Noname manuscript No.
(will be inserted by the editor)

Adaptive disparity computation using local and non-local cost aggregations

Qicong Dong · Jieqing Feng*

Received: date / Accepted: date

Abstract A new method is proposed to adaptively compute the disparity of stereo matching by choosing one of the alternative disparities from local and non-local disparity maps. The initial two disparity maps can be obtained from state-of-the-art local and non-local stereo algorithms. Then, the more reasonable disparity is selected. We propose two strategies to select the disparity. One is based on the magnitude of the gradient in the left image, which is simple and fast. The other utilizes the fusion move to combine the two proposal labelings (disparity maps) in a theoretically sound manner, which is more accurate. Finally, we propose a texture-based sub-pixel refinement to refine the disparity map. Experimental results using Middlebury datasets demonstrate that the two proposed selection strategies both perform better than individual local or non-local algorithms. Moreover, the proposed method is compatible with many local and non-local algorithms that are widely used in stereo matching.

Keywords stereo matching · adaptive disparity computation · fusion move · disparity selection · texture-based sub-pixel refinement

1 Introduction

Stereo matching is one of the fundamental problems in stereo vision. Three dimensional (3D) objects or scenes can be reconstructed with stereo matching, which is widely applied in navigation, robotics, and autonomous driving. The disparity map generated from stereo matching is useful in many application fields. Augmented reality [29] and tracking [24] have shown a particular interest for disparity

Qicong Dong
State Key Lab of CAD&CG, Zhejiang University,
Hangzhou, 310058, China
E-mail: qicongdong@zju.edu.cn

✉ Jieqing Feng
State Key Lab of CAD&CG, Zhejiang University,
Hangzhou, 310058, China
E-mail: jqfeng@cad.zju.edu.cn

maps for occlusion tolerance [39] or for the problem of hand segmentation from the scene and gesture recognition [35,21,6]. Consider two images captured by two horizontal cameras, where one image is regarded as the reference image and the other image is the target image, and the camera parameters are known. The goal is to obtain the disparity d of a pixel at position (x, y) in the reference image so that the same pixel appears at position $(x - d, y)$ in the target image. Once the disparity d is obtained, we can compute the depth of the pixel in the 3D scene as $z = fB/d$, where f is the focal length of the cameras and B is the baseline length (the distance between the centers of the two cameras).

According to the survey in [33], a general process of stereo matching primarily consists of four steps: 1. matching cost computation, 2. cost aggregation, 3. disparity computation, and 4. disparity refinement. An important class of stereo matching algorithms is called the local approach [18,52,9]. The local approach primarily focuses on the cost aggregation step. A local support window is usually determined for each pixel and the costs are aggregated in the window to obtain more accurate results. There is another class of methods called the non-local approach [44,17,41] which also focuses on the cost aggregation step, but each pixel can receive information concerning the entire image. The latter class of methods addresses the matching problem more globally because the "support window" is non-local. The main difference between the local cost aggregation and the non-local cost aggregation is the size of the support window. Local methods aggregate the matching costs in a finite local window, while non-local methods aggregate the matching costs considering all pixels in the whole image.

The methods above have their own advantages and disadvantages and are suitable for different types of scenes. If the regions are rich in local details such as abundant colors and textures, local methods will generally generate a superior stereo matching result. The goal of local methods is to find an accurate estimation in these regions. Conversely, if the regions are textureless, local methods suffer from weaknesses [8], and non-local methods will perform superior to local algorithms in this case. The goal of non-local methods is to evaluate disparities in these regions more reasonably. Conversely, the contribution of far-away pixels will sometimes produce an error propagation problem for sharp edges and thin objects in non-local methods [41]. Fig. 1 shows some results from local and non-local algorithms. In Fig. 1(a), the green square region on the left is textureless. A local algorithm may generate incorrect matches as shown in Fig. 1(b), whereas the blue square regions in Fig. 1(a) contain abundant local structure information, which may often cause problems for non-local algorithms, as shown in Fig. 1(c).

