Cross-Scale Cost Aggregation for Stereo Matching

Kang Zhang, Yuqiang Fang, Dongbo Min, *Senior Member, IEEE*, Lifeng Sun, *Member, IEEE*, Shiqiang Yang, *Senior Member, IEEE*, and Shuicheng Yan, *Senior Member, IEEE*

***Abstract*— This paper proposes a generic framework that enables a multiscale interaction in the cost aggregation step of stereo matching algorithms. Inspired by the formulation of image filters, we first reformulate cost aggregation from a weighted least-squares (WLS) optimization perspective and show that different cost aggregation methods essentially differ in the choices of similarity kernels. Our key motivation is that while the human stereo vision system processes information at both coarse and fine scales interactively for the correspondence search, state-of-the-art approaches aggregate costs at the finest scale of the input stereo images only, ignoring inter-consistency across multiple scales. This motivation leads us to introduce an inter-scale regularizer into the WLS optimization objective to enforce the consistency of the cost volume among the neighboring scales. The new optimization objective with the inter-scale regularization is convex, and thus, it is easily and analytically solved. Minimizing this new objective leads to the proposed framework. Since the regularization term is independent of the similarity kernel, various cost aggregation approaches, including discrete and continuous parameterization methods, can be easily integrated into the proposed framework. We show that the cross-scale framework is important as it effectively and efficiently expands state-of-the-art cost aggregation methods and leads to significant improvements, when evaluated on Middlebury, Middlebury Third, KITTI, and New Tsukuba data sets.**

***Index Terms*— Cost aggregation, local stereo matching, multiscale.**

Manuscript received January 26, 2015; revised May 10, 2015, June 23, 2015, and October 23, 2015; accepted November 29, 2015. Date of publication December 30, 2015; date of current version May 3, 2017. The work of K. Zhang, L. Sun, and S. Yang was supported in part by the National Natural Science Foundation of China under Grant 61272231, Grant 61472204, and Grant 61210008, in part by the Beijing Key Laboratory of Networked Multimedia, and in part by the Tsinghua Samsung Joint Laboratory. The work of Y. Fang and S. Yan was supported by the Singapore National Research Foundation under its International Research Centre at Singapore Funding Initiative and administered by the IDM Programme Office. The work of D. Min was supported by the Institute for Information and Communications Technology Promotion within the Ministry of Science, ICT and Future Planning through the Korean Government under Grant R0115-15-1007. This paper was recommended by Associate Editor S. Ci. *(Corresponding authors: Dongbo Min and Lifeng Sun.)*

K. Zhang, L. Sun, and S. Yang are with the Beijing Key Laboratory of Networked Multimedia, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: zk54188@gmail.com; sunlf@mail.tsinghua.edu.cn; yangshq@mail.tsinghua.edu.cn).

Y. Fang is with the National University of Defense Technology, Changsha 410073, China (e-mail: yqfang.cs@gmail.com).

D. Min is with Chungnam National University, Daejeon 305-764, Korea (e-mail: dbmin@cnu.ac.kr).

S. Yan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: eleyans@nus.edu.sg).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2513663

I. Introduction

ESTABLISHING dense correspondence between two images is one of the most important problems in computer vision [19]. When it comes to the case of using two images taken at the same scene, the dense correspondence task becomes the well-known stereo matching problem [33]. A stereo matching algorithm generally takes four steps: 1) *cost computation*; 2) *cost (support) aggregation*; 3) *disparity computation*; and 4) *disparity refinement* [33]. In the cost computation, a 3D cost volume (also known as the disparity space image [33]) is generated by computing matching costs for each pixel at all possible disparity levels. In the cost aggregation, the costs are then aggregated, which enforces *piecewise coherency* of a resultant disparity map, over the support region of each pixel. Then, disparity for each pixel is computed with local or global optimization methods and refined by various postprocessing methods in the last two steps, respectively. Among these steps, the quality of cost aggregation has a significant impact on the success of stereo algorithms. It is a key ingredient for state-of-the-art local algorithms [4], [24], [29], [44], [47] and a primary building block for some top-performing global algorithms [42], [45]. In this paper, we primarily concentrate on *cost aggregation*.

Most cost aggregation methods can be viewed as joint filtering over the cost volume [29]. Actually, even simple linear image filters such as box or Gaussian filter can be used for cost aggregation, but as isotropic diffusion filters, they tend to blur the depth boundaries [33]. To avoid such oversmoothing artifacts, a number of edge-preserving filters such as bilateral filter (BF) [39] and guided image filter [13] were introduced for cost aggregation. Yoon and Kweon [47] adopted the BF into cost aggregation, which generated appealing disparity maps on the Middlebury data set [33]. However, their method is computationally expensive due to a straightforward aggregation over a large kernel size (e.g., 35×35). To address the computational limitation of the BF, Rhemann *et al.* [29] introduced the guided image filter into cost aggregation, whose computational complexity is independent of the kernel size. Recently, Yang [44] proposed a *nonlocal* (NL) cost aggregation method, which extends the kernel size to the entire image. By computing a minimum spanning tree (MST) over the image graph, the NL cost aggregation can be performed extremely fast. Mei *et al.* [24] extended the NL cost aggregation idea by constructing the MST over the segment graph instead of the image graph, and they showed that better disparity maps are obtained.

All these state-of-the-art cost aggregation methods have made great contributions to stereo vision. A common property

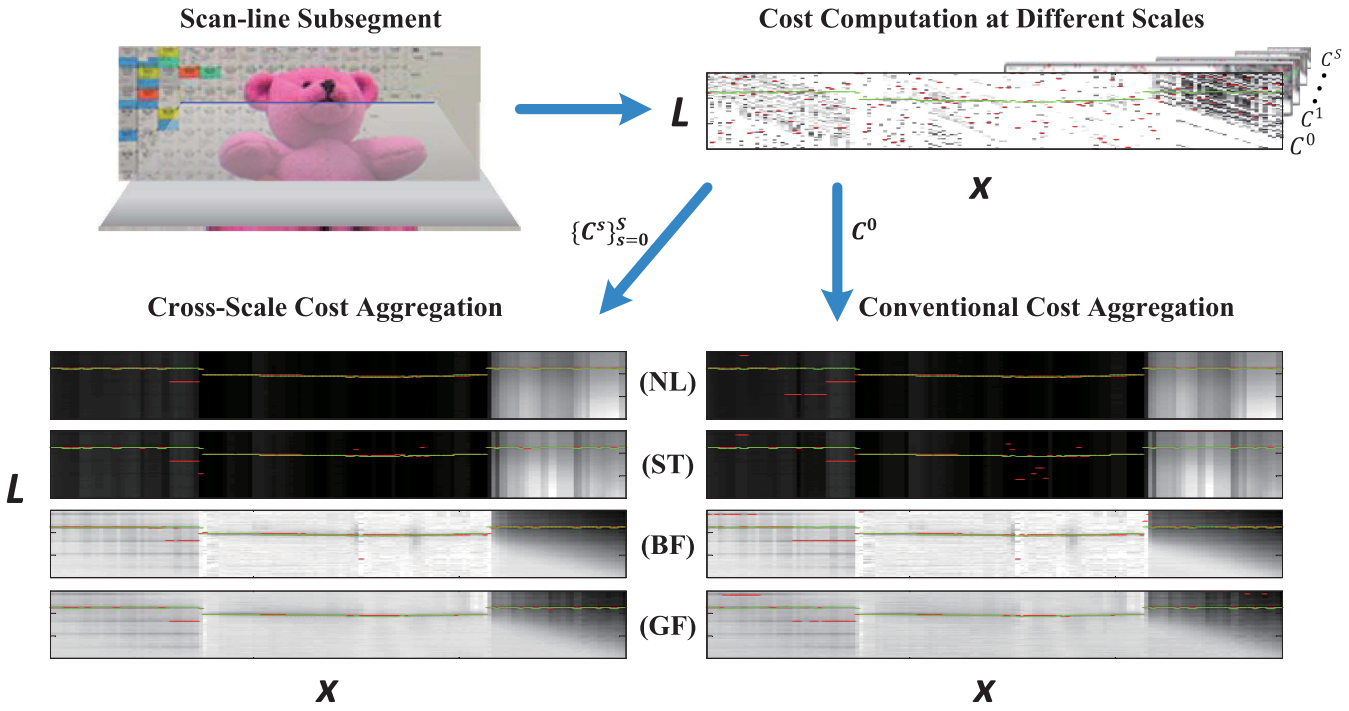


Fig. 1. Cross-scale cost aggregation. Top-left: enlarged view of a scan-line subsegment from Middlebury [33] *Teddy* stereo pair. Top-right: cost volumes ($\{\mathbf{C}^s\}_{s=0}^S$) after cost computation at different scales, where the *intensity* + *gradient* cost function is adopted as in [24], [29], and [44]. The horizontal axis x indicates different pixels along the subsegment, and the vertical axis L represents different disparity labels. The red line indicates disparity generated by current cost volume while the green line is the ground truth. Bottom-right: cost volumes after applying different cost aggregation methods at the finest scale (from top to bottom: NL [44], ST [24], BF [47], and GF [29]). Bottom-left: cost volumes after integrating different methods into our cross-scale cost aggregation framework, where cost volumes at different scales are adopted for aggregation. (Best viewed in color.)

of these methods is that costs are aggregated at the finest scale of the input stereo images. However, human beings generally process stereoscopic correspondence across multiple scales [22], [23], [25]. According to [22], information at coarse and fine scales is processed interactively in the correspondence search of the human stereo vision system. Thus, from this bioinspiration, it is reasonable that costs should be aggregated across multiple scales rather than the finest scale as done in conventional methods (Fig. 1).

