Multimed Tools Appl (2018) 77:12469–12491
DOI 10.1007/s11042-017-4897-1

Efficiency-enhanced cost-volume filtering featuring coarse-to-fine strategy

Ryosuke Furuta[1] · **Satoshi Ikehata**[2] · **Toshihiko Yamaskai**[1] · **Kiyoharu Aizawa**[1]

Received: 21 October 2016 / Revised: 20 May 2017 / Accepted: 1 June 2017 /
Published online: 16 June 2017

Abstract Cost-volume filtering (CVF) is one of the most widely used techniques for solving general multi-labeling problems based on a Markov random field (MRF). However it is inefficient when the label space size (i.e., the number of labels) is large. This paper presents a coarse-to-fine strategy for cost-volume filtering that efficiently and accurately addresses multi-labeling problems with a large label space size. Based on the observation that true labels at the same coordinates in images of different scales are highly correlated, we truncate unimportant labels for cost-volume filtering by leveraging the labeling output of lower scales. Experimental results show that our algorithm achieves much higher efficiency than the original CVF method while maintaining a comparable level of accuracy. Although we performed experiments that deal with only stereo matching and optical flow estimation, the proposed method can be employed in many other applications because of the applicability of CVF to general discrete pixel-labeling problems based on an MRF.

Keywords Cost-volume filtering · Markov random field · Multi-labeling problems · Coarse-to-fine

✉ Ryosuke Furuta
furuta@hal.t.u-tokyo.ac.jp

Satoshi Ikehata
sikehata@seas.wustl.edu

Toshihiko Yamaskai
yamasaki@hal.t.u-tokyo.ac.jp

Kiyoharu Aizawa
aizawa@hal.t.u-tokyo.ac.jp

[1] Department of Information Communication and Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, 113-8656, Tokyo, Japan

[2] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan

1 Introduction

Many low-level computer-vision problems (e.g., stereo matching and optical flow estimation) are formulated as multi-labeling problems, where discrete labels (e.g., disparity and motion vector) are assigned to pixels. In general, there are two approaches to solve these problems: global and local. The former models a labeling problem as a Markov random field (MRF), where global optimization techniques [7, 8, 14, 19, 30, 32, 35, 37] are used to minimize the energy function. Although such an approach is effective, using it to solve a large optimization problem makes the inference intractable when the image size or label space is large. Rhemann et al. [28] presented a local approach called *cost-volume filtering* (CVF), which efficiently solves general multi-labeling problems by performing MRF optimization via fast local filtering of label costs instead of global smoothing. CVF is easy to implement and provides high-quality results; therefore, it has been widely used to solve various multi-labeling problems [10, 12, 18, 20, 40]. However, a limitation of CVF is that it does not scale to extremely large label sets (e.g., sub-pixel stereo matching and up-sampling of 16-bit depth maps captured by a Kinect sensor).

To overcome this limitation, Lu et al. [22] proposed the PatchMatch filter (PMF), which performs CVF iteratively on local superpixels with compact label subsets instead of performing it on the entire image coordinate space. In general, the average size of local label subsets is much smaller than the size of the entire label space; therefore, although PMF and CVF provide similar levels of accuracy, the efficiency of PMF is considerably higher. Nevertheless, PMF relies on global optimization based on the complex PatchMatch approach [3, 6] to estimate a label subset for each superpixel. Thus, the computational complexity of PMF increases with the number of superpixels, and therefore, PMF becomes less effective when an image is divided into many superpixels.

This paper presents an alternative coarse-to-fine strategy for efficiently estimating compact label subsets to solve the label space problem in cost-volume filtering. Based on the observation that true labels at the same coordinates in an image of different scales are highly correlated, we propose that lower-scale labeling outputs be leveraged for estimating higher-scale local label subsets. Starting with an image of very low-resolution, we iteratively truncate unimportant labels at each higher scale, and finally, we assign compact and approximately optimal label subsets to local regions of the original scale. The advantage of the proposed framework is a simple and efficient coarse-to-fine strategy, which does not require any global optimization as in [22]; moreover, its computational complexity is not significantly affected by the number of local regions. Extensive experiments described in Section 4 show that our algorithm achieves higher efficiency than PMF and CVF while providing a comparable or often superior level of accuracy.

The fundamental algorithm of our method and the experimental results of stereo matching have already been presented in our preliminary study [15]. In this paper, we provide detailed explanations and present the results of additional experiments for optical flow estimation. Note that we are not proposing a better algorithm for stereo matching and optical flow estimation, but proposing a coarse-to-fine method to drastically reduce the computational time of CVF while preserving its accuracy. As presented in [10, 12, 18, 20, 28], the CVF can be used for wide range of applications and the stereo matching and the optical flow estimation presented in this paper is just an example.

Our proposed algorithm can be directly applied to not only original CVF [28] but also several of its variants picked up in Section 2. In addition, our proposed algorithm can be implemented on GPU similar to the original CVF. However, in this paper, we did not perform those implementations, and compared with only the original CVF because we focus on “how to deal with the large label space efficiently”, not to improve the accuracy and not the real-time application.

The reminder of this paper is organized as follows. Section 2 reviews related studies. Section 3 briefly reviews CVF [28] and describes the details of the proposed coarse-to-fine strategy. Section 4 presents the experimental results and describes their evaluation using the Middlebury benchmark [24, 25]. Finally, Section 5 summarizes our findings and concludes the paper.

2 Related work

In this section, we mainly focus on related works about stereo matching and optical flow estimation, because they are main problems among multi-labeling problems and a lot of methods using cost-volume filtering techniques have been proposed in stereo matching and optical flow estimation. However, as mentioned in Section 1, the cost-volume filtering technique is not only used for them but also applied to wide range of multi-labeling problems such as image segmentation [20], and depth-map up-sampling [12].

2.1 Cost aggregation methods for labeling problems

First, we review cost aggregation methods for correspondence field estimation. Yoon and Kweon [44, 45] proposed a cost aggregation method using an adaptive weighted window such as an edge-preserving bilateral filter [36]. This method is slow because it needs to perform naive bilateral filtering iteratively, where the number of iterations is equal to the number of disparity candidates. To address this problem, Richard et al. [29] proposed an approximate bilateral filtering technique that reduces the computational complexity of adaptive support weight calculation. However, this approach provides low-quality results, as compared to state-of-the-art stereo matching methods. On the other hand, Yang [40, 41] proposed a tree-based non-local cost aggregation method using a minimum spanning tree. This method aggregates the cost values based on a tree structure constructed using input images, and the final disparity refinement process is also performed on the basis of the tree structure. Bai et al. [2] proposed an algorithm based on loop-erased random walk to improve the support weighted window of [40] near depth discontinuities. As stated in Section 1, Rhemann et al. [28] proposed CVF for general multi-labeling problems. By using an $O(1)$ edge-preserving filter called a guided filter (GF) [16] for cost aggregation, CVF can efficiently solve general multi-labeling problems and achieve high-quality results. Lu et al. [21] proposed a new edge-preserving filter called a cross-based local multipoint filter (CLMF), which is an extension of the GF. Although the shape of the local support region of the GF is a square, that of the CLMF can be an adaptively derived from a reference image. Further, Lu et al. [21] showed that higher-quality stereo matching results can be achieved by applying the CLMF instead of the GF for cost aggregation. Zhang et al. [49] proposed a cross-scale cost aggregation algorithm based on CVF [28] for stereo matching. They showed that higher-quality disparity maps can be obtained by adding a regularization term between the cost values of different scales, and that the computational time of cross-scale aggregation is not significantly greater than that of the original CVF [28]. This method [49] is similar to ours in terms of multi-scale cost-volume utilization, but its purpose is to improve the quality of the disparity maps, not to reduce the computational complexity. Recently, Zhan et al. [48] proposed some techniques for local stereo matching methods to improve the accuracy: mask filtering as a pre-processing, an improved matching cost function, and multi-step disparity refinement as a post-processing. Inspired by the great success of convolutional neural networks (CNNs) in image recognition task, CNNs are recently used for computing the label costs (matching costs in stereo matching and optical flow estimation) instead of hand-crafted cost functions [11, 13, 23, 46, 47], which has led to significant improvement in terms of accuracy. In MC-CNN [46, 47], the CNN directly outputs the matching cost of two input patches. Cross-based cost aggregation and semi-global matching are preformed for the obtained cost-volume to produce accurate disparity map. To speed up computing the matching cost, Chen et al. [11] and Luo et al. [23] proposed similar ideas, where the matching cost is defined as the inner product of two features from CNN. In FlowNet [13], the matching costs are defined as the correlation between two patches of feature maps, and the final flow map is obtained by upconvolution operation. The computation of the correlations is implemented as correlation layer, which is incorporated into CNN.