This observation inspires us to utilize both local and non-local methods to achieve improved stereo matching performance. We propose a method to compute disparities via both local and non-local algorithms and adaptively select the superior disparities for different regions of a scene. The same example is shown in Fig. 1(d), where the disparities in the rectangular regions are computed more robustly. In the cost aggregation step, both local and non-local algorithms are applied simultaneously. Then, two disparity maps are generated using the winner-take-all (WTA) computation.

The key question is how to select the more reasonable disparities between two disparity maps. In this paper, we propose two strategies. One strategy selects disparities based on the magnitude of the gradient of each pixel. If the magnitude of the gradient of a pixel is higher than a threshold, the disparity computed by

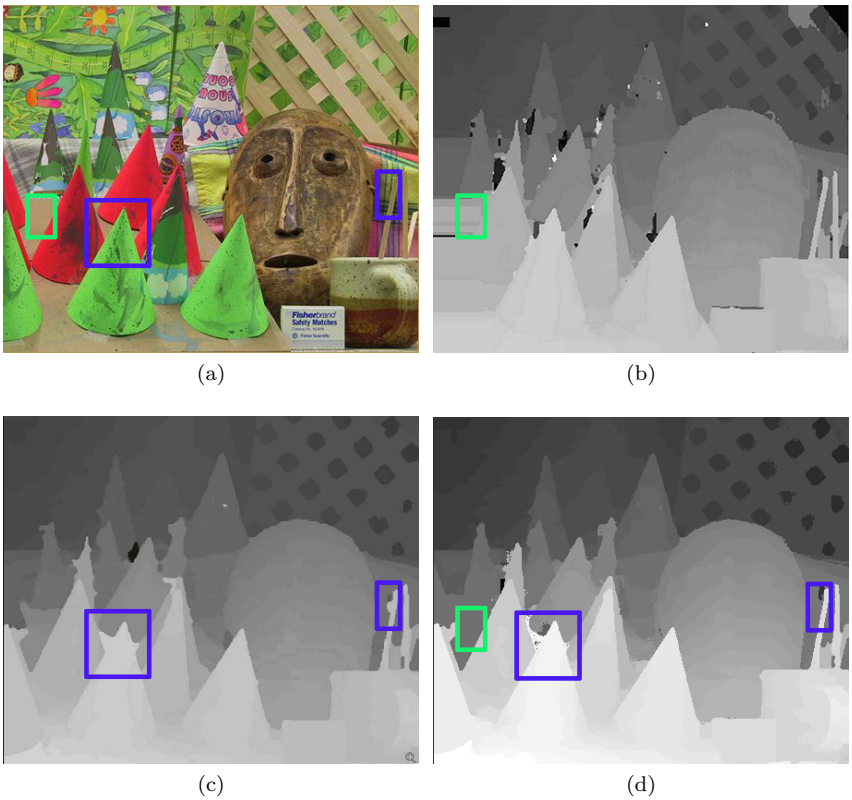

Fig. 1 Examples of local, non-local, and the proposed stereo matching algorithms. (a) Raw image, (b) Disparity map generated by the local algorithm [52], (c) Disparity map generated by the non-local algorithm [44], and (d) Disparity map generated by the proposed method. The disparities in the square regions demonstrate that the proposed method can generate superior results to individual local or non-local algorithms

the local algorithm is selected; otherwise, the disparity computed by the non-local algorithm is selected for this pixel. This strategy is based directly on the observation above and the experimental results demonstrate that this strategy performs superior to individual local or non-local algorithms. However, there is no guarantee that this observation is always valid for each pixel. Therefore, we also propose a more accurate strategy. This strategy considers the local and non-local disparity maps as two proposal labelings and combines them using the fusion move algorithm [14]. The fusion move belongs to an important class of optimization problems that minimizes energies from pairwise Markov random fields (MRFs) with discrete labels. The fusion move can combine two proposal labelings in a reasonable manner by employing graph cut based algorithms (also known as the QPBO-graph cut [11]), which is in practice often globally optimal [14].

Both of these two strategies can achieve results superior to individual local or non-local algorithms. Furthermore, the proposed method is a general framework and is also compatible with many local and non-local algorithms. The method can

exploit either local or non-local algorithms in specific regions and achieves superior results.

