COARSE-TO-FINE STRATEGY FOR EFFICIENT COST-VOLUME FILTERING

Ryosuke Furuta, Satoshi Ikehata, Toshihiko Yamasaki and Kiyoharu Aizawa

The University of Tokyo

ABSTRACT

Cost-volume filtering is one of the most widely known techniques to solve general multi-label problems, however it is problematically inefficient when the label space size is extremely large. This paper presents a coarse-to-fine strategy of the cost-volume filtering that handles efficiently and accurately multi-label problems with a large label space size. Based upon the observation that true labels at the same image coordinate of different scales are highly correlated, we truncate unimportant labels for the cost-volume filtering by leveraging the labeling output of lower scales. Experimental results show that our algorithm achieves much higher efficiency than the original cost-volume filtering while enjoying the comparable accuracy to it.

***Index Terms*—** cost-volume filtering, Markov random fields, label selection, coarse-to-fine

1. INTRODUCTION

Many low-level computer vision problems (*e.g*., stereo matching and optical flow estimation) are formulated as multi-label problems where discrete labels (*e.g*., disparity and motion vector) are assigned to pixels. There are generally 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 [1, 2, 3] are used to minimize the energy function. While effective, solving a large optimization problem makes the inference intractable when the image size is high or the label space is large. More recently, Rhemann *et al*. [4] presented a local approach called *cost volume filtering* (CVF), which efficiently solves general multi-label problems. The trick is that CVF substitutes the fast local filtering of label costs for the global smoothing in the MRF optimization. Since CVF is easy to implement yet provides high-quality results, it has been widely used to solve various multi-label problems [5, 6, 7, 8, 9]. However, a limitation of CVF is that it does not scale to the extremely large label set (*e.g*., sub-pixel stereo matching and upsampling of the 16-bit depth map from the kinect sensor).

To tackle this difficulty, Lu *et al*. [10] have recently proposed PatchMatch Filter (PMF) algorithm which performs CVF iteratively on local superpixels with compact label subsets instead of performing on the entire image coordinate. Since the average size of local label subsets is generally much smaller than the size of the entire label subset, PMF is usually much more efficient than CVF while keeping its accuracy. Nevertheless, PMF relies on the complex patch-match based global optimization to estimate a label subset for each superpixel whose computational complexity increases in response to the number of superpixels, therefore 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 of the cost-volume filtering. Based upon the observation that true labels at the same image coordinate of different scales are highly correlated, we propose to leverage the labeling output of the lower scale for estimating local label subsets of the higher scale. Starting from the very low-resolution image, we iteratively truncate unimportant labels of each higher scale and finally, we assign compact and approximately optimal label subsets to local regions of the original scale. The proposed framework benefits from simple, efficient coarse-to-fine strategy, which does not require any global optimization like [10] and its computational complexity is less affected by the number of local regions. Extensive experiments in Sec. 3 show that our algorithm achieves the higher efficiency than PMF and CVF while enjoying the comparable or often superior accuracy to them.

2. COARSE-TO-FINE STRATEGY FOR EFFICIENT COST VOLUME FILTERING

In this section, we present a coarse-to-fine strategy of the cost-volume filtering (CVF) [4] for handling multi-label problems with a large label space. Given a label set $\mathcal{L} = \{l_0, \cdots, l_{L-1}\}$, the goal of a multi-label 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 I which minimizes the label costs that are encoded in the energy function [4].

2.1. Cost-volume Filtering

CVF [4] solves multi-label problems by three steps. First, a 3-D cost volume C is constructed as a collection of the cost for choosing label l at each pixel i which is based on the data term in the energy function. Then each slice of the cost volume is independently filtered by an edge preserving

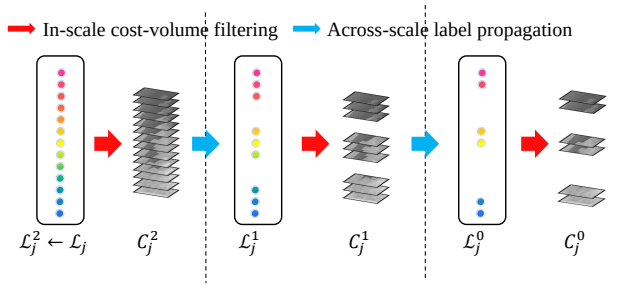

Fig. 1. A framework of proposed method.

filter [11, 12] which substitutes for the smoothness term in the energy function. Finally, the label at pixel i is simply chosen in a winner-take-all strategy. When $O(1)$ edge preserving filter (*e.g.*, guided filter [11]) is used, the computational complexity of filtering an entire cost volume is $O(ML)$: M is the number of pixels in I and L is the size of $\mathcal{L}$, which leads to the difficulty in handling an extremely large label space.