Most of local methods perform cost aggregation for all the labels (disparities) at every pixel. Therefore, those methods are limited in that they do not scale to extremely large label sets. To overcome this problem, with regard to stereo matching, Min et al. [26, 27] proposed a technique to estimate a compact disparity subset for every pixel by considering disparities with the local minima of the pre-filtered cost values. Although this method efficiently achieves high-quality results with the Middlebury stereo benchmark [25], it cannot be applied to general multi-labeling problems directly. Wang et al. [38] adapted the sequential probability ratio test to reduce the disparity search range with the sufficient confidence in stereo matching problem. Helala and Qureshi [17] proposed the Accelerated CVF using an occulusion handling technique for stereo matching problem. For general multi-labeling problems, Lu et al. [22] proposed PMF, which is based on CVF [28]. As mentioned in Section 1, PMF estimates a compact label subset for every superpixel using the PatchMatch [3, 6] strategy; therefore, it is usually much more efficient than CVF while maintaining a similar level of accuracy. However, because PMF relies on complex PatchMatch-based global optimization to estimate a label subset for each superpixel, it becomes less effective when an image is divided into many superpixels.

2.2 Coarse-to-fine strategy

Coarse-to-fine strategies have been employed in a variety of methods for labeling problems such as stereo matching and optical flow estimation. We can classify them into two types: the coarse-to-fine strategies where the cost aggregation results from all resolution are merged in order to obtain more accurate results such as [33, 42, 49], and ones where the results of lower resolution are propagated to higher resolution in order to reduce the search range of labels such as [43, 50]. We focus on the latter because our method is classified into latter group.

Brox et al. [9] employed a coarse-to-fine strategy in their global optimization framework to estimate a optical flow field. They obtain an output flow field as the solution of their energy minimization formulation by solving Euler-Lagrange equations. They supplied a theoretical explanation that justifies their coarse-to-fine strategy by regarding it as a part of the two nested iterations for non-convex optimization, and argued that their coarse-to-fine strategy helps the convergence to the global minimum by setting the solution of coarser scale to the initialization of the next finer scale. Similar to [9], Wedal et al. [39] employed a coarse-to-fine strategy in their optical flow estimation framework, where the flow field is obtained by solving the total variation (L1 norm) minimization problem using linear approximation and alternating optimization scheme. They argued that their coarse-to-fine strategy has the advantage of avoiding poor local minimum by propagating the solution of coarser scale to the finer scale. Those coarse-to-fine strategies such as [9, 39] are tailored for global optimization techniques. These method iteratively update one solution for the entire image and propagate it to the next scale after the predetermined number of iterations. Therefore, their coarse-to-fine approaches cannot be used for CVF which needs pixel-wise cost computation for all possible labels and obtains pixel-wise solutions by winner-take-all strategy. Yang et al. [43] proposed a coarse-to-fine technique for belief propagation (BP), which reduces the computational complexity in both spatial and depth domain. This method is tailored for BP and cannot be directly applied to CVF. Different from these approaches, we propose a coarse-to-fine strategy for the cost-volume filtering technique that is categorized in local methods.

Next, we discuss the coarse-to-fine strategies employed in local cost aggregation methods which are close to our method. Zhao et al. [50] employed a coarse-to-fine strategy in their elegant implementation on GPGPU for real-time stereo. They limit the search range within ±2 pixels of the disparity value obtained in lower resolution. The main difference between their method and ours is that the reduction of the search range is performed per pixel in their method, while it is done in each local region in our method. In addition, the comparison with their method has little meaning because their objective is the efficient disparity estimation in only foreground region and their algorithm is optimized for it. Their experimental results on Middlebury stereo datasets with the assumption that whole image area is foreground show the poor accuracy especially around the object boundaries (Disc. in Table 1 [50]). Tao et al. [34] proposed a multiscale local cost aggregation method for optical flow estimation called SimpleFlow. They upsampled the flow field obtained at the coarser scale and skipped the cost computation by interpolating the flow using simple bilinear interpolation in the regions where the flow was smooth. Therefore, their method can obtain a flow field with sublinear time with respect to the size of input images. Thier coarse-to-fine strategy is different from ours because our method estimates a compact label set in each local region to handle the large label space. In addition, without the refinement using the global optimization [31], the accuracy of the flow fields obtained by SimpleFlow [34] is much lower than that of CVF [28]. Although the SimpleFlow with the refinement can obtain the comparable accuracy to the CVF, the running time drastically increases because the global optimization in the refinement process is computationally expensive (Table 4 in [4]). On the other hand, our method can obtain comparable accuracy to the CVF [28] and is several times faster than CVF. Bao et al. [4, 5] proposed a fast edge-preserving PatchMatch for optical flow estimation. Their method estimates an approximate nearest neighbor field (NNF) using PatchMatch search at the coarsest scale, and repeats upsampling the NNF and the refinement of it within a small search range (3 × 3 pixels) until the original resolution. Their method is very fast and can achieve high-quality results for large displacement optical flow. However, for the datasets with small displacement optical flow, their coarse-to-fine strategy obtains the less accurate results than when without it (Table 4 [4]) because their method is tailored for large displacement optical flow. In contrast, our coarse-to-fine strategy for general multi-labeling problems obtains the comparable or more accurate results than the original CVF both when the label space is small and large.

3 Coarse-to-fine strategy for efficient CVF

In this section, we present a coarse-to-fine strategy for CVF [28] in order to address multi-labeling problems with a large label space. Given a label set $\mathcal{L} = \{l_0, \cdots, l_{L-1}\}$, the objective of a multi-labeling problem is to assign a label $l_i \in \mathcal{L}$ to each pixel

$i \triangleq [x_i,\ y_i]$ $(i = 0, \ldots, M-1)$ in the image coordinate space I such that it minimizes the label costs encoded in the energy function [28]. Here, L and M denote the number of labels and the number of pixels, respectively.

3.1 CVF

The outline of CVF [28] is shown in Fig. 1. CVF solves a multi-labeling problem in three steps. First, a 3-D cost volume C is constructed as a collection of costs $C(i, l)$ for selecting label l at each pixel i on the basis of the data term in the energy function. Then, each slice of the cost volume is independently filtered by an edge-preserving filter [16, 21], which is substituted for the smoothness term in the energy function:

$$C(i, l) \leftarrow \sum_{i' \in \omega_i} W_{ii'} C(i', l), \tag{1}$$

where ω_i is the squared window centered at the pixel i. Finally, the label at pixel i is simply selected by the winner-takes-all (WTA) strategy:

$$l_i = \arg\min_{l \in \mathcal{L}} C(i, l). \tag{2}$$

When an $O(1)$ edge-preserving filter (e.g., guided filter [16]) is used, the computational complexity of filtering an entire cost volume is $O(ML)$; thus, it is difficult to handle an extremely large label space.

One possible strategy for handling a large label space is to locally change the label space in order to reduce its size. Because the true label configuration is generally smooth in space (e.g., disparities are smooth except for object boundaries), the label space required for performing CVF on a local region should be smaller than the entire label space. As an example, we present a colored true disparity map of *cones* (see Fig. 2) that is divided into local regions by regular rectangular grids. In addition, we show a histogram of the true disparities l in the entire image and the ones in the local regions S_i^0 and S_j^0. We observe that the types of true labels in a local region are fewer than those in the entire label space.

However, the problem is of course that we do not know *a priori* which labels are important for each local region, and thus, the estimation of local label subsets is required [22].