In this paper, a general cross-scale cost aggregation framework is proposed. First, inspired by the formulation of image filters in [26], we show that various cost aggregation methods can be uniformly formulated as weighted least-squares (WLS) optimization problems. Then, from this unified optimization perspective, by adding a generalized Tikhonov regularizer into the WLS optimization objective, we enforce the consistency of the cost volume among the neighboring scales, i.e., inter-scale consistency. The new optimization objective with inter-scale regularization is convex and can be easily and analytically solved. As conventional cost aggregation methods can preserve the intra-scale consistency of the cost volume, many of them can be integrated into our framework to generate more robust cost volumes and better disparity maps.

Fig. 1 shows the effect of the proposed framework. Slices of the cost volumes of four representative cost aggregation methods, including the NL method [44], the segment tree (ST) method [24], the BF method [47], and the guided filter (GF) method [29], are visualized. We use red dots to denote disparities generated by local winner-take-all (WTA) optimization in each cost volume and green dots to denote ground truth disparities. It can be found that more robust cost volumes and more accurate disparities are produced by adopting cross-scale cost aggregation. Extensive experiments on Middlebury [33], Middlebury Third [30], KITTI [9], and New Tsukuba [28] data sets also reveal that better disparity maps can be obtained using cross-scale cost aggregation.

We extend our preliminary work [49] by applying our framework into the patch match (PM) stereo algorithm [4], and then evaluate this new method (named S + PM) on various stereo benchmarks. The PM stereo method adopts continuous parameterization for the solution space, unlike existing cost aggregation approaches based on the piecewise constant assumption, thus alleviating staircase artifacts on slanted surfaces. The disparity labeling task is defined with continuous plane parameters, and the randomized search algorithm, introduced by the original PM algorithm [3], is employed to efficiently traverse such an infinite continuous solution space. In short, the PM stereo performs the cost aggregation (e.g., using the BF) on only a subset of label candidates, while conventional local stereo approaches should perform the cost aggregation on all (discretized) label candidates. It should be noted that in the PM stereo [4], the subset of label candidates varies for each pixel. We will show that our framework is also applicable to the PM stereo method with spatially varying, partial label hypotheses, and substantially improves its disparity accuracy.

In summary, the contributions of this paper are threefold:

1) a unified WLS formulation of various cost aggregation methods, including discrete and continuous parameterization methods, from an optimization perspective;
2) a novel and effective cross-scale cost aggregation framework;
3) quantitative evaluation of representative cost aggregation methods on four data sets.

The remainder of this paper is organized as follows. In Section II, we summarize the related work. The WLS formulation for cost aggregation is given in Section III. Our inter-scale regularization is described in Section IV. Then we detail the implementation of our framework in Section V and the formulation of cross-scale PM stereo is also shown in this section. Finally, the experimental results and analyses are presented in Section VI and the conclusive remarks are made in Section VII.

II. Related Work

Recent surveys [11], [16], [40] give sufficient comparison and analysis for various cost aggregation methods. We refer the reader to these surveys to get an overview of different cost aggregation methods and we will focus on stereo matching methods involving multiscale information, which are very relevant to our idea but have substantial differences.

Early researchers of stereo vision adopted the coarse-to-fine (CTF) strategy for stereo matching [23]. Disparity of a coarse resolution was assigned first, and coarser disparity was used to reduce the search space for calculating finer disparity. This CTF (hierarchical) strategy has been widely used in global stereo methods. Van Meerbergen *et al.* [41] adopted the CTF strategy in dynamic programming, where the disparity map of a coarser scale is used as offset disparity map at a finer scale. Hermann and Klette [14] proposed to calculate a disparity map from half-resolution images and used this disparity map to restrict the disparity search space for full-resolution semiglobal stereo matching. Felzenszwalb and Huttenlocher [7] used the CTF strategy to reduce the number of message passing iterations in belief propagation and Yang *et al.* [45] adopted the same belief propagation approach in their algorithm. Other global methods like simulated annealing [6] and partial-differential-equation-based approach [2] also utilize the CTF strategy for the purpose of accelerating convergence and avoiding unexpected local minima.

Not only global methods but also local methods adopt the CTF strategy. Yang and Pollefeys [46] improved traditional sum-of-square-differences (SSD) dissimilarity measure by combining SSD measurements for windows of different sizes, which can achieve real-time performance on GPU. Hu *et al.* [17] proposed to reduce the search space of local stereo matching by introducing a candidate set from disparities of neighboring pixels of the corresponding coarser scale pixel. Jen *et al.* [18] introduced an adaptive scale selection mechanism by convolving the surface prior image with a Laplacian of Gaussian kernel. The scale selection results helped to determine the starting scale level for CTF approach. Magarey and Dick [21] adopted the CTF framework based on the complex discrete wavelet transform. Sizintsev [35] proposed to perform CTF stereo matching in a generalization of the Laplacian pyramid to solve the problem of poor recovery of thin structures—a common drawback of CTF approach [36]. Tang *et al.* [38] proposed a robust multiscale stereo matching algorithm to handle fundus images with radiometric differences. They invented the multiscale pixel feature vector and performed matching in the neighboring scales to generate a disparity map in each scale. The main purpose of adopting the CTF strategy in local stereo methods is to reduce the search space [17], [18] or take the advantage of multiscale related image representations [35], [38], [46]. However, there is one exception in local CTF approaches. Min and Sohn [27] modeled the cost aggregation by anisotropic diffusion and solved the proposed variational model efficiently by the multiscale approach. The motivation of their model is to denoise the cost volume which is very similar to our model, but our method enforces the inter-scale consistency of cost volumes by regularization.

Overall, most CTF approaches share a similar property. They explicitly or implicitly model the disparity evolution process in the scale space [38], i.e., *disparity consistency* across multiple scales. Different from previous CTF methods, our method models the evolution of the cost volume in the scale space, i.e., *cost volume consistency* across multiple scales. From the optimization perspective, CTF approaches narrow down the solution space by considering a subset of the disparity search range in different scales, while our method does not alter the solution space but adds inter-scale regularization into the optimization objective. Thus, incorporating multiscale prior by regularization is the originality of our approach. Another point worthy of mentioning is that local CTF approaches perform no better than state-of-the-art cost aggregation methods [17], [18], while our method shows significant improvements over those cost aggregation methods [24], [29], [44].

After publication of the conference version of [49], Tan *et al.* [37] proposed a multiscale cost aggregation approach which is conceptually similar to ours. A key difference is that they employ a soft fusion scheme based on a min convolution [8], which iteratively aggregates costs from different scales. It is worth noting that their computational complexity is larger than ours due to the min convolution operator.

III. Cost Aggregation as Optimization

In this section, we show that the cost aggregation can be formulated as a WLS optimization problem. Under this formulation, different choices of similarity kernels [26] in the optimization objective lead to different cost aggregation methods.

First, the cost computation step is formulated as a function $f: \mathbb{R}^{W\times H\times 3} \times \mathbb{R}^{W\times H\times 3} \mapsto \mathbb{R}^{W\times H\times L}$, where W and H are the width and height of input images, 3 represents color channels, and L denotes the number of disparity levels. Thus, for a stereo color pair: $\mathbf{I}, \mathbf{I}' \in \mathbb{R}^{W\times H\times 3}$, by applying cost computation

$$\mathbf{C} = f(\mathbf{I}, \mathbf{I}') \tag{1}$$

we can get the cost volume $\mathbf{C} \in \mathbb{R}^{W\times H\times L}$, which represents matching costs for each pixel at all possible disparity levels. For a single pixel $i = (x_i, y_i)$, where x_i and y_i are pixel locations, its cost at the disparity level l can be denoted as a scalar, $\mathbf{C}(i, l)$. Various methods can be used to compute the cost volume. For example, the *intensity + gradient* cost function [24], [29], [44] can be formulated as

$$\begin{aligned}\mathbf{C}(i,l) = &(1-\alpha)\cdot\min(\|\mathbf{I}(i)-\mathbf{I}'(i_l)\|,\tau_1)\\ &+\alpha\cdot\min(\|\nabla_x\mathbf{I}(i)-\nabla_x\mathbf{I}'(i_l)\|,\tau_2).\end{aligned} \tag{2}$$

Here, $\mathbf{I}(i)$ denotes the color vector of the pixel i. ∇_x is the grayscale gradient in the x-direction. i_l is the corresponding pixel of i with a disparity l, i.e., $i_l = (x_i - l, y_i)$. α balances the color and gradient terms and τ_1 and τ_2 are truncation values.

