PatchMatch Filter: Edge-Aware Filtering Meets Randomized Search for Visual Correspondence

Jiangbo Lu, *Senior Member, IEEE,* Yu Li, Hongsheng Yang, Dongbo Min, *Senior Member, IEEE,* Weiyong Eng, and Minh N. Do, *Fellow, IEEE*

Abstract—Though many tasks in computer vision can be formulated elegantly as pixel-labeling problems, a typical challenge discouraging such a discrete formulation is often due to computational efficiency. Recent studies on fast cost volume filtering based on efficient edge-aware filters provide a fast alternative to solve discrete labeling problems, with the complexity independent of the support window size. However, these methods still have to step through the entire cost volume exhaustively, which makes the solution speed scale linearly with the label space size. When the label space is huge or even infinite, which is often the case for (subpixel-accurate) stereo and optical flow estimation, their computational complexity becomes quickly unacceptable. Developed to search approximate nearest neighbors rapidly, the PatchMatch method can significantly reduce the complexity dependency on the search space size. But, its pixel-wise randomized search and fragmented data access within the 3D cost volume seriously hinder the application of efficient cost slice filtering. This paper presents a generic and fast computational framework for general multi-labeling problems called PatchMatch Filter (PMF). We explore effective and efficient strategies to weave together these two fundamental techniques developed in isolation, i.e., PatchMatch-based randomized search and efficient edge-aware image filtering. By decompositing an image into compact superpixels, we also propose superpixel-based novel search strategies that generalize and improve the original PatchMatch method. Further motivated to improve the regularization strength, we propose a simple yet effective cross-scale consistency constraint, which handles labeling estimation for large low-textured regions more reliably than a single-scale PMF algorithm. Focusing on dense correspondence field estimation in this paper, we demonstrate PMF's applications in stereo and optical flow. Our PMF methods achieve top-tier correspondence accuracy but run much faster than other related competing methods, often giving over 10-100 times speedup.

Index Terms—Approximate nearest neighbor, edge-aware filtering, stereo matching, optical flow

1 Introduction

Many computer vision tasks such as stereo, optical flow and dense image alignment [24] can be formulated elegantly as pixel-labeling problems. In general, the common goal is to find a labeling solution that is spatially smooth and discontinuity-preserving, while matching the observed data/label cost at the same time. To achieve this goal, a Markov Random Field (MRF)-based energy function is often employed which involves a data term and a pairwise smoothness term [38]. However, a serious challenge posed to this discrete optimization framework is computational complexity, as global energy minimization algorithms such as graph cut or belief propagation become very slow when the image resolution is high or the label space is large. Recently, edge-aware filtering (EAF) of the cost volume [34], [25] has emerged as a competitive and fast alternative to energy-based global approaches. Though simple, cost volume filtering techniques can achieve high-quality labeling results efficiently. However, despite their runtime being independent of the filter kernel size, EAF-based methods do not scale well to large label spaces.

J. Lu and Y. Li are with the Advanced Digital Sciences Center, Singapore (e-mail: jiangbo.lu@adsc.com.sg; li.yu@adsc.com.sg); H. Yang is with Google, USA (e-mail: yhs@google.com); D. Min is with Chungnam National University, Korea (e-mail: dbmin@cnu.ac.kr); W. Eng is with Multimedia University, Malaysia (e-mail: engweiyong@gmail.com); M. N. Do is with the University of Illinois at Urbana-Champaign, USA (e-mail: minhdo@illinois.edu). *This work was done when Hongsheng, Dongbo and Weiyong were working at ADSC. This study is supported by the HCCS research grant from A*STAR.*

Almost concurrently, computing approximate nearest-neighbor field (ANNF) has been advanced remarkably by the recent PatchMatch method [6] and methods improving it [7], [20], [16]. The goal of ANNF computation is to find for each image patch P centered at pixel p one or k closest neighbors in appearance from another image. In the energy minimization context, ANNF's sole objective is to search for one or k patches that minimize the dissimilarity or the data term with a given query patch, but the spatial smoothness constraint is not enforced at all. This fact is consistent with ANNF's desire of *mapping incoherence* [20] that is crucial for image reconstruction quality. The complexity of ANNF methods is only marginally affected by the label space size i.e., the number of correspondence candidates, which is vital for interactive image editing tasks [6].

Then a motivating question that follows is – whether these two independently developed fast algorithms, i.e., PatchMatch-based randomized search and EAF, can be seamlessly woven together to address the curse of large label spaces very efficiently, while still maintaining or even improving the solution quality. For the very first time, this paper is positioned to solve this interesting yet challenging problem of general applicability to many vision tasks. However, this goal is nontrivial. First, these two algorithms have different objective functions to optimize for. As shown in Fig. 1(c, d), ANNF estimated by PatchMatch [6] is very "noisy" and dramatically inferior to the desired true flow map. Second, their computation

and memory access patterns are significantly disparate. In fact, the random and fragmented data access strategy within the cost volume effected by PatchMatch is drastically opposed to the highly regular and deterministic computing style of EAF methods.

Our main contribution is to propose a generic and fast computational framework for general multi-labeling problems called PatchMatch Filter (PMF). We take compact superpixels and subimages parsimoniously containing them as the atomic data units, and perform random search, label propagation and efficient cost aggregation collaboratively for them. This enables the proposed PMF framework to benefit from the complementary advantages of PatchMatch and EAF while keeping the overhead at a minimum. PMF's run-time complexity is independent of the aggregation kernel size and only proportional to the logarithm of the search range [6]. We further propose superpixel-based efficient search strategies that generalize and improve the original PatchMatch method [6]. Though not limited to the correspondence field estimation, PMF's applications in stereo matching and optical flow estimation are instantiated and evaluated in this paper. The label space considered is often huge or even infinite due to e.g., two-dimensional motion search space, displacement in subpixel accuracy, or over-parameterized surface or motion modeling [9]. Experiments show our PMF methods achieve top-tier correspondence accuracy also with a superior advantage of over 10-100x speedup over other competing methods.

An early version of this work was published in CVPR'13 [26]. The current paper presents this technique in more depth and detail. In addition, we propose a computationally efficient cross-scale labeling consistency constraint, which brings noticeable quality improvements for challenging low-textured image regions while maintaining the advantages of the original PMF method [26]. Furthermore, we also evaluate the proposed algorithm on the challenging MPI Sintel optical flow datasets [12], and report its performance comparison with other leading methods. Based on these evaluations, some distinctive features of the PMF algorithm can be summarized. First, PMF is able to achieve top-tier performance on a few image matching tasks, even compared with the leading task-specific approaches, such as DeepFlow [41] and PPM [46] for the Sintel optical flow, and PM-Huber [18] and PM-PM [43] for subpixel accurate stereo. Second, PMF has an easy-to-implement workflow without involving complex energy terms or optimization. Compared to other recent MRF inference methods [8], [39] only tested on a single matching task, PMF shows its strong results on *both* continuous stereo matching and large displacement optical flow, while running two orders of magnitude faster than [8], [39].

2 Related Work

Here we review the work most related to our method.

Cost-volume filtering and EAF. Though the MRF-based energy minimization formulation for discrete labeling problems is elegant [38], the energy minimization process is still time-consuming even with modern global optimization algorithms. Leveraging the significant recent advance in edge-aware image filtering, e.g. [40], [30], [17], several methods have been proposed for fast cost-volume filtering [34], [25]. They often achieve labeling results as good as those obtained by global energy-based approaches but at much faster speed, with the complexity typically independent of the filter kernel size. However, filtering each cost slice individually, albeit allowing straightforward application of various efficient EAF techniques, makes the runtime scale linearly with the label space size. This makes discrete approaches very slow in the case of large label spaces.

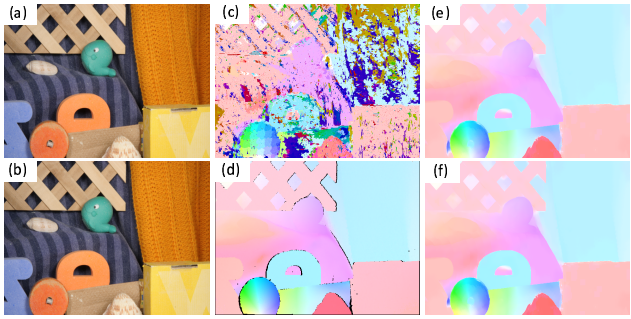

Fig. 1. Problems with PatchMatch [6] and CostFilter [34] for correspondence field estimation. (a,b) Input images. (c) ANNF of PatchMatch (with the same color coding for optical flow). (d) Ground-truth flow [1]. (e) Flow map of CostFilter [34]. (f) Flow map of our PMF method, running 10-times faster than [34] under fair settings. Average endpoint error of (e) 0.0837 and (f) 0.0825.

ANNF computation and PatchMatch. As explained before, computing ANNF for every patch in a given image with another image is computationally challenging, due to the large search space. Recent years have witnessed significant progress in accelerating this computation, which is key to non-parametric patch sampling used in many vision and graphics tasks. Motivated by the coherent natural structure in images, the PatchMatch method [6], [7] devised a very efficient randomized search and nearest-neighbor propagation approach, achieving substantial improvements in speed and memory efficiency over the prior arts. Inspired by PatchMatch, a few faster algorithms [20], [16] have been proposed which in one way or another allow efficient propagation from patches similar in appearance. However, with its objective to find the nearest neighbors, the computed ANNF is very different from the true visual correspondence field which is spatially smooth and discontinuity-preserving.