2 Related work

Stereo matching is a well-studied problem in computer vision and many algorithms have been proposed to address this issue. In this section, we discuss only the most related algorithms that are widely used in stereo matching.

The commonly used matching cost is usually defined as the Sum of Absolute Differences (SAD) or the Sum of Squared Differences (SSD) [13]. Hirschmuller [7] computes mutual information among pixels to define a matching cost. Some studies apply Gradient Differences (GD) [31,4] or a census transform [48] which consider the information of pixels' neighborhoods. Mei et al. [18] combine the Absolute Differences (AD) and census transform methods to compute a matching cost, which generates superior results to using the AD or census transform methods individually. Jiao et al. [9] combine the AD, GD, and census transform methods to obtain a more robust matching cost. Zhan et al. [50] introduce the concept of guidance image, which is a filter-based image. Matching costs (the AD, GD and census transform) are computed on both the raw image and the guidance image, and then these costs are combined to obtain a more robust matching cost.

Since the above initial matching costs are not sufficiently robust to achieve accurate stereo matching in general, the costs are often aggregated in a support region in local methods. In the cost aggregation step, the main problem is how to determine the shape of the support region and the weights of neighboring pixels contributing to the center pixel. The simplest solution is a fixed rectangular window and constant weights in the window. Yoon and Kweon [46] use adaptive weights based on color and spatial differences between the neighboring pixel and the center pixel in a fixed window. Certain algorithms focus on the shape of the support region, such as cross-based cost aggregation [52], which defines a cross region for each pixel, and the weights in the cross region are constant. Mei et al. [18] propose rules that determine the shape of the cross region more strictly and achieve a superior result. Another study adds a rule called the space constraint to define the cross region [27]. Rhemann et al. utilize the guided filter [30] to aggregate the initial matching costs. This algorithm provides the weights between two pixels according to their statistical analogy based on the averages and the variances of several squared windows on the guided (reference) image [20].

In summary, the cost aggregation techniques used in local methods can generally be classified into four types: (1) fixed window size and constant weight (in the support window), (2) fixed window size and adaptive weight, (3) adaptive window size and constant weight, and (4) adaptive window size and adaptive weight. However, most local approaches are quite fragile and are prone to experiencing difficulties within textureless regions [42].

Yang [44] presents a non-local cost-aggregation method for stereo matching. In this method, a Minimum Spanning Tree (MST) is built for the entire image based on color differences among neighboring pixels; then, the costs are aggregated on that MST. Each node in the MST receives information from all other nodes on this tree. Mei et al. create a Segment Tree (ST) to aggregate the matching costs [17]. This method segments the image into regions, and a tree graph is created in

each segment. Then, these tree graphs are linked to form the ST. The ST enforces the connectivity within the segment because pixels in the same segment tend to have similar disparities. Vu et al. [41] extend this algorithm hierarchically with a hybrid tree to aggregate the costs at the pixel and region levels simultaneously. The aggregation in the region level MST is a coarse aggregation, whereas the aggregation in the pixel level MST is a finer aggregation. Psota et al. [28] employ the MST structure for global optimization by message passing. The disparity map is represented as a collection of hidden states on some MSTs, and each MST is modeled as a hidden Markov tree. Li et al. [15] combine MST-based algorithm and PatchMatch stereo algorithm [1] to obtain an accurate disparity map. The above methods use more information in stereo matching than local methods, and they do not become trapped in local optimum. Essentially, the non-local algorithm is an adaptive weight algorithm whose supporting window is the entire image. However, this class of algorithms might not perform well in certain regions such as highly textured regions and at object boundaries.

