compact superpixels. We note that this collaborative processing scheme based on superpixels is conceptually similar to that in PMF [25]. However, SPM-BP deals with a generic computational framework for efficiently minimizing continuous MRFs through particle-based message passing.

The proposed SPM-BP method begins by partitioning an input image into B non-overlapping superpixels $I = \{S(b), b = 1, 2, \ldots, B\}$, for instance, using the SLIC algorithm [4]. A superpixel-level graph $\mathcal{G}$ is then built with the set of superpixels $\{S(b)\}$. Similar to existing approaches such as PBP and PMBP, a pixel-level graph is also constructed using adjacent 4-neighbors for each pixel on the original image grid. Fig. 3 shows the graph structure, consisting of two different types of nodes.

On the superpixel-level graph, the label candidates (particles), which are sampled from adjacent superpixels, are propagated into the reference superpixel $S(b)$. The role of the super-pixel graph is to generate new particle proposals to be tested on the pixel-level graph. Each superpixel is slightly enlarged and treated as a sub-image, in which the aggregated data cost E_p is jointly obtained for each pixel. The continuous MRF formulation is still solved on the full image grid, but the message passing algorithm in (5) proceeds in the inner loop of each superpixel, while outgoing messages on the boundary of $S(b)$ are fixed (see Fig. 3). Note that the message updating order of SPM-BP is conceptually similar to that of tile-based BP [22], which divides an image into a set of tiles to make it suitable for a parallel implementation.

From a computational perspective, we choose to use the superpixel as a basic unit for both computing the data cost E_p in (2) and updating the message in (5). As motivated earlier, EAF has been widely used for efficiently computing E_p in local labeling approaches. The computational efficiency of EAF primarily comes from reusing a shared computation together with neighboring pixels within the local window. This kind of computational scheme requires simultaneously updating the output values of all pixels to be filtered. To meet such a requirement while leveraging the computational efficiency of EAF, we compute the unary term E_p and update the message m_{qp} of (5) simultaneously for all pixels belonging to the same superpixel $S(b)$. It is worth noting that though all of the operations are executed for each superpixel individually, the final labeling solution is computed on a full pixel grid, thus enabling the estimation of fine-grained label maps such as non-rigid motion or depth.

3.3. SPM-BP Algorithm

After constructing the graph structure in Fig. 3, we assign each superpixel K labels randomly sampled within the label set $\mathcal{L}$. Namely, all pixels belonging to each superpixel start with the same particles.

After the initialization, the SPM-BP iterates the label propagation and center-biased random sampling on the superpixel-level graph. The computation of E_p in (2) and the message update in (5) are on the pixel-level graph. This process is performed in an interleave order. During even iterations, the SPM-BP algorithm proceeds from the top-left superpixel $S(1)$ to the bottom-right superpixel $S(B)$, and *vice versa*. In the inner loop of $S(b)$ for $b \in \{1, ..., B\}$, the message passing algorithm (5) runs from the top-left pixel to the bottom-right pixel during even iterations, and *vice versa*. For clarity sake, we first define useful notations.

- $S(b_i)$, $b_i \in \mathcal{N}(b)$: a spatially neighboring superpixel of the reference superpixel $S(b)$.
- $\mathcal{R}_p$: a pixel-varying set of particle proposals. This stores the set of current best particles at pixel p.
- $\mathcal{R}_N(b)$: the set of particles generated for superpixel $S(b)$. This is obtained using the particle propagation (**Step 1**) or the random search (**Step 4**), and is commonly used for all pixels $p \in S(b)$.
- $\text{argminK}_{l \in \mathcal{R}'_p} B_p(l)$: function that returns the top K labels yielding the smallest disbeliefs $B_p(l)$.

Start: We first initialize all messages $m_{qp}^{(0)}(l)$ for all $p, q \in I$ and $l \in \mathcal{L}$ to zero. At the t^{th} iteration, we do the following

four steps for each superpixel $S(b)$, $b = 1, 2, \ldots, B$. For simplicity, we omit the superscript t.