3.2 Problem statement

Here, we present a simple but efficient label subset estimation algorithm. Unlike Lu et al. [22], we do not rely on global optimization for estimating local label subsets; instead,

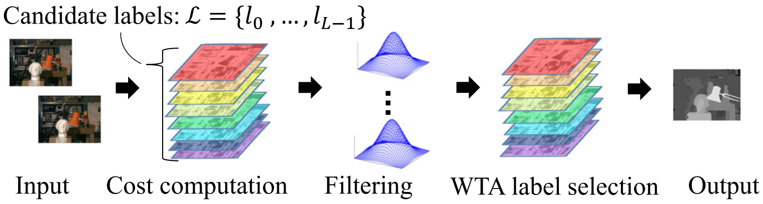


Fig. 1 Framework of CVF [28]

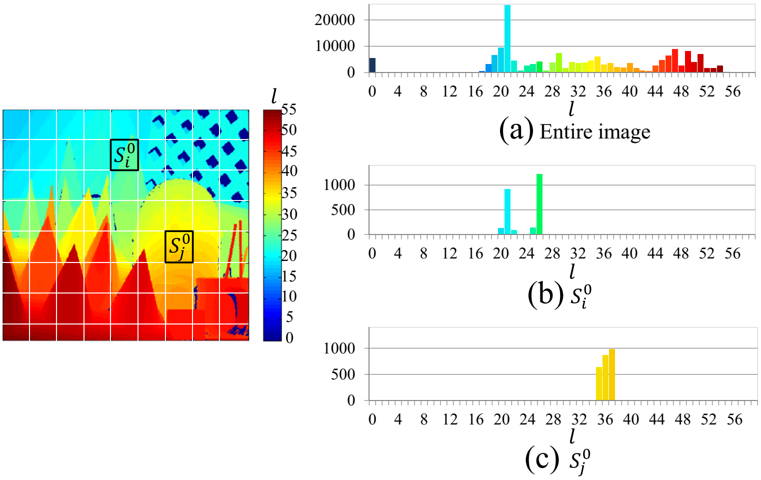


Fig. 2 Colored true disparity map of *cones*, and a histogram of the true disparities l in the entire image and the ones in local regions S_i^0 and S_j^0. The disparities l are rounded off to integer values

we leverage the coarse-to-fine framework. An overview of the proposed method is shown in Fig. 3. Our algorithm mainly involves two steps (i) in-scale cost-volume filtering and (ii) across-scale label propagation. The latter is an essential feature of our approach, whereby a local label subset is estimated from the CVF output at a low-resolution. Because the computational cost of CVF for a low-resolution image is negligibly small, we perform CVF using a large label space with a low-resolution and truncate unimportant labels using the output.

Let $I^k (k = 0, \ldots, n-1)$ denote a cascade of images of decreasing resolution ranging from the original scale (i.e., $I^{k+1} = I^k_{\downarrow s}$, where $\downarrow$ is a down-scaling operator with a scale

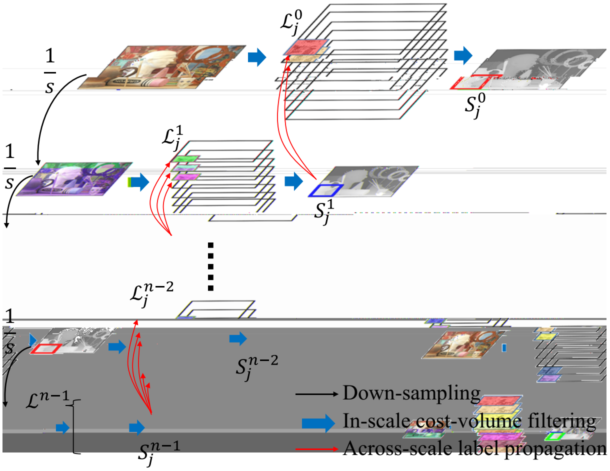


Fig. 3 Framework of proposed method

factor $s \in (1, \infty)$),[1] and let $\mathcal{L}^k$ denote the set of all possible labels at the k-th scale. Then, we divide I^0 $(= I)$ into m non-overlapping local regions S_j^0 and partition $I^k (k \geq 1)$ into local regions $S_j^k (j = 0, \dots, m-1)$ such that $S_j^{k+1} = S_{j\,\downarrow s}^k$. In addition, we represent a label subset for S_j^k as $\mathcal{L}_j^k$ and its size as L_j^k. The total computational complexity of CVF from the lowest scale ($k = n-1$) to the original scale ($k = 0$) is expressed as

$$O\left(\sum_{k=0}^{n-1}\sum_{j=0}^{m-1} M_j^k L_j^k\right), \tag{3}$$

where M_j^k is the number of pixels in S_j^k (i.e., $M_j^k = s^{-2k} M_j^0$). Therefore, our objective is to estimate compact label subsets $\mathcal{L}_j^k$ such that $\sum_{k=0}^{n-1}\sum_{j=0}^{m-1} M_j^k L_j^k \ll ML$ while maintaining the accuracy of CVF. The optimal m and n values will be discussed in Section 4.

3.3 Across-scale label propagation

In this section, we present an algorithm for estimating compact label subsets ($\mathcal{L}_j^k$) that sufficiently reduce the computational cost in (3) without truncating important labels. Our algorithm begins with the coarsest scale (i.e., $k = n-1$). At this scale, we set $\forall j \ \mathcal{L}_j^{n-1} \leftarrow \mathcal{L}^{n-1}$ and simply perform CVF [28] to acquire the filtered cost volume C^{n-1} at the $(n-1)$-th scale. Note that although we use a complete label set, the computational complexity of CVF at this scale is $O(s^{-2(n-1)} ML)$, which is generally negligible (e.g., if we set s to 2 and n to 4, $O(s^{-2(n-1)} ML) \approx O(10^{-2} \times ML)$). Then, we initialize the label subset at the higher resolution ($\tilde{\mathcal{L}}_j^{n-2}$) by merging labels having the smallest cost values in C^{n-1} at the corresponding local regions S_j^{n-1}. Strictly speaking, the initialization is expressed as

$$\tilde{\mathcal{L}}_j^{n-2} = \bigcup_{i \in S_j^{n-1}} f(l_i), \quad l_i = \arg\min_l C^{n-1}(i, l), \tag{4}$$

where $C^{n-1}(p, q)$ is the value of the cost volume at the $(n-1)$-th scale with regard to the position p and the label q, and f is a *projection function* that normalizes the label space if required. In general, the projection function is represented as a constant scale factor giving $f = s$. For instance, a disparity l at the k-th scale corresponds to sl at the $(k-1)$-th scale in the stereo matching problem.[2] The initialization method based on across-scale label propagation is motivated by a reasonable observation that true labels at the same coordinates in images of different scales are highly correlated; in particular, they are very close when the difference in scales is small.

Although the initial estimation $\tilde{\mathcal{L}}_j^{n-2}$ is a good approximation of the optimal label subset $\mathcal{L}_j^{n-2}$, the problem is that $\tilde{\mathcal{L}}_j^{n-2}$ does not consist of labels that are not included in $f(\mathcal{L}_j^{n-1})$, which results in aliasing artifacts when the intermediate labels of $\mathcal{L}_j^{n-1}$ should be included

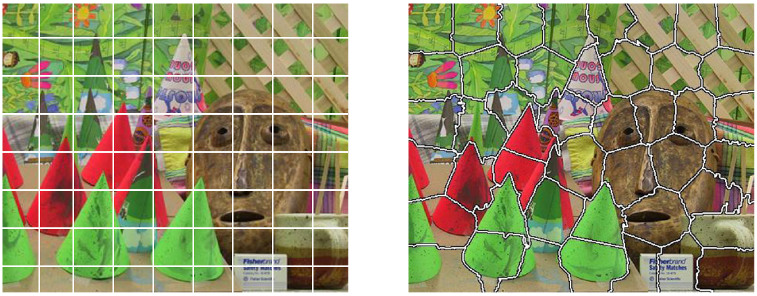
[1] We used the “buildPyramid” function in OpenCV to down-sample images.

[2] In some cases, the label space does not need to be normalized because the scale of a label does not depend on the image coordinate space. Examples include depth-map up-sampling [12] and image segmentation [20].

in $\mathcal{L}_j^{n-2}$ (artifacts become more problematic as the scale difference increases). In addition, the filtered cost volume C^{n-1} often contains numerical errors due to occlusion boundaries or insufficient energy modeling. We adopt two strategies to overcome these difficulties. First, we down-sample images with a relatively small scale factor (e.g., $s \leq 2$), such that the scale difference between two layers becomes sufficiently small. Second, we complete the initial label subset by adding the supporting labels within $\pm s/2$. Note that our algorithm supports floating labels (e.g., sub-pixel disparity values). For instance, if the scale factor is 2 and the disparity unit is 0.5, the initial estimation $\tilde{\mathcal{L}}_j^{n-2} = \{2, 5\}$ is extended as $\mathcal{L}_j^{n-2} = \{1, 1.5, 2, 2.5, 3, 4, 4.5, 5, 5.5, 6\}$. Once a compact label subset $\mathcal{L}_j^{n-2}$ has been constructed, the target layer is shifted to the higher scale (i.e., $k \leftarrow n-2$). Similarly to the case of the coarsest scale, CVF is performed on S_j^{n-2} with regard to $\mathcal{L}_j^{n-2}$. Cost-volume filtering with respect to $\mathcal{L}_j^k$ and the estimation of $\mathcal{L}_j^{k-1}$ from C^k are iterated $n-1$ times until $\mathcal{L}_j^0$ is obtained. Then, the final label at each pixel in S_j^0 is selected by a simple WTA strategy, as in the case of CVF [28].