PatchMatch-based correspondence field estimation. Realizing PatchMatch's power in efficient search, Bleyer *et al.* [9] proposed to overparameterize disparity by estimating an individual 3D plane at each pixel. They showed that this method can deal with slanted surfaces much better than previous methods and achieved leading subpixel disparity accuracy. This idea has also been integrated into a global optimization framework to accel-

erate the message passing speed [8]. To handle disparity discontinuities, adaptive-weight cost aggregation [48] in 35×35 windows is used in [9]. Though PatchMatch can significantly reduce the complexity dependency on the label space size, such a brute-force adaptive-weight summation has a linear complexity dependent on the window size and it slows down the overall runtime greatly. In addition, other challenging dense correspondence problems such as optical flow are not addressed in these methods [9], [8]. It is also worth noting that the histogram-based disparity prefiltering scheme [29] was proposed to reduce the complexity caused by large label spaces down to processing only e.g. 10% plausible disparities detected for each pixel. But this reduction is not as aggressive as in PatchMatch, and also efficient local cost aggregation was not supported.

Since the publication of our early work [26], other interesting works have also been proposed to leverage the PatchMatch idea for visual correspondence field estimation. For instance, Heise *et al.* [18] applied the Huber regularization to the PatchMatch stereo approach [9] and solved it using a convex optimization. Recently, Xu *et al.* [43] proposed a convex formulation of the multi-label Potts Model with [9] as well. Though both techniques demonstrated very competitive results in subpixel accurate stereo reconstruction, they are still much slower than the proposed PMF method. It is explicitly discussed in [43] that accelerating the cost aggregation step (e.g. using a window of 41×41) through a PMF-like algorithm remains as a future work. In addition to stereo matching, PatchMatch or ANNF techniques have also been used in recent optical flow estimation algorithms. For instance, Chen *et al.* [14] designed a complex motion segmentation pipeline together with continuous flow refinement, which computes NNF to generate initial motion matches. Though achieving a high estimation accuracy, this method is still too slow for practical applications. Bao *et al.* [5] used a local PatchMatch-like data aggregation with a coarse-to-fine framework, but this method tends to lose fine-grained motion details and also has difficulties in handling large textureless regions. Based on a simple and more general-purpose computational framework, the proposed PMF algorithm demonstrates strong estimation results and fast runtimes on both subpixel stereo matching and large displacement optical flow benchmark datasets.

3 Cost Volume Filtering

We briefly present a general framework and notations of cost volume filtering-based methods for discrete labeling problems, and focus particularly on visual correspondence field estimation. As in [34], given a pair of images I and I', the goal is to assign each pixel $p = (x_p, y_p)$ a label l from the label set $\mathcal{L} = \{0, 1, ..., L-1\}$. L denotes the label space size. For general pixel-labeling problems, the label l to be assigned can represent different local quantity [38]. For stereo and optical flow problems considered here, $l = (u, v)$, where u and v correspond to the displacement in x and y directions. Stereo degenerates to assigning a disparity d ($u = d$) to pixel p, where $v = 0$.

Unlike global optimization-based discrete methods [38], local window-based methods stress reliable cost aggregation from the neighborhood and evaluate exhaustively every single hypothetical label $l \in \mathcal{L}$. The final label l_p for each pixel p is decided with a Winner-Takes-All (WTA) scheme. To achieve spatially smooth yet discontinuity-preserving labeling results, edge-aware smoothing filters have been adopted in the local cost aggregation step of several leading local methods [34], [25]. Given the raw cost slice $C(l)$ computed for a label l, we denote its edge-aware filtered output as $\tilde{C}(l)$. Then the filtered cost value at pixel p is given as:

$$\tilde{C}_p(l) = \sum_{q \in W_p(r)} \omega_{q,p}(I) C_q(l) \ . \tag{1}$$

$W_p(r)$ is the local aggregation window centered at p with a filter kernel radius r. $\omega_{q,p}(I)$ is the normalized adaptive weight of a support pixel q, which is defined based on the structures of the image I. Various EAF methods [40], [30], [17], [25] can be applied here, and they differ primarily in the ways of defining and evaluating $\omega_{q,p}(I)$.

Though EAF is very efficient, the linear complexity dependency on the label space size L requires repeated filtering of $C(l)$ as in Eq. (1), and $C(l)$ is of the same size of I. This makes the runtime unacceptably slow when L is large. To largely remove this complexity dependency, recent techniques such as PatchMatch [6] appear helpful conceptually. However, it can be discerned that PatchMatch's randomized label space visit pattern for each individual pixel p is very incompatible with the regular image-wise cost filtering routine that is essential to the efficiency of EAF-based methods.

4 PatchMatch Filter Using Superpixels

This section proposes a superpixel-based computational framework for fast correspondence field estimation by exploiting PatchMatch-like random search and EAF-based cost aggregation synergistically. Our key motivation draws from the observation that labeling solutions for natural images are often spatially smooth with discontinuities aligned with image edges, in contrast to the very "noisy" ANNF (see Fig. 1). The very nature of spatially coherent ground-truth labeling solutions actually advocates a collaborative label search and propagation strategy for similar pixels covered in the same compact superpixel, without necessarily going to the pixel-wise fine granularity in PatchMatch [6].

Another key motivation from a computing perspective is that the efficiency of EAF essentially comes from the high computational redundancy or the vast opportunity for shared computation reuse among neighboring pixels when filtering an image or cost slice. However, PatchMatch processes each pixel with its random set of label candidates individually in raster scan order. This renders EAF techniques not applicable and the cost aggregation

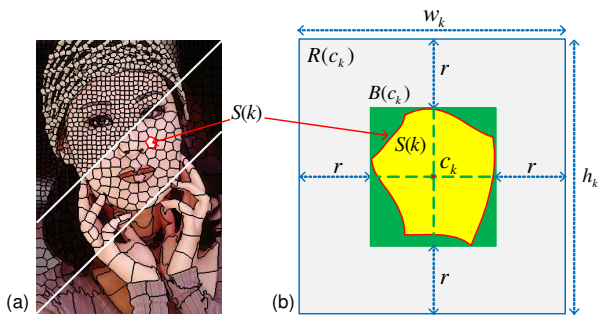

Fig. 2. (a) SLIC superpixels of approximate size 64, 256 and 1024 pixels. Fig. courtesy from [3]. (b) Bounding-box $B(c_k)$ containing the superpixel $S(k)$ centered at pixel c_k and r-pixel extended subimage $R(c_k)$.

runtime to grow linearly with the filter kernel size $m = (2r+1)^2$ [9], resulting in heavy computational loads.

Based on the above analysis, we propose to partition the input image into non-overlapping superpixels, and use them as the basic units for performing random search, propagation and subimage-based efficient cost aggregation collaboratively. As a spatially regularized labeling solution is favored, such a superpixel-based strategy, adapting to the underlying image structures, is more consistent with the goal of correspondence field estimation than its pixel-based counterpart. Compared to the propagation from the immediate causal pixels [6], taking superpixels as the basic primitive also effectively extends the propagation range and ameliorates the issue of being trapped in local optimum. More importantly, superpixel-based collaborative processing creates desired chances for computation reuse and speedup.

4.1 Superpixel-Based Image Representation

As a key building block to many computer vision algorithms, superpixel decomposition of a given image has been actively studied. In this paper, we choose the recently proposed SLIC superpixel algorithm [3] to decompose an input color image I into K non-overlapping superpixels or segments, i.e., $S = \{S(k) | \bigcup_{k=1}^{K} S(k) = I \text{ and } \forall k \neq l, S(k) \cap S(l) = \emptyset\}$. Compared to other graph-based superpixel algorithms e.g. [15], the SLIC method yields state-of-the-art adherence to image boundaries, while having a faster runtime linear in the number of pixels M. Another important advantage is that SLIC superpixels are compact and of more regular shapes and sizes (M/K on average), giving a low overhead when their bounding-boxes are sought as discussed later. Spatial compactness also assures that the pixels from the same superpixels are more likely to share similar optimal labels. Fig. 2(a) shows SLIC superpixels generated with different parameters. For the convenience of presentation, we also define two additional variables. As shown in Fig. 2(b), for a given segment $S(k)$, $B(c_k)$ represents its minimum *bounding-box* centered at pixel c_k and $B(c_k) \in I$. We use $R(c_k)$ to denote the *subimage* that contains $B(c_k)$, but with its borders extended outwards by r pixels while being restricted to remain within I.

4.2 PatchMatch Filter Algorithm

Now we present the PatchMatch filter (PMF) – a general computational framework to efficiently address discrete labeling problems, exploiting superpixel-based PatchMatch search and efficient edge-aware cost filtering. The PMF framework is general and allows the integration of various ANNF and EAF techniques. We will present improved superpixel-based search strategies in Sect. 4.3.

Unlike the regular image grid that has a default neighbor system, an adjacency (or affinity) graph is first built for an input image decomposed into K superpixels in a preprocessing step. We use a simple graph construction scheme here: every segment serves as a graph node, and an edge is placed between two segments if their boundaries have an overlap. Similar to PatchMatch [6], a random label is then assigned to each node. After this initialization, we process each superpixel $S(k)$ roughly in scan order. The PMF algorithm iterates two search strategies in an interleaved manner, i.e., *neighborhood propagation* and *random search*.