Step 1: Particle propagation
Generate particle proposals for the reference superpixel $S(b)$ using spatially neighboring superpixels in the superpixel-level graph. As [25], we randomly pick one pixel $p_i \in S(b_i)$ from each neighboring superpixel $S(b_i), b_i \in \mathcal{N}(b)$, and generate a new proposal set $\mathcal{R}_N(b) = \bigcup \mathcal{R}_{p_i}$.

Step 2: Computation of E_p
Compute the data cost E_p for the new proposal set $\mathcal{R}_N(b)$. $E_p(l; W)$ is obtained by applying the EAF to the raw matching cost $C_p(l)$ as in (2) for $p \in S(b)$ with $l \in \mathcal{R}_N(b)$.

Step 3: Message update
Update the message and disbelief. For the reference pixel $p \in S(b)$ and its neighboring pixel $q \in \mathcal{N}_4(p)$, their particles are defined respectively as follows: $l_p \in \mathcal{R}'_p = \mathcal{R}_p \bigcup \mathcal{R}_N(b), \quad l_q \in \mathcal{R}_q$.
Step 3.1: The incoming messages $m_{qp}(l_p)$ of $p \in S(b)$ from $q \in \mathcal{N}_4(p)$ are first computed using (5).

Step 3.2: The log disbelief $B_p(l_p)$ is computed using (4).

Step 3.3: Compute and update new particles for all pixels $p \in S(b)$ by selecting top K disbeliefs only: $\mathcal{R}_p = \text{argminK}_{l \in \mathcal{R}'_p} B_p(l)$.

Step 4: Random search
Similar to the original PM [6], center-biased random sampling is performed to prevent the solution from being trapped in a local minima. We randomly pick one pixel $p \in S(b)$ within the reference superpixel $S(b)$, and then generate a new proposal set $\mathcal{R}_N(b) = \{l + \frac{R}{2^i} | i = 1, \ldots, M\}$, $\forall\, l \in \mathcal{R}_p$. The term R is a random variable uniformly sampled from the entire label space. Namely, a new label is sampled M times within a distance $\frac{|R|}{2^i}$ centered at l.

Step 4.1: Repeat **Step 2 and 3** using the new proposal set $\mathcal{R}_N(b)$.

The optimization process described above is performed iteratively until the labeling result converges or the maximum number of iterations is reached. The final labeling solution is obtained as follows:

$$l_p = \text{argmin}_{l \in \mathcal{R}_p} B_p(l). \tag{6}$$

In **Step 2**, different EAF methods such as the guided filter (GF) [16] or cross-based local multipoint filtering (CLMF) [24] can be used. Similar to the design in PMF [25], for a given superpixel $S(b)$, we define a minimum bounding box covering $S(b)$. We then apply the EAF to a subimage I^b which contains the bounding box, but with its borders extended outwards by a radius w of the local window W, in order to avoid filtering artifacts that may occur around superpixel boundaries.

Table 1. Complexity comparison of three different techniques. In [25], PMF stores only one best particle ($K = 1$) per pixel node, thus requiring more iterations than the other two methods.

	PMF* [32]	PMBP [8]	**SPM-BP**
Data Cost	$O(N\log L)$	$O(\|W\|KN\log L)$	$O(KN\log L)$
Message Passing	-	$O(K^2N\log L)$	$O(K^2N\log L)$

3.4. Complexity

Given an image size N, the local window size W and the label space size $L = |\mathcal{L}|$, we compare the computational complexity of PMF, PMBP and our SPM-BP. Table 1 summarizes the complexity analysis, consisting of the data cost computation and message passing.