For the entirety of the coarse-to-fine process, we fix the radius of the edge-preserving filter to smooth the cost-volumes; in other words, the radius is not changed when the target scale is shifted to a higher scale. Therefore, the lower the scale, the more strongly is the cost-volume smoothed. Thus, incorrect labels that accidentally have low costs are truncated during our coarse-to-fine process. In the original CVF [28], especially near object boundaries, the low costs of such incorrect labels are sometimes not sufficiently smoothed, and these incorrect labels are selected by the WTA strategy. Therefore, in such cases, our coarse-to-fine strategy sometimes increases the accuracy of the output at the finest scale, as compared to the original CVF. The results will be presented in Section 4.1.2.

It is possible to generate S_j^0 in various ways, e.g., using regular rectangular grids or superpixels [1], as shown in Fig. 4. The former is simple and suitable for edge-preserving filters using integral images, e.g., a guided filter [16]. In contrast, when S_j^0 are generated by superpixels, some additional computational time is required because we need to apply the edge-preserving filter to the bounding-box containing each region, as in the case of [22]. However, in such cases, it is easier to estimate the local label subsets because the local regions based on the superpixels are less likely to cross object boundaries than regular grids.



(a) Rectangular regular grids

(b) SLIC super-pixels [3]

Fig. 4 Examples of local regions S_j^0

For these reasons, we use both regular rectangular grids and superpixels for generating local regions S_j^0, as described in Section 4.1.2.

The proposed algorithm is summarized as Algorithm 1.

Algorithm 1 The proposed coarse-to-fine strategy

```
INPUT: image pyramid $I^k$ $(k = 0, \cdots, n-1)$
OUTPUT: labeling at the original scale $(k = 0)$.
  set $k \leftarrow n-1$ and $\mathcal{L}_*^{n-1} \leftarrow \mathcal{L}^{n-1}$ # start from the coarsest scale.
  while scales $k \geq 0$ do
      divide the image $I^k$ into $m$ local regions $S_j^k$
      for regions $j = 0$ to $m-1$ do
          for all $i \in S_j^k$ and $l \in \mathcal{L}_j^k$ do
              compute the cost value $C^k(i, l)$.
          end for
          for all $i \in S_j^k$ and $l \in \mathcal{L}_j^k$ do
              $C^k(i, l) \leftarrow \sum_{i' \in \omega_i} W_{ii'} C^k(i', l)$ # filter the cost volume.
          end for
          $\tilde{\mathcal{L}}_j^{k-1} = \bigcup_{i \in S_j^k} f(l_i), \quad l_i = \arg\min_{l \in \mathcal{L}_j^k} C^k(i, l)$ # across scale label propagation.
          $\mathcal{L}_j^{k-1} = \tilde{\mathcal{L}}_j^{k-1} + supporting\ labels$
      end for
      $k \leftarrow k-1$ # move to the next higher scale.
  end while

  # at the original scale $(k = 0)$
  for regions $j = 0$ to $m-1$ do
      for all $i \in S_j^k$ do
          $l_i = \arg\min_{l \in \mathcal{L}_j^0} C^0(i, l)$ # get the final labeling.
      end for
  end for
```

4 Results

In this paper, we demonstrate the validity of our coarse-to-fine approach for CVF by applying it to stereo matching and optical flow estimation. Important to note that our technical contribution is the computational efficiency as compared to the original CVF algorithm, not the accuracy improvement. Besides, the application of CVF is not limited to stereo matching and optical flow estimation.

4.1 Middlebury stereo

Experiments were conducted to evaluate the performance of our proposed method using the Middlebury stereo matching benchmark [25]. In stereo matching, the label l corresponds to

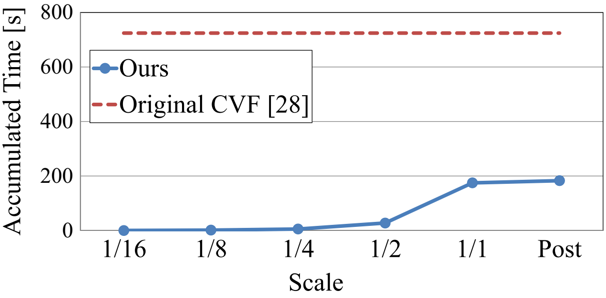


Fig. 5 Evaluation of the computational time. The results of eight Middlebury stereo datasets are averaged. *Post* indicates the total computational time after weighted median filtering for the final disparity-map refinement

the integer disparity between a pixel i in the target image I and its equivalent in the reference image I' shifted by the disparity. In the same manner, the cost function is selected as [28]:

$$\begin{aligned} C(i,l) = {} & (1-\alpha)\min\left[\|I'_{i+l} - I_i\|, \tau_1\right] \\ & + \alpha \min\left[\|\nabla_x I'_{i+l} - \nabla_x I_i\|, \tau_2\right], \end{aligned} \tag{5}$$

where ∇_x is the gradient in the x direction. The model parameters α, τ_1, and τ_2 are set to 0.89, 0.0027, and 0.0078, respectively.[3] We divide eight test image pairs of the Middlebury stereo datasets [25] into two categories according to their size: *small* and *large*. The *small* category includes *cones* (450 × 375), *teddy* (450 × 375), *tsukuba* (384 × 288), and *venus* (434 × 383). Further, the *large* category includes *art* (1390 × 1110), *books* (1390 × 1110), *moebius* (1390 × 1110), and *reindeer* (1342 × 1110). The label space size L is set to 60 for *small* datasets and 240 for *large* datasets. All the experiments were performed using an Intel Core i7-2600 (3.4GHz, single thread) machine with 16 GB of RAM, and they were implemented in C++. As in the original study of CVF [28], we use the guided filter [16] to smooth the cost volume (the radius of the filter is fixed at 9).

4.1.1 Evaluation of label selection

We begin by evaluating the efficiency of our coarse-to-fine strategy, as compared to that of CVF [28]. Here, we apply our method ($n = 5, s = 2, m = 30$) and CVF [28] to both *small* and *large* datasets; the results are averaged as shown in Fig. 5. We observe that overall, our coarse-to-fine strategy takes much less time than CVF [28]. As expected, the computational time for small scales (e.g., 1/16, 1/8, 1/4×) is negligible as compared to that for the original resolution (1/1×).

Further, we present the average size of local label subsets estimated in our coarse-to-fine process, as compared to the size of the entire label space (see Fig. 6). We observe that although the latter increases exponentially with the scale, there is no significant increase in the former, which is much smaller than the latter in the original scale. As a result, our method is much more efficient than CVF [28].

[3] Parameters have been provided by the authors of [28].

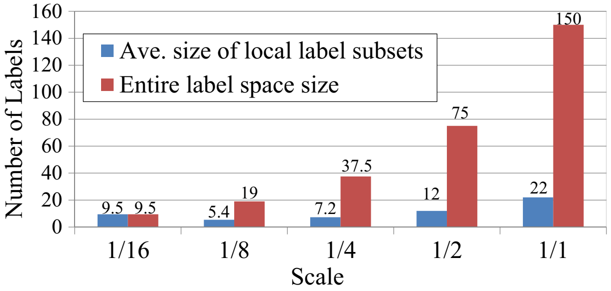


Fig. 6 Evaluation of label set size at each scale. The results of eight Middlebury stereo datasets are averaged

However, an important question arises, which directly addresses the accuracy of the final label selection: “Are the estimated label subsets of the original scale really correct?” To answer this question, we define two metrics for measuring the correctness of the final label subset:

$$P(j) = \frac{|\mathcal{L}_j^0 \cap \mathcal{L}_j|}{|\mathcal{L}_j^0|}, \quad R(j) = \frac{|\mathcal{L}_j^0 \cap \mathcal{L}_j|}{|\mathcal{L}_j|}, \tag{6}$$

where $\mathcal{L}_j$ is the subset of ground truth labels at the original scale (i.e., a collection of ground truth disparity values that emerge in the j-th region), and we recall that $\mathcal{L}_j^0$ is the subset of estimated labels at the original scale. These two metrics evaluate the estimated label subset in two different aspects: $P(j) \in [0, 1]$ measures the *precision* of $\mathcal{L}_j^0$, which implies how correctly unimportant labels are removed, and $R(j) \in [0, 1]$ measures the *recall* of $\mathcal{L}_j^0$, which implies how correctly important labels are maintained. Note that the ideal situation of course occurs when $\forall j \; \mathcal{L}_j^0 = \mathcal{L}_j$. For $\mathcal{L}_j$, we used the ground truth of the disparity maps precomposed in the Middlebury stereo datasets [25].