First, for a current segment $S(k)$, we denote its set of spatially adjacent neighbors as $\mathcal{N}(k) = \{S(i)\}$. A *candidate pixel* $t \in S(i)$ is then randomly sampled from every neighboring segment, totaling a number of $|\mathcal{N}(k)|$. As a result, a set of current best labels $\mathcal{L}_t = \{l_t\}$ assigned to the sampled pixel set $\{t\}$ can be retrieved, and they are propagated to the superpixel $S(k)$ under consideration. Given this set of propagated labels $\mathcal{L}_t$, EAF-based cost aggregation in Eq. (1) is then performed for the subimage $R(c_k)$ defined for $S(k)$, but the filtering result is used only for the pixels in $B(c_k)$. The reason is that pixels in $R(c_k) \backslash B(c_k)$ are not supplied with all possible support pixels needed for a reliable full-kernel filtering, and also they tend to have a lower chance of sharing similar labels with pixels in $S(k)$. We denote such a subimage-based cost filtering process over a selected set of labels with a function $\mathbf{f}$, which is defined as follows,

$$\mathbf{f} : \mathbf{C}\left(R(c_k), \{l \in \mathcal{L}_t\}\right) \mapsto \tilde{\mathbf{C}}\left(B(c_k), \{l \in \mathcal{L}_t\}\right) , \quad (2)$$

where $\mathbf{C}$ and $\tilde{\mathbf{C}}$ represent the raw and filtered cost volume of cross-section size of $|R(c_k)|$ and $|B(c_k)|$, respectively. For any pixel $p \in B(c_k)$, its current best label l_p is updated by a new label $l \in \mathcal{L}_t$ if $\tilde{\mathbf{C}}(p, l) < \tilde{\mathbf{C}}(p, l_p)$.

After the preceding propagation step, a center-biased random search as in PatchMatch [6] is performed for the current segment $S(k)$. It evaluates a sequence of random labels $\mathcal{L}_r$ sampled around the current best label l^* at an exponentially decreasing distance. We set the fixed ratio α between two consecutive search scopes [6] to $1/2$. Different ways exist to define l^*. Here we randomly pick a *reference pixel* $s \in S(k)$ to promote the label propagation within a segment. We set $l^* = l_s$, where l_s is the current best label for s. The function $\mathbf{f}$ is then applied again to filter those cost subimages specified by $\mathcal{L}_r$ by substituting for $\mathcal{L}_t$ in Eq. (2).

To remove unnecessary computation, a list recording the labels that have been visited for each segment $S(k)$ is maintained. Therefore, no subimage filtering will be

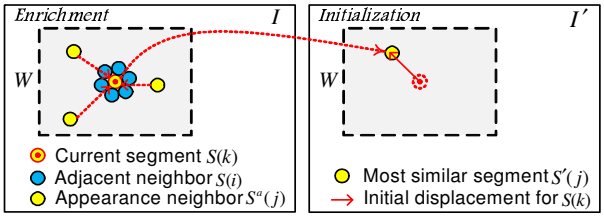

Fig. 3. Generalized affinity graph and improved strategies: superpixel-induced *enrichment* and *initialization*.

needed if a candidate label has been visited before. It is also clear from Fig. 2(b) that compact superpixels $S(k)$ are favored in our PMF algorithm, as the filtering overhead incurred by the stretched sizes of $R(c_k)$ and $B(c_k)$ will be kept low.

Discussion. Note that prior stereo or optical flow methods [49], [21] often take segments as the matching units and infer a single displacement for each segment. To achieve pixel-wise accuracy, further (continuous) optimization is still required that makes them even slower. In contrast, our PMF method works like other cost-volume filtering methods [34]. It directly estimates and decides the optimal label for each pixel independently, while leveraging their shared spatial neighbors and plausible label candidates for fast computation. Also, the common weakness of segmentation-based methods, i.e., they cannot recover from segmentation errors, does not apply.

To be emphasized is that the proposed superpixel-based PatchMatch method does not reduce the number of label evaluations performed for each pixel per iteration, when compared to the original pixel-based PatchMatch methods [6], [9]. The main difference is that our PMF method performs EAF-based cost aggregation *collaboratively* for all pixels contained in a superpixel together over a set of shared label candidates, while a pixel-based PatchMatch method [9] evaluates the label candidates generated for each pixel *individually*. With our more densely connected graph edges (involving causal and non-causal spatial neighbors plus non-local appearance neighbors to be presented shortly), the number of label candidates attempted per graph node (i.e. each superpixel) in one iteration actually increases. More importantly, a superpixel-based PatchMatch scheme can take advantage of image segmentation to implicitly promote more (long-range) spatial regularization, and allow plausible label candidates to be propagated over distance effectively. The performance gain brought by our superpixel-based algorithm over pixel-based PatchMatch methods will be shown in Sect. 6.

4.3 Superpixel-Induced Efficient Search Strategies

For the clarity sake, we presented the proposed PMF framework in Sect. 4.2 based on a baseline search and propagation strategy conceptually close to the original PatchMatch principle [6]. We further propose some improved search strategies induced by the superpixel-based image representation (see Fig. 3). Compared to the baseline PatchMatch method [6], the new strategies are more effective and efficient in finding and propagating plausible candidates.

Enrichment. First, we generalize the adjacency graph in Sect. 4.2 to add at most κ new *appearance neighbors* to every node or segment. Specifically, given a segment $S(k)$, we search within a predefined window the top κ segments $\mathcal{N}^a(k) = \{S^a(j), j = 1, 2, ..., \kappa\}$ most similar to $S(k)$. Due to arbitrary shapes and uneven sizes of different segments, we use a loose form to define the inter-segment similarity $H(S(k), S(j))$ as follows,

$$H(S(k),S(j)) = \sum_{s\in S(k), t\in S(j)} \exp\left(-\frac{\|s-t\|^2}{\sigma_s^2} - \frac{\|I_s - I_t\|^2}{\sigma_r^2}\right). \quad (3)$$

s and t denote pixels randomly sampled from segment $S(k)$ and $S(j)$, respectively. We repeat this random pair sampling for a fixed number of times, e.g. 10% of the average superpixel size. σ_s and σ_r control the spatial and color similarity. Picking the top κ segments $\{S^a(j)\}$ closest to $S(k)$ and also above a similarity threshold, $\mathcal{N}^a(k)$ augments the original spatial neighbor set $\mathcal{N}(k)$ for $S(k)$ by non-local neighbors similar in appearance. We set $\kappa = 3$ and $\sigma_s = \infty$ here. This enrichment scheme allows effective and fast propagation of plausible label candidates from similar segments. Note that other methods such as color histograms can also be used to evaluate the similarity between two superpixels in Eq. (3).

Initialization. As image representation in superpixels greatly reduces the graph complexity, this motivates us to design a better label initialization strategy than the random initialization [6]. The basic idea is to assign a potentially good candidate label rather than a random label to each segment $S(k)$. Given the maximum label search range W, we select for segment $S(k)$ in image I a closest segment $S'(j)$ from the target image I' within a slightly enlarged range. The similarity between segments is evaluated as in Eq. (3), but with σ_s decreased to 100 to favor spatially close segments. The displacement vector between the centroids of $S(k)$ and $S'(j)$ is used as the initial label for $S(k)$. Such a preprocessing method of low complexity makes PMF converge faster and tackles small objects with large displacements better.

4.4 Adaptive Cross-Scale Consistency Constraint

Up to this point, the PMF technique is designed as a fast labeling algorithm that takes advantage of EAF for cost aggregation and randomized label search and refinement. Though it works quite well as a significantly accelerated alternative to cost volume filtering, PMF still faces the same challenge when dealing with large textureless regions (see Fig. 4). This is largely due to the limited labeling regularization power provided by local cost aggregation, where a global smoothness constraint is not explicitly enforced. With the aim of tackling this challenge in a computationally efficient way, we propose a cost-effective approach to improve the matching accuracy of the PMF algorithm, which is termed **fPMF**.

The key idea originates from a general observation that correspondences estimated at a coarse image scale tend to be more reliable for weakly-textured regions, where a stronger regularization helps resolving ambiguous visual matches. However, on the other hand, visual correspondences estimated at a fine image scale localize and preserve structure or motion details much better. With the goal of estimating a high-quality correspondence field with both coherence and details ultimately at the full image scale, we propose to incorporate a spatially adaptive, cross-scale consistency constraint into a hierarchical image matching workflow. Basically, we construct an image pyramid for each image of a given pair, and then apply a slightly modified PMF algorithm to each image scale, allowing pixels on a fine scale to integrate the "guidance" from the labeling results of their parents estimated at a coarse scale.

Specifically, for a fine scale of the constructed image pyramid (we empirically set the number of image scales to 2 in this paper), an approximate texture and textureless region classification map Υ is quickly computed at first. The binary classification map $\Upsilon = \{\Upsilon_p\}$: $\mathbb{Z}^2 \mapsto \{0, 1\}$ classifies a pixel p as either from a textured region ($\Upsilon_p = 1$) or from a textureless region ($\Upsilon_p = 0$). The key motivation is that for textureless regions, label estimation from a coarse layer should enforce a stronger smoothness constraint over the corresponding child nodes at an adjacent fine scale; while for the textured regions, this constraint should be attenuated to favor detail-preserving estimation results from the fine layers.

Based on this guideline, given a pixel p and a candidate label l[1], we slightly modify the aggregated cost $\tilde{C}_p(l)$ by adding a cross-scale consistency cost

$$\hat{C}_p(l) = \tilde{C}_p(l) + \lambda_p \cdot \left\| l - l^*_{p_a} \right\|_1 \ , \tag{4}$$

where p_a denotes the pixel p's parent node in the coarse scale and the label assigned to it is $l^*_{p_a}$. The weighting parameter λ_p is adaptively decided as follows:

$$\lambda_p = \Upsilon_p \cdot \lambda_1 + (1 - \Upsilon_p) \cdot \lambda_2 \ . \tag{5}$$

The two constants λ_1 and λ_2 (with $\lambda_1 \ll \lambda_2$) are set to control the parent-child label regularization strength adaptively for pixels in the textured and textureless regions, respectively. From Eq. (4), it is easy to see the computation of $\hat{C}_p(l)$ incurs only a minimal complexity overhead over computing $\tilde{C}_p(l)$, based on a precomputed classification map Υ. When this cross-scale consistency constraint is turned on, for any pixel p at a fine image scale, the new cost $\hat{C}_p(l)$ rather than $\tilde{C}_p(l)$ is used in the label update process with the WTA scheme.