On the data computation, our SPM-BP can remove the dependency on the window size $|W|$ by employing the EAF in computing E_p of (2). PMF [25] is a special case of SPM-BP without the pairwise term E_{pq} in (1) and thus, has no computational dependency on the window size $|W|$ as well. It should be noted that PMF typically uses a single particle $K = 1$ and therefore requires more iterations than PMBP and SPM-BP. PMBP can reduce the complexity on the label space L, but its complexity still depends on the local window size $|W|$. As a result, it often faces a serious computational challenge when a larger window size is needed. For message passing, both PMBP and SPM-BP have the same complexity of $O(K^2N\log L)$ while PMF, as a local method, does not have this part. The $\log L$ complexity in all terms was discussed in the PatchMatch paper [6].

4. SPM-BP for Correspondence Estimation

We evaluate the proposed SPM-BP algorithm by applying it to two visual correspondence tasks: sub-pixel accurate stereo and optical flow estimation on continuous MRFs.

4.1. Subpixel-Accurate Stereo Reconstruction

Following PatchMatch stereo [8], our SPM-BP method estimates a sub-pixel disparity at each pixel p by searching for an over-parameterized 3D plane $\mathbf{l}_p = [a_p, b_p, c_p]^\top$. This representation can handle slanted surfaces better. The search range $|\mathcal{L}|$ contains an infinite number of 3D planes. A support pixel $q = (x_q, y_q) \in W_p$ of the left image I is projected to $q' = (x_{q'}, y_{q'})$ in the right image I' as:

$$x_{q'} = x_q - [x_q, y_q, 1] \cdot \mathbf{l}_p \ , \text{ and } \ y_{q'} = y_q \ . \tag{7}$$

The raw matching cost is computed with two pixels q and q' [32, 40] as:

$$\begin{aligned} C_q(l) = (1-\alpha) \cdot \min\left(\left\|I_q - I'_{q'}\right\|, \tau_{col}\right) \\ + \alpha \cdot \min\left(\left\|\nabla_x I_q - \nabla_x I'_{q'}\right\|, \tau_{grad}\right) \ , \end{aligned} \tag{8}$$

where τ_{col} and τ_{grad} are truncation thresholds. The color and gradient dissimilarities are combined using a user-specified weight α. In experiments, we set $\tau_{col} = 10$,

$\tau_{grad} = 2$, and $\alpha = 0.9$. The smoothness term is defined with two neighboring pixels p and $q \in \mathcal{N}_p$ as

$$E_{pq}(l_p, l_q) = \omega_{pq}(|\mathbf{n}_p \cdot (\mathbf{x}_q - \mathbf{x}_p)^\top| + |\mathbf{n}_q \cdot (\mathbf{x}_p - \mathbf{x}_q)^\top|), \quad (9)$$

where $\mathbf{n}_p = \text{unit}(a_p, b_p, -1)$ and $\mathbf{x} = [x, y, 1]$ are the plane normal at pixel p and a point on the plane. This cost allows small variation between two planes. The parameter $\omega_{pq} = \lambda \exp(-\|I_p - I_q\|/\sigma)$ becomes higher when p and q are similar, in which λ is a constant controlling the ratio of the smoothness term w.r.t. the data term.

Post-processing: After obtaining initial disparity maps, we detect unreliable disparity estimates by conducting cross-checking, and then fill them in by extrapolating the plane parameter $\mathbf{l}_p$ with reliable estimates [8]. Finally, a weighted median filter is applied to refine the resulting disparity map.

4.2. Large-Displacement Optical Flow

We now present our optical flow method based on SPM-BP, in which the label l represents a 2D displacement vector $l = (u, v)$. Given a candidate label l, a pixel q in a reference image I is matched to a pixel $q' = q + (u, v)$ in the target image I'. Although SPM-BP deals with continuous MRFs, we constrain the subpixel precision of the flow vector l up to $1/8$ as in [25]. In our experiments, the flow vectors estimated on this scale showed a satisfactory quality. In fact, estimating flow vectors on a higher precision than $1/8$ does not improve the flow quality noticeably due to image quality degradation that may occur when applying image upscaling (e.g. bicubic interpolation).

