Soft Cost Aggregation with Multi-Resolution Fusion

Xiao Tan[1,2], Changming Sun[1], Dadong Wang[1], Yi Guo[1], and Tuan D. Pham[3]

[1]CSIRO Computational Informatics, North Ryde, NSW 1670, Australia
[2]The University of New South Wales, Canberra, ACT 2600, Australia
[3]The University of Aizu, Fukushima, Japan
{tanxchong@gmail.com} {changming.sun@csiro.au} {dadong.wang@csiro.au}
{yi.guo@csiro.au}{tdpham@u-aizu.ac.jp}

Abstract. This paper presents a simple and effective cost volume aggregation framework for addressing pixels labeling problem. Our idea is based on the observation that incorrect labelings are greatly reduced in cost volume aggregation results from low resolutions. However, image details may be lost in the low resolution results. To take advantage of the results from low resolution for reducing these incorrect labelings while preserving details, we propose a multi-resolution cost aggregation method (MultiAgg) by using a soft fusion scheme based on min-convolution. We implement our MultiAgg in applications on stereo matching and interactive image segmentation. Experimental results show that our method significantly outperforms conventional cost aggregation methods in labeling accuracy. Moreover, although MultiAgg is a simple and straightforward method, it produces results which are close to or even better than those from iterative methods based on global optimization.

Keywords: Multi-resolution fusion, Cost aggregation, Stereo matching, Interactive segmentation

1 Introduction

Many early vision problems, such as stereo matching and image segmentation, can be formulated as pixel-labeling problems. The labels represent some specified local quantities [13] such as disparity for stereo matching or background/object index for segmentation. Generally, a good labeling should be both locally smooth and edge-preserving while being consistent with the observed data. The labeling methods can be generally categorized into two classes. One is the local cost aggregation methods such as the recently developed cost volume filtering [11] and non-local aggregation [16]. In these methods, the cost volume is aggregated within a local region by implicitly making a spatial smoothness assumption. Another alternative to the local method is the global method. In global methods, the labeling problem is solved by minimizing an energy function which explicitly incorporates local smoothness constraints. In general, global methods produce more satisfactory results at a cost of running time. Conversely, local aggregation methods are more efficient but yield less accurate results.

Local cost aggregations typically use adaptive supports to achieve edge preserving aggregation. One good example of these methods is the bilateral filter [14] based supports as proposed in the adaptive window method [21]. However, due to the high computational complexity of the full kernel implementation to the bilateral filter, many methods [17, 18] are proposed for speeding up the implementation with the cost of a lower accuracy. In addition to these bilateral filter based methods, various methods based on different types of adaptive support are developed for cost aggregation, such as those in [6], [8], and [11]. Recently, non-local cost aggregation methods are proposed based on tree structures [16]. Unlike the local aggregation methods as mentioned above, the non-local methods propagate the contribution of a pixel to all other pixels. These methods are robust in low texture regions.

Multi-resolution image processing is an old but still widely used scheme [15]. One of its important characteristics in solving pixel-labeling problem is that the incorrect labeling can be reduced in a lower resolution version of the original image, but the risk of losing important details increases as the resolution goes down. The balance between low resolution and high resolution is found by incorporating the multi-resolution or coarse-to-fine methods into an optimization framework [4, 7, 19]. Yang and Pollefeys propose a multi-resolution cost aggregation method by summing up the matching scores computed from several kernels in different resolutions [20]. Another method [22] uses the results from the lower resolution to guide the search range at a higher resolution. These methods are very efficient and can be easily implemented in hardware with parallel acceleration. However, they do not produce satisfactory results.

Despite the fact that current cost aggregation methods achieve great success by introducing edge-aware filtering methods into adaptive local cost aggregation or by using minimum spanning tree (MST) for non-local aggregation, all these methods are sensitive to the local property of the images. For example, unreliable results by local adaptive cost aggregation methods are usually observed in textureless regions. By aggregating cost on a MST, non-local methods perform well in textureless regions, but they are vulnerable in regions containing too much texture, particularly in regions containing repetitive patterns. Because the contribution of a pixel to another is measured by the distance of the path between two pixels on the tree, a pixel in highly textured regions can hardly receive any contribution from other pixels. Then a challenging question that follows is: whether the multi-resolution technique can be introduced to break the bottleneck of both local and non-local cost aggregation methods, providing comparable or even better results than global methods, while still maintaining the computational efficiency without using any iterative optimization methods.

In this paper, we present a multi-resolution cost aggregation method (MultiAgg) to achieve this goal. In our method, a cost volume is computed at the original resolution. The guidance image (e.g., the reference image in stereo matching) and the cost volume are both down sampled from the original resolution to the lowest resolution. Then a soft aggregation is carried out from the lowest resolution. The aggregation results from the low resolution are passed to the next

higher resolution and the results are fused with the cost volume there for the next round of soft aggregation. The final labeling is decided from the soft aggregation results at the original resolution by the winner-take-all (WTA) method.

A great advantage of our soft aggregation scheme is that it takes both the advantages of reducing incorrect labeling from low resolutions and preserving details from fine resolutions. The proposed method is a generic framework that works well with many current state-of-the-art cost aggregation methods including both local and non-local aggregation methods. The proposed method boosts the robustness of these methods against different local features, such as the lack of texture or too much texture. We implement our methods to address two vision tasks: stereo matching and interactive segmentations. Experimental results show that our method outperforms current state-of-the-art cost aggregation methods both quantitatively and visually.