The cost volume $\mathbf{C}$ is typically very noisy (Fig. 1). Inspired by the WLS formulation of the denoising problem [26], the cost aggregation can be formulated with the noisy input $\mathbf{C}$ as

$$\tilde{\mathbf{C}}(i,l) = \arg\min_z \frac{1}{Z_i}\sum_{j\in N_i} K(i,j)\|z-\mathbf{C}(j,l)\|^2 \tag{3}$$

where N_i defines a neighboring system of i. $K(i, j)$ is the similarity kernel [26], which measures the similarity between pixels i and j, and $\tilde{\mathbf{C}}$ is the (denoised) cost volume. $Z_i = \sum_{j\in N_i} K(i, j)$ is a normalization constant. The solution of this WLS problem is

$$\tilde{\mathbf{C}}(i,l) = \frac{1}{Z_i}\sum_{j\in N_i} K(i,j)\mathbf{C}(j,l). \tag{4}$$

Thus, like image filters [26], a cost aggregation method corresponds to a particular instance of the similarity kernel. For example, the BF method [47] adopted the spatial and photometric distances between two pixels to measure the similarity, which is the same as the kernel function used in the BF [39]. Rhemann *et al.* [29] (GF) adopted the kernel defined in the GF [13], whose computational complexity is independent of the kernel size. The NL method [44] defined a kernel based on a geodesic distance between two pixels in a tree structure. This approach was further enhanced by making use of color segments, called an ST approach [24]. A major difference between filter-based [29], [47] and tree-based [24], [44] aggregation approaches is the action scope of the similarity kernel, i.e., N_i in (4). In filter-based methods, N_i is a local window centered at i, but in tree-based methods, N_i is a whole image. Fig. 1 visualizes the effect of different action scopes. The filter-based methods hold some local similarity after the cost aggregation, while tree-based methods tend to produce hard edges between different regions in the cost volume.

After showing that representative cost aggregation methods can be formulated within a unified framework, let us recheck the cost volume slices in Fig. 1. The slice, coming from *Teddy* stereo pair in the Middlebury data set [34], consists of three typical scenarios: 1) low-texture; 2) high-texture; and 3) near textureless regions (from left to right). The four state-of-the-art cost aggregation methods all perform very well

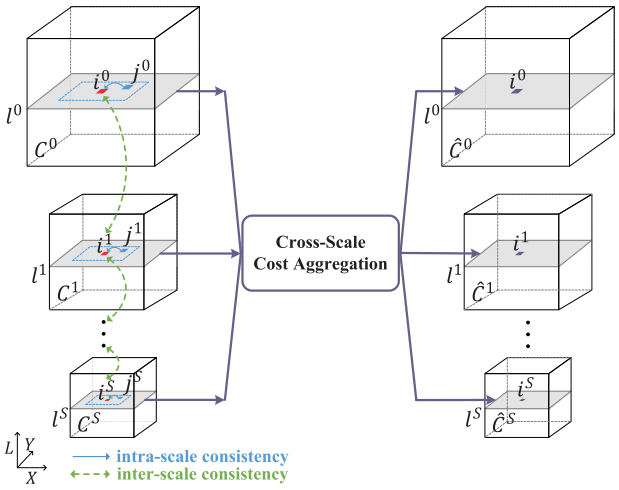


Fig. 2. Flowchart of cross-scale cost aggregation. $\{\hat{\mathbf{C}}^s\}_{s=0}^S$ is obtained by utilizing a set of input cost volumes, $\{\mathbf{C}^s\}_{s=0}^S$, together. Corresponding variables $\{i^s\}_{s=0}^S$, $\{j^s\}_{s=0}^S$, and $\{l^s\}_{s=0}^S$ are visualized. The blue arrow represents an *intra-scale* consistency (commonly used in the conventional cost aggregation approaches), while the green dashed arrow denotes an *inter-scale* consistency. (Best viewed in color.)

in the high-texture area, but most of them fail in either low-texture or near textureless regions. To yield highly accurate correspondence in those low-texture and near textureless regions, the correspondence search should be performed at the coarse scale [25]. However, under the formulation of (3), costs are always aggregated at the finest scale, making it impossible to adaptively utilize information from multiple scales. Hence, we need to reformulate the WLS optimization objective from the scale space perspective.

IV. Cross-Scale Cost Aggregation Framework

It is obvious that directly using (3) to tackle multiscale cost volumes is equivalent to performing cost aggregation at each scale separately. First, we add a superscript s to $\mathbf{C}$, and denote the cost volumes at different scales of a stereo pair as $\mathbf{C}^s$, where $s \in \{0, 1, \ldots, S\}$ is the scale parameter. $\mathbf{C}^0$ represents the cost volume at the finest scale. The multiscale cost volume $\mathbf{C}^s$ is computed using the downsampled images with a factor of η^s. Note that this approach also reduces the search range of the disparity. The multiscale version of (3) can be easily expressed as

$$\tilde{\mathbf{v}} = \arg\min_{\{z^s\}_{s=0}^S}\sum_{s=0}^{S}\frac{1}{Z^s_{i^s}}\sum_{j^s\in N_{i^s}} K(i^s,j^s)\|z^s-\mathbf{C}^s(j^s,l^s)\|^2. \tag{5}$$

Here, $Z^s_{i^s} = \sum_{j^s\in N_{i^s}} K(i^s, j^s)$ is a normalization constant. $\{i^s\}_{s=0}^S$ and $\{l^s\}_{s=0}^S$ denote a sequence of corresponding variables at each scale (Fig. 2), where i^s denotes a single pixel at scale s and l^s represents the disparity level. The relationships of these variables across different scales are $i^{s+1} = i^s/\eta$ and $l^{s+1} = l^s/\eta$.

N_{i^s} is a set of neighboring pixels at the sth scale. In our work, the size of N_{i^s} remains the same for all the scales to enforce more smoothing at the coarser scale.

We use the vector $\tilde{\mathbf{v}} = [\tilde{\mathbf{C}}^0(i^0, l^0), \tilde{\mathbf{C}}^1(i^1, l^1), \ldots, \tilde{\mathbf{C}}^S(i^S, l^S)]^T$ with $S + 1$ components to denote the aggregated

cost at each scale. The solution of (5) is obtained by performing cost aggregation at each scale independently as follows:

$$\forall s, \tilde{\mathbf{C}}^s(i^s, l^s) = \frac{1}{Z^s_{i^s}} \sum_{j^s \in N_{i^s}} K(i^s, j^s)\mathbf{C}^s(j^s, l^s). \tag{6}$$

Previous CTF approaches typically constrain the disparity search space at the current scale using a disparity map estimated from the cost volume at the coarser scale, which often provokes the loss of small disparity details. Alternatively, we directly enforce the inter-scale consistency on the cost volume by adding a generalized Tikhonov regularizer into (5), leading to the following optimization objective:

$$\hat{\mathbf{v}} = \arg\min_{\{z^s\}_{s=0}^S} \left(\sum_{s=0}^{S} \frac{1}{Z^s_{i^s}} \sum_{j^s \in N_{i^s}} K(i^s, j^s)\|z^s - \mathbf{C}^s(j^s, l^s)\|^2 + \lambda \sum_{s=1}^{S} \|z^s - z^{s-1}\|^2 \right) \tag{7}$$

where λ is a constant parameter to control the amount of regularization. Similar to $\tilde{\mathbf{v}}$, the vector $\hat{\mathbf{v}} = [\hat{\mathbf{C}}^0(i^0, l^0), \hat{\mathbf{C}}^1(i^1, l^1), \ldots, \hat{\mathbf{C}}^S(i^S, l^S)]^T$ also has $S + 1$ components to denote the costs at each scale. The above optimization problem is convex. Hence, we can get the solution by finding the stationary point of the optimization objective. Let $F(\{z^s\}_{s=0}^S)$ represent the optimization objective in (7). For $s \in \{1, 2, \ldots, S-1\}$, the partial derivative of F with respect to z^s is

$$\begin{aligned} \frac{\partial F}{\partial z^s} &= \frac{2}{Z^s_{i^s}} \sum_{j^s \in N_{i^s}} K(i^s, j^s)(z^s - \mathbf{C}^s(j^s, l^s)) \\ &\quad + 2\lambda(z^s - z^{s-1}) - 2\lambda(z^{s+1} - z^s) \\ &= 2(-\lambda z^{s-1} + (1 + 2\lambda)z^s - \lambda z^{s+1} - \tilde{\mathbf{C}}^s(i^s, l^s)). \end{aligned} \tag{8}$$

Setting $(\partial F/\partial z^s) = 0$ and using $\sum_{j^s \in N_{i^s}} K(i^s, j^s) = Z^s_{i^s}$, we get

$$-\lambda z^{s-1} + (1 + 2\lambda)z^s - \lambda z^{s+1} = \tilde{\mathbf{C}}^s(i^s, l^s). \tag{9}$$

It is easy to get similar equations for $s = 0$ and $s = S$. Thus, we have $S + 1$ linear equations in total, which can be expressed concisely as

$$A\hat{\mathbf{v}} = \tilde{\mathbf{v}}. \tag{10}$$

The matrix A is an $(S + 1) \times (S + 1)$ tridiagonal constant matrix, which can be easily derived from (9). Since A is tridiagonal, its inverse always exists. Thus

$$\hat{\mathbf{v}} = A^{-1}\tilde{\mathbf{v}}. \tag{11}$$

The final cost volume is obtained through the adaptive combination of the results of cost aggregation performed at different scales. Such an adaptive combination enables the multiscale interaction of the cost aggregation in the context of optimization.

Fig. 3 shows the effect of inter-scale regularization. In this example, without cross-scale cost aggregation, there are many local minima in the cost vector, yielding erroneous disparity. Information from the finest scale is not enough to estimate an accurate disparity, but when the inter-scale regularization is adopted, useful information from coarse scales reshapes the cost vector, generating disparity closer to the ground truth.