Now we turn to the task of precomputing the classification map Υ for the input image I. In fact, it is not necessary to compute an exact texture/textureless classification map, because the imprecise smoothness constraint caused by small misclassified regions is insufficient to make a wrong label to be favored. The reason is that such misclassifications (if any) often occur near object boundaries, where a highly reliable aggregated cost $\tilde{C}_p(l)$ providing a strong discriminative power is usually available. This means the side effect of inappropriately using a soft consistency constraint is typically not on par with the strong matching evidence collectively contributed by neighboring pixels within a local support window. Moreover, our post-processing steps such as weighted median filtering presented in Sect. 5 is particularly good at correcting this kind of outliers. Therefore, we use a simple method to calculate $\{\Upsilon_p\}$ efficiently. First, we evaluate the density of the Canny edge pixels [13] in a local neighborhood window (3×3) for each pixel. A hard thresholding is then applied to classify pixels with a high edge density as pixels from textured regions, while the rest of the image as textureless regions.

It is worth noting that our cross-scale consistency constraint differs a lot from the conventional practice of applying a coarse-to-fine estimation procedure [11], [21], [24], [5], which has well-known issues such as loss of structure/motion details and difficulty in capturing small objects undergoing large displacements [42]. Instead of strictly committing to a local neighborhood search based on label results from a coarse level, the cross-scale constraint in Eq. (4) actually allows for a full-range label search at a fine scale while taking sensible consideration of the coarse-scale label assignment. We notice such a cross-scale regularization scheme is some-

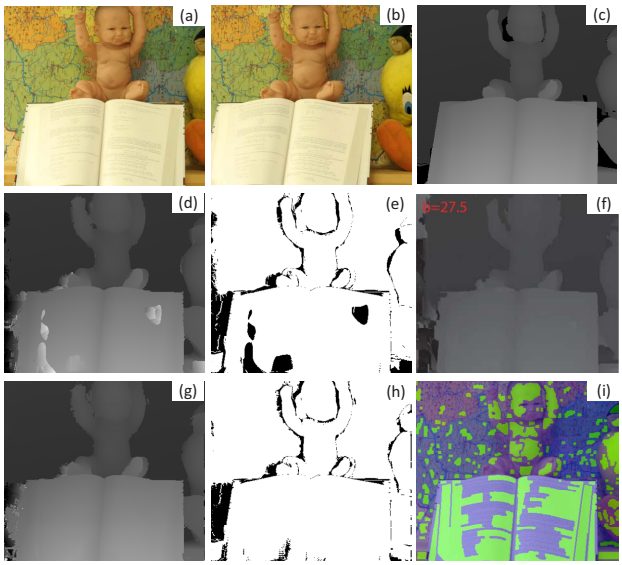

Fig. 4. Strength of the cross-scale consistency constraint in matching large low-textured regions. (a, b) Input *Baby2* stereo image pair. (c) Ground-truth depth map. (d, e) Depth map and error map of the PMF algorithm (without post-processing). (f) Depth map of the PMBP method [8] with a strong regularization weight b (When $b = 0$, the resulting PM-stereo method [9] struggles with the low-textured regions.). (g, h) Depth map and error map of the fPMF algorithm (without post-processing). (i) The binary classification map Υ superimposed on the left input image (green pixels denote the classified textureless regions, otherwise textured regions). It is generated to adaptively adjust the cross-scale consistency constraint in fPMF.

1. For simplicity, the converted disparity is used instead of the plane parameters for the L_1 distance in Eq. (4) in our slanted-surface stereo.

TABLE 1
Complexity comparison of three different techniques

	CostFilter [34]	PatchMatch [9]	PMF
Complexity	$O(ML)$	$O(mM\log L)$	$O(M\log L)$
Memory	$O(M)$	$O(M)$	$O(M)$

what similar to the inter-layer motion smoothness term used in a global optimization formulation [19]. However, our cross-scale regularization constraint is adjusted in a content-sensitive manner for different image regions, and also it is cheap to compute and well compatible with the fast PMF routine. We also make a distinction from a very recent work improving EpicFlow [33] for optical flow estimation [4], where a hierarchical correspondence search strategy is proposed. Though their purpose [4] is to propagate potentially good flow values from non-local pixels (due to the subsampled neighborhood structures at coarse image levels) as a *data term* issue, our design focuses on improving the end labeling coherence of the proposed PMF as a general discrete labeling approach.

4.5 Overall Algorithm and Complexity

The PMF algorithm integrated with the cross-scale consistency constraint is summarized in Algorithm 1.

Next, we discuss the complexity of the single-scale PMF algorithm. Given an image of size M, the label space size L and the superpixel number K, we further denote the total area size of subimages by $\tilde{R} = \sum_{k=1}^{K} |R(c_k)|$. Enabling the integration of linear-time EAF techniques for cost filtering, our PMF approach removes the complexity dependency on the matching window size m, in contrast to the PatchMatch methods [6], [9]. Consequently the complexity of our PMF is $O(K^2 + \tilde{R}\log L)$, with $O(K^2)$ accounting for the complexity upper bound of the new initialization strategy in Sect. 4.3. This overhead is negligible, because searching for similar segments can be well constrained in a predefined search window. The dominant part of PMF is then $O(\tilde{R}\log L) \approx O(M\log L)$, as $\tilde{R}$ is larger than M by a factor of a small leading constant. Table 1 gives the comparison, where the $\log L$ terms (thanks to the use of PatchMatch) were discussed in its original paper [6].

The memory complexity of the PMF method is $O(M + K\log L)$. $O(M)$ is used to hold the filtered cost associated with the current best label at each pixel. Much less than $O(M)$, $O(K\log L)$ records the list of the labels that have been visited for each segment $S(k)$. In our implementation, we pre-organize all the subimages $\{R(c_k)\}$ of the input image I into an array of compact 2D buffers, which facilitates cost computation and filtering next.

Algorithm 1: The PMF algorithm for a given scale

Input: (1) A pair of images I and I' for dense correspondence estimation. (2) The label map estimated with PMF from the immediate coarse scale, when the cross-scale consistency constraint (Sect. 4.4) is turned on.
Discrete label search space: $\mathcal{L} = \{0, 1, ..., L-1\}$.
Output: The estimated pixel-wise label map $L = \{l(p)\}$.

/ Initialization */*
1: Partition I into a set of disjoint K segments $I = \{S(k), k = 1, 2, ..., K\}$ and build adjacency graph $\mathcal{G}$.
2: Assign a random label l_k to each segment $S(k)$. For each pixel $p \in S(k)$, set $l_p = l_k$. (Optionally, the improved initialization scheme in Sect. 4.3 can be applied.)
3: **if** *the cross-scale consistency constraint is turned on & the current scale is not the coarsest scale* **then**
  Estimate a binary map Υ to classify pixels into textured or textureless regions for I.

/ Iterative label search and optimization */*
repeat
  for $k = 1 : K$ **do**
    4: Propagate a set of labels $\mathcal{L}_t$ randomly sampled from neighboring segments to the segment $S(k)$. (The enrichment scheme in Sect. 4.3 can be optionally applied here to augment $\mathcal{L}_t$ with plausible label candidates.)
    for $l \in \mathcal{L}_t$ **do**
      5: Evaluate the raw matching cost $C_q(l)$ for each pixel $q \in R(c_k)$ with Eq. (7) (or Eq. (8)) .
      6: Compute the aggregated cost $\tilde{C}_p(l)$ for each pixel $p \in B(c_k)$ with Eq. (1).
      7: **if** *the cross-scale consistency constraint is turned on & the current scale is not the coarsest scale* **then**
        Compute $\hat{C}_p(l)$ with Eq. (4).
        $\tilde{C}_p(l) \longleftarrow \hat{C}_p(l)$.
      8: **if** $\tilde{C}_p(l) < \tilde{C}_p(l_p), \forall p \in B(c_k)$ **then**
        $l_p \longleftarrow l$.
    9: Decide for $S(k)$ a representative label l_k^* and generate a set of random labels $\mathcal{L}_r$ around l_k^*.
    10: Perform random label candidates evaluation and update by following Step **5**–**8** for $l \in \mathcal{L}_r$.
until *convergence or the maximum iteration number.*

5 Applications

We present two applications of the proposed PMF framework: stereo matching and optical flow estimation. As for the EAF techniques, we use the guided filter (GF) [17] and the zero-order cross-based local multipoint filter (CLMF-0) [25] in this paper, though other methods can be easily employed in our framework as well. Both techniques have a linear time complexity to compute Eq. (1), depending only on the image size M but not on the filter kernel size m.

5.1 Subpixel Stereo with Slanted Support Windows

We present two different PMF-based stereo methods that model the scene disparity and parameterize the corresponding label space differently. Like most stereo methods [34], [25], the first approach makes an assumption of fronto-parallel local support windows, whereby pixels inside are matched to pixels in another view at a constant (integer) disparity. We call this method **PMF-C**. Similar to [9], the second approach attempts to estimate a 3D plane Q_p at each pixel p, so pixels lying on the same slanted surfaces can then be used for reliable cost aggregation with high subpixel precision. This method

is called **PMF-S**. Both methods can benefit from the PMF technique, as the disparity search range can be quite large due to high-resolution stereo images or an infinite number of possible 3D planes. Since PMF-S solves a more generalized and challenging labeling problem than PMF-C, we focus on presenting and evaluating PMF-S.