In addition to the local and non-local methods, there is also an important class of algorithms called global methods in stereo. These methods solve the stereo problem by minimizing pairwise MRF energies. In stereo vision, the pairwise MRFs are usually with either multiple discrete or continuous labels. Thus, this class of optimization problems is NP-hard in general. Various approximate solutions are proposed, such as graph cuts [12,19,37,2] and belief propagation [47,22]. In the proposed method, we utilize fusion moves [14] to fuse the local and non-local disparity maps into a more accurate disparity map and solve this binary-labeled MRF minimization problem utilizing the QPBO-graph cut [11]. Some approaches [23,43] generate several plausible initial proposals and combine these proposals to obtain a superior disparity map. These approaches use many initial proposals and conduct a number of fusions because the accuracy of these initial proposals might not be very high. In the proposed method, we only need to solve a binary-labeled MRF minimization problem once since we have already obtained two relatively accurate disparity maps (proposals) after the cost aggregation step. In fact, the method to balance the accuracy and number of input proposals is a crucial problem in fusion approaches. We will show in the experimental results that the proposed method can achieve an accurate disparity map without many input proposals.

Recently many deep learning approaches are utilized to solve depth estimation problems. Žbontar and LeCun [49] define a new matching cost by training a Convolutional Neural Network (CNN) in a supervised way. The matching cost is refined with some post-processing steps to give an accurate depth estimation. Ummenhofer et al. [40] train a CNN end-to-end to compute depth and camera motion. There are also some approaches that handle the depth problem utilizing unsupervised learning [53,54]. Shu et al. [34] develop a new network to transfer labeling information across heterogeneous domains, especially from text domain to image domain.

3 Adaptive Disparity Computation for Stereo Matching

In this section, we will first introduce our stereo matching framework. The proposed framework is generally consistent with the traditional framework used in stereo matching [33]. The difference is that the matching costs are aggregated us-

ing local and non-local cost aggregations simultaneously, and the superior disparity of each pixel is selected reasonably to generate a disparity map. Fig. 2 illustrates the proposed framework. The matching cost is initialized by MC-CNN-acrt [49], a state-of-the-art method. Then, the matching costs are aggregated using two methods: local cost aggregation and non-local cost aggregation. Two initial disparity maps are generated using WTA and more reasonable disparities are selected. Finally, texture-based sub-pixel refinement is performed and the final disparity map is generated. A detailed description of each step is provided in the following subsections.

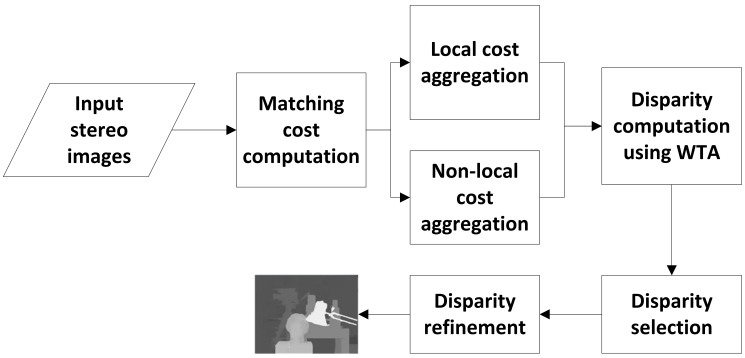

Fig. 2 Framework of the proposed method

3.1 Matching Cost Computation

The matching cost is initialized by the state-of-the-art cost predicted by a convolution neural network [49]. Let $\mathbf{p}$ denote the image coordinates of pixel p in the image. Let $C(\mathbf{p}, d)$ define the cost of the disparity d at position $\mathbf{p}$; the cost is formulated as follows:

$$C(\mathbf{p}, d) = C_{CNN}(R^L(p_x, p_y), R^R(p_x - d, p_y)) \tag{1}$$

where $C_{CNN}(\cdot, \cdot)$ predicts the similarity cost. This cost is computed between the $11*11$ image patch R^L centered at pixel p of the left image and the image patch R^R centered at the corresponding pixel of the right image.

3.2 Local and Non-local Cost Aggregations

As described above, two cost aggregation approaches, i.e., a local approach and a non-local approach, are adopted to compute two cost maps in the proposed method. The two candidate disparity maps will be selected optimally in the subsequent steps.