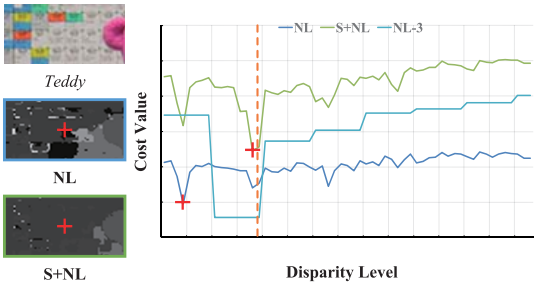


Fig. 3. Effect of inter-scale regularization. Right: we visualize three cost vectors (one in coarse scale) of a single pixel [pixel location (295, 49)] of *Teddy* stereo pair. The blue line denotes the cost vector computed by NL method [44]. The green line is the cost vector after applying cross-scale cost aggregation (S + NL). The cyan line is the cost vector of NL in the fourth ($S = 3$) scale, which is interpolated to have a size equal to that of the finest scale cost vectors. The red cross represents the minimal cost location for each cost vector and the vertical dashed line denotes the ground truth disparity. Left: image and disparity patches centering on this pixel are shown. (Best viewed in color.)

Algorithm 1 Pseudocode for Cross-Scale Cost Aggregation

Input: Stereo Color Image $\mathbf{I}$, $\mathbf{I}'$.

1) Build Gaussian pyramid of input $\mathbf{I}^s$, $\mathbf{I}'^s$, $s \in \{0, 1, \ldots, S\}$.
2) Generate initial cost volume $\mathbf{C}^s$ for each scale by cost computation according to Equation (1).
3) Aggregate costs at each scale separately according to Equation (6) to get cost volume $\tilde{\mathbf{C}}^s$.
4) Aggregate costs across multiple scales according to Equation (11) to get final cost volume $\hat{\mathbf{C}}^s$.

Output: Robust cost volume: $\hat{\mathbf{C}}^0$.

V. Implementation and Complexity

A. Scale Space Implementation

To build cost volumes for different scales (Fig. 2), we need to extract stereo image pairs at different scales. In our implementation, we choose the Gaussian pyramid [5], which is a classical representation in the scale space theory. The Gaussian pyramid is obtained by successive smoothing and subsampling ($\eta = 2$). One advantage of this representation is that the image size decreases exponentially as the scale level increases, and thus the computational cost of cost aggregation is reduced at the coarser scale exponentially.

One may want to adopt other multiscale representations to build cost volumes for different scales. During our experiments, we also test the nonlinear scale space utilized in [1]. The quality gain of the nonlinear scale space [1] over the linear Gaussian pyramid is, however, less than 1%, while it is much slower than the linear Gaussian pyramid. Thus, we choose to use the Gaussian pyramid for scale space representation, when considering both the computational complexity and accuracy improvement.

B. Computational Complexity

The basic workflow of the cross-scale cost aggregation is shown in Algorithm 1, in which we can utilize any existing cost aggregation method at Step 3. Note that the complexity

analysis explained here is for the methods that employ the discrete front-parallel formulation for performing cost aggregation, e.g., NL, ST, BF, or GF. The continuous approach assuming a slanted surface, e.g., cross-scale PM stereo, which will be explained in Section V-C, may have a different complexity due to practical implementation issues like the memory requirement.

The computational complexity of our algorithm using the discrete front-parallel formulation just increases by a small constant factor, compared with conventional cost aggregation methods. Specifically, let us denote the computational complexity of conventional cost aggregation methods as $O(m\mathrm{WHL})$, where m differs for different cost aggregation methods. The number of pixels and disparities at the scale s is $\lfloor(\mathrm{WH}/4^s)\rfloor$ and $\lfloor(L/2^s)\rfloor$, respectively. Thus, the computational complexity of Step 3 increases at most by $(1/7)$, compared with conventional cost aggregation methods, as explained in the following:

$$\sum_{s=0}^{S}\left(m\left\lfloor\frac{\mathrm{WHL}}{8^s}\right\rfloor\right)\leq\lim_{S\to\infty}\left(\sum_{s=0}^{S}\frac{m\mathrm{WHL}}{8^s}\right)=\frac{8}{7}m\mathrm{WHL}. \quad (12)$$

Step 4 involves the inversion of the matrix A with a size of $(S+1)\times(S+1)$, but A is a spatially invariant matrix, with each row consisting of at most three nonzero elements, and thus its inverse can be precomputed. Also, in (11), the cost volume at the finest scale, $\hat{\mathbf{C}}^0(i^0,l^0)$, is used to yield a final disparity map, and thus we need to compute only

$$\hat{\mathbf{C}}^0(i^0,l^0)=\sum_{s=0}^{S}A^{-1}(0,s)\tilde{\mathbf{C}}^s(i^s,l^s) \quad (13)$$

and not $\hat{\mathbf{v}}=A^{-1}\tilde{\mathbf{v}}$. This cost aggregation across multiple scales requires only a small amount of extra computational load. In the following section, we will analyze the runtime efficiency of our method in more detail.

C. Cross-Scale Patch Match Stereo Algorithm

So far, we have studied the cross-scale cost aggregation using the discrete front-parallel formulation. As mentioned in Section I, we can also integrate the PM stereo method [4] with continuous plane parameters into our framework. The key difference in employing discrete front-parallel and continuous slanted surfaces lies in the formulation of cost computation and aggregation.

For conciseness, we just show the continuous version of (2) and (4). Let us denote the continuous plane parameter as $f=(a,b,c)$. We first build a cost volume $\mathbf{C}(i,f)$ using $\mathbf{I}(i)$, $\nabla_x\mathbf{I}(i)$, $\mathbf{I}'(i_f)$, and $\nabla_x\mathbf{I}'(i_f)$ in a way similar to (2). Here, $i_f=(x_i-(ax_i+by_i+c),y_i)$ is the corresponding pixel of i in the right view. Since the x-coordinate of i_f lies in the continuous domain, the original PM stereo method [4] calculates $\mathbf{I}'(i_f)$ and $\nabla_x\mathbf{I}'(i_f)$ with a linear interpolation. The aggregated cost $\tilde{\mathbf{C}}(i,f)$ is then obtained through the single-scale cost aggregation using (4). The similarity kernel is defined with a bilateral kernel $K(i,j)=e^{-((\|\mathbf{I}(i)-\mathbf{I}(j)\|)/\gamma)}$ as in [4].

The extension into the multiscale version is similar to the previous derivation, so we omit its details here. There is, however, one implementation issue due to the tradeoff between the memory requirement and computational complexity. To be specific, in the conventional discrete approaches using the front-parallel assumption, when the disparity $l^0=l$ is given at the finest scale, a set of *2D cost slices* $\tilde{\mathbf{C}}^s(i^s,l^s)$ for all i^s $(s=1\sim S-1)$ can be reused to compute the final cost slice $\hat{\mathbf{C}}^0(i^0,l^0)$ at the finest scale. Contrarily, the PM stereo algorithm evaluates the cost volume $\tilde{\mathbf{C}}(i,f)$ only on the subset consisting of partial disparity hypotheses f, and this subset varies for each pixel i. Thus, the cross-scale PM stereo algorithm requires saving a set of *3D cost volumes* $\tilde{\mathbf{C}}^s(i^s,f^s)$ for all i^s and the partial candidates f^s evaluated $(s=0\sim S-1)$, but it is too memory intensive. In our implementation, we hence decide to recalculate $\tilde{\mathbf{C}}^s(i^s,f^s)$, whenever the pixel i^s and the label f^s are reached at each scale s. Therefore, when the computational complexity of single-scale PM stereo is $O(m\mathrm{WH}\log(L))$ [20], the complexity of cross-scale PM stereo becomes $O(m(S+1)\mathrm{WH}\log(L))$ in our implementation. Actually, this issue is related to a tradeoff between memory requirement and computational complexity. A better design choice would be possible, but we reserve this for future work. In the following section, we will show that the cross-scale PM stereo method substantially improves the disparity accuracy on three data sets. Our code of the PM stereo and cross-scale PM stereo methods is publicly available.[1]

VI. Experimental Result and Analysis

In this section, we use Middlebury [33], KITTI [9], and New Tsukuba [28] data sets to validate that when integrating state-of-the-art cost aggregation methods, such as BF [47], GF [29], NL [44], ST [24], and their continuous counterpart PM stereo method [4], into our framework, there will be significant performance improvements. Furthermore, we also implement the simple box filter aggregation method (named BOX, window size is 7×7) to serve as a baseline, which also becomes very powerful when integrated into our framework. For NL and ST, we directly use the C++ codes provided by the authors,[2,3] and thus all the parameter settings are identical as those used in their implementations. For GF, we implement our own C++ code by referring to the author-provided software (implemented in MATLAB)[4] in order to process high-resolution images from KITTI and New Tsukuba data sets efficiently. For BF, we implement the asymmetric version as suggested by [16]. For the PM stereo method, we implement our own C++ code by referring to [4]. The local WTA strategy is adopted to generate a disparity map. In order to compare different cost aggregation methods fairly, no disparity refinement technique is employed, unless we explicitly declare. S is set to 4, i.e., totally five scales are used in our framework. For the regularization parameter λ,