The raw matching cost at pixel q is computed using both an absolute distance (AD) and Census transform [26] as:

$$C_q(l) = \rho(C_q^{census}(l), \tau_{cs}) + \rho(C_q^{AD}(l), \tau_{ad}), \quad (10)$$

where $\rho(C, \tau) = 1 - \exp(-C/\tau)$ is a robust function. In experiments, we set $\tau_{cs} = 30$ and $\tau_{ad} = 60$. The smoothness cost is defined with two pixels p and $q \in \mathcal{N}_p$ by the modified Potts model with the truncation threshold $\tau_s = 2$:

$$E_{pq}(l_p, l_q) = \omega_{pq} \min(\|l_p - l_q\|_2^2, \tau_s). \quad (11)$$

Post-processing: After estimating the bidirectional flow field between two images, we perform the cross-checking [32] between two fields to detect occluded regions. Simple extrapolation does not provide satisfactory results when the occlusion region is large. Thus, we employed post-processing based on a quadratic optimization that is easily solved by a sparse matrix solver (*e.g.* [27]). Inspired by the work of [29], we define the data term using the initial flow vector l_p^* and the generated occlusion map:

$$E_p(l_p) = \begin{cases} \|l_p - l_p^*\|_2^2, & p \text{ is visible} \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

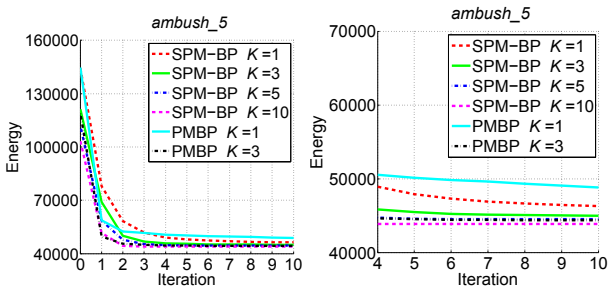

Figure 4. Convergence study for SPM-BP and PMBP under different settings of particle numbers K. (a) shows the energy per iteration. (b) shows the zoomed-in region after the $4th$ iteration.

For all occluded pixels, the cost value is always zero and thus, their outputs completely depend on visible pixels. The smoothness term is defined as

$$E_{pq}(l_p, l_q) = \omega_{pq}\|l_p - l_q\|_2^2, \quad (13)$$

where ω_{pq} is the confident weight defined by the color similarity of neighboring pixels p and q [29]. This smoothness term can help propagate the flow vectors from visible pixels to occluded pixels depending on their color similarities.

5. Experiments

Our experiments were performed on a PC with Intel I7 CPU (3.4GHz) and 8GB RAM. The implementation was done in C++. We chose GF [16] and CLMF-0 [24] as the edge aware filters for cost aggregation in stereo and optical flow tasks, respectively. For PMBP, we used the source C++ code provided by the authors. We have also implemented PMF for comparison. The parameter values follow the settings in the two papers. For our SPM-BP, we have always used 500 superpixels, and set $\sigma = 10$ and $\lambda = 0.01$ for ω_{pq}. The window size $|W|$ was set to 31×31 and $|W| = 19 \times 19$ for stereo and optical flow, respectively. We will make our code publicly available.

5.1. Parameter Evaluation

We have tested the convergence of our SPM-BP. We plot the energy produced by SPM-BP on an optical flow task in Fig. 4(a) with a zoomed-in view after iteration 4 in Fig. 4(b). As can be seen, SPM-BP almost converges after 5 iterations. In addition to SPM-BP, we have also plotted the energy by PMBP ($K = 1, 3$) using the bilateral cost as exactly in [7] . The energy produced by PMBP is close to that by SPM-BP. Note that we plot the energy value per one iteration. For each iteration, SPM-BP takes a much shorter runtime thanks to the highly efficient data cost computation. Fig. 4 also shows the effect of using different particle numbers $K = 1, 3, 5, 10$. Using more particles yields a converged solution with a slightly lower energy. However,

Table 2. Quantitative stereo result evaluation (w/o post processing) on eight Middleburry 2006 datasets with error threshold 0.5.