One possible strategy for handling a large label space problem is to locally change the label space for reducing the size of label space. Because the true label configuration is generally smooth in space (*e.g.*, disparities are smooth except for object boundaries), the necessary label space for performing CVF on a local region should be much smaller than the entire label space. However, the problem is of course we do not know a priori which labels are important for each local region, and thus the estimation of local label subsets is required [10].

2.2. Problem Statement

Here we present a simple but efficient label subset estimation algorithm. Unlike Lu *et al*. [10], we do not rely on the global optimization for estimating local label subsets instead leveraging the coarse-to-fine framework. The overview of the proposed method is illustrated in Fig. 1. Our algorithm consists of mainly two steps (i) in-scale cost-volume filtering and (ii) across-scale label propagation. An essential ingredient is the latter step, where a local label subset is estimated from the CVF output at its lower resolution. Since the computational cost of CVF for a low-resolution image is less noticeable, we perform CVF with a large label space at low-resolution and then 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 factor $s \in (0, 1)$)[1] and $\mathcal{L}^k$ denote a set of all possible labels at k-th scale which is given by $\mathcal{L}$. Then we divide I^0 $(= I)$ into m non-overlapping local regions S_j^0 and then partition $I^k(k \geq 1)$ into local regions $S_j^k(j = 1, \ldots, m-1)$ such that $S_j^{k+1} = S^k_{j\downarrow s}$.[2] In addition, we represent a label subset for S_j^k as $\mathcal{L}_j^k$ and its size as L_j^k. The entire computational complexity of CVF from the lowest scale $(k = n)$ to the original scale $(k = 0)$ is described as

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

where M_j^k is a number of pixels in S_j^k (*i.e.*, $M_j^k = s^{2k} M_j^0$). Therefore, our goal is to estimate compact label subsets $\mathcal{L}_j^k$ such that $\sum_{k=0}^{n}\sum_{j=0}^{m} M_j^k L_j^k \ll ML$ while keeping the accuracy of CVF.

2.3. Across-scale Label Propagation

In this section, we present an algorithm to estimate compact label subsets $(\mathcal{L}_j^k)$ which sufficiently reduce Eq. (1) without truncating important labels. Our algorithm is begun by 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 [4] to acquire the filtered cost volume C^{n-1} at $(n-1)$-th scale[3]. Note that though we use a complete label set, the computational complexity of CVF at this scale is $O(s^{2(n-1)}ML)$, which is generally neglectable (*e.g.*, if we set s by 0.5 and n by 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 which have the smallest cost values in C^{n-1} at corresponding local regions S_j^{n-1} . Strictly speaking, the initialization is represented as

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

where $C^{n-1}(p, q)$ is a value of the cost volume at $(n-1)$-th scale with regard to the position p and the label q, and f is a *projection function* which normalizes the label space if necessary. The projection function is generally represented as a constant scale factor giving $f = s^{-1}$. For instance, a disparity l at the k-th scale corresponds to $s^{-1}l$ at the $(k-1)$-th scale in the stereo matching problem[4]. The initialization method based on the across-scale label propagation is motivated by a reasonable observation that true labels at the same image coordinate of different scales are highly correlated, especially they are very close when the difference of scales is small.

While 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 $\tilde{\mathcal{L}}_j^{n-2}$

[1] We used "buildPyramid" function in OpenCV to downsample images.

[2] S_j^0 can be generated in various ways *e.g.*, rectangular regular grids or super-pixels [13] as shown in Sec. 3.

[3] When local regions are not rectangular (*e.g.*, superpixels with varying shapes), we perform the edge-aware filter on the bounding-box containing each region in the same manner with [10].

[4] There are some cases where the label space does not need to be normalized since the scale of a label does not depend on the image coordinate. Some examples are depth-map upsampling [9] and image segmentation [8].

does not consist of labels that are not included in $f(\mathcal{L}_j^{n-1})$, which results in the aliasing artifacts when the intermediate labels of $\mathcal{L}_j^{n-1}$ should be included in $\mathcal{L}_j^{n-2}$ (artifacts become more problematic as the scale difference increases). In addition, filtered cost volume C^{n-1} often contains numerical errors due to occlusion boundaries or insufficient energy modeling. We overcome these difficulties by two strategies. First, we downsample images with relatively large scale factor (*e.g.*, $s \geq 0.5$), so that the scale difference between two layers becomes sufficiently small. Second, we complete the initial label subset by adding the supporting labels within $\pm 1/(2s)$. We should note that our algorithm supports floating labels (*e.g.*, sub-pixel disparity values). For instance, if the scale factor is 0.5 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.*, $k \leftarrow n-2$). In the similar manner with 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 $\mathcal{L}^k$ and the estimation of $\mathcal{L}^{k-1}$ from C^k are iterated $n-1$ times until $\mathcal{L}_j^0$ is acquired Then the final label at each pixel in S_j^0 is selected by a simple winner-takes-all strategy in the same manner with CVF [4].