[1] https://github.com/rookiepig/CrossScalePatchMatch
[2] http://www.cs.cityu.edu.hk/~qiyang/publications/cvpr-12/code/
[3] http://xing-mei.net/resource/page/segment-tree.html
[4] https://www.ims.tuwien.ac.at/publications/tuw-202088

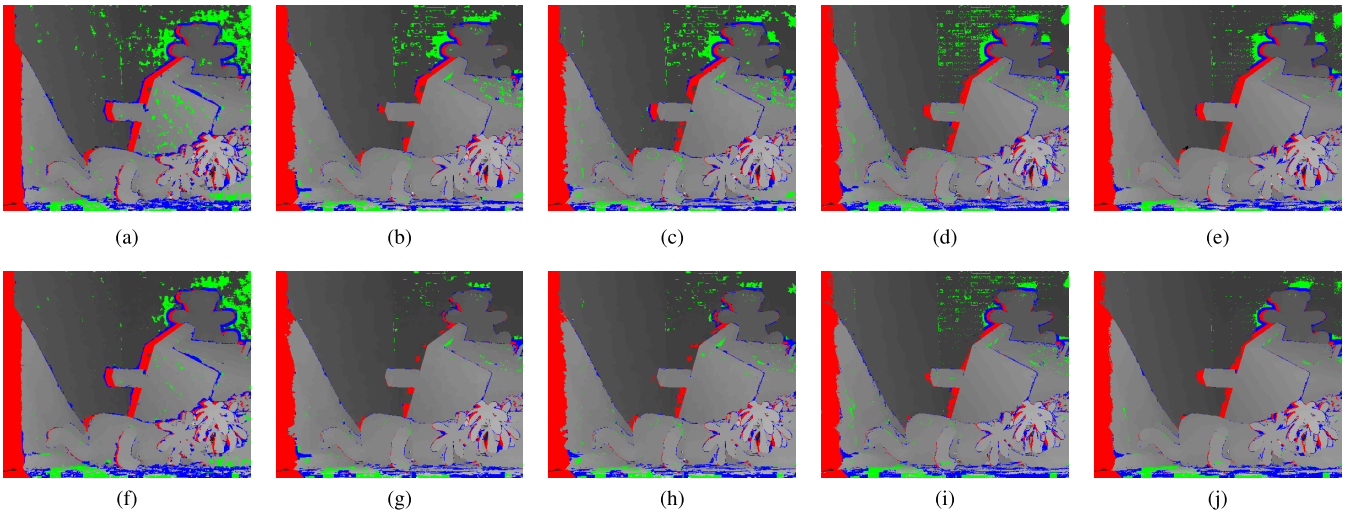


Fig. 4. Disparity maps of *Teddy* for all cost aggregation methods (with no disparity refinement techniques). The *non-occ* error rate is shown in each subtitle, where the absolute disparity error is larger than 1. Red indicates *all* error. Green indicates *non-occ* error and blue indicates *disc* error. (Best viewed in color.) (a) BOX (14.23%). (b) NL (8.60%). (c) ST (9.78%). (d) BF (10.24%). (e) GF (8.25%). (f) S + BOX (11.18%). (g) S + NL (5.74%). (h) S + ST (6.22%). (i) S + BF (8.17%). (j) S + GF (6.99%).

we set it to 0.3 for the Middlebury data set, and 1.0 on the KITTI and New Tsukuba data sets for more regularization, considering these two data sets contain a large portion of textureless regions.

A. Middlebury Dataset

The Middlebury benchmark [34] is a *de facto* standard for comparing existing stereo matching algorithms. In the benchmark [34], four stereo pairs (*Tsukuba*, *Venus*, *Teddy*, and *Cones*) are used to rank more than 100 stereo matching algorithms. In our experiment, we adopt these four stereo pairs. In addition, we use Middlebury 2005 [32] (6 stereo pairs) and Middlebury 2006 [15] (21 stereo pairs) data sets, which involve more complex scenes. Thus, we have 31 stereo pairs in total, denoted as *M31*. It is worth mentioning that during our experiments, all local cost aggregation methods perform rather badly [error rate of nonocclusion (*non-occ*) area is more than 20%] in four stereo pairs from the Middlebury 2006 data set, i.e., *Midd1*, *Midd2*, *Monopoly*, and *Plastic*. A common property of these four stereo pairs is that they all contain large textureless regions, making local stereo methods fragile. In order to alleviate bias toward these four stereo pairs, we exclude them from *M31* to generate another collection of stereo pairs, which we call *M27*. We evaluate all methods on both *M31* and *M27* (Table I). We adopt the *intensity* + *gradient* cost function in (2), which is widely used in state-of-the-art cost aggregation methods [24], [29], [44].

In Table I, we show the average error rates of *non-occ* regions for different discrete cost aggregation methods on both *M31* and *M27* data sets. We use the prefix **S+** to denote the integration of existing cost aggregation methods into the cross-scale cost aggregation framework. Avg Non-occ is an average percentage of bad matching pixels in *non-occ* regions, where the absolute disparity error is larger than 1. The results are encouraging: all cost aggregation methods see an improvement when using cross-scale cost aggregation, and even the simple BOX method becomes very powerful (comparable with state of the art on *M27*) when using cross-scale cost aggregation. Disparity maps of *Teddy* stereo pair for all these methods are shown in Fig. 4, while others are shown in the supplementary material due to space limit.

TABLE I

QUANTITATIVE EVALUATION OF COST AGGREGATION METHODS ON MIDDLEBURY DATASET. PREFIX S+ DENOTES OUR CROSS-SCALE COST AGGREGATION FRAMEWORK. FOR RANK PART (COLUMN 4∼5), DISPARITY RESULTS WERE REFINED WITH THE SAME DISPARITY REFINEMENT TECHNIQUE [44]

Method	Avg Non-occ(%)		Avg	Avg	Time
	M31	*M27*	Rank	Err(%)	(s)
BOX	15.45	10.7	59.6	6.2	0.11
S+BOX	**13.09**	**8.55**	**51.9**	**5.93**	0.15
NL [44]	12.22	9.44	41.2	5.48	0.29
S+NL	**11.49**	**8.73**	**39.4**	**5.2**	0.37
ST [24]	11.52	8.95	31.6	5.35	0.2
S+ST	**10.51**	**8.07**	**27.9**	**4.97**	0.29
BF [47]	12.26	8.77	48.1	5.89	60.53
S+BF	**10.95**	**8.04**	**40.7**	**5.56**	70.62
GF [29]	10.5	6.84	40.5	5.64	1.16
S+GF	**9.39**	**6.20**	**37.7**	**5.51**	1.32

Furthermore, to follow the standard evaluation metric of the Middlebury benchmark [34], we show each cost aggregation method's rank on the website (evaluated at October 2013) in Table I. Avg Rank and Avg Err indicate the average rank and error rate measured using *Tsukuba*, *Venus*, *Teddy*, and *Cones* images [34]. Here, each method is combined with the state-of-the-art disparity refinement technique from [44] (for ST [24], we list its original rank reported in the Middlebury benchmark [34], since the same results were not reproduced using the author's C++ code). The rank also validates the effectiveness of our framework. We also report the running time for *Tsukuba* stereo pair on a PC with a 2.83-GHz CPU and 8 GB of memory. As mentioned before, the computational

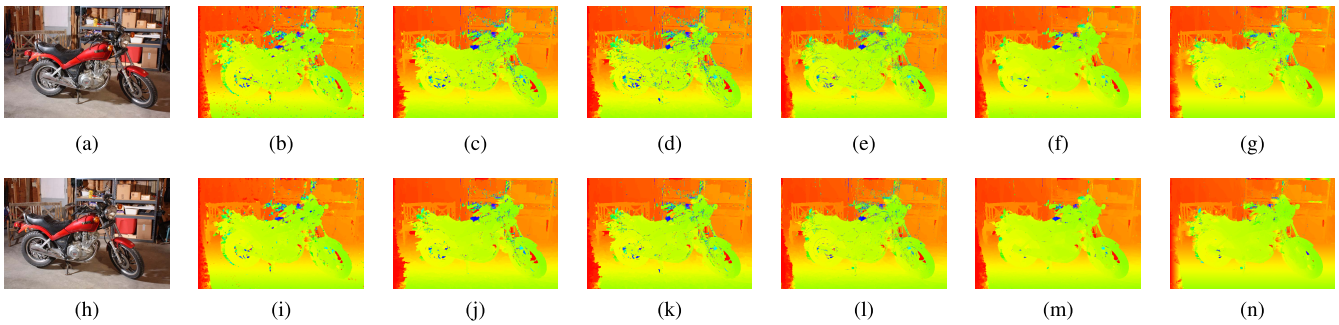


Fig. 5. Disparity maps of *Motorcycle* from the third version of the Middlebury stereo evaluation for all cost aggregation methods. Disparity maps are visualized using HSV colormap. The *non-occ* error rate is shown in each subtitle, where the absolute disparity error is larger than 1. (Best viewed in color.) (a) Left image. (b) BOX (12.74%). (c) NL (11.03%). (d) ST (13.31%). (e) BF (12.26%). (f) GF (7.21%). (g) PM (12.18%). (h) Right image. (i) S + BOX (9.26%). (j) S + NL (8.96%). (k) S + ST (10.61%). (l) S + BF (10.76%). (m) S + GF (6.66%). (n) S + PM (9.42%).