Using these metrics, we evaluate our method with a varying scale factor s and number of layers n using only *small* datasets, as shown in Tables 1 and 2. Here, the results are averaged over all the datasets in this category.

Table 1 shows the evaluation of the label subset estimation with a fixed lowest scale and varying scale differences. We observe that when the scale difference between two layers is small (down-scale factor $s = 2$), our algorithm successfully maintains around 90% of ground truth labels and truncates more than 50% of unnecessary labels, on average, whereas the original label subset contains 90% of unnecessary labels. When the scale difference is large ($s = 16$), our method maintains more than 70% of unnecessary labels, on average. Therefore, we select a small down-scale factor ($s = 2$) in the following.

Next, Table 2 shows the case of a fixed scale difference and varying number of layers. We observe that when the number of layers n is set to 4, the performance of our method is optimal, considering both the precision and the recall. In such cases, our algorithm maintains

Table 1 Evaluation of label subset estimation with fixed lowest scale and varying scale differences

Transition of scale	Ave. Precision	Ave. Recall
1/16→1/8→1/4→1/2→1/1 (s=2, n=5)	0.58	0.89
1/16→1/4→1/1 (s=4, n=3)	0.48	0.89
1/16→1/1 (s=16, n=2)	0.23	0.93
1/1 (CVF[28])	0.13	1.00

Table 2 Evaluation of label subset estimation with fixed scale difference and varying number of layers

Transition of scale	Ave. Precision	Ave. Recall
1/16→1/8→1/4→1/2→1/1 (s=2, n=5)	0.58	0.89
1/8→1/4→1/2→1/1 (s=2, n=4)	0.58	0.91
1/4→1/2→1/1 (s=2, n=3)	0.57	0.90
1/2→1/1 (s=2, n=2)	0.49	0.92
1/1 (CVF[28])	0.13	1.00

more than 90% of ground truth labels and truncates more than 50% of unnecessary labels, on average. Further, we observe that when the number of layers is small ($n = 2$), the precision is low (less than 50%).

In summary, our observations are in good agreement with our experiments: the improvement in precision is generally limited when the number of layers is too small or the scale difference between two layers is too large. When setting the appropriate number of layers ($n = 4$) and scale difference ($s = 2$), our method successfully maintains important labels and removes unimportant labels using the coarse-to-fine strategy. Therefore, in the experiments described below, we fix n to 4 and s to 2.

4.1.2 Comparison with patchmatch filter

Here, we evaluate the performance of our method by comparing it with PatchMatch filter (PMF) [22] using both *small* and *large* datasets of the Middlebury stereo benchmark [25]. We did not compare the performance of our method with other algorithms dedicated for stereo matching because the stereo matching is merely one of the applications of our method for general multi-labeling problems. For a fair comparison, our method and PMF are performed using the same superpixels clustered by SLIC [1], the cost function, and post-processing based on left-right cross-checking and median-filtering (for further details, see [28]).[4] Further, we evaluate the performance of our method on the basis of a regular image grid with varying block size. Note that the number of local regions is inversely proportional to the block size. The results are presented in Tables 3 and 4. Here, the percentage disparity errors (threshold is set as one for *small* datasets, and one and four for *large* datasets) are averaged over all images within the same category. We observe that although our method, PMF [22], and CVF [28] provide nearly the same level of accuracy, our method is the most efficient method for both categories. In particular, for *large* datasets, our method achieves 6× faster performance than CVF [28], while providing a similar (or higher level) accuracy. We also observe that our method outperforms PMF when the number of local regions is large (e.g., superpixels with $K = 200, 500$) or when the image is divided into local regions on the basis of a simple image grid. This is because unlike the case of PMF [22], we do not consider any spatial smoothness of label subsets within the scale; instead, we consider the cross-scale smoothness of the local label subset, which is independent of the spatial coherence.

The estimated disparity maps of the *teddy* and *art* datasets are shown in Figs. 7 and 8, respectively. These are compared with those obtained by PMF [22] and CVF [28]. We observe that our method succeeds in estimating smoother and more reasonable disparity maps than CVF and PMF, especially in the case of the *teddy* dataset. Near object boundaries,

[4] Post-processing is performed on our method only in the original resolution.

Table 3 Comparison with PMF using *small* datasets

Method	Time[s]	Err. %: thre.=1.0		
		nonocc	all	disc
CVF[28]	35.38	3.30	6.17	9.74
PMF[22] (K=50)	23.43	3.19	**5.97**	9.56
PMF[22] (K=100)	28.97	3.23	6.03	9.32
PMF[22] (K=200)	43.14	3.27	6.04	9.36
PMF[22] (K=500)	73.21	3.30	6.08	**9.31**
Ours (Superpixels, K=50)	15.98	3.51	6.31	10.8
Ours (Superpixels, K=100)	16.56	3.46	6.23	10.7
Ours (Superpixels, K=200)	18.48	3.69	6.48	11.3
Ours (Superpixels, K=500)	23.55	4.15	7.03	12.2
Ours (Grid, 150×150)	17.67	**3.11**	5.98	10.1
Ours (Grid, 75×75)	**12.47**	3.22	6.02	10.4

Bold emphasis means the best performance in each column, which helps readers find the best method in each evaluation metric

CVF and PMF assign many incorrect labels, whereas our method does not. The reason is that our coarse-to-fine strategy successfully truncates incorrect labels that accidentally have low costs, as mentioned in Section 3.3.

Finally, we present the estimated disparity maps of *small* and *large* datasets in Figs. 9 and 10, respectively.

4.2 KITTI stereo 2015

We also conducted experiments on the KITTI stereo 2015 benchmark, which is more difficult than the Middlebury stereo dataset in Section 4.1 in terms of disparity range and image resolution. All the parameters and the cost function are exactly same as those in Section

Table 4 Comparison with PMF using *large* datasets

Method	Time[s]	Err. % (all)	
		Err. thre.=1	Err. thre.=4
CVF[28]	1413	21.5	14.8
PMF[22] (K=50)	266	22.7	15.6
PMF[22] (K=100)	322	22.5	15.5
PMF[22] (K=200)	484	22.5	15.6
PMF[22] (K=500)	802	23.3	16.2
Ours (Superpixels, K=50)	269	22.5	15.3
Ours (Superpixels, K=100)	249	23.0	15.7
Ours (Superpixels, K=200)	262	23.6	16.0
Ours (Superpixels, K=500)	304	24.5	17.2
Ours (Grid, 600×600)	1186	21.1	14.4
Ours (Grid, 300×300)	796	**20.5**	**13.4**
Ours (Grid, 150×150)	371	21.6	14.4
Ours (Grid, 75×75)	**246**	25.2	17.6

Bold emphasis means the best performance in each column, which helps readers find the best method in each evaluation metric

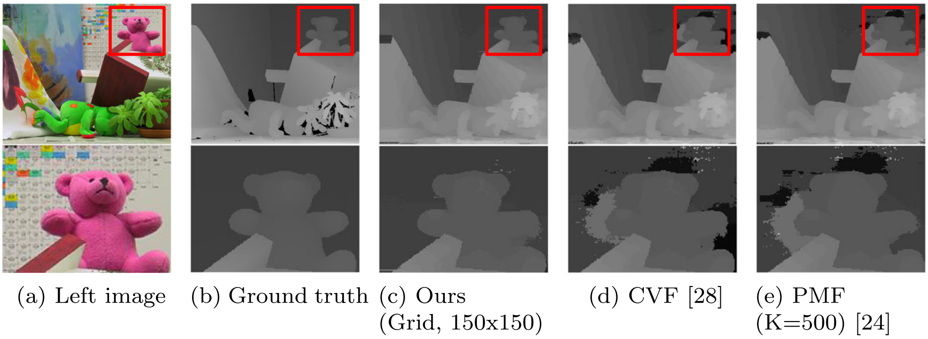


Fig. 7 Qualitative comparison with regard to estimated disparity maps of *Teddy* dataset

4.1. We used 200 training images with ground truth disparity maps. The resolutions of all images are 1241 × 376. In this dataset, we did not perform the post-processing (weighted median filtering) in order to compare the pure performance of each method. The search range of disparity was set to 256 in all methods.