Dataset	PMF [25]	PMBP [7]	**SPM-BP**
Baby2	15.34	16.85	**12.82**
Books	**22.15**	27.57	22.52
Bowling2	15.95	15.20	**14.35**
Flowerpots	**24.59**	27.97	24.80
Lampshade1	25.02	30.22	**23.39**
Laundry	**26.77**	33.90	27.32
Moebius	21.47	25.09	**21.09**
Reindeer	**15.04**	21.57	16.02
Mean	20.79	24.79	**20.29**

Table 3. Middlebury stereo evaluation [1] for error threshold = 0.5.

Method	Avg. Rank	Avg. Error	Runtime(s)
PM-PM [41]	8.2	7.58	34 (GPU)
PM-Huber [17]	8.4	7.33	52 (GPU)
SPM-BP	12.1	7.71	30
PMF [25]	12.3	7.69	20
PMBP [7]	19.8	8.77	3100

as discussed in Sec. 3.4, the computational complexity increases with more particle number. Therefore each iteration takes a longer runtime when using more particles. In practice, we found choosing $K = 3$ and the number of iterations $T = 5$ provides satisfactory results. As such, we fixed this for the following experiments for SPM-BP and PMBP. PMF, using a single particle, needs more iterations. We fixed PMF's iteration number to 10, as suggested in the the original PMF paper. We fixed the number of superpixels to 500 as using more superpixels leads to a little improvement but causing more runtime. With this configuration, PMF takes **20s** for stereo (443×370 resolution) and **27s** for optical flow (1024×436 resolution). The average runtime of SPM-BP is **30s** for stereo and **42s** for optical flow, which has a relatively marginal runtime overhead as compared with PMF. PMBP is much slower and the runtimes are **3100s** (stereo) and **2103s** (optical flow). In optical flow estimation, our SPM-BP are 50 times faster then PMBP. Moreover, the runtime gain of stereo matching task becomes even more, as it requires using a larger window size $|W| = 31 \times 31$.

5.2. Results of Sub-pixel Stereo Estimation

We evaluated our SPM-BP based stereo method using the Middlebury stereo dataset [1]. Table 2 measured error rates with the threshold of 0.5, and compared them with those of PMF and PMBP. Post-processing was not applied to all the methods. It is shown that PMF tends to produce less errors than PMBP, which was also reported in the PMF paper [25]. SPM-BP works better than PMF, but the quality gain is relatively small as the majority of test images has no large textureless regions. Nevertheless, the superior advantage of the SPM-BP over PMF is still observed in *e.g. Bowling2* containing large textureless regions. See also Fig. 1 for a subjective evaluation. Next, we compare SPM-BP with some leading methods based on the standard Middlebury benchmark datasets (*Tsukuba, Venus, Teddy* and *Cones*) in Table 3. Our method is highly ranked out of over 150 methods on the leaderboard, and compares favorably with those leading PatchMatch-based approaches in Table 3.

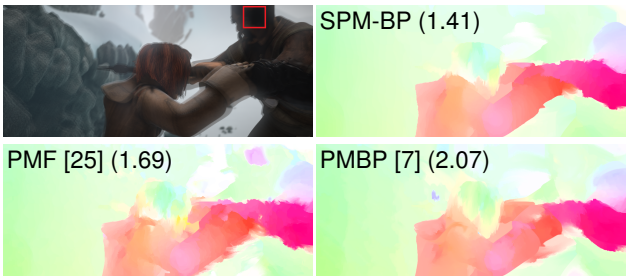

Figure 5. Visual and EPE comparison of the optical flow by PMF [25], PMBP [7]. Note the errors in PMF's flow map.