TABLE II

QUANTITATIVE EVALUATION OF PM AND S + PM ON MIDDLEBURY DATASET. FOR RANK PART (COLUMN 4∼5), ERROR THRESHOLD IS 0.5, WHICH CAN REFLECT SUBPIXEL DISPARITY ACCURACY [4] DIFFERENT FROM THAT OF TABLE I, WHERE ERROR THRESHOLD IS 1.0. AS MENTIONED BEFORE, WE DO NOT SAVE A SET OF 3D COST VOLUMES. THUS, COMPUTATIONAL COMPLEXITY OF S + PM IS ABOUT S TIMES LARGER THAN THAT OF PM

Method	Avg Non-occ(%)		Avg	Avg	Time
	M31	*M27*	Rank	Err(%)	(s)
PM [4]	11.92	7.29	24.4	9.91	61.32
S+PM	**10.10**	**6.25**	**18.8**	**9.32**	226.66

TABLE III

QUANTITATIVE EVALUATION OF COST AGGREGATION METHODS ON THIRD VERSION OF MIDDLEBURY STEREO EVALUATION. PREFIX S+ DENOTES OUR CROSS-SCALE COST AGGREGATION FRAMEWORK

Method	Avg Non-occ (%)
BOX	32.18
S+BOX	**26.59**
NL [44]	24.46
S+NL	**21.31**
ST [24]	25.26
S+ST	**22.27**
BF [47]	24.87
S+BF	**21.34**
GF [29]	21.54
S+GF	**21.34**
PM [4]	30.19
S+PM	**24.60**

overhead is relatively small. To be specific, it consists of the cost aggregation of $\tilde{\mathbf{C}}^s$ ($s \in \{0, 1, \ldots, S\}$) and the computation of (13).

We show the evaluation results of PM and S + PM on the Middlebury data set in a separate table (Table II) since their rank results are based on a different error threshold. In Table II, we adopt the postprocessing methods of the PM method to get the rank results and the running time for *Tsukuba* pair is reported. As can be seen from the table, cross-scale cost aggregation improves PM stereo in all evaluation metrics, but as mentioned before, the computational complexity of S + PM is about $S + 1$ times larger than that of the conventional PM method.

Finally, the third version of the Middlebury stereo evaluation has been released recently [30], [31]. The new benchmark provides 30 stereo image pairs which are split into test and training sets with 15 image pairs each. The new image pairs contain more complex scenes and take the effect of rectification error and radiometric changes into account, providing a more challenging benchmark than the previous one. We evaluate all the cost aggregation methods on the training set of the new benchmark, where we adopt the quarter resolution input and the error threshold is 1. The evaluation results are shown in Table III. Again all cost aggregation methods are improved with cross-scale cost aggregation. Cross-scale cost aggregation can consistently improve all cost aggregation methods' error rate by at least 3%, which can help to get better performance on this more challenging evaluation benchmark. Disparity maps of all methods on the *Motorcycle* stereo pairs from this data set are shown in Fig. 5.

B. KITTI Dataset

The KITTI data set [9] contains 194 training image pairs and 195 test image pairs for evaluating stereo matching algorithms. For the KITTI data set, image pairs are captured under the real-world illumination condition and almost all image pairs have a large portion of textureless regions, e.g., walls and roads [9]. During our experiments, we use the whole 194 training image pairs with ground truth disparity maps available. The evaluation metrics are the same as the KITTI benchmark [10] with an error threshold 3. Besides, since BF is too slow for high-resolution images (requiring more than one hour to process one stereo pair), we omit BF from evaluation.

Considering the illumination variation on the KITTI data set, we adopt *census transform* [48], which is proved to be powerful for robust optical flow computation [12]. We show the performance of different methods when integrated into cross-scale cost aggregation in Table IV. Some interesting points are worth noting. First, for BOX, GF, and PM, there

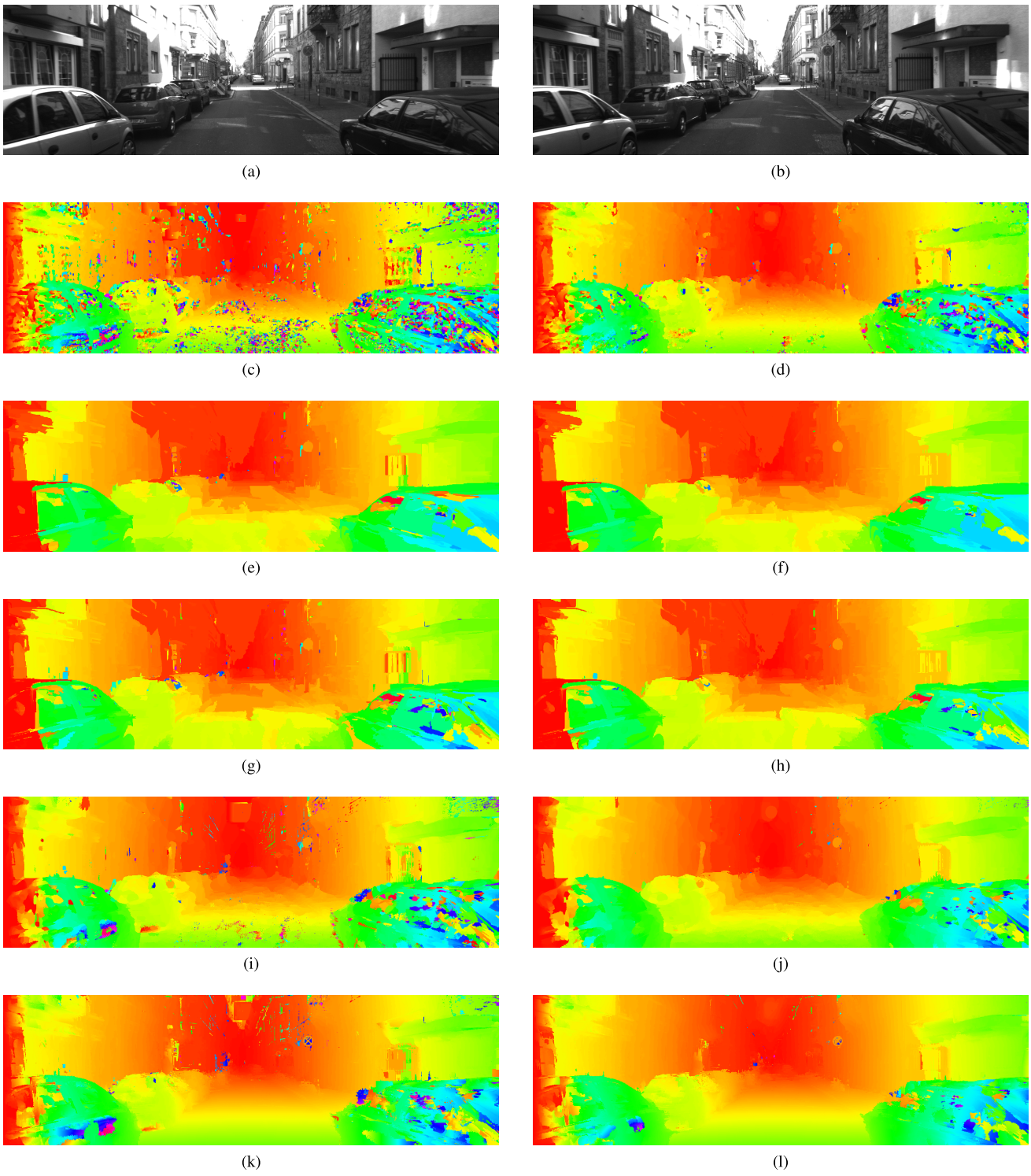


Fig. 6. Disparity maps of frame No. 83 in KITTI training pairs. First row: left and right images. Disparity maps are visualized using HSV colormap and evaluation results are shown in each subtitle. Left to right: the evaluation metrics are percentage of erroneous pixels in nonoccluded areas, percentage of erroneous pixels in all areas, average disparity error in nonoccluded areas, and average disparity error in all areas. Note how the road and car regions are improved by cross-scale cost aggregation. (a) No. 83 left image. (b) No. 83 right image. (c) BOX (26.84% 27.74% 14.50 px 15.21 px). (d) S + BOX (16.52% 17.54% 4.93 px 5.81 px). (e) NL (28.71% 29.58% 4.04 px 5.01 px). (f) S + NL (26.87% 27.76% 3.44 px 4.42 px). (g) ST (28.96% 29.83% 4.31 px 5.24 px). (h) S + ST (28.12% 29.00% 3.57 px 4.53 px). (i) GF (16.04% 17.07% 5.08 px 5.95 px). (j) S + GF (13.99% 15.04% 2.92 px 3.86 px). (k) PM (13.62% 14.68% 3.99 px 4.81 px). (l) S + PM (12.23% 13.31% 2.57 px 3.4 px).

are significant improvements when using cross-scale cost aggregation. Again, like on the Middlebury data set, the simple BOX method becomes very powerful using cross-scale cost aggregation. However, for S + NL and S + ST, their performances are almost the same as those without cross-scale cost aggregation, which are even worse than that of S + BOX. This may be due to the NL property of tree-based cost aggregation methods. As shown in Fig. 6, for textureless