We show the comparison of computational time and accuracy in Table 5. Following the official evaluation rule of KITTI stereo 2015, we computed the percentage of error pixels. We regarded the pixel to be correctly estimated if the disparity error is less than 3 pixel or 5% at each pixel. The results of 200 images are averaged in Table 5. We observe that the accuracy of PMF [22] ($K = 50$) is worse than the original CVF [28] although PMF [22] is the fastest. In this dataset, our method with superpixel division is 5 or 6 times faster than the original CVF [28], and our method ($K = 500$) is much more accurate in both non-occluded and all regions. We observe that the patchmatch search did not work effectively in this dataset as opposed to our coarse-to-fine strategy. However, our method with regular grid division is worse than that with superpixel division in terms of both efficiency and accuracy.

The estimated disparity maps are shown in Fig. 11. Compared with CVF [28] and PMF [22], our method achieved smoother and more reasonable results by truncating unnecessary labels with the coarse-to-fine strategy. We observe that our method is better especially in less or repeated texture regions (e.g., on the road and in the sky).

4.3 Middlebury optical flow

We also carried out experiments using the Middlebury optical flow benchmark [24]. In optical flow estimation, the label l corresponds to the 2-D motion vector (u, v) between the target image and the reference image. Further, u and v denote the displacements along the x

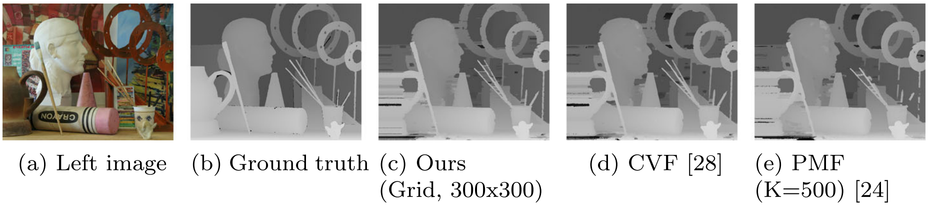
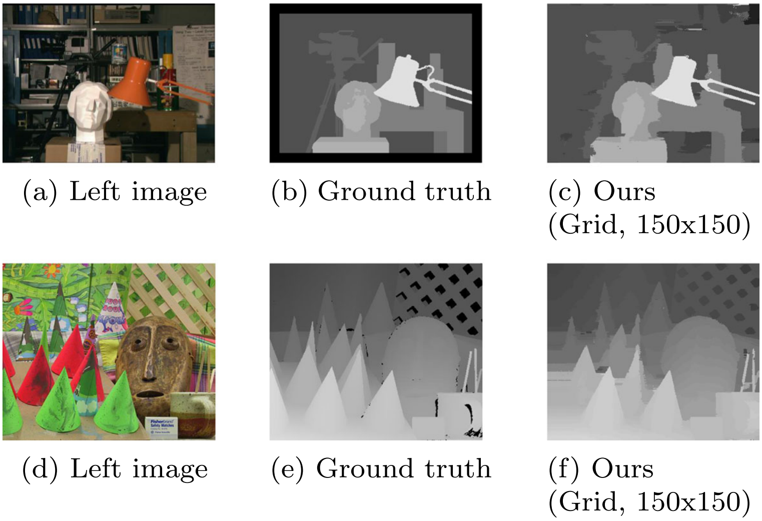


Fig. 8 Qualitative comparison with regard to estimated disparity maps of *Art* dataset



(a) Left image

(b) Ground truth

(c) Ours (Grid, 150x150)

(d) Left image

(e) Ground truth

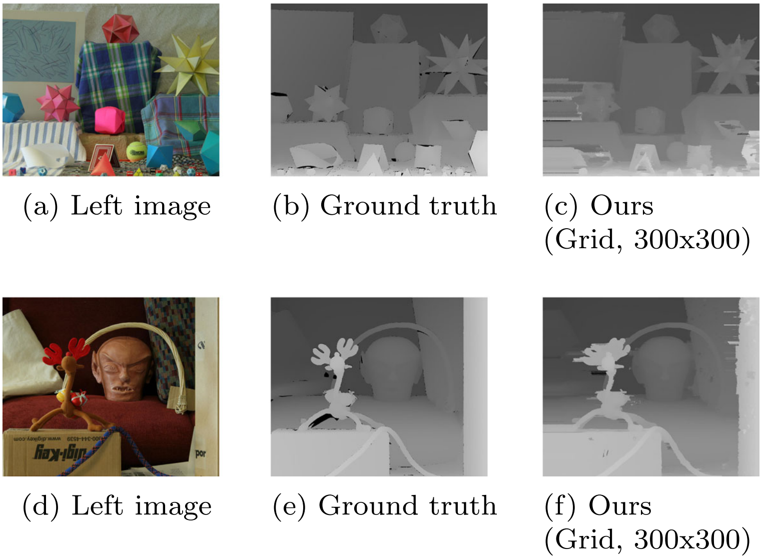
(f) Ours (Grid, 150x150)

Fig. 9 Qualitative results on the *small* datasets

and y directions, respectively, and they take floating values. We use the same cost function as that in the original CVF [28]:

$$\begin{aligned} C(i,l) \;=\; & (1-\alpha)\min\left[\|I'_{i+l} - I_i\|, \tau_1\right] \\ & + \alpha\min\left[\|\nabla_x I'_{i+l} - \nabla_x I_i\| + \|\nabla_y I'_{i+l} - \nabla_y I_i\|, \tau_2\right], \end{aligned} \tag{7}$$

where ∇_x and ∇_y are the gradients in x and y direction, respectively. The parameters are set to the same values as those in the experiments for stereo matching; only τ_2 is changed



(a) Left image

(b) Ground truth

(c) Ours (Grid, 300x300)

(d) Left image

(e) Ground truth

(f) Ours (Grid, 300x300)

Fig. 10 Qualitative results on the *large* datasets

Table 5 Comparison of computational time and accuracy using KITTI stereo 2015 datasets

Method	Time[s]	Err. %	
		Err. Nonocc.	Err. All
CVF[28]	244	32.0	33.2
PMF[22] (K=50)	**28.5**	35.9	36.6
PMF[22] (K=100)	30.7	35.5	36.2
PMF[22] (K=200)	37.7	35.1	35.8
PMF[22] (K=500)	53.5	35.2	35.9
Ours (Superpixels, K=50)	58.2	24.2	25.5
Ours (Superpixels, K=100)	50.9	23.5	24.8
Ours (Superpixels, K=200)	45.2	22.6	23.8
Ours (Superpixels, K=500)	44.5	**22.3**	**23.6**
Ours (Grid, 300×300)	89.4	26.3	27.5
Ours (Grid, 150×150)	67.3	27.2	28.4
Ours (Grid, 75×75)	39.3	24.3	25.6

Bold emphasis means the best performance in each column, which helps readers find the best method in each evaluation metric

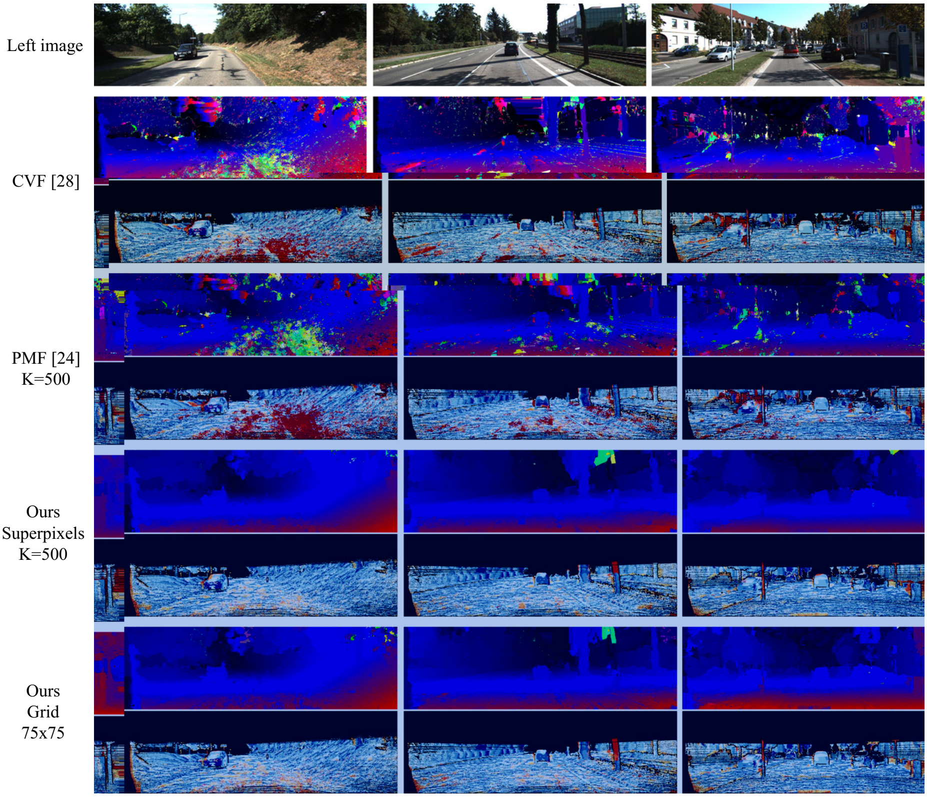


Fig. 11 Qualitative comparison with regard to estimated disparity maps of KITTI stereo 2015 dataset. Disparity maps (*upper rows*) and error maps (*lower rows*)