Table 4. Optical flow performance on MPI Sintel Dataset (http://sintel.is.tue.mpg.de/results, captured on 16 Apr. 2015). For those methods without providing public code, we report their time on KITTI. *use GPU.

Method	EPE all		EPE all		Runtime
	Clean	Rank	Final	Rank	(Sec)
EpicFlow [31]	4.115	1	6.285	1	17
PH-Flow [43]	4.388	2	7.423	8	800
SPM-BP	5.202	5	7.325	6	42
DeepFlow [38]	5.377	7	7.212	4	19
LocalLayering [34]	5.820	13	8.043	13	-
MDP-Flow2 [40]	5.837	14	8.445	21	754
EPPM [5]	6.494	18	8.377	20	0.95*
S2D-Matching [21]	6.510	19	7.872	10	2000
Classic+NLP [35]	6.731	21	8.291	19	688
Channel-Flow [33]	7.023	24	8.835	26	>10000
LDOF [10]	7.563	25	9.116	28	30

5.3. Large-Displacement Optical Flow Results

We evaluate our SPM-BP based optical flow estimation using the MPI Sintel dataset [12], a modern and challenging optical flow evaluation benchmark. Note that the Middlebury optical flow benchmark [2] provides the dataset with small motion only, and the KITTI dataset [3] was specially targeted on road driving scenes. Thus, we focus our evaluation on the Sintel dataset with large displacement flow vectors and more complex non-rigid motions. The evaluation is performed on two types of rendered frames, *i.e.* clean pass and final pass. The final pass adds more complex effects such as specular reflections, motion blur, defocus blur, and atmospheric effects. We fixed the search range of flow vectors to $[-200, 200]^2$. The floating precision of flow vectors was set to $1/8$ for both x and y directions. This results in label space, $\mathcal{L}$, with over 10 million labels.