3.2.1 Local Cost Aggregation

Local cost aggregation algorithms aggregate the initialized matching costs in a local window. As stated in Section 2, the weights of neighboring pixels contributing to the center pixel can be fixed or adaptive, as is the shape of the supporting window. The general form of the local algorithms can be formulated as follows:

$$C_l(\mathbf{p}, d) = \frac{\sum_{\mathbf{q} \in S_d(\mathbf{p})} w(\mathbf{p}, \mathbf{q}) C(\mathbf{q}, d)}{\sum_{\mathbf{q} \in S_d(\mathbf{p})} w(\mathbf{p}, \mathbf{q})} \tag{2}$$

where $C_l(\mathbf{p}, d)$ is the aggregated cost and $C(\mathbf{q}, d)$ is the initial matching cost of the neighboring pixel q. $S_d(\mathbf{p})$ is the supporting window of the center pixel p and $w(\mathbf{p}, \mathbf{q})$ is the supporting weight of the neighboring pixel q to the center pixel p.

In this paper, we implement two local cost aggregation algorithms, i.e., Cross-based Cost Aggregation [18] and Guided Filter [36], to generate accurate disparity maps.

3.2.2 Non-local Cost Aggregation

The initial matching cost is also aggregated utilizing the non-local cost aggregation method. We use Yang's MST approach in [44] as an example to introduce this type of method. The method aggregates the cost based on pixel similarity on a tree structure and is not confined to a local optimum. In this method, the input image is represented as an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges in the graph. The nodes in V are pixels in the image and the edges in E connect neighboring pixels, where a four-connected graph is adopted in the proposed method. The edge weights are defined according to the color differences, as follows:

$$\omega_e(\mathbf{s}, \mathbf{t}) = |I_{\mathbf{s}} - I_{\mathbf{t}}| \tag{3}$$

where $\mathbf{s}$ and $\mathbf{t}$ are the positions of a pair of neighboring pixels, $I_{\mathbf{s}}$ and $I_{\mathbf{t}}$ are intensities of pixel $\mathbf{s}$ and pixel $\mathbf{t}$, respectively. The weight is computed from the average absolute difference of three channels of the reference image in the proposed method. Then, an MST is built based on the edge weights. The distance between the pixels p and q on this tree is defined as $D(\mathbf{p}, \mathbf{q})$, and it is the summation of the edge weights of the connected edges (shortest path) between them. Then,

$$W(\mathbf{p}, \mathbf{q}) = exp(\frac{-D(\mathbf{p}, \mathbf{q})}{\sigma}) \tag{4}$$

denotes the similarity between p and q, where σ is a user-defined parameter. The matching cost can be aggregated on the tree structure, as follows:

$$C_n(\mathbf{p}, d) = \frac{\sum\limits_{\mathbf{q}} W(\mathbf{p}, \mathbf{q}) C(\mathbf{q}, d)}{\sum\limits_{\mathbf{q}} W(\mathbf{p}, \mathbf{q})} \tag{5}$$

where $C_n(\mathbf{p}, d)$ is the aggregated cost and $C(\mathbf{q}, d)$ is the initial matching cost. This cost is non-local because each pixel receives supporting weights from the

entire image. The aggregated costs of each pixel can be efficiently computed by traversing the tree structure in two sequential passes. For detailed information concerning this algorithm, we refer readers to [44]. Addtionally, in this paper, we implement a Segment-tree [17] to generate a disparity map.

3.3 Disparity Computation from Two Cost Volumes

We compute two cost volumes in the previous steps and generate two disparity maps using WTA, i.e., a local map and a non-local map, as follows:

$$d^l(\mathbf{p}) = \underset{d}{argmin} C_l(\mathbf{p}, d) \tag{6}$$

$$d^n(\mathbf{p}) = \underset{d}{argmin} C_n(\mathbf{p}, d) \tag{7}$$

where $C_l(\mathbf{p}, d)$ and $C_n(\mathbf{p}, d)$ are the aggregated costs computed by local and non-local cost aggregations.

3.4 Disparity Selection from Two Disparity Maps

In this step, the superior disparity is selected from the two disparity maps obtained above. The choice of the superior disparity will be important in the proposed stereo matching framework. Here, two selection strategies are proposed. One utilizes the texture information of the left image to select the superior disparity, while the other uses the fusion move [14] to combine the two disparity maps obtained above in a sound manner. The detailed descriptions of these two strategies will be presented in the following sub-sections.