Another advantage of our method is its computational efficiency which is inherited from the current well developed fast cost aggregation methods. In addition, our method is straightforward, without involving any iterative process, and the fusion process is carried out independently for each pixel, which means that it can be easily embedded into parallelized acceleration systems.

2 Method

2.1 Adaptive Cost Aggregation

In this section, we firstly review several adaptive cost aggregation (ACA) methods. Assuming that a cost volume have been computed and denoted by $C_l(p)$ for pixel p at label l. ACA methods compute the aggregation results by

$$\hat{C}_l(p) = \sum_{q \in \Omega_p} w_{q,p} C_l(q) \tag{1}$$

where $\hat{C}_l(p)$ is the aggregation results, $w_{q,p}$ is the weight between pixel p and q measured in the guidance image I, and Ω_p is the support region of p. Different methods are used in defining $w_{q,p}$ and Ω_p. For example, authors in [11] and [21] use the bilateral filtering weights and the guided filtering weights as $w_{q,p}$ respectively. In [8], Ω_p is delineated by the cross based skeleton, and all pixels are used as Ω_p for non-local aggregation [16].

2.2 Multi-Resolution Aggregation and Soft Cost Fusion

Unlike conventional aggregation methods, we first build two pyramids by recursively half down-sampling the cost volume and the guidance image before carrying out cost aggregation (see Fig. 1). The down-sampling of the cost volume is performed at each individual label in the image space. Denote the pyramids of guidance images and cost volumes by $\left\{I^{(1)}, I^{(2)}, I^{(3)}, ..., I^{(N)}\right\}$ and $\left\{C^{(1)}, C^{(2)}, C^{(3)}, ..., C^{(N)}\right\}$ where N is the total number of level of the pyramids; $I^{(1)}$ and

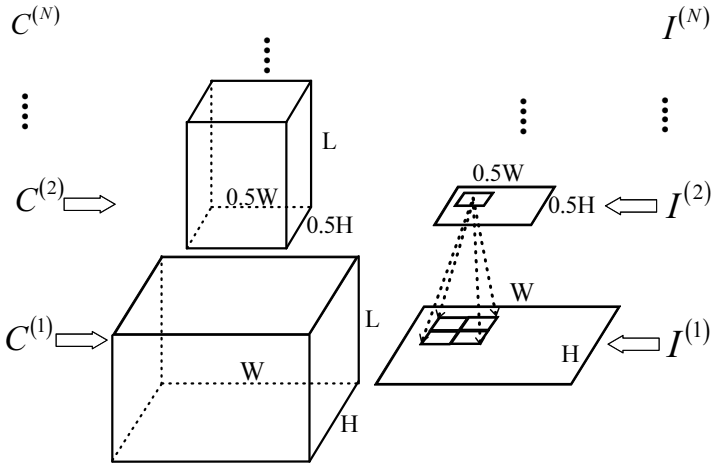


Fig. 1. Down-sampling is performed sequentially on both the guidance image and the cost volume. The size of the original image is $W \times H$, and the size of label space is L. The down-sampling of the cost volume is only carried out on image space. One pixel in a lower resolution corresponds four pixels in the next higher resolution

$C^{(1)}$ are the original guidance image and cost volume. The multi-resolution aggregation starts from the lowest resolution: aggregating $C^{(N)}$ under the guidance by $I^{(N)}$ using the ACA method in Eq. (1). The aggregation results in the nth level, $\hat{C}^{(n)}$, are passed to the next higher resolution and are fused with $C^{(n-1)}$. We expect that incorrect labelings are reduced while the details are preserved in the fused cost volume. Therefore, the fusion should hold some properties:

1. Fusion results encourage suggested labels from the results of lower resolution. The suggested labels are those where the value of $\hat{C}^{(n)}$ is low.
2. The extremely low value of $C^{(n-1)}$ should stay low in the fusion results.
3. Labels which are close to the suggested labels should also have low cost values in fusion results.

The functionality of the first requirement is obvious: it helps to reduce incorrect labelings in higher resolution by considering its lower resolution results. The second requirement is necessary for preserving details and boundaries in the higher resolution images. Generally, when a label l of a pixel p has an extremely low value in the cost function in the higher resolution but l does not have a low value in the aggregation results of its corresponding pixel p' at the lower resolution, it is very likely that p lies in regions containing details which are lost in the lower resolution results. The third requirement is useful in some applications, such as stereo matching and optical flow estimation, where the label of a pixel is close to those of its corresponding pixels in higher resolutions.

Even though the fusion method which satisfies the three points as mentioned above may not be unique, we found that the following method is very