Slanted surface modeling. For each pixel p, we search for a 3D plane Q_p defined by a three-parameter vector $\mathbf{l}_p = (a_p, b_p, c_p)$. Given such a plane, a support pixel $q = (x_q, y_q)$ in p's neighborhood $W_p(r)$ in the left view I will be projected to $q' = (x_{q'}, y_{q'})$ in the right view I' as:

$$x_{q'} = x_q - d_q = x_q - \mathbf{l}_p \cdot (x_q, y_q, 1)^\top \ , \text{ and } \ y_{q'} = y_q \ . \quad (6)$$

In Eq. (6) d_q is computed from the plane equation whose value exists in a continuous domain. This enables PMF-S to handle slanted scene objects much better than PMF-C by avoiding discretization of disparities.

Raw matching cost. For PMF-C and PMF-S, we compute the raw matching cost between a pair of hypothetical matching pixels q and q' in the similar way as [34]:

$$\begin{aligned} C_q(l) = (1-\beta) \cdot \min\left(\left\|I_q - I'_{q'}\right\|, \gamma_1\right) \\ + \beta \cdot \min\left(\left\|\nabla I_q - \nabla I'_{q'}\right\|, \gamma_2\right) \ . \end{aligned} \quad (7)$$

For PMF-C, the label l represents a disparity candidate d, while l corresponds to the three parameters (a_p, b_p, c_p) of a plane evaluated for the center pixel p in PMF-S. For stereo, ∇ evaluates only the gradient in x direction in Eq. (7). The color and gradient dissimilarity is combined using a user-specified parameter β. γ_1 and γ_2 are truncation thresholds. Since q' generally takes fractional x-coordinates in PMF-S, linear interpolation is used to derive its color and gradient.

PMF-based cost aggregation. We apply the PMF algorithm described in Sect. 4.2 to perform superpixel-based collaborative random search, propagation and cost subimage filtering. The implementation of cost aggregation for PMF-C is straightforward, whereas more care needs to be taken for the random plane initialization and iterative random search steps in PMF-S[2]. To this end, we adopt the approach presented in [9], and use a random unit normal vector (n_x, n_y, n_z) plus a random disparity value sampled from the allowed continuous range as proxy for the plane representation. View propagation [9] is also used in PMF-S to propagate the plane parameters of the matching pixels.

Post-processing. After deciding an initial disparity map using a WTA strategy, we detect unreliable disparity estimates by conducting a left-right cross-checking. Then, these unreliable pixels are filled by background disparity extension [34] in PMF-C, and plane extrapolation [9] in PMF-S. Finally, a weighted median filter is applied to refine the resulting disparity map.

5.2 Optical Flow

We now present a PMF-based optical flow method named **PMF-OF**. Its main work flow closely resembles that of PMF-C, but a label l represents a displacement vector (u, v) in x and y directions. The label space for optical flow is therefore often much larger than typical label spaces tackled in stereo matching. Based on a discrete labeling formulation, PMF-OF solves for subpixel accurate flow vectors by upscaling the label dimension to allow fractional displacements along both x and y directions. As in [34], an upscaling factor of 8 is used in this paper, and the pixel colors at subpixel locations are obtained from bicubic interpolation. To tackle more challenging photometric variations and large occlusion regions between the two given images seen in the MPI Sintel datasets [12], we present additional improvements for the raw cost evaluation, cost aggregation, and post-processing modules, respectively.

Raw matching cost. Given a candidate label l, a pixel q in image I is matched to the pixel $q' = q + (u, v)$ in the second image I'. We compute the raw matching cost between two pixels q and q' using both an absolute distance (AD) and Census transform [27] as:

$$C_q(l) = \rho(C_q^{AD}(l), \tau_{ad}) + \rho(C_q^{census}(l), \tau_{cs}) \ . \quad (8)$$

$\rho(C, \tau) = 1 - \exp(-C/\tau)$ is a robust function. In our experiments, we set $\tau_{ad} = 60$ and $\tau_{cs} = 30$. The window used in the Census transform is 11×11.

PMF-based cost aggregation. The PMF-based label search and cost filtering algorithm is then applied in a manner similar to PMF-C, but PMF-OF includes the improved strategies presented in Sect. 4.3 to more effectively tackle the huge motion search space.

Quadratic optimization-based post-processing. After estimating the bidirectional flow fields between two images with a WTA strategy, we detect occluded regions through the cross-checking [34] between two fields. A simple extrapolation used in PMF-C and PMF-S is not so effective when the occluded region is big due to a large displacement optical flow. Thus, we proposed to perform a post-processing step based on a quadratic optimization, in which an objective is defined using reliable estimates and is then efficiently minimized by a sparse matrix solver (e.g. [28]). Interestingly, this method is also similar to the non-local disparity refinement used in [47] in spirit, though more principled.

We define an objective function consisting of the data term E_p and the smoothness term E_{pq} as follows,

$$E = \sum_p E_p(l_p) + \sum_p \sum_{q \in \mathcal{N}_p} E_{pq}(l_p, l_q) \ , \quad (9)$$

where $\mathcal{N}_p$ represents a set of pairwise neighbors for pixel p. Similar to [31], [47], we define the data term using the initial flow vector l_p^* and the occlusion map computed from the cross-checking technique:

$$E_p(l_p) = \begin{cases} \|l_p - l_p^*\|_2^2, & p \text{ is visible,} \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

When the pixel p is occluded, the cost value $E_p(l_p)$ is always zero. Thus, its output is determined by flow

2. Our improved strategies are not used for fair comparison with [9].

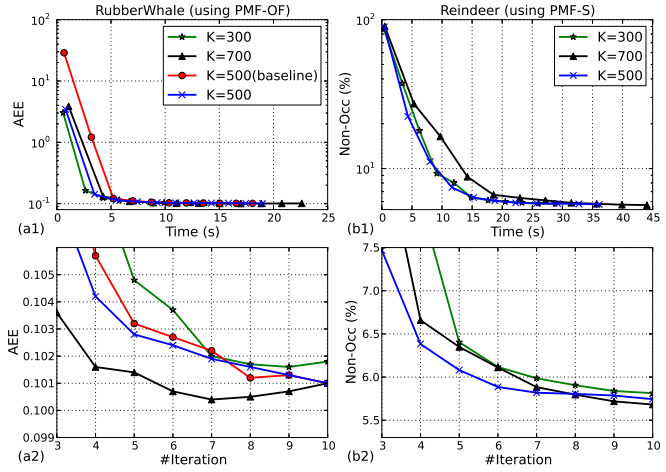

Fig. 5. Time-accuracy trade-off study of PMF methods.

vectors of reliable neighboring pixels by taking into account the following smoothness term:

$$E_{pq}(l_p, l_q) = \theta_{pq} \|l_p - l_q\|_2^2 \,, \tag{11}$$

where θ_{pq} is an adaptive weight defined by the color similarity between neighboring pixels p and q. The objective function E holds a quadratic form, and its solution is easily obtained by solving a linear system based on a large sparse matrix. We perform this post-processing independently for u and v. Solving the linear system can help propagate the flow vectors from visible pixels to occluded pixels depending on their color similarities.

6 Experimental Results

We implemented the PMF algorithm in C++, and GF [17] and CLMF-0 [25] used for EAF in Eq. (1). The following same parameter settings are used across all stereo and optical flow datasets: $\{r, \sigma_r, \beta, \gamma_1\} = \{9, 0.1, 0.9, 0.039\}$. As [34], $\gamma_2 = 0.008$ (0.016) in Eq. (7) is used for stereo (optical flow). We set the smoothness parameter $\epsilon = 0.01^2$ in GF, and the inlier threshold $\tau = 0.1$ in CLMF-0. The segment number K is set to 500.

When the cross-scale consistency constraint is enabled, we set $\lambda_1 = 0.01$ and $\lambda_2 = 0.1$ in Eq. (5). We also fix the number of image scales to 2 in our experiments. The coarse scale image is downscaled from the original images (fine scale) by reducing each side length by half. For the coarse scale correspondence estimation, the number of superpixels used and the search range along each spatial axis are also reduced by half, while all the other parameters are kept the same. All of our experiments were run on an Intel Core i5 2.5GHz CPU with a single-core implementation.

6.1 Time-Accuracy Trade-off Evaluation of PMF

First, we present a time-accuracy trade-off study of our PMF approaches in Fig. 5. Two test image pairs *RubberWhale* and *Reindeer* from the Middlebury optical flow/stereo datasets [2], [1] are used to evaluate the PMF-OF and PMF-S methods (using CLMF-0), respectively. It can be observed that for a reasonable range of K settings, optical flow or stereo results have almost always converged after 8-10 iterations. This also holds true for other images tested with GF not shown here. Fig. 6 shows the optical flow estimation results after each iteration (without applying any post-processing) for a pair of *RubberWhale* images. In addition, Fig. 5(a1) shows that our improved search strategies in Sect. 4.3 lead to a faster convergence speed than the baseline method, especially for the first few iterations. For the same iteration number, choosing a larger K (namely a smaller superpixel size) gives a better gain in accuracy on optical flow estimation than stereo, due to intrinsically more complex 2D motions. However, this is at a price of a longer runtime per iteration caused by the increased adjacency graph size and increased subimage processing overhead. In general, we find that $K = 500$ gives a good balance between the complexity of each iteration and the iteration number for a target accuracy level.