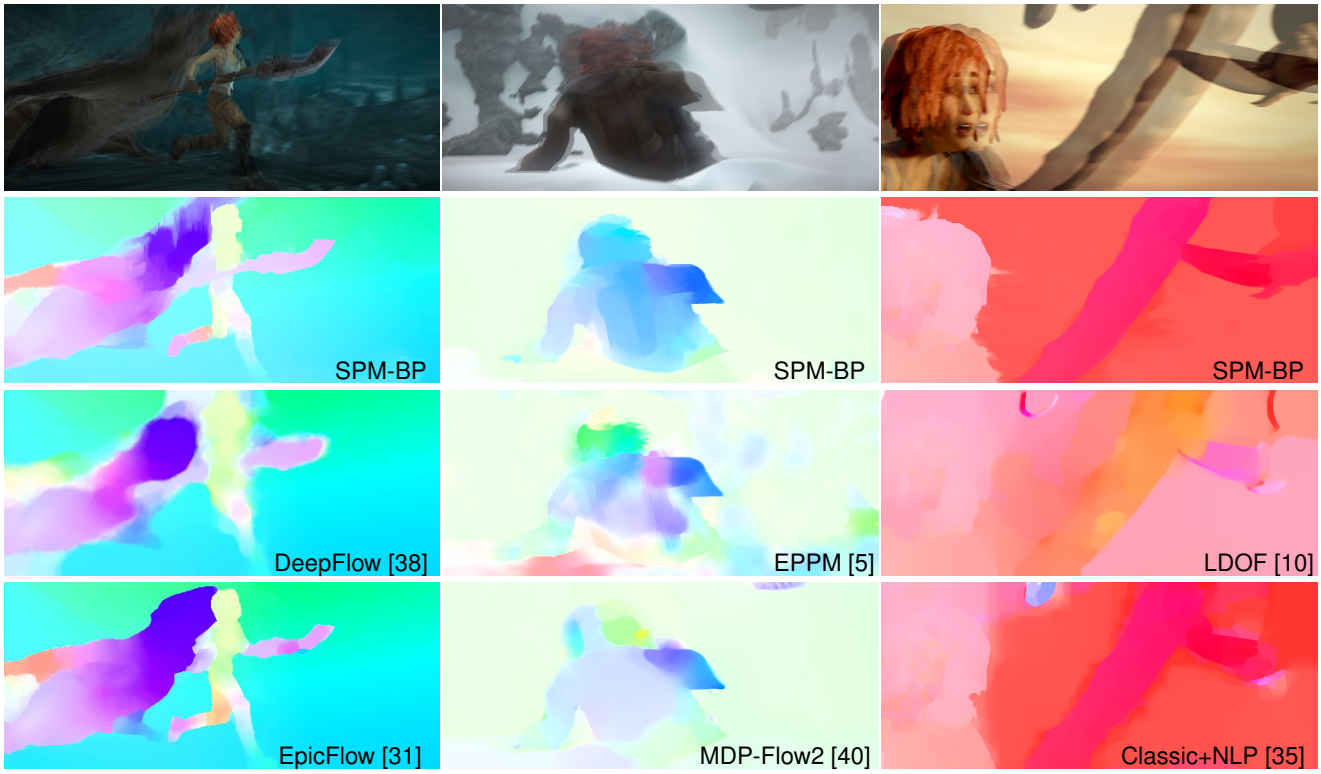

Figure 6. Visual comparison of optical flow maps generated by different methods. Without using more complex formulations and initializations or a coarse-to-fine framework, SPM-BP produces high-quality estimates for large displacements and fine motions with both global smoothness and structure preservance.

We sampled the training set every other second frame (totally 331 frames), and used them to train all parameters. For clean/final passes, the SPM-BP achieved average End Point Error (EPE) of $2.91/3.90$, while the EPEs of PMF (also using the post-processing of Sec. 4.2) are $3.31/4.66$. The quality gain of using the global optimization is obvious, since the Sintel dataset contains many textureless regions in which global approaches are more favorable than local methods. A representative scene in Fig. 5 clearly demonstrates the limitation of local methods, e.g. the man's head part. While PMBP overcomes this limitation at the cost of a much longer runtime, SPM-BP addresses the global labeling coherence issue much more efficiently, achieving a 50 times speedup over PMBP. For instance, PMF, PMBP and our SPM-BP takes $27s$, $2103s$ and $42s$ respectively for Sintel image pairs of 1024×436.

The Sintel test set contains 12 long sequences consisting of totally 564 frames where the ground truth flows are hidden. Table 4 lists the quantitative evaluation results on the Sintel benchmark website. On the benchmark, our SPM-BP ranks 5/6 (clean/final) among all 39 methods at the time of submission. Fig. 6 shows the flow results of some state-of-the-arts methods. Note that EPPM [5] uses a local PatchMatch-like data aggregation together with the coarse-to-fine framework. It loses fine-grained details of flow vectors and still has difficulties in handling large textureless regions even with such a coarse-to-fine scheme.

6. Conclusion

We have proposed a novel SPM-BP algorithm to efficiently solve continuous MRFs consisting of an aggregated data cost term and a pairwise smoothness term. By exploiting a two-layer graph structure, SPM-BP takes the best computational advantages of efficient edge-aware cost filtering and superpixel-based particle-sampling for message passing. As an efficient global MRF optimizer, SPM-BP outperforms the existing methods such as PMF [25] in solution quality and PMBP [7] in runtime. Though simple in its formulation, SPM-BP demonstrated superior standings on optical flow and stereo benchmark tests. SPM-BP may be used to accelerate [18] that used PMBP [7]. We also plan to use SPM-BP with the aim of significantly improving the previous cross-scene matching work [42] based on the PMF [25]. It is interesting to note a very recent work [15] also used a superpixel-based CRF model with a similar scheme to propagate neighboring particle proposals, but its single superpixel-level graph may be unable to handle complex non-rigid motions well. Extending SPM-BP to include high-order terms [15] is an interesting topic for future work.