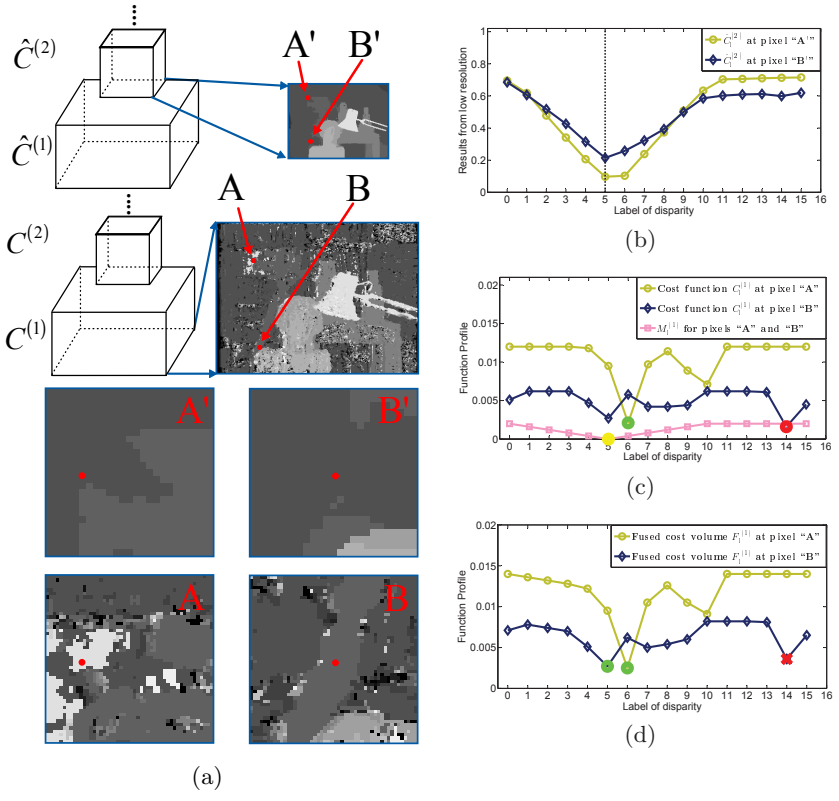


Fig. 2. (a) Applying a WTA method on the aggregation result at lower resolution: $\hat{C}^{(2)}_{l'}$, and the cost volume at higher resolution: $C^{(1)}$. (b) Aggregation result at pixels "A′" and "B′". Label which has the lowest cost is denoted by the vertical dash line. (c) Cost function of A and B in $C^{(1)}$ and min-convolution results between the aggregation results in (b) and a truncated linear function. Lowest cost values are denoted in "green", "red", and "yellow". (d) Correct labels (green points) have the lowest cost after fusion. Incorrect label (denoted by red cross) where the value of the original cost is the lowest does not stay to be the lowest after fusion

efficient and effective for applications at hand. In this method, $\hat{C}^{(n)}$ is firstly min-convoluted [5] with a robust function and then the results are added to $C^{(n-1)}$ to generate a fused cost volume $F^{(n-1)}$ at level $n-1$. The result of min-convolution [5] at pixel p is the lower envelop of functions by rooting the robust function at points of $\left(l', \hat{C}^{(n)}_{l'}(p')\right)$ for all l'. That is

$$M^{(n-1)}_{l}(p) = \min_{l'}\left(\widehat{C}^{(n)}_{l'}(p') + V^{(n)}(l-l')\right) \tag{2}$$

Fused cost volume is then generated by

$$F_l^{(n-1)}(p) = C_l^{(n-1)}(p) + M_l^{(n-1)}(p) \tag{3}$$

where p' is the corresponding pixel of p in the lower resolution; $V^{(n)}(l - l')$, a robust function, is chosen according to applications. For example, the Potts model function for interactive segmentation and the truncated linear function for stereo matching. $V^{(n)}$ is augmented by the sampling scale at the nth level: $V^{(n)} = 2^{n-1} \times V$, where V is the robust function at the original level: $V = V^{(1)}$. To explain why the proposed fusion scheme works for both preserving details and reducing the incorrect labelings, we now show an example of min-convolution with a truncated linear function in stereo matching problem (see Fig. 2). In Fig. 2, $M_l^{(1)}(A)$ and $M_l^{(1)}(B)$ are both subtracted by a normalization constant for all l. Disparity "5" from the lower resolution leads to details missing around pixel "A′". The correct labeling to pixel "A" (disparity "6") has an extremely low value in the cost function, which is preserved in the fusion results. Although incorrect labeling (disparity "14") has the lowest value of the cost function of pixel "B", the correct labeling (disparity "5") has a comparable low value. By considering results in (b), i.e., adding $M_l^{(1)}$ to the cost volume, the correct labeling to pixel "B" now has the lowest value in the fused cost function.

2.3 Multi-Resolution Soft Aggregation

Conventional ACA methods can be carried out directly on the fused cost volume $F_l^{(n)}$ for cost aggregation at the nth level; however, we found that conventional ACA methods may introduce wider artifacts near weak boundaries of the guidance image. The reason is as follows: $F_l^{(n)}$ at a pixel contains the summation of the cost values over all corresponding pixels at the original resolution, which leads to large value differences of $F_l^{(n)}$ for different labels. This large difference is likely to over-penalize the labeling discontinuity when using conventional ACA methods. Recall the scheme for soft cost fusion in the previous section, we propose a soft aggregation method: instead of directly using the cost volume in aggregation, the min-convolution [5] results between F and a robust function are aggregated (see Fig. 3). Empirically, $V^{(n)}(l - l')$, the robust function in the previous section, can be directly used here. That is

$$E_l^{(n)}(q) = \min_{l'} \left(F_l^{(n)}(q) + V^{(n)}(l - l') \right) \tag{4}$$

and

$$\hat{C}_l^{(n)}(p) = \sum_{q \in \Omega_p} w_{q,p} E_l^{(n)}(q) \tag{5}$$

2.4 Implementation Issues and Algorithm Steps

Being a generic framework, our method works very well with many advanced cost aggregation methods. Theoretically, different ACA methods or different