3. RESULTS

Experiments were carried out to evaluate the performance of proposed method using the Middlebury stereo matching benchmark [14]. In stereo matching, the label l corresponds to the integer disparity between a pixel i in the target image I and the correspondence in the reference image I' shifted by the disparity. The cost function is chosen in the same manner with [4] where model parameters of α, τ_1 and τ_2 are set by 0.89, 0.0027, 0.0078 respectively[5]. We divide eight test image pairs in Middlebury stereo datasets [14] into two categories with their size: *small* and *large*. The *small* category includes *cones* (450×375), *teddy* (450×375), *tsukuba* (384×288) and *venus* (434×383). And the *large* category includes *art* (1390×1110), *books* (1390×1110), *moebius* (1390×1110) and *reindeer* (1342×1110). The label space size L is set by 60 in the *small* datasets and 240 in the *large* datasets. All experiments were performed on Intel Core i7-2600 (3.4GHz, single thread) machine with 16 GB of RAM and were implemented in C++. Like the original work of CVF [4], we use the guided image filter [11] to smooth the cost volume (the radius of the filter is fixed by 9).

3.1. Evaluation of Label Selection

We begin by evaluating the efficiency of our coarse-to-fine strategy comparing with CVF [4]. Here we apply our method

[5]Parameters are given by authors of [4]

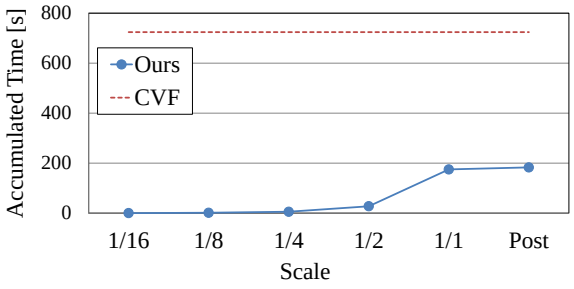

Fig. 2. Evaluation of the computation time. The results of eight Middlebury stereo datasets are averaged. *Post* indicates the entire computational time after the weighted median filtering for the final disparity-map refinement.

($n = 5, s = 0.5, m = 30$) and CVF [4] to both *small* and *large* datasets and results are averaged as shown in Fig. 2. We observe that our coarse-to-fine strategy entirely takes much less time than CVF [4]. As we expected, the computational time for small scales (*e.g.*, $1/16, 1/8, 1/4\times$) is neglectably smaller than that of the original resolution ($1/1\times$).

However, one important question arises, "Estimated label subsets of the original scale are really correct?" which directly addresses the accuracy of the final label selection. To answer this question, we define two metrics for measuring the correctness of the final label subset as

$$P(j) = \frac{|\mathcal{L}_j^0 \cap \mathcal{L}_j|}{|\mathcal{L}_j^0|}, \quad R(j) = \frac{|\mathcal{L}_j^0 \cap \mathcal{L}_j|}{|\mathcal{L}_j|}, \tag{3}$$

where $\mathcal{L}_j$ is the subset of ground truth labels at the original scale (*i.e.*, a collection of ground truth disparity values emerged in the j-th region) and we remind 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 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 $\forall j \ \mathcal{L}_j^0 = \mathcal{L}_j$.

Using these metrics, we evaluate our method with varying scale factor s and number of layers n using only *small* datasets as shown in Table 1 and Table 2. Here results are averaged over all datasets in this category. In summary, we observe that our algorithm successfully maintains more than 80% of ground truth labels and truncate more than 50% unnecessary labels in average while the original label subset contains 90% of unnecessary labels. We also observe that as expected, the improvement of the precision is generally limited when the number of layers is too small or the scale differences between two layers are too large. Therefore in the following experiments, we fix n by 4 and s by 0.5.

Table 1. Evaluation of the 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=0.5,n=5)	0.58	0.89
1/16→1/4→1/1(s=0.25,n=3)	0.48	0.89
1/16→1/1(s=0.0625,n=2)	0.23	0.93
1/1(CVF[4])	0.13	1.00

Table 2. Evaluation of the 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=0.5,n=5)	0.58	0.89
1/8→1/4→1/2→1/1(s=0.5,n=4)	0.58	0.91
1/4→1/2→1/1(s=0.5,n=3)	0.57	0.90
1/2→1/1(s=0.5,n=2)	0.49	0.92
1/1(CVF[4])	0.13	1.00