TABLE 2
Middlebury stereo evaluation [2] for error threshold = 0.5. * use GPU. ^{o} We used the source C++ code provided by the authors of [8]. For [9], we report the runtime after setting the regularization weight to zero in PMBP [8]. [captured on 29/07/2015]

Algorithm	Avg. Rank	Avg. Error	Runtime (s)
GC+LSL [39]	6.2	6.63	400*
PM-PM [43]	8.5	7.58	34*
PM-Huber [18]	8.6	7.33	52*
PMF-S	12.5	7.69	**20**
PMBP [8]	19.8	8.77	3100^o
PatchMatch [9]	28.4	9.91	1005^o

TABLE 3
Stereo evaluation results for *Teddy* and *Cones* when error threshold = 0.5 [captured on 29/07/2015]

Algorithm	*Teddy*			*Cones*		
	nocc	all	disc	nocc	all	disc
GC+LSL [39]	$\mathbf{4.20_1}$	$\mathbf{7.12_2}$	$\mathbf{12.9_3}$	3.77_8	9.16_9	10.4_{12}
PM-PM [43]	5.21_6	11.9_{11}	15.9_8	3.51_7	8.86_7	9.58_7
PM-Huber [18]	5.53_8	$\mathbf{9.36_5}$	15.9_9	$\mathbf{2.70_1}$	$\mathbf{7.90_2}$	$\mathbf{7.77_1}$
PMF-S	$\mathbf{4.45_3}$	9.44_7	$\mathbf{13.7_4}$	$\mathbf{2.89_2}$	$\mathbf{8.31_3}$	$\mathbf{8.22_2}$
PMBP [8]	5.60_9	12.0_{12}	15.5_6	3.48_6	8.88_8	9.41_6
PatchMatch [9]	5.66_{10}	11.8_{10}	16.5_{10}	3.80_9	10.2_{11}	10.2_{10}

6.2 Sub-Pixel Stereo Reconstruction Results

We first focus on evaluating the proposed PMF-S stereo method combined with the GF filtering technique [17], using the Middlebury standard stereo benchmark [2] in Table 2. (GF is found to perform slightly better than CLMF-0 [25] in the subpixel-accurate stereo task in [26]). For this evaluation, we report those leading stereo algorithms designed specifically to tackle slanted surfaces with subpixel precision, and set the Middlebury error threshold to 0.5. Table 2 shows that our PMF-S method performs better than PatchMatch stereo [9] and PMBP [8], while the latter uses belief propagation for

Fig. 6. After applying PMF for a few iterations, optical flow estimation for the *RubberWhale* images quickly converges.

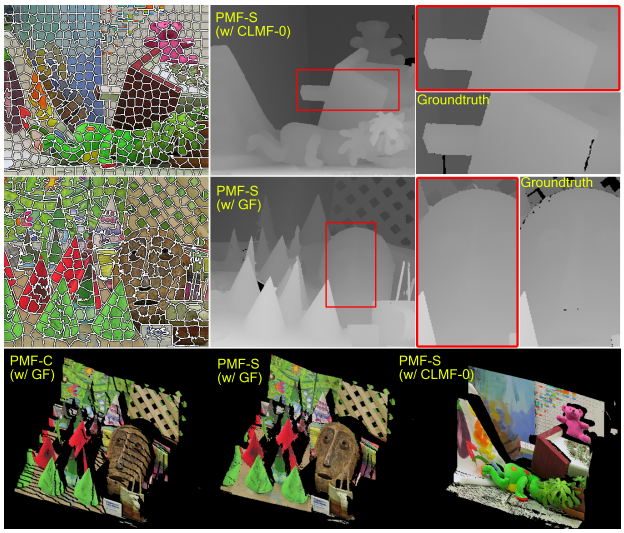

Fig. 7. Visual results. Top row (left to right): Segmented *Teddy* image, PMF-S (w/ CLMF-0) result and close-up comparison. Middle row (left to right): Segmented *Cones* image, PMF-S (w/ GF) result and close-up comparison. Bottom row (left to right): Synthesized novel-view images with PMF-C and PMF-S.

TABLE 4
Quantitative stereo result evaluation (w/o post-processing) on seven Middleburry 2006 datasets with error threshold 0.5.

Dataset	M [9]	PMBP [8]	**PMF-S**	**fPMF-S**
Baby2	18.80	16.85	12.42	**8.94**
Books	31.52	27.58	21.17	**20.31**
Bowling2	15.01	15.10	11.41	**10.86**
Lampshade1	31.67	30.22	**27.46**	28.60
Laundry	31.97	33.90	24.86	**22.44**
Moebius	22.92	25.08	20.35	**18.28**
Reindeer	21.54	21.57	**14.29**	15.18
Mean	24.78	24.33	18.85	**17.80**

global optimization. The performance of PMF-S is also close to that of recent PatchMatch-based stereo methods, i.e., PM-PM [43] and PM-Huber [18]. In particular, our PMF-S methods ranks high in performance on the more complex datasets of *Teddy* and *Cones* among all top Middlebury stereo methods as shown in Table 3.

In terms of runtime speed, Table 2 shows that PMF-S achieves about $50-100$ times speedup over PatchMatch stereo [9] and PMBP [8], when measured on the same CPU. PMF-S is also much faster than other top algorithms [39], [43], [18] which use GPUs for acceleration. For visual examination, Fig. 7 shows the disparity maps estimated by our PMF-S methods, which preserve depth discontinuities while generating spatially smooth disparities with high subpixel accuracy. Compared to the fronto-parallel version i.e., PMF-C, PMF-S reconstructs the slanted surfaces at much higher quality, as shown by the rendered novel views.

Next, we use some Middlebury 2006 stereo datasets to demonstrate the effectiveness of integrating the cross-scale consistency constraint presented in Sect. 4.4, our new strategy called fPMF-S in dealing with large textureless regions. Table 4 shows the numerical comparisons of PatchMatch Stereo (PM) [9], PMBP [8], PMF-S [26], and fPMF-S. The comparisons are done by setting the disparity error threshold to 0.5 and evaluating the results without post-processing. Overall, fPMF-S obtains the lowest average stereo estimation error among all the four methods. Particularly, it shows better performance over PMF-S on the datasets containing large textureless regions. The visual comparisons of two such examples (*Baby2* and *Bowling2*) are shown in Fig. 8. Note that the single-scale, local aggregation-based methods i.e., PM and PMF-S struggle at flat regions on *Baby2*'s book and *Bowling2*'s ball while fPMF-S can overcome this limitation. Our fPMF-S also performs better than the global belief propagation based method PMBP [8]. As we will show later, the computational overhead of fPMF over PMF is very minor.

6.3 Optical Flow Results on the Middlebury Datasets

We first evaluate our PMF-OF methods (with GF filtering) using the Middlebury flow benchmark [1]. In the following tests, we have fixed the motion search range to $[-40, 40]^2 \times 8^2$ (about 410,000 labels) and the number of iterations to 10. Following [34], [26], the raw matching cost is computed as given in Eq. (7). Table 5 lists the average ranks of a few competing methods also based on discrete optimization as well as the top-performing MDP-Flow2 [42] and NN-Field [14] measured in the average endpoint error (AEE). PMF-OF, though simple and free of a large number of parameters, has a very competitive ranking out of over 110 methods. In particular, it outperforms CostFilter [34] (see also Fig. 1), even though image-wise cost filtering has been exhaustively performed for every single label in [34]. This very fact of a label space subsampling method giving better results was also observed and explained from the information representation perspective in [29]. Also, using compact superpixels as the atomic units tends to have better spatial regularization than [34], without compromising the accuracy along motion discontinuities. Table 5 shows that PMF-OF performs quite well for the three challenging scenes with fine details and strong motion discontinuities. In Fig. 9, we compare visually the flow maps estimated by PMF-OF and other competing methods. Our method preserves fine motion details and strong

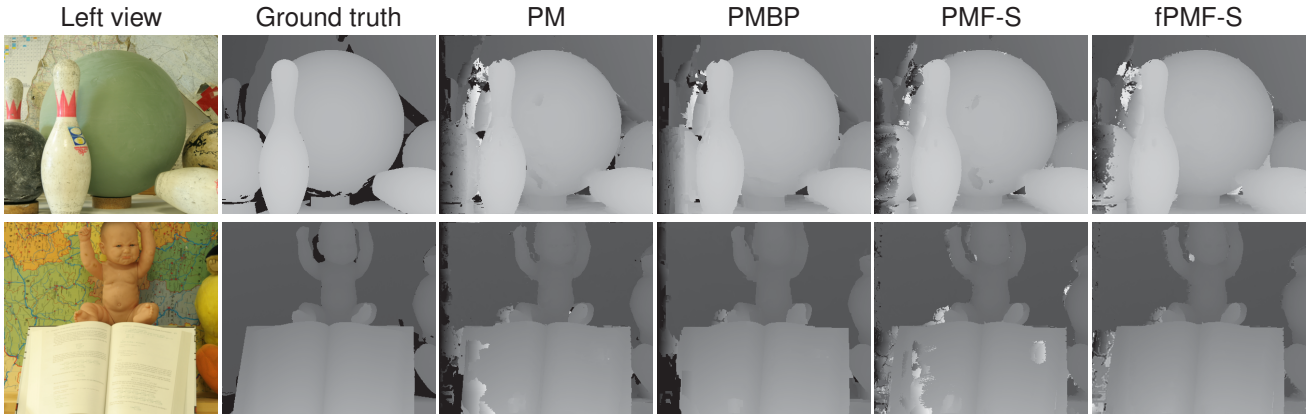

Fig. 8. Visual comparison of the stereo results estimated by PatchMatch Stereo (PM) [9], PMBP [8], PMF-S [26], and fPMF-S for *Bowling2* (top) and *Baby2* (bottom) that contain large textureless regions.