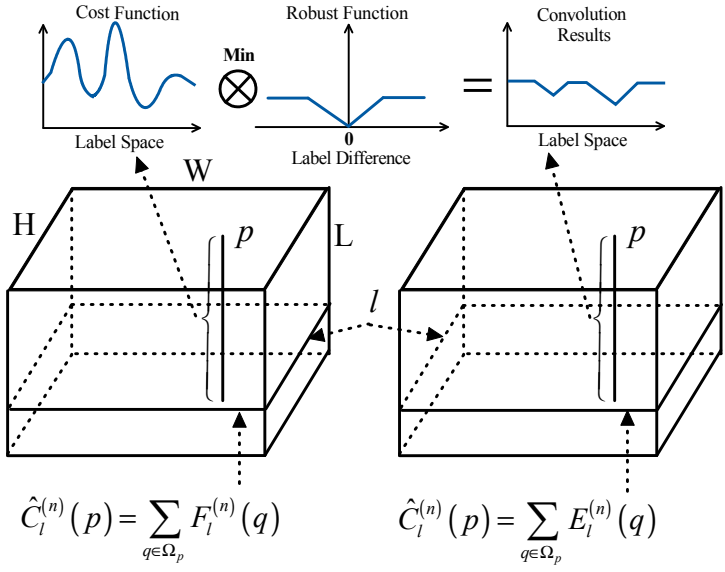


Fig. 3. Conventional ACA methods carried out the aggregation on the cubic of fused cost volume of the nth level. Soft aggregation is carried out on the cubic of min-convolution results

parameters can be used for cost aggregation in different levels. In this study, we use the same ACA method with constant parameters, such as window size and weighting parameters, in different levels for simplicity.

The first issue is the computational complexity. The complexity of our method is depended on the adopted ACA method and the robust function V. We focus on two types of V: truncated linear function and Potts model function. According to [4], the min-convolution requires 3 operations (add or minus) for each pixel at each label when using the truncated linear function and requires 2 operations when using the Potts model function. As the total number of pixels is reduced by 4 times in the next lower resolution, the operations for all resolutions is $\frac{4}{3}$ times of that in the original resolution. For an image being processed with size $W \times H$ and the label space size L, the total number of operations of min-convolution in cost fusion is $\left(\frac{4}{3} + \frac{1}{4}\left(3 \times \frac{4}{3}\right)\right) WHL$ (for truncated linear model) or $\left(\frac{4}{3} + \frac{1}{4}\left(2 \times \frac{4}{3}\right)\right) WHL$ (for Potts model). The total number of operations of min-convolution in soft aggregation is $\left(3 \times \frac{4}{3}\right) WHL$ (for truncated linear model) or $\left(2 \times \frac{4}{3}\right) WHL$ (for Potts model). As the guidance image is the same for all labels, guidance image down-sampling is performed only once and the overhead is negligible. The number of operations for building the cost volumes pyramid is $\frac{4}{3} WHL$. Assume that the ACA method being employed requires O operations for aggregating cost volume in the original resolution level, our method requires

$\frac{4}{3}O + 8WHL$ (for truncated linear model) or $\frac{4}{3}O + 6WHL$ (for Potts model) operations in total.

Another issue to be discussed is how many levels of hierarchical images are needed. We found that for local aggregation methods, the larger the area covered by Ω_p in the lowest resolution is, the better the results become. Thus, we set the number of levels N to a value so that Ω_p just covers the whole image at the lowest resolution. For the non-local aggregation method [16], the level N is empirically set to 5.

Algorithm 1 shows the steps of the proposed method.

Algorithm 1. Steps of Multi-Resolution Soft Aggregation:

Inputs: Input image I, cost volume C, robust function V, number of multi-resolutions N.

Outputs: A labeling for all pixels in I.

1. Build pyramids of guidance images $I^{(1)}, I^{(2)}, ..., I^{(N)}$ and cost volumes $\left\{C^{(1)}, C^{(2)}, ..., C^{(N)}\right\}$. Set current level, $n = N$, and set $M_l^{(N)}$ to zeros.
2. From $n = N$ to $n = 1$ iteratively perform step (a) to step (d).
   a. Compute the fused cost volume, $F_l^{(n)}$, using Eq. (3).
   b. Compute the min-convolution results, $E_l^{(n)}$, using Eq. (4), then carry out ACA on $E_l^{(n)}$ and obtain $\hat{C}^{(n)}$ as given in Eq. (5).
   c. If $n = 1$, go to step (3). Otherwise, compute $M_l^{(n-1)}$ using Eq. (2).
   d. Set $n := n - 1$.
3. Output the labeling for all pixels in I by using the WTA method on $\hat{C}^{(1)}$.

3 Applications and Experiments

3.1 Stereo Matching

We evaluated our method combined with three popular ACA techniques: CostFilter [11], MST [16], and CLMF-0 [8], using the Middlebury stereo benchmark [1]. These methods are denoted by MultiAgg (GF), MultiAgg (MST), and MultiAgg (CLMF-0) respectively. All methods are implemented in C++ on a PC with 2.0 GHz CPU and 4 GB RAM using single-core implementation. The comparison between our method and the conventional ACA methods is conducted.

For comparison, the same method [11] is employed for calculating all cost volumes. The default parameters in [11] and [16] are used for guided filtering and MST based ACA in both the conventional ACA methods and those in the MultiAgg methods. As CLMF-0 uses a different cost volume calculation method in its original work [8], the parameters for CLMF-0 and MultiAgg (CLMF-0) are tuned with care so that the best results are presented. The truncated linear function is used as the robust function: $V(l_1, l_2) = \rho \min(|l - l'|, d)$. We set $\{\rho, d\} = \left\{2 \times 10^{-4}, 5\right\}$ in MultiAgg (GF) and MultiAgg (MST) and $\{\rho, d\} = \left\{1 \times 10^{-3}, 5\right\}$ in MultiAgg (CLMF-0). Results from different methods are shown in Fig. 4. We further applied the weighted median filter based occlusion handling