Table 6 Comparison in optical flow estimation

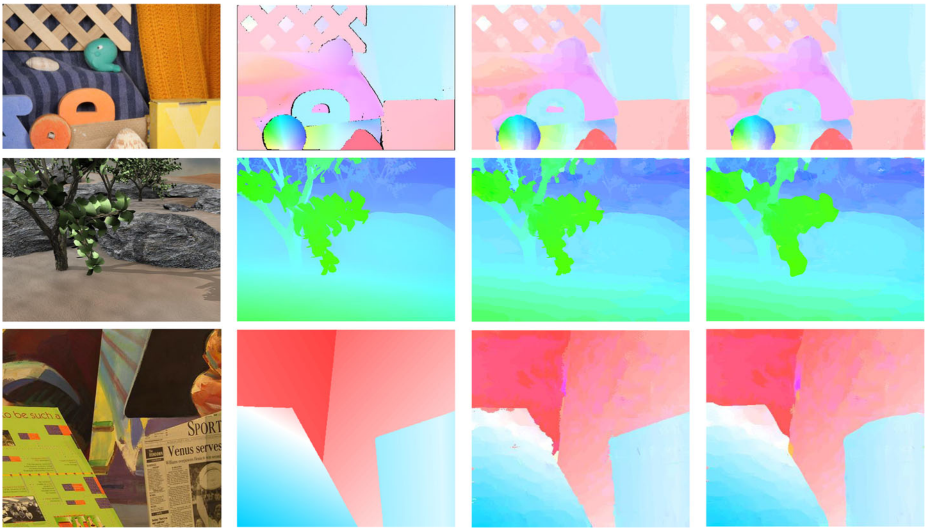
Method	RubberWhale			Grove2			Venus		
	Time[s]	AAE	AEE	Time[s]	AAE	AEE	Time[s]	AAE	AEE
CVF[28]	6978	5.20	0.165	10792	3.65	0.258	5353	6.60	0.432
Ours (Superpixels, K=50)	537	**4.26**	**0.139**	768	**2.59**	**0.191**	493	4.21	0.318
Ours (Superpixels, K=100)	523	4.36	0.143	748	2.63	0.194	485	4.13	0.315
Ours (Superpixels, K=200)	588	4.45	0.145	786	2.63	0.194	524	4.24	0.317
Ours (Superpixels, K=500)	769	4.44	0.145	1009	2.67	0.197	752	**4.07**	**0.306**
Ours (Grid, 150×150)	574	4.32	0.140	739	2.62	0.193	902	4.18	0.312
Ours (Grid, 75×75)	**376**	4.29	**0.139**	**555**	2.61	0.193	**448**	4.10	0.308

Bold emphasis means the best performance in each column, which helps readers find the best method in each evaluation metric

to 0.0156 in the same manner as in [28]. In all the datasets, the search ranges of both u and v are set to the interval of -10 to 10 pixels. To achieve sub-pixel accuracy, the units are set to 0.25 pixel (i.e., each space of u and v is $\{-10, -9.75, -9.5, \ldots, 0, \ldots, 9.5, 9.75, 10\}$). Therefore, the size of the entire label space is $81 \times 81 = 6561$.

The results are listed in Table 6. Here, the average angle error (AAE) and average end-point error (AEE) are used for evaluation. They are defined, respectively, as

$$AE = cos^{-1}\left(\frac{1.0 + u \times u_{GT} + v \times v_{GT}}{\sqrt{1.0 + u^2 + v^2}\sqrt{1.0 + u_{GT}^2 + v_{GT}^2}}\right), \tag{8}$$



(a) First frame (b) Ground truth (c) Ours (Grid, 150x150) (d) CVF [28]

Fig. 12 Qualitative comparison with regard to estimated flow maps of Middlebury optical flow dataset

$$EE = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2}, \tag{9}$$

where (u, v) is the estimated flow and (u_{GT}, v_{GT}) is the ground truth flow. We did not compare the performance of our method with other algorithms dedicated for optical flow estimation for the same reason as stereo matching. From Table 6, we observe that our method is superior to and much faster than the original CVF [28] in all cases. In particular, by using superpixel division ($K = 50$), our method achieves the most accurate results and much faster performance than CVF (10× or more). Further by using regular grid division, our method achieves a higher level of accuracy than CVF, and it is the most efficient.

The estimated flow maps of the Middlebury optical flow dataset are shown in Fig. 12. We observe that our method estimates the flow around boundaries more accurately than CVF. As in the case of our stereo matching results, this is because erroneous flow vectors, which yield minimum costs even though they are the wrong choices, are efficiently removed by our hierarchical approach.

5 Conclusion

In this paper, we proposed a coarse-to-fine strategy to reduce the large label space for efficient cost-volume filtering. The proposed method truncates redundant labels in each local region by using the labeling output of lower scales. Our method demonstrated higher efficiency than CVF while maintaining a comparable level of accuracy in stereo matching and optical flow estimation. Compared with PMF, our method showed comparable performance. Although PMF estimates compact label sets to reduce the computational cost by complex patchmatch search, our method does by simple coarse-to-fine strategy. Therefore, our method is yet another approach to optimize the label sets for efficient cost-volume filtering, which is much easier to implement than PMF. Moreover, we will make our source code publicly available.

In future work, as the performance of our method depends on the shape and number of local regions, we intend to explore the optimal division of local regions. In addition, we plan to investigate the GPU implementation of our method for real-time applications.

Acknowledgements The authors would like to thank T. Taniai of The University of Tokyo for his help with PatchMatch Filter code.

References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Sùsstrunk S (2012) SLIC Superpixels compared to state-of-the-art superpixel methods. IEEE Trans PAMI 34(11):2274–2281
2. Bai X, Luo X, Li S, Lu H (2014) Adaptive stereo matching via loop-erased random walk. In: ICIP
3. Bames C, Shechtman E, Goldman DB, Finkelstein A (2010) The generalized patchmatch correspondence algorithm. In: ECCV
4. Bao L, Yang Q, Jin H (2014) Fast edge-preserving PatchMatch for large displacement optical flow. In: CVPR