References

[1] http://vision.middlebury.edu/stereo/.

[2] http://vision.middlebury.edu/flow/.

[3] http://www.cvlibs.net/datasets/kitti/.

[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11), 2012.

[5] L. Bao, Q. Yang, and H. Jin. Fast edge-preserving PatchMatch for large displacement optical flow. In *CVPR*, 2014.

[6] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. In *SIGGRAGH*, 2009.

[7] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. PMBP: PatchMatch belief propagation for correspondence field estimation. *IJCV*, 110(1):2–13, 2014.

[8] M. Bleyer, C. Rhemann, and C. Rother. PatchMatch Stereo - Stereo matching with slanted support windows. In *BMVC*, 2011.

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

[10] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *CVPR*, 2009.

[11] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[12] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[13] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

[14] E. S. L. Gastal and M. M. Oliveira. Domain transform for edge-aware image and video processing. In *SIGGRAGH*, 2011.

[15] F. Güney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *CVPR*, 2015.

[16] K. He, J. Sun, and X. Tang. Guided image filtering. In *ECCV*, 2010.

[17] P. Heise, S. Klose, B. Jensen, and A. Knoll. PM-Huber: PatchMatch with Huber regularization for stereo matching. In *ICCV*, 2013.

[18] M. Hornek, F. Besse, J. Kautz, A. Fitzgibbon, and C. Rother. Highly overparameterized optical flow using PatchMatch belief propagation. In *ECCV*, 2014.

[19] R. Kothapa, J. Pacheco, and E. Sudderth. Max-product particle belief propagation. *Technical report, Brown University*, 2011.

[20] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011.

[21] M. Leordeanu, A. Zanfir, and C. Sminchisescu. Locally affine sparse-to-dense matching for motion and occlusion estimation. In *ICCV*, 2013.

[22] C.-K. Liang, C.-C. Cheng, Y.-C. Lai, L.-G. Chen, and H. H. Chen. Hardware-efficient belief propagation. In *CVPR*, 2009.

[23] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *TPAMI*, 33(5), 2011.

[24] J. Lu, K. Shi, D. Min, L. Lin, and M. N. Do. Cross-based local multipoint filtering. In *CVPR*, 2012.

[25] J. Lu, H. Yang, D. Min, and M. N. Do. PatchMatch Filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *CVPR*, 2013.

[26] X. Mei, X. Sun, M. Zhou, shaohui Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *ICCV Workshop*, 2011.

[27] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do. Fast global image smoothing based on weighted least squares. *TIP*, 23(12):5638–5653, 2014.

[28] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*, 4(1):1–74, 2009.

[29] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon. High quality depth map upsampling for 3d-tof cameras. In *ICCV*, 2011.

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

[31] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.

[32] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, 2011.

[33] L. Sevilla-Lara, D. Sun, E. G. Learned-Miller, and M. J. Black. Optical flow estimation with channel constancy. In *ECCV*, 2014.

[34] D. Sun, C. Liu, and H. Pfister. Local layering for joint motion estimation and occlusion detection. In *CVPR*, 2014.

[35] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *IJCV*, 106(2):115–137, 2014.

[36] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, 1998.

[37] H. Trinh and D. McAllester. Unsupervised learning of stereo vision with monocular cues. In *BMVC*, 2012.

[38] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013.

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

[40] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *TPAMI*, 34(9):1744–1757, 2012.

[41] S. Xu, F. Zhang, X. He, X. Shen, and X. Zhang. PM-PM: PatchMatch with Potts model for object segmentation and stereo matching. *TIP*, 24(7):2182–2196, 2015.

[42] H. Yang, W.-Y. Lin, and J. Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *CVPR*, 2014.

[43] J. Yang and H. Li. Accurate optical flow estimation with piecewise parametric model. In *CVPR*, 2015.

[44] K. Yoon and I. Kweon. Adaptive support-weight approach for correspondence search. *TPAMI*, 2006.