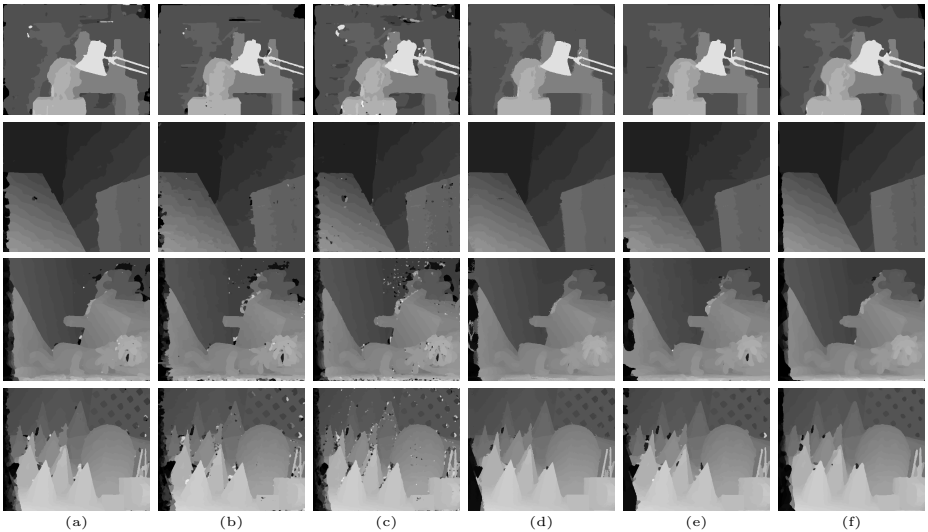


Fig. 4. Experimental results on the Middlebury datasets. (a)-(c) are results from conventional aggregation methods: (d)-(f) are results from our MultiAgg methods. The average percentage of bad pixels over four images are given at the bottom of the results. Compared with conventional aggregation methods, the MultiAgg methods achieve globally better performance. For instance, see the highly textured regions around the top right corner of the "Tsukuba" dataset and the regions around the head of the teddy bear in the "Teddy"dataset

method [11] to the results. With post-processing, the results from MultiAgg (GF) (see Fig. 6) are the best among the six methods. The quantitative evaluation is presented in Table 1 for the comparison with other methods (including global optimization based methods) on the Middlebury datasets. Table 1 and Fig. 4 show that our method outperforms the conventional ACA methods which use hard aggregation in a single resolution. Our method is also close to or even better than many iterative methods which are based on global optimization, such as [9] and [23]. For the four Middlebury datasets, the average running time (excluding occlusion handling) of CostFilter, MST, CLMF-0, MultiAgg (GF), MultiAgg (MST), MultiAgg (CLMF-0) are 1.5 s, 0.14 s, 0.42 s, 2.2 s, 0.31 s, and 0.71 s respectively. The running time of MultiAgg is about 1.5 to 2.2 times slower than corresponding ACA methods in the original resolution.

Since methods in [11] and [23] use local linear model for addressing stereo matching problem, we explicitly compared these two methods with our MultiAgg (GF) which also uses the local linear model. Table 1 shows that the ranking of our method is similar to the method in [23] (ranked 15 versus ranked 14) and both methods significantly outperform the method in [11] (ranked 35). More