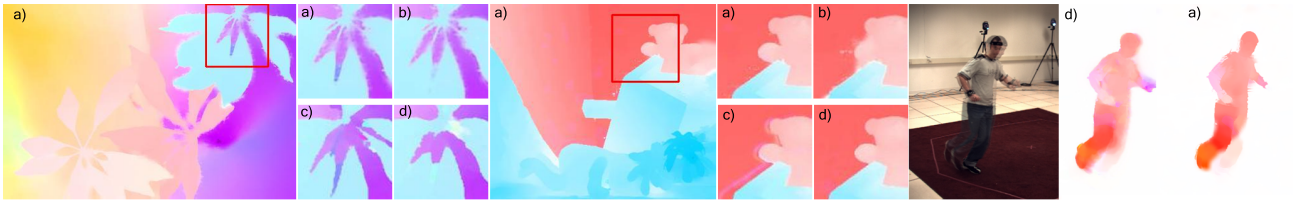

Fig. 9. Results on *Schefflera*, *Teddy* and *HumanEva* by a) PMF-OF b) CostFilter [34] c) DPOF [21] d) MDP-Flow2 [42].

TABLE 5
Middlebury quantitative flow evaluation results measured with average endpoint error (AEE) for three challenging scenes. In brackets are the ranks for (all, disc, untext). Runtime is given for the *Urban* sequence. *use GPU. [captured on 01/08/2015]

Algorithm	μRank	*Schefflera*	*Grove*	*Teddy*	sec
MDP-Flow2 [42]	9.7	(5,5,2)	(19,19,20)	(6,5,6)	342
NN-Field [14]	10.3	(3,4,7)	(1,1,1)	(3,8,1)	362
PMF-OF	34.2	(11,11,14)	(11,11,5)	(7,3,13)	35
EPPM [5]	39.6	(29,34,14)	(19,19,9)	(15,18,18)	2.5*
CostFilter [34]	41.7	(10,10,14)	(13,16,7)	(17,30,15)	55*
DPOF [21]	51.8	(14,12,46)	(25,29,16)	(32,30,9)	287

discontinuities, and handles nonrigid large-displacement flow without changing any parameters. Fig. 10 verifies the strength of our superpixel-induced initialization and search strategies over the baseline approach.

As shown in Table 5, our PMF method has a significant runtime advantage and often gives an order of magnitude speedup over the previous methods. Tested on the same CPU, PMF-OF runs even over 30-times faster than CostFilter [34] on the *Urban* sequence, thanks to slashing the complexity dependency on the huge label space size.

6.4 Optical Flow Results on the MPI Sintel Datasets

Now we focus on evaluating large-displacement optical flow estimation results obtained by the proposed algorithms including PMF-OF, fPMF-OF, and fPMF-OF (with global post-processing) on the MPI Sintel dataset [12], a modern and challenging optical flow evaluation benchmark containing large displacement flow vectors and more complex non-rigid motions. Note that in this

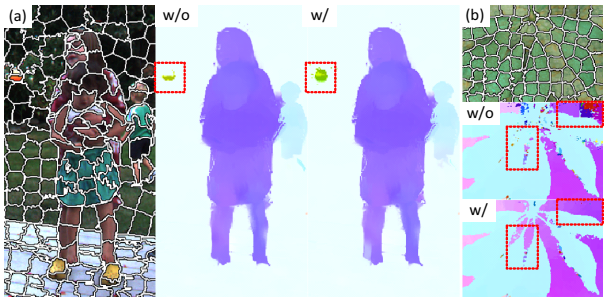

Fig. 10. Advantages of our improved search strategies proposed in Sect. 4.3. a) Better initialization. b) Non-local neighbor propagation (# iteration = 3).

TABLE 6
Evaluation of different PMF-based approaches on the MPI Sintel training dataset. Average end point errors (EPE) are reported. "QO" indicates the quadratic optimization presented in Sect. 5.2 is applied.

Algorithm	PMF-OF	fPMF-OF	fPMF-OF (w/ QO)
Clean pass	3.373	3.094	2.728
Final pass	4.768	4.739	4.210
Runtime (s)	29	37	39

section we compute the raw matching cost by using ADCensus in Eq. (8) in all our methods in Table 6, and we use CLMF-0 [25] for cost aggregation, which is found to provide the optimal accuracy-complexity trade-off on the Sintel's resolution. The prefix 'f' indicates the cross-scale smoothness constraint presented in Sect. 4.4 is used. We fixed the search range of flow vectors to $[-200, 200]^2$. The floating precision of flow vectors was set to $\frac{1}{8}$ for

Fig. 11. Visual and EPE comparison of the optical flow results by PMF-OF, fPMF-OF, and fPMF-OF (w/ QO).

TABLE 7
Optical flow performance on the MPI Sintel Dataset. For those methods without providing public code, we report their time on KITTI. *use GPU. [captured on 12/08/2015]

Method	Clean	Final	Runtime(s)
EpicFlow [33]	4.115	6.285	17
PH-Flow [46]	4.388	7.423	800
DeepFlow [41]	5.377	7.212	19
fPMF-OF	5.378	7.630	39
LocalLayering [36]	5.820	8.043	-
MDP-Flow2 [42]	5.837	8.445	754
EPPM [5]	6.494	8.377	0.95*
S2D-Matching [22]	6.510	7.872	2000
Classic+NLP [37]	6.731	8.291	688
Channel-Flow [35]	7.023	8.835	>10000
LDOF [10]	7.563	9.116	30

both x and y directions. This results in a huge label space with over 10 million labels.

Table 6 shows the comparison of the three PMF-based methods on the Sintel training set. It is clear that our new strategy with the cross-scale constraint (i.e., fPMF-OF) obtains lower optical flow estimation errors than the original single-scale PMF-OF method, incurring only a relatively marginal runtime overhead. Our global optimization based post-processing (i.e., fPMF-OF (w/ QO)) leads to further accuracy improvements. Fig. 11 shows two example cases in the Sintel training set. Compared to PMF-OF and fPMF-OF, fPMF-OF (w/ QO) handles large motion and large occlusions better both visually and quantitatively. Therefore, in the rest of this section, we use fPMF-OF to simply denote our best PMF variant with the quadratic optimization-based post-processing.

Next, we move on to test on the MPI Sintel test dataset. Table 7 shows the quantitative comparison of several published optical flow methods with our fPMF-OF method. Without being specially tailored for this correspondence task, the proposed fPMF-OF achieves a very competitive standing on the MPI Sintel benchmark evaluation. The visual comparison of our fPMF-OF with other popular optical flow methods (using the authors' public source code) is provided in Fig. 12. Our results are visually close to the results of EpicFlow [33], a leading optical flow method on the MPI Sintel benchmark, while others have problems in handling large motions. Note that EpicFlow is a specially designed, multi-pass method for optical flow that involves both dense interpolation and variational energy minimization, while our PMF is based on a general framework for discrete labeling problems. The advantage of fPMF-OF over EPPM [5] is also quite obvious: though EPPM uses a local PatchMatch-like data aggregation with a coarse-to-fine framework, it tends to lose fine-grained motion details and still has difficulties in handling large textureless regions.

7 Conclusions and Future Work

This paper proposed a generic PMF framework of solving discrete multi-labeling problems efficiently. We have particularly demonstrated its effectiveness in estimating smoothly varying yet discontinuity-preserving subpixel-accurate stereo and optical flow maps. Additionally, we proposed a hierarchical matching scheme to extend the PMF approach, which incorporates a cross-scale consistency constraint in a spatially adaptive manner. We justified its effectiveness in handling large textureless regions, while keeping the strength of the original single-scale PMF that effectively captures fine-grained details.

Future work broadly include the following aspects. First, a theoretic study on approximate inference techniques for continuous MRFs either with a local or global optimization approach [8], [23], to best exploit particle sampling and cost aggregation, is very interesting. Second, we plan to apply and optimize the PMF algorithm also for other tasks or datasets, such as the KITTI dataset featuring more structured rigid road scenes. Yamaguchi *et al.* [44] presented a well-designed pipeline specifically for this dataset and achieved excellent results. It will be interesting to evaluate whether MotionSLIC proposed in [44] can be used similarly to initialize our label estimates. In addition, our recent work [45] based on the PMF framework shows some initial success in tackling general scene matching. Lastly, optimizing the PMF algorithm on GPUs or a multi-core CPU for further speedups will be helpful, for which several acceleration possibilities exist [6], [7], [32].

8 Acknowledgments

We thank the Associate Editor and reviewers for constructive suggestions that help improve the paper.

References

[1] http://vision.middlebury.edu/flow/.
[2] http://vision.middlebury.edu/stereo/.
[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. PAMI*, 34(11), 2012.
[4] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proc. of ICCV*, 2015.