3.4.1 Selection via Texture information

We compute the gradient map of the left image via the Sobel gradient operator. The disparity is selected based on the magnitude of the gradient. For each pixel, when the magnitude of the gradient is higher than a threshold, we select the disparity computed by the local algorithm; otherwise, the disparity computed by the non-local algorithm is selected. Moreover, when one pixel's local and non-local disparities are nearly equal (the difference between them is not larger than one), we compute the average value of the local and non-local disparities and use this average value as the final disparity to achieve a sub-pixel accuracy disparity. More formally, the disparity of each pixel is obtained as follows:

$$d(\mathbf{p}) = \begin{cases} \frac{d^l(\mathbf{p}) + d^n(\mathbf{p})}{2} & if \ \left| d^l(\mathbf{p}) - d^n(\mathbf{p}) \right| \leq 1 \\ d^l(\mathbf{p}) & else \ if \ \sqrt{G_x(\mathbf{p})^2 + G_y(\mathbf{p})^2} \geq \delta_{tex} \\ d^n(\mathbf{p}) & otherwise \end{cases} \tag{8}$$

where $G_x(\mathbf{p})$ and $G_y(\mathbf{p})$ denote the gradient components at pixel p along the x-direction and the y-direction, respectively. $d^l(\mathbf{p})$ and $d^n(\mathbf{p})$ are two disparity

maps obtained via local and non-local cost aggregations, respectively. δ_{tex} is a user-defined threshold.

The local algorithm usually successfully processes a region that contains abundant texture information. In general, the magnitude of the gradient is large in these regions because the color variation is great. Thus, when the region is rich in texture, the disparity computed by the local algorithm is selected; otherwise, when the region is textureless or has weak texture, the magnitude of the gradient is generally small. The local algorithm cannot address the disparity well; thus, the disparity computed by the non-local algorithm is selected to alleviate this problem. Additionally, when the difference between the local disparity and the non-local disparity is small, we consider these two disparities to be both reasonable and use their average value to achieve a sub-pixel accuracy disparity. In Section 4, the experimental results show that this strategy can generate a satisfactory disparity map.

3.4.2 Selection via Fusion Move

Selecting reasonable disparities via the texture information is simple and fast. However, there is no guarantee that a local algorithm always successfully processes highly textured regions superior to a non-local algorithm, and vice versa. In this sub-section, we propose a more elaborate approach to combine the local disparity map and the non-local disparity map in a sound manner to generate a more accurate disparity map.

First, we introduce the concept of the optimization problem in stereo vision. Let x_p denote the value x at pixel p. The stereo problem can be considered as solving energies associated with pairwise MRFs, which take the following form:

$$E(\mathbf{x}) = \sum_{p \in \nu} U_p(x_p) + \sum_{p,q \in \tau} V_{pq}(x_p, x_q), \quad \mathbf{x} \in \mathbf{L}^{\nu} \tag{9}$$

where ν is a set of nodes that correspond to the pixels in the image and τ is a set of undirected edges connecting pairs of nodes that contain pairs of adjacent pixels in four-connected neighborhoods. The labeling $\mathbf{x}$ assigns a label (disparity) x_p from the label space $\mathbf{L}$ to each node $p \in \nu$. In stereo, the function $U_p : \mathbf{L} \rightarrow \mathbf{R}$ is called the *unary term,* which usually corresponds to the matching costs. The function $V_{p,q} : \mathbf{L}^2 \rightarrow \mathbf{R}$ is called the *pairwise term,* which usually corresponds to the costs of disparity changes. However, we have obtained two cost volumes aggregated by the local and the non-local algorithms, which is different from the majority of stereo algorithms. We use the combined aggregated costs as the unary term in our stereo method. The unary term is reformulated as follows:

$$U_p(x_p) = \alpha_t C_l(\mathbf{p}, x_p) + (1 - \alpha_t) C_n(\mathbf{p}, x_p) \tag{10}$$

and

$$\alpha_t = \frac{\sqrt{G_x(\mathbf{p})^2 + G_y(\mathbf{p})^2}}{T_{tex}} \tag{11}$$

where $C_l(\mathbf{p}, x_p)$ and $C_n(\mathbf{p}, x_p)$ are normalized as defined in Equations (2) and (5). T_{tex} is a user-defined threshold but we limit T_{tex} to be slightly larger than the maximum gradient of the image to ensure that α_t is between zero and one.