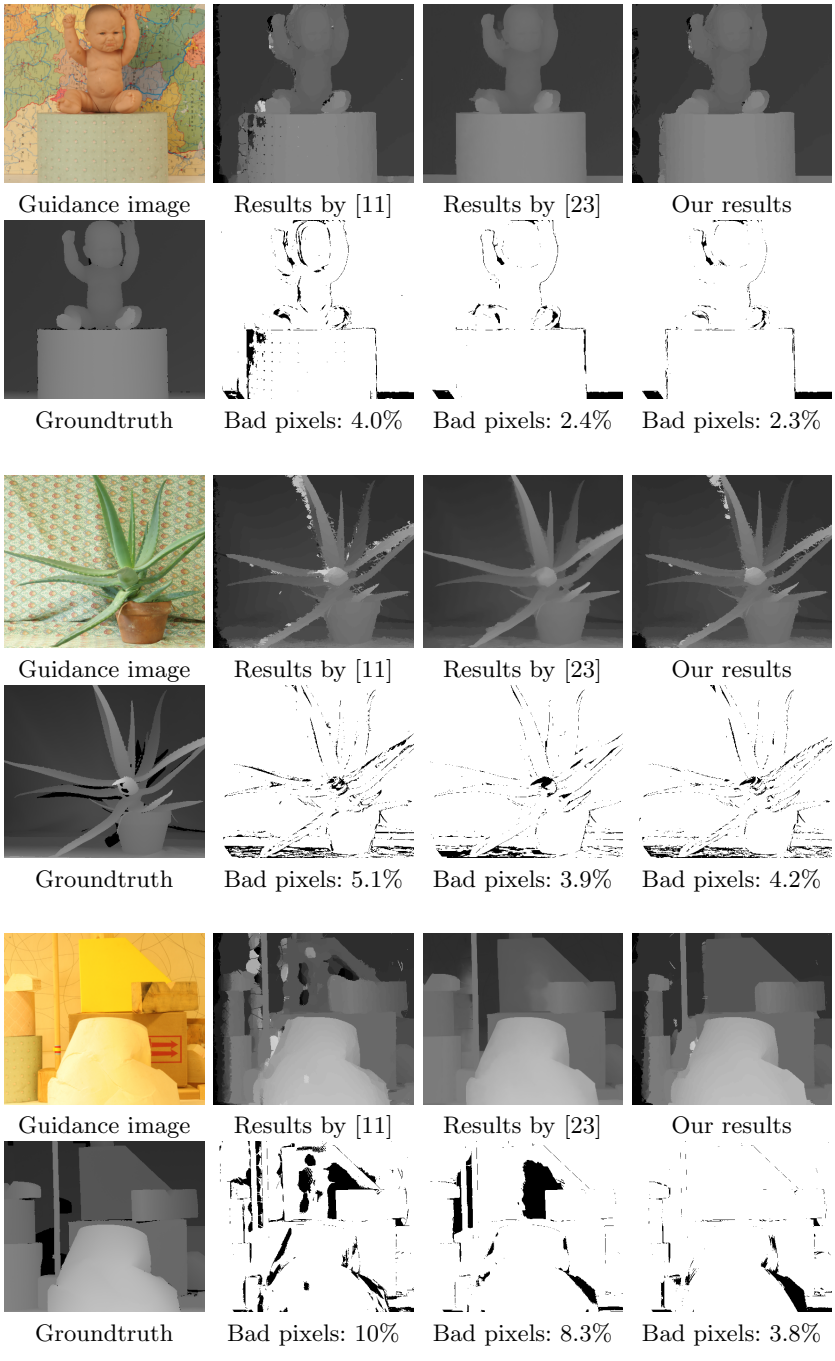


Fig. 5. Comparison among our method MultiAgg (GF), and methods in [11], and [23]. Ratio of bad pixels in non-occluded regions are given (error $>$ 1 pixel). Our method preserves details and boundaries very well while reducing large mount of incorrect labelings in textureless regions

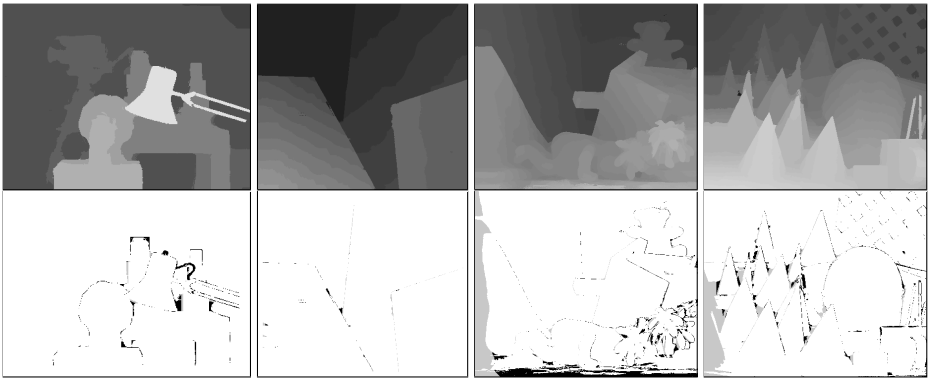


Fig. 6. Stereo matching results. First row: results from MultiAgg (GF) with occlusion handling. All results are obtained using constant parameters. Second row: error maps (error > 1 pixel). Errors in the occluded regions are colored in gray and errors in non-occluded regions are colored in black

Table 1. Evaluation on the Middlebury benchmarks

Methods	Total Rank	Average Rank	Tsukuba			Venus			Teddy			Cones		
			nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc
ADCensus [10]	1	10.9	1.07	1.48	5.73	0.09	0.25	1.15	4.10	6.22	10.9	2.42	7.25	6.95
LLR [23]	14	28.3	1.05	1.65	5.64	0.29	0.81	3.07	4.56	9.81	12.2	2.17	8.02	6.42
MultiAgg (GF)	15	30.0	1.52	1.82	8.20	0.16	0.39	2.03	5.09	10.5	13.8	2.27	7.49	6.71
PMF [9]	22	34.6	1.74	2.04	8.07	0.33	0.49	4.16	2.52	5.87	8.30	2.13	6.80	6.32
CostFilter [11]	35	42.1	1.51	1.85	7.61	0.20	0.39	2.42	6.16	11.8	16.0	2.71	8.24	7.66

comparisons on the widely used stereo datasets are given in Fig. 5 where results without occlusion handling from MultiAgg (GF) and [11] are presented.

3.2 Interactive Image Segmentation

In image segmentation, we evaluated MultiAgg (GF) and MultiAgg (MST). For comparison, we implement three other popular methods: CostFilter [11], Grabcut [12], and MST [16]. Although the non-local aggregation method in [16] is proposed typically for stereo matching, it can be naturally adopted for interactive image segmentation.

For pixels whose labels l' are given by the user we define the cost as

$$C_l(p) = \begin{cases} 0 & l = l', \\ K & \text{otherwise.} \end{cases} \tag{6}$$

where K is a very large value. To compute the cost value of pixels not labeled by the user, we build a color histogram ($8 \times 8 \times 8$ bins) for each label based on the provided strokes from users. The value in the histogram H_l is normalized so that the summation over all bins equals to 1. Then the cost of assigning label l_p to p is defined as