5. Bao L, Yang Q, Jin H (2014) Fast edge-preserving patchmatch for large displacement optical flow. IEEE Trans Image Process 23(12):4996–5006
6. Barnes C, Shechtman E, Finkelstein A, Goldman DB (2009) Patchmatch: a randomized correpondence algorithm for structual image editing. In: ACM SIGGRAGH
7. Boykov Y, Kolmogorov V (2004) An experimental comparison of min-cut/max-flow algorithm for energy minimization in vision. IEEE Trans PAMI 26(9):1124–1137
8. Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. IEEE Trans PAMI 23(11):1222–1239
9. Brox T, Bruhn A, Papenberg N, Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. In: ECCV
10. Brunton A, Lang J, Dubois E (2012) Efficient multi-scale stereo of high-resolution planar and spherical images. In: 3 DIMPVT
11. Chen Z, Sun X, Wang L, Yu Y, Huang C (2015) A deep visual correspondence embedding model for stereo matching costs. In: ICCV
12. Cho J, Ikehata S, Yoo H, Gelautz M, Aizawa K (2013) Depth map up-sampling using cost-volume filtering. In: IVMSP Workshop
13. Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, van der Smagt P, Cremers D, Brox T (2015) Flownet: learning optical flow with convolutional networks. In: ICCV
14. Felzenszwalb PF, Huttenlocher DP (2006) Efficient belief propagation for early vision. IJCV 70(1):41–54
15. Furuta R, Ikehata S, Yamasaki T, Aizawa K (2014) Coarse-to-fine strategy for efficient cost-volume filtering. In: ICIP
16. He K, Sun J, Tang X (2010) Guided image filtering. In: ECCV
17. Helala MA, Qureshi FZ (2014) Accelerating cost volume filtering using salient subvolumes and robust occlusion handling. In: ACCV
18. Hosni A, Rhemann C, Bleyer M, Gelautz M (2012) Temporally consistent disparity and optical flow via efficient spatio-temporal filtering. In: Advances in image and video technology, pp 165–177
19. Kolmogorov V (2006) Convergent tree-reweighted message passing for energy minimization. IEEE Trans PAMI 28(10):1568–1583
20. Kramarev V, Demetz O, Schroers C, Weickert J (2013) Cross anisotropic cost volume filtering for segmentation. In: ACCV
21. Lu J, Shi K, Min D, Lin L, Do MN (2012) Cross-based local multipoint filtering. In: CVPR
22. Lu J, Yang H, Min D, Minh ND (2013) Patchmatch filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In: CVPR
23. Luo W, Schwing AG, Urtasun R (2016) Efficient deep learning for stereo matching. In: CVPR
24. Middlebury optical flow database. http://vision.middlebury.edu/flow/
25. Middlebury stereo database. http://vision.middlebury.edu/stereo/
26. Min D, Lu J, Do MN (2011) A revisit to cost aggregation in stereo matching: How far can we reduce its computational redundancy? In: ICCV
27. Min D, Lu J, Do MN (2013) Joint histogram-based cost aggregation for stereo matching. IEEE Trans PAMI 35(10):2539–2545
28. Rhemann C, Hosni A, Bleyer M, Rother C, Gelautz M (2011) Fast cost-volume filtering for visual correspondence and beyond. In: CVPR
29. Richardt C, Orr D, Davies I, Criminisi A, Dodgson NA (2010) Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In: ECCV
30. Sun J, Zheng NN, Shum HY (2003) Stereo matching using belief propagation. IEEE Trans PAMI 25(7):787–800
31. Sun D, Roth S, Black MJ (2010) Secrets of optical flow estimation and their principles. In: CVPR
32. Szeliski R, Zabih R, Scharstein D, Veksler O, Kolmogorov V, Agarwala A, Tappen M, Rother C (2008) A comparative study of energy minimization methods for markov random fields with smoothnes-based priors. IEEE Trans PAMI 30(6):1068–1080
33. Tan X, Sun C, Wang D, Guo Y, Pham TD (2014) Soft cost aggregation with multi-resolution fusion. In: ECCV
34. Tao M, Bai J, Kohli P, Paris S (2012) Simpleflow: a non-iterative, sublinear optical flow algorithm. In: Computer graphics forum, vol 31, pp 345–353

35. Tappen MF, Freeman WT (2003) Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In: ICCV
36. Tomasi C, Manduchi R (1998) Bilateral filtering for gray and color images. In: ICCV
37. Wainwright MJ, Jaakkola TS, Willsky AS (2005) Map estimation via agreement on trees: Message-passing and linear-programming approaches. IEEE Trans Inf Theory 51(11):3697–3717
38. Wang Y, Wang K, Dunn E, Frahm JM (2014) Stereo under sequential optimal sampling: a statistical analysis framework for search space reduction. In: CVPR
39. Wedel A, Pock T, Zach C, Bischof H, Cremers D (2009) An improved algorithm for tv-l 1 optical flow. In: Statistical and geometrical approaches to visual motion analysis, pp 23–45
40. Yang Q (2012) A non-local cost aggregation method for stereo matching. In: CVPR
41. Yang Q (2015) Stereo matching using tree filtering. IEEE Trans PAMI 37(4):834–846
42. Yang R, Pollefeys M (2003) Multi-resolution real-time stereo on commodity graphics hardware. In: CVPR
43. Yang Q, Wang L, Ahuja N (2010) A constant-space belief propagation algorithm for stereo matching. In: CVPR
44. Yoon KJ, Kweon IS (2005) Locally adaptive support-weight approach for visual correspondence search. In: CVPR
45. Yoon KJ, Kweon IS (2006) Adaptive support-weight approach for correspondence search. IEEE Trans PAMI 28(4):650–656
46. Zbontar J, LeCun Y (2015) Computing the stereo matching cost with a convolutional neural network. In: CVPR
47. Zbontar J, LeCun Y (2016) Stereo matching by training a convolutional neural network to compare image patches. JMLR 17(1-32):2
48. Zhan Y, Gu Y, Huang K, Zhang C, Hu K (2016) Accurate image-guided stereo matching with efficient matching cost and disparity refinement. IEEE Trans CSVT 26(9):1632–1645
49. Zhang K, Fang Y, Min D, Sun L, Yang S, Yan S, Tian Q (2014) Cross-scale cost aggregation for stereo matching. In: CVPR
50. Zhao Y, Taubin G (2011) Real-time stereo on gpgpu using progressive multi-resolution adaptive windows. Image Vis Comput 29(6):420–432

Ryosuke Furuta received the B.S. and M.S. degrees in information and communication engineering from The University of Tokyo, in 2014 and 2016, respectively. He is currently working towards the Ph. D. degree at The University of Tokyo. His research interests include computer vision, machine learning, and image processing, especially MRF optimization. He is a member of IEEE, ACM, ITE.

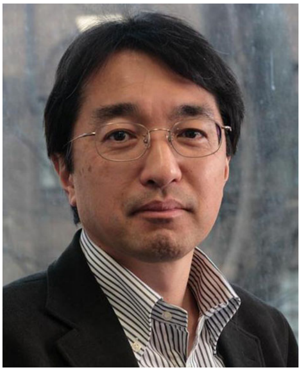
Satoshi Ikehata received the BA degree in psychology and the MS and PhD degrees in information science and technology all from the University of Tokyo, in 2009, 2011 and 2014, respectively. He worked as a postdoctoral researcher at Washington University in St. Louis. Currently, he is an assistant professor at National Institute of Informatics in Tokyo, Japan. His main interests include image-based 3D scene reconstruction, time-of-flight imaging, and human cognition and perception.

Toshihiko Yamasaki received the B.S. degree in electronic engineering, the M.S. degree in information and communication engineering, and the Ph.D. degree from The University of Tokyo in 1999, 2001, and 2004, respectively.

From April 2004 to Oct. 2006, he was an Assistant Professor at Department of Frontier Informatics, Graduate School of Frontier Sciences, The University of Tokyo. He is currently an Associate Professor at Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo. He was a JSPS Fellow for Research Abroad and a visiting scientist at Cornell University from Feb. 2011 to Feb. 2013.

His current research interests include attractiveness computing based on multimedia big data analysis, pattern recognition, machine learning, and so on. His publication includes three book chapters, more than 60 journal papers, more than 160 international conference papers, more than 500 domestic conference papers. He has received around 50 awards.

Dr. Yamasaki is a member of IEEE, ACM, IEICE, ITE, IPSJ.



Prof. Kiyoharu Aizawa received the B.E., the M.E., and the Dr.Eng. degrees in Electrical Engineering all from the University of Tokyo, in 1983, 1985, 1988, respectively. He is currently a Professor at Department of Information and Communication Engineering of the University of Tokyo. He was a Visiting Assistant Professor at University of Illinois from 1990 to 1992. His research interest is in image processing and multimedia applications.

He received the 1987 Young Engineer Award and the 1990, 1998 Best Paper Awards, the 1991 Achievement Award, 1999 Electronics Society Award from IEICE Japan, and the 1998 Fujio Frontier Award, the 2002 and 2009 Best Paper Award, and 2013 Achievement award from ITE Japan. He received the IBM Japan Science Prize in 2002.

He is currently a Senior Associate Editor of IEEE Tras. Image Processing, and on Editorial Board of ACM TOMM, APSIPA Transactions on Signal and Information Processing, and International Journal of Multimedia Information Retrieval. He served as the Editor in Chief of Journal of ITE Japan, an Associate Editor of IEEE Trans. Image Processing, IEEE Trans. CSVT and IEEE Trans. Multimedia. He has served a number of international and domestic conferences; he was a General co-Chair of ACM Multimedia 2012.