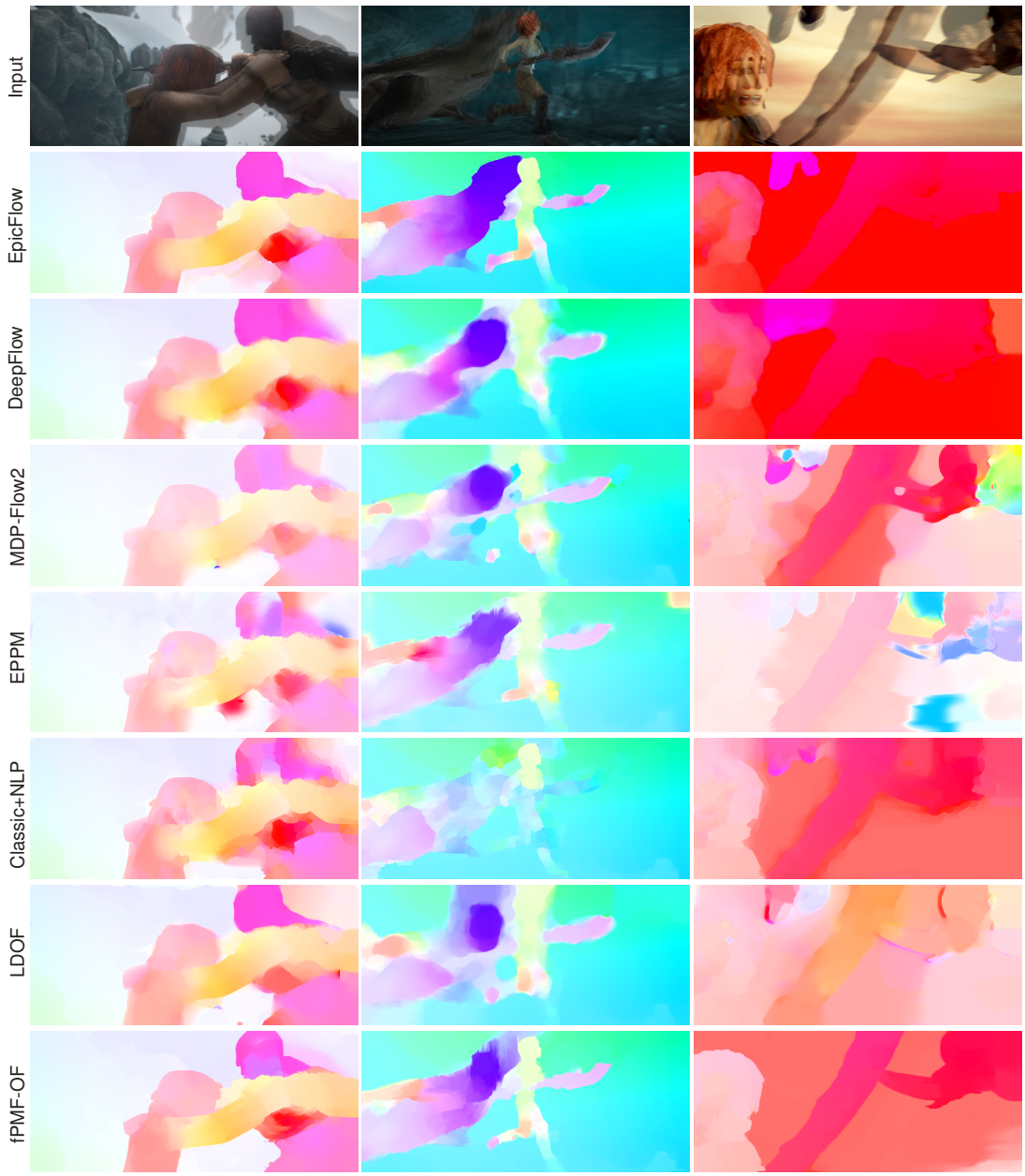

Fig. 12. Visual comparison on the MPI Sintel dataset with EpicFlow [33], DeepFlow [41], MDP-Flow2 [42], EPPM [5], Classic+NLP [37], LDOF [10], and our fPMF-OF.

[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 *Proc. of ACM SIGGRAGH*, 2009.
[7] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *ECCV*, 2010.
[8] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. Pmbp: Patchmatch belief propagation for correspondence field estimation. *IJCV*, 110(1):2–13, 2014.
[9] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - Stereo matching with slanted support windows. In *Proc. of BMVC*, 2011.
[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] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June 1986.
[14] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu. Large displacement optical flow from nearest neighbor fields. In *CVPR*, 2013.
[15] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.
[16] K. He and J. Sun. Computing nearest-neighbor fields via propagation-assisted KD-trees. In *CVPR*, 2012.
[17] K. He, J. Sun, and X. Tang. Guided image filtering. In *Proc. of ECCV*, 2010.
[18] P. Heise, S. Klose, B. Jensen, and A. Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *ICCV*, 2013.
[19] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013.
[20] S. Korman and S. Avidan. Coherency sensitive hashing. In *Proc. of ICCV*, 2011.
[21] C. Lei and Y.-H. Yang. Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In *Proc. of ICCV*, 2009.
[22] M. Leordeanu, A. Zanfir, and C. Sminchisescu. Locally affine sparse-to-dense matching for motion and occlusion estimation. In *ICCV*, 2013.
[23] Y. Li, D. Min, M. S. Brown, M. N. Do, and J. Lu. Spm-bp: Sped-

up patchmatch belief propagation for continuous mrfs. In *ICCV*, 2015.
[24] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5), 2011.
[25] J. Lu, K. Shi, D. Min, L. Lin, and M. N. Do. Cross-based local multipoint filtering. In *CVPR*, 2012.
[26] 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.
[27] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *ICCV Workshop*, 2011.
[28] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do. Fast global image smoothing based on weighted least squares. *IEEE Trans. Image Processing*, 23(12):5638–5653, Dec. 2014.
[29] D. Min, J. Lu, and M. N. Do. A revisit to cost aggregation in stereo matching: How far can we reduce its computational redundancy? In *Proc. of ICCV*, 2011.
[30] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand. Bilateral filtering: Theory and applications. *Foundations and Trends in Comp. Graphics and Vision*, 4(1):1–73, 2008.
[31] 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.
[32] V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer, and S. Bathiche. MonoFusion: Real-time 3d reconstruction of small scenes with a single web camera. In *ISMAR*, 2013.
[33] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.
[34] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, 2011.
[35] L. Sevilla-Lara, D. Sun, E. G. Learned-Miller, and M. J. Black. Optical flow estimation with channel constancy. In *ECCV*, 2014.
[36] D. Sun, C. Liu, and H. Pfister. Local layering for joint motion estimation and occlusion detection. In *CVPR*, 2014.
[37] 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.
[38] 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 with smoothness-based priors. *IEEE TPAMI*, 30(6):1068–1080, 2008.
[39] T. Taniai, Y. Matsushita, and T. Naemura. Graph cut based continuous stereo matching using locally shared labels. In *CVPR*, 2014.
[40] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. of ICCV*, 1998.
[41] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013.
[42] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. PAMI*, 34(9):1744–1757, 2012.
[43] 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.
[44] K. Yamaguchi, D. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. In *CVPR*, 2013.
[45] H. Yang, W.-Y. Lin, and J. Lu. Daisy filter flow: A generalized discrete approach to dense correspondences. In *CVPR*, 2014.
[46] J. Yang and H. Li. Accurate optical flow estimation with piecewise parametric model. In *CVPR*, 2015.
[47] Q. Yang. A non-local cost aggregation method for stereo matching. In *CVPR*, 2012.
[48] K. Yoon and I. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. PAMI*, 2006.
[49] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 2007.

Jiangbo Lu (M'09-SM'15) received the Ph.D. degree in electrical engineering, Katholieke Universiteit Leuven, Leuven, Belgium, in 2009. Since then, he has been working with the Advanced Digital Sciences Center, Singapore, which is a joint research center between the University of Illinois at Urbana-Champaign, Urbana, and the Agency for Science, Technology and Research (A*STAR), Singapore, where he is leading a few research projects as a Senior Research Scientist. His research interests include computer vision, visual computing, image processing, and robotics. He received the 2012 Best Associate Editor Award from IEEE Transactions on Circuits and Systems for Video Technology (TCSVT).

Yu Li received his B.Eng. degree from Beijing University of Posts and Telecommunications in 2011. He is now working at Advanced Digital Sciences Center, Singapore. Meanwhile he is working towards his Ph.D. degree in National University of Singapore. His research interests include computer vision, computational photography, and computer graphics.

Hongsheng Yang received his B.Eng. degree in electronic information engineering from the University of Electronic Science and Technology of China (UESTC) in 2011. He worked in Advanced Digital Sciences Center (ADSC) as a R&D software engineer between 2011-2013. He finished his M.S. degree in computer science in University of North Carolina at Chapel Hill (UNC), and he is currently working for Google, an Alphabet company.

Dongbo Min (M'09-SM'15) received the B.S., M.S., and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, in 2003, 2005, and 2009, respectively. From 2009 to 2010, he was with the Mitsubishi Electric Research Laboratories. From 2010 to 2015, he was with the Advanced Digital Sciences Center, Singapore. Since 2015, he has been an Assistant Professor with the Department of Computer Science and Engineering at Chungnam National University, Daejeon, Korea. His research interests include computer vision, 2D/3D video processing, computational photography, and continuous/discrete optimization.

Weiyong Eng received the B.S. degree in electronics engineering from Multimedia University, Melaka, Malaysia in 2009, and the M.S. degree in vision and robotics from the Heriot-Watt University, Edinburgh, UK, in 2011. She was with the Advanced Digital Sciences Center, Singapore, as a Software Engineer till 2014. She is currently pursuing the Ph.D. degree in computer vision with the Multimedia University, Malaysia. Her current research interests include 3D computer vision, image and video processing.

Minh N. Do (M'01-SM'07-F'14) received the B.Eng. degree in computer engineering from the University of Canberra, Australia, in 1997, and the Dr.Sci. degree in communication systems from the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland, in 2001. Since 2002, he has been on the faculty at the University of Illinois at Urbana-Champaign (UIUC), where he is currently a Professor in the Department of Electrical and Computer Engineering. His research interests include image and multidimensional signal processing, wavelets and multiscale geometric analysis, computational imaging, and visual information representation. He was an Associate Editor of the IEEE Transactions on Image Processing.