$$C_l(p) = (1 - H_l(B_p)) \tag{7}$$

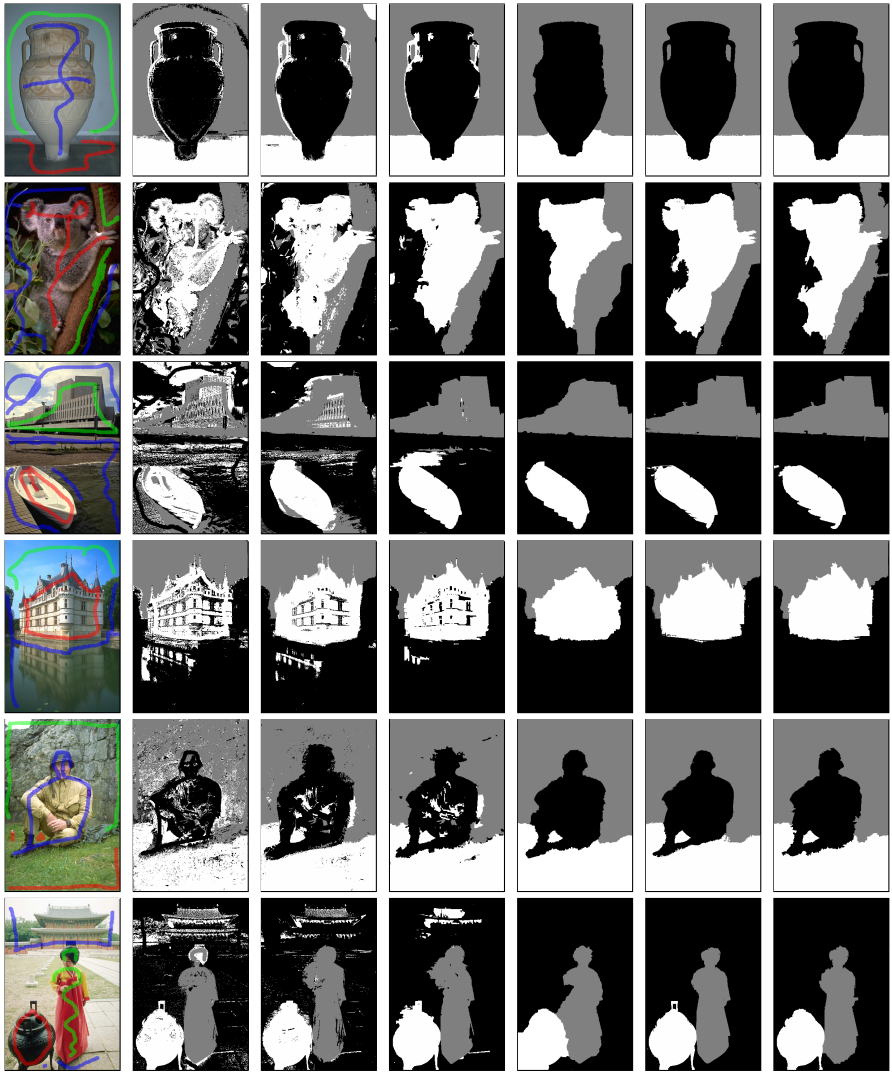


Fig. 7. Interactive segmentation results. Input images courtesy from [2]. From the left to the right are: (a) strokes by users, (b) cost volume, (c) results from CostFilter [11], (d) results from MST aggregation [16], (e) results from Graph Cuts (one iteration of [12]), (f) results from MultiAgg (GF), and (g) results from MultiAgg (MST). Note that the large amount of incorrect labeling disappear in our results and boundaries are also better preserved

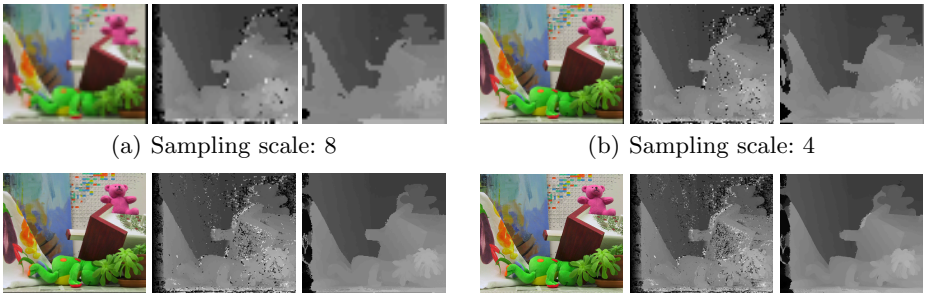
where B_p is the index of the bin where p falls into. Unlike [12], where the image is segmented only into background and foreground, our implementation segments image into multiple (three) parts. The robust function $V(l_1, l_2)$ is a Potts model function with value P if the two labels are not equal. Fig. 7 shows the results from different methods on the same cost volume. The parameters of the ACA methods in CostFilter [11] and MultiAgg (GF) are the same as those in [11]. The parameters of the ACA methods in MST [16] and MultiAgg (MST) are the same as those in [16]. We set $P = 0.25$ for MultiAgg (GF), and $P = 2.0$ for MultiAgg (MST).

Being a local aggregation method in a single resolution, method in [11] does not handle the large amount of incorrectly labeled pixels well. By using MST, method in [16] outperforms the method in [11] in textureless regions. Unfortunately it does not performs well in regions containing too much texture. GrabCut [12] does well in reducing incorrect labeling in both highly textured and textureless regions thank to the global optimization; however, its smoothness term which penalizes labeling inconsistency among 4 or 8 neighborhoods is prone to introducing boundary shrinking artifacts. Results show that our method produces the best results where the total number of incorrect labelings is minimum while boundaries are well preserved.