TABLE IV

QUANTITATIVE COMPARISON OF COST AGGREGATION METHODS ON KITTI DATASET. OUT-NOC: PERCENTAGE OF ERRONEOUS PIXELS IN NONOCCLUDED AREAS. OUT-ALL: PERCENTAGE OF ERRONEOUS PIXELS IN TOTAL. AVG-NOC: AVERAGE DISPARITY ERROR IN NONOCCLUDED AREAS. AVG-ALL: AVERAGE DISPARITY ERROR IN TOTAL

Method	Out-Noc	Out-All	Avg-Noc	Avg-All
BOX	22.51 %	24.28 %	12.18 px	12.95 px
S+BOX	**12.06** %	**14.07** %	**3.54** px	**4.57** px
NL [44]	**24.69** %	**26.38** %	4.36 px	5.54 px
S+NL	25.41 %	27.08 %	**4.00** px	**5.20** px
ST [24]	**24.09** %	**25.81** %	4.31 px	5.47 px
S+ST	24.51 %	26.22 %	**3.82** px	**5.02** px
GF [29]	12.50 %	14.51 %	4.64 px	5.69 px
S+GF	**9.66** %	**11.73** %	**2.19** px	**3.36** px
PM [4]	9.11 %	11.19 %	2.75 px	3.69 px
S+PM	**7.09** %	**9.21** %	**1.58** px	**2.50** px

slant planes, e.g., roads, tree-based methods tend to overuse the *piecewise constancy* assumption and may generate erroneous fronto-parallel planes. Thus, even though the cross-scale cost aggregation is adopted, errors in textureless slanted planes are not fully addressed. More results using other stereo pairs are presented in the supplementary material.

C. New Tsukuba Dataset

The New Tsukuba data set [28] contains 1800 stereo pairs with ground truth disparity maps. These pairs consist of a 1-min photorealistic stereo video, generated by moving a stereo camera in a computer generated 3D scene. Besides, there are four different illumination conditions: 1) *Daylight*; 2) *Fluorescent*; 3) *Lamps*; and 4) *Flashlight*. In our experiments, we use the *Daylight* scene, which bears a challenging real-world illumination condition [28]. Since neighboring frames usually share similar scenes, we sample the 1800 frames every second to get a subset of 60 stereo pairs, which saves the evaluation time. We test both *intensity* + *gradient* and *census transform* cost functions, and *intensity* + *gradient* cost function gives better results on this data set. Disparity level of this data set is the same as the KITTI data set, i.e., 256 disparity levels, making BF [47] too slow, so we omit BF from evaluation.

Table V shows evaluation results for different cost aggregation methods on the New Tsukuba data set. We use the same evaluation metrics as the KITTI benchmark [10] (error threshold is 3). Again, all cost aggregation methods see an improvement when using cross-scale cost aggregation.

D. Parameters Effect

The key parameter in (7) is the regularization parameter λ. By adjusting this parameter, we can control the strength of inter-scale regularization as shown in Fig. 7. The error rate is evaluated on *M31*. When λ is set to 0, inter-scale regularization is prohibited, which is equivalent to performing cost aggregation at the finest scale. When regularization is introduced, there are improvements for all methods. As λ becomes large,

TABLE V

QUANTITATIVE COMPARISON OF COST AGGREGATION METHODS ON NEW TSUKUBA DATASET

Method	Out-Noc	Out-All	Avg-Noc	Avg-All
BOX	31.08 %	37.70 %	7.37 px	10.72 px
S+BOX	**18.82** %	**26.50** %	**3.92** px	**7.44** px
NL [44]	21.88 %	26.72 %	4.12 px	6.40 px
S+NL	**19.84** %	**24.50** %	**3.65** px	**5.73** px
ST [24]	21.68 %	27.07 %	4.33 px	7.02 px
S+ST	**18.99** %	**24.16** %	**3.60** px	**5.96** px
GF [29]	23.42 %	30.34 %	6.35 px	9.86 px
S+GF	**14.40** %	**21.78** %	**3.10** px	**6.38** px
PM [4]	21.53 %	28.55 %	7.02 px	9.77 px
S+PM	**17.05** %	**21.51** %	**3.45** px	**4.72** px

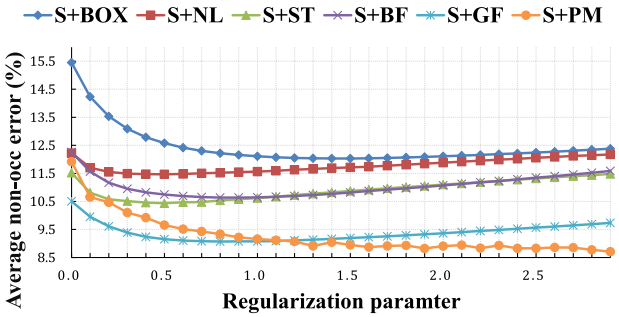


Fig. 7. Effect of varying inter-scale regularization parameter for different methods.

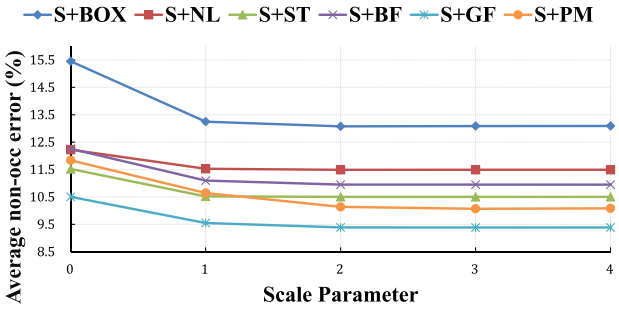


Fig. 8. Effect of varying scale numbers for different methods.

the regularization term dominates the optimization, causing the cost volume of each scale to be purely identical. As a result, fine details of disparity maps are missing and error rate increases. One may note that it will generate better results by choosing different values of λ for different cost aggregation methods, though we use consistent λ for all methods.

Another parameter that can influence the accuracy and running speed of the proposed framework is the scale parameter S. In Fig. 8, we vary the parameter from 0 to 4 and show the average error rate on *M31*. It can be seen that when $S = 1$, i.e., using two scales, all methods achieved significant quality improvements compared with the original cost aggregation methods. As more than two scales are used, the improvements become rather marginal. The resolution of stereo images from *M31* is about 400×300, and thus, under the Gaussian pyramid representation, coarser scale images (i.e., $s \geq 2$) become too small to produce meaningful estimates. However, for

stereo images from Middlebury Third version benchmark, KITTI, and New Tsukuba, the resolution is bigger, and thus coarser scales can produce meaningful estimates. Considering the marginal computational overhead introduced using more scales, we consistently set the scale parameter to 4, i.e., totally five scales for all data sets.

VII. Conclusion

In this paper, we have proposed a cross-scale cost aggregation framework for stereo matching. This paper is not intended to present a completely new cost aggregation method that yields a highly accurate disparity map. Rather, we investigate the scale space behavior of various cost aggregation methods. Extensive experiments on three data sets validated the effect of cross-scale cost aggregation. Almost all methods saw improvements and even the simple box filtering method combined with our framework achieved very good performance.

Recently, a new trend in stereo vision has been to solve the correspondence problem in the continuous plane parameter space rather than in the discrete disparity label space.

We have shown that the cross-scale cost aggregation further improves the accuracy of the PM stereo method [4]. Along this direction, it would be valuable to investigate how to integrate multiscale information into other stereo approaches [20], [43] that utilize the continuous plane parameters but employ more complicated data structures, e.g., super-pixel-based representation. One possible solution is to adopt multiscale super-pixel segmentation instead of the Gaussian pyramid used in our approach.

References

[1] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *Proc. 12th ECCV*, vol. 7577. 2012, pp. 214–227.

[2] L. Alvarez, R. Deriche, J. Sánchez, and J. Weickert, "Dense disparity map estimation respecting image discontinuities: A PDE and scale-space based approach," *J. Vis. Commun. Image Represent.*, vol. 13, nos. 1–2, pp. 3–21, Mar. 2002.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," in *Proc. ACM SIGGRAPH*, 2009, Art. ID 24.

[4] M. Bleyer, C. Rhemann, and C. Rother, "PatchMatch stereo—Stereo matching with slanted support windows," in *Proc. BMVC*, 2011, pp. 14.1–14.11.

[5] P. J. Burt, "Fast filter transform for image processing," *Comput. Graph. Image Process.*, vol. 16, no. 1, pp. 20–51, 1981.

[6] C. Chang and S. Chatterjee, "Multiresolution stereo by simulated annealing," in *Proc. IJCNN*, vol. 2. Jun. 1990, pp. 885–890.

[7] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," in *Proc. IEEE CVPR*, vol. 1. Jun./Jul. 2004, pp. 261–268.

[8] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory Comput.*, vol. 8, pp. 415–428, 2012.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 3354–3361.

[10] A. Geiger, P. Lenz, and R. Urtasun. (2012). *The KITTI Vision Benchmark Suite*. [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo

[11] M. Gong, R. Yang, L. Wang, and M. Gong, "A performance study on different cost aggregation approaches used in real-time stereo matching," *Int. J. Comput. Vis.*, vol. 75, no. 2, pp. 283–296, 2007.