Here, we add the weight α_t to consider the texture information. When the pixel's gradient is large, the cost of the local aggregation will have a larger proportion in the unary term since the cost of the local aggregation may be more accurate, and vice versa. Note that the weight α_t only plays a guided role but not a dominant role in this algorithm, which is distinct from selecting disparities based on the texture information in Section 3.4.1.

The pairwise term is also called the smooth term, which penalizes the change in disparities between neighboring pixels. We use the first-order smooth prior as the pairwise term in our algorithm, as follows:

$$V_{pq}(x_p, x_q) = min(|x_p - x_q|, \lambda_s) \tag{12}$$

where λ_s is a constant positive penalty.

Then we utilize the fusion move algorithm in [14] to combine the local and non-local disparity maps. Given two labelings (disparity maps) $\mathbf{d}^l \in \mathbf{L}^\nu$ and $\mathbf{d}^n \in \mathbf{L}^\nu$, which are obtained from Section 3.3, the fusion move combines these two labelings by using the label (disparity) of each pixel either from $\mathbf{d}^l$ or $\mathbf{d}^n$. In this case, the fusion move can be expressed as a binary-labeled MRF minimization problem. This auxiliary binary-labeled MRF can be optimized efficiently using non-submodular graph cuts, yielding a new combined labeling with decreased energy, i.e., a superior solution (disparity map). Moreover, the problem is a binary-labeled one, which is more efficient than a traditional multiple-labeled MRF problem.

To formulate the problem more formally, a combination $\mathbf{d^c}$ is defined by an auxiliary binary vector $\mathbf{y} \in \{0,1\}^\nu$, such that the following:

$$\mathbf{d}^c(\mathbf{y}) = \mathbf{d}^l \cdot (\mathbf{1} - \mathbf{y}) + \mathbf{d}^n \cdot \mathbf{y} \tag{13}$$

where $\cdot$ denotes the Hadamard product, and $\mathbf{1}$ denotes the full one vector. For instance, for each pixel p, if $y_p = 0$, then $d_p^c(y_p) = d_p^l$, and if $y_p = 1$, then $d_p^c(y_p) = d_p^n$. Equation (9) can be reformulated as the following [14]:

$$E^f(\mathbf{d}) = E(\mathbf{d}^c(\mathbf{y})) = \sum_{p \in \nu} U_p^f(y_p) + \sum_{p,q \in \tau} V_{pq}^f(y_p, y_q) \tag{14}$$

where $U_p^f(i) = U_p(d_p^i), V_{pq}^f(i,j) = V_{pq}(d_p^i, d_q^j)$. We minimize Equation (14) using QPBO-graph cuts [11] and compute the resulting labeling $\hat{\mathbf{y}}$. Once $\hat{\mathbf{y}}$ is obtained, the new disparity map $\hat{d}(\mathbf{p})$ is obtained based on Equation (13), as follows:

$$\hat{d}(\mathbf{p}) = d^l(\mathbf{p}) \cdot (1 - \hat{y}_p) + d^n(\mathbf{p}) \cdot \hat{y}_p \tag{15}$$

In addition, when the difference between the local disparity and the non-local disparity of each pixel is less than or equal to one, we compute the average value of these two disparities as stated in Section 3.4.1. Considering this case and Equation (15), the final disparity of each pixel is computed as follows:

$$\hat{d}(\mathbf{p}) = \begin{cases} \frac{d^l(\mathbf{p}) + d^n(\mathbf{p})}{2} & if \ \left| d^l(\mathbf{p}) - d^n(\mathbf{p}) \right| \leq