3.2. Comparison with PatchMatch Filter

Here we evaluate the performance of our method by a numerical comparison with PatchMatch filter (PMF) [10] using both *small* and *large* datasets in Middlebury stereo benchmark [14]. For a fair comparison, our method and PMF are performed with same superpixels clustered by SLIC [13], the cost function, and the post-processing based on left-right cross-checking and median-filtering (see details in [4])[6]. We also evaluate the performance of our method based on the regular image grid with varying block size. Note that the number of local regions is in inverse proportion to the block size. The results are displayed in Table 3 and Table 4. Here the percentage disparity errors (threshold is set by one for *small*, and one and four for *large*) are averaged over all images with the same category. We observe that while the accuracy of our method, PMF [10] and CVF [4] are almost same, our method is most efficient method of all for both categories. Especially as for *large* datasets, our method achieves $6\times$ faster performance than CVF [4] while keeping (or even better) accuracy. We also observe that our method works when the number of local regions is large (*e.g.*, superpixels with $K = 200, 500$) or when the image is divided into local regions as a simple image grid. That is because unlike PMF [10], we do not consider any spatial smoothness of label subsets within the scale, rather consider the cross-scale smoothness of the local label subset which is independent of the spatial coherence.

Finally, we illustrate the estimated disparity maps of *Teddy* dataset in Fig. 3. We observe that our method succeeds in estimating smoother and more reasonable disparity maps than CVF and PMF in this condition.

[6]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[4]	35.38	3.30	6.17	9.74
PMF[10] (K=50)	23.43	**3.19**	**5.97**	9.56
PMF[10] (K=100)	28.97	3.23	6.03	**9.32**
PMF[10] (K=200)	43.14	3.27	6.04	9.36
PMF[10] (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, 150x150)	17.67	**3.11**	**5.98**	10.1
Ours (Grid, 75x75)	**12.47**	3.22	6.02	10.4

Table 4. Comparison with PMF using *large* datasets.

Method	Time[s]	Err. % (all)	
		Err. thre.=1	Err. thre.=4
CVF[4]	1413	21.5	14.8
PMF[10] (K=50)	266	22.7	15.6
PMF[10] (K=100)	322	22.5	15.5
PMF[10] (K=200)	484	22.5	15.6
PMF[10] (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, 600x600)	1186	**21.1**	**14.4**
Ours (Grid, 300x300)	796	**20.5**	**13.4**
Ours (Grid, 150x150)	371	21.6	**14.4**
Ours (Grid, 75x75)	**246**	25.2	17.6

Fig. 3. Qualitative comparison with regard to estimated disparity maps of *Teddy* dataset.

4. CONCLUSION

In this paper, we have proposed a coarse-to-fine strategy to reduce the large label space for the efficient cost volume filtering. Our proposed method has demonstrated the highest efficiency, while keeping the accuracy in stereo matching. In future work, we will apply our method to other applications of discrete labeling problems which is hard to solve for the original cost volume filtering because of the huge label space.

5. REFERENCES

[1] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. PAMI*, vol. 23, no. 11, pp. 1222–1239, 2001.

[2] J. Sun, H.-Y. Shun, and N.-N. Zheng, "Stereo matching using belief propagation," in *ECCV*, 2002.

[3] M. F. Tappen and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters," in *Proc. of ICCV*, 2003.

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

[5] Q. Yang, "A non-local cost aggregation method for stereo matching," in *CVPR*, 2012.

[6] A. Hosni, C. Rhemann, M. Bleyer, and M. Gelautz, "Temporally consistent disparity and optical flow via efficient spatio-temporal filtering," in *Advances in Image and Video Technology*, pp. 165–177. Springer Berlin Heidelberg, 2012.

[7] A. Brunton, J. Lang, and E. Dubois, "Efficient multi-scale stereo of high-resolution planar and spherical images," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*. IEEE, 2012.

[8] V. Kramarev, O. Demetz, C. Schroers, and J. Weickert, "Cross anisotropic cost volume filtering for segmentation," in *Computer Vision-ACCV 2012*, pp. 803–814. Springer Berlin Heidelberg, 2013.

[9] J. Cho, S. Ikehata, H. Yoo, M. Gelautz, and K. Aizawa, "Depth map up-sampling using cost-volume filtering," in *Proc. of IVMSP Workshop*, 2013.

[10] J. Lu, H. Yang, and N. D. Min, D. Minh, "PatchMatch filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation," in *CVPR*, 2013.

[11] K. He, J. Sun, and X. Tang, "Guided image filtering," in *ECCV*, 2010.

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

[13] 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*, vol. 34, no. 11, pp. 2274–2281, 2012.

[14] "Middlebury stereo database," `http://vision.middlebury.edu/stereo/`.

[15] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correpondence algorithm for structual image editing," in *Proc. of ACM SIGGRAGH*, 2009.