3.3 Discussions

The benefit of using MultiAgg is clearly demonstrated in our experiments. For a better understanding for the reason behind the good performance of MultiAgg, let us look at Fig. 8. A first observation is that the guidance image becomes smooth as the resolution goes low. This is very important for eliminating incorrect labeling in the textured regions. Since the value of a pixel in low resolution is the averaging value of all corresponding pixel in the original resolution, it is expected that regions with similar texture have a similar color in the low resolution image. As a result, pixels in highly textured regions are able to receive contributions from other pixels in regions with similar texture when carrying out ACA in low resolution. However, since pixel values of highly textured regions in original resolution change dramatically, a pixel in these regions can hardly receive contribution from other pixels even for non-local aggregation method [16]. Two other observations are as follows: (1) The low resolution cost volume contains less noises than that of high resolution. (2) Pixels become closer to each other in low resolution. In virtue of these three points, incorrect labelings caused by local characteristics can be greatly reduced in results of low resolution. The downside of carrying out ACA in low resolution is the ambiguity of boundaries and the missing of details owing to the averaging effect in the sampling process. Based on the fact that the cost function of pixels at object boundaries usually has an extremely low value at the correct label, our soft fusion strategy (in Section 2.2) takes into account low resolution results and fuse them into a new cost volume where the incorrect labelings are reduced while the details and boundaries are also well preserved.

The next question that follows is how the robust function influences the results. The robust function controls the strength of the results from low resolution



(a) Sampling scale: 8 (b) Sampling scale: 4

(c) Sampling scale: 2 (d) Original resolution

Fig. 8. Cost aggregation in multi-resolutions. From the left to the right for each subfigure are guidance images, cost volumes, and soft aggregation results. Note that the highly textured region around the head of the teddy bear in the original guidance image is smoothed in the low resolution; and ACA in low resolution produces accurate results in these regions. Despite details are lost in the low resolution results, these details are recovered in high resolution results thanks to the soft cost fusion strategy

fusing into the cost volume at the next higher resolution. When the strength of the robust function increases, the final results will therefore be biased towards the low resolution results where the labeling is smooth but details may be lost. On the other hand, weak robust function will help preserve details, but may fail to eliminate the large amount of incorrect labelings.

3.4 Limitations

Our method has a common limitation as other cost aggregation methods on the application of stereo matching – it may produce incorrect disparity values for highly slant surfaces by smoothing the values. The smoothing is caused by the fronto-parallel surface model which is implicity used in cost aggregation methods. One example can be found at the bottom of the "Teddy" dataset where the disparity values of pixels on a highly slant plane are smoothed. This artifact is unavoidable for cost aggregation methods when piecewise smoothness is forced. Slant plane based methods (e.g., [3] and [9]) would be used for finding slant plane models at the cost of using a complex iterative optimization process.

4 Conclusions

This paper has presented MultiAgg, a generic cost volume aggregation framework for effectively addressing pixel-labeling problems. The key contribution is the idea of adaptively fusing the cost aggregation results from multi-resolutions in a coarse-to-fine manner. Experimental results have shown that MultiAgg produces more accurate results than current state-of-the-art methods both visually and quantitatively. In addition to its effectiveness, another advantage is its computational efficiency which is inherited from the fast cost aggregation methods.

In future, we will explore the implementation of MultiAgg in addressing a more challenging problem of approximate nearest-neighbor field where the size of the labels space is huge.

References

1. http://www.vision.middlebury.edu/stereo/ (2013)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. PAMI 33(5), 898–916 (2011)
3. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: BMVC. vol. 11, pp. 1–11 (2011)
4. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. IJCV 70(1), 41–54 (2006)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. Theory of Computing 8(1), 415–428 (2012)
6. Hosni, A., Bleyer, M., Gelautz, M., Rhemann, C.: Local stereo matching using geodesic support weights. In: ICIP. pp. 2093–2096 (2009)
7. Lei, C., Yang, Y.H.: Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In: ICCV. pp. 1562–1569 (2009)
8. Lu, J., Shi, K., Min, D., Lin, L., Do, M.N.: Cross-based local multipoint filtering. In: CVPR. pp. 430–437 (2012)
9. Lu, J., Yang, H., Min, D., Do, M.: Patchmatch filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In: CVPR. pp. 1854–1861 (2013)
10. Mei, X., Sun, X., Zhou, M., Jiao, S., Wang, H., Zhang, X.: On building an accurate stereo matching system on graphics hardware. In: ICCV. pp. 467–474 (2011)
11. Rhemann, C., Hosni, A., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. In: CVPR. pp. 3017–3024 (2011)
12. Rother, C., Kolmogorov, V., Blake, A.: GrabCut: Interactive foreground extraction using iterated graph cuts. In: ACM Transactions on Graphics (TOG). vol. 23, pp. 309–314 (2004)
13. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. PAMI 30(6), 1068–1080 (2008)
14. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV. pp. 839–846 (1998)
15. Willsky, A.S.: Multiresolution Markov models for signal and image processing. Proceedings of the IEEE 90(8), 1396–1458 (2002)
16. Yang, Q.: A non-local cost aggregation method for stereo matching. In: CVPR. pp. 1402–1409 (2012)
17. Yang, Q.: Recursive bilateral filtering. In: ECCV, pp. 399–413. Springer (2012)
18. Yang, Q., Tan, K.H., Ahuja, N.: Real-time O(1) bilateral filtering. In: CVPR. pp. 557–564 (2009)
19. Yang, Q., Wang, L., Yang, R., Stewénius, H., Nistér, D.: Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. PAMI 31(3), 492–504 (2009)
20. Yang, R., Pollefeys, M.: Multi-resolution real-time stereo on commodity graphics hardware. In: CVPR. vol. 1, pp. I–211 (2003)
21. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. PAMI 28(4), 650–656 (2006)
22. Zhao, Y., Taubin, G.: Real-time stereo on GPGPU using progressive multi-resolution adaptive windows. Image and Vision Computing 29(6), 420–432 (2011)
23. Zhu, S., Zhang, L., Jin, H.: A locally linear regression model for boundary preserving regularization in stereo matching. In: ECCV, pp. 101–115 (2012)