[12] D. Hafner, O. Demetz, and J. Weickert, "Why is the census transform good for robust optic flow computation?" in *Proc. 4th Int. Conf. SSVM*, vol. 7893. 2013, pp. 210–221.

[13] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. 11th ECCV*, 2010, pp. 1–14.

[14] S. Hermann and R. Klette, "Evaluation of a new coarse-to-fine strategy for fast semi-global stereo matching," in *Advances in Image and Video Technology*, vol. 7087. Springer, 2011, pp. 395–406.

[15] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[16] A. Hosni, M. Bleyer, and M. Gelautz, "Secrets of adaptive support weight techniques for local stereo matching," *Compt. Vis. Image Understand.*, vol. 117, no. 6, pp. 620–632, Jun. 2013.

[17] W. Hu, K. Zhang, L. Sun, and S. Yang, "Comparisons reducing for local stereo matching using hierarchical structure," in *Proc. IEEE ICME*, Jul. 2013, pp. 1–6.

[18] Y.-H. Jen, E. Dunn, P. Fite-Georgel, and J.-M. Frahm, "Adaptive scale selection for hierarchical stereo," in *Proc. BMVC*, 2011, pp. 95.1–95.10.

[19] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.

[20] J. Lu, H. Yang, D. Min, and M. N. Do, "PatchMatch filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 1854–1861.

[21] J. Magarey and A. Dick, "Multiresolution stereo image matching using complex wavelets," in *Proc. 14th ICPR*, vol. 1. Aug. 1998, pp. 4–7.

[22] H. A. Mallot, S. Gillner, and P. A. Arndt, "Is correspondence search in human stereo vision a coarse-to-fine process?" *Biol. Cybern.*, vol. 74, no. 2, pp. 95–106, Feb. 1996.

[23] D. Marr and T. Poggio, "A computational theory of human stereo vision," *Proc. Roy. Soc. London B, Biol. Sci.*, vol. 204, no. 1156, pp. 301–328, May 1979.

[24] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang, "Segment-tree based cost aggregation for stereo matching," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 313–320.

[25] M. D. Menz and R. D. Freeman, "Stereoscopic depth processing in the visual cortex: A coarse-to-fine mechanism," *Nature Neurosci.*, vol. 6, no. 1, pp. 59–65, Jan. 2003.

[26] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[27] D. Min and K. Sohn, "Cost aggregation and occlusion handling with WLS in stereo matching," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1431–1442, Aug. 2008.

[28] M. Peris, A. Maki, S. Martull, Y. Ohkawa, and K. A. Fukui, "Towards a simulation driven stereo vision system," in *Proc. 21st ICPR*, Nov. 2012, pp. 1038–1042.

[29] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3017–3024.

[30] D. Scharstein and H. Hirschmüller. (2014). *Middlebury Stereo Evaluation—Version 3 (Beta)*. [Online]. Available: http://vision.middlebury.edu/stereo/eval3/

[31] D. Scharstein *et al.*, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. 36th GCPR*, vol. 8753. 2014, pp. 31–42.

[32] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[33] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, Apr. 2002.

[34] D. Scharstein and R. Szeliski. (2002). *Middlebury Stereo Vision Website*. [Online]. Available: http://vision.middlebury.edu/stereo/

[35] M. Sizintsev, "Hierarchical stereo with thin structures and transparency," in *Proc. CCCRV*, May 2008, pp. 97–104.

[36] M. Sizintsev and R. P. Wildes, "Efficient stereo with accurate 3-D boundaries," in *Proc. BMVC*, 2006, pp. 237–246.

[37] X. Tan, C. Sun, D. Wang, Y. Guo, and T. D. Pham, "Soft cost aggregation with multi-resolution fusion," in *Proc. 13th ECCV*, vol. 8693, 2014, pp. 17–32.

[38] L. Tang, M. K. Garvin, K. Lee, W. L. M. Alward, Y. H. Kwon, and M. D. Abràmoff, "Robust multiscale stereo matching from fundus images with radiometric differences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2245–2258, Nov. 2011.

[39] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th ICCV*, Jan. 1998, pp. 839–846.

[40] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

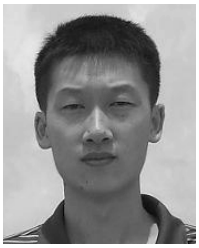
[41] G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool, "A hierarchical symmetric stereo algorithm using dynamic programming," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 275–285, Apr. 2002.

[42] Z.-F. Wang and Z.-G. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.
[43] K. Yamaguchi, D. McAllester, and R. Urtasun, "Robust monocular epipolar flow estimation," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 1862–1869.
[44] Q. Yang, "A non-local cost aggregation method for stereo matching," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 1402–1409.
[45] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 492–504, Mar. 2009.
[46] R. Yang and M. Pollefeys, "Multi-resolution real-time stereo on commodity graphics hardware," in *Proc. IEEE CVPR*, vol. 1. Jun. 2003, pp. I-211–I-217.
[47] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650–656, Apr. 2006.
[48] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proc. 3rd ECCV*, 1994, pp. 151–158.
[49] K. Zhang, Y. Fang, D. Min, L. Sun, S. Yang, S. Yan, and Q. Tian, "Cross scale cost aggregation for stereo matching," in *Proc. IEEE CVPR*, 2014, pp. 1590–1597.

Kang Zhang received the B.E. degree from the Department of Computer Science and Technology, Beijing Institute of Technology, Beijing, China, in 2010. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing.

He was sponsored by the NeXT program to visit the National University of Singapore, Singapore, from 2013 to 2014. He has authored papers on stereo matching in top conferences of the relevant field. His current research interests include 3D reconstruction and stereo matching.

Dr. Zhang served as a Reviewer of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.



Yuqiang Fang received the master's degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 2010, where he is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems.

He was with the Learning and Vision Research Group, National University of Singapore, Singapore, as a Research Assistant in 2013. His current research interests include machine learning and computer vision, and their applications to autonomous vehicle.

Dongbo Min (M'09–SM'15) received the B.S., M.S., and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea, in 2003, 2005, and 2009, respectively.

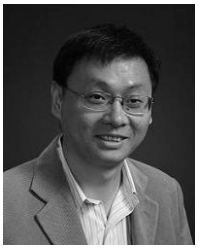
He was with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, as a Post-Doctoral Researcher from 2009 to 2010, where he developed a prototype of 3D video system. From 2010 to 2015, he was with the Advanced Digital Sciences Center, Singapore, which was jointly founded by the University of Illinois at Urbana–Champaign, Champaign, IL, USA, and the Agency for Science, Technology and Research, Singapore, a Singapore Government Agency. Since 2015, he has been an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University, Daejeon, Korea. His current research interests include computer vision, 2D/3D video processing, computational photography, augmented reality, and continuous/discrete optimization.

Lifeng Sun (M'05) received the B.S. and Ph.D. degrees in system engineering from the National University of Defense Technology, Changsha, China, in 1995 and 2000, respectively.

He was involved in post-doctoral research with the Department of Computer Science and Technology, Tsinghua University, Beijing, China, from 2001 to 2003. He is currently a Professor with the Department of Computer Science and Technology, Tsinghua University. His current research interests include online social network, video streaming, interactive multiview video, 3D computer vision, and distributed video coding.

Shiqiang Yang (M'00–SM'08) received the B.E. and M.E. degrees in computer science from Tsinghua University, Beijing, China, in 1977 and 1983, respectively.

He was an Assistant Professor with Tsinghua University from 1980 to 1992, where he served as an Associate Professor from 1994 to 1999, and has been a Professor since 1999. From 1994 to 2011, he was the Associate Head of the Department of Computer Science and Technology with Tsinghua University. He is currently the President of the Multimedia Committee of the China Computer Federation, Beijing, and the Co-Director of the Microsoft-Tsinghua Multimedia Joint Laboratory with Tsinghua University. His current research interests include multimedia procession, media streaming, and online social network.



Shuicheng Yan (M'06–SM'09) received the Ph.D. degree from the School of Mathematical Sciences, Peking University, Beijing, China, in 2004.

He spent three years as a Post-Doctoral Fellow with the Chinese University of Hong Kong, Hong Kong, and the University of Illinois at Urbana–Champaign, Urbana, IL, USA. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore, and the Founding Leader of the Learning and Vision Research Group. He has authored or co-authored over 300 technical papers over a wide range of research topics, with Google Scholar citation over 8900 times, and has an H-index of 41. His current research interests include computer vision, multimedia, and machine learning.

Dr. Yan received the best paper awards from ACM Multimedia (MM) in 2012 (demo), the Pacific-Rim Conference on Multimedia in 2011, ACM MM in 2010, the International Conference on Multimedia & Expo in 2010, and the International Conference on Internet Multimedia Computing and Service in 2009, the Winner Prizes of the Classification Task in PASCAL VOC from 2010 to 2012, the Winner Prize of the Segmentation Task in PASCAL VOC in 2012, the Honorable Mention Prize of the Detection Task in PASCAL VOC in 2010, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Best Associate Editor Award in 2010, the Young Faculty Research Award in 2010, the Singapore Young Scientist Award in 2011, and the NUS Young Researcher Award in 2012. He is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the *ACM Transactions on Intelligent Systems and Technology*. He has served as the Guest Editor of the special issues of the IEEE TRANSACTIONS ON MULTIMEDIA and *Computer Vision and Image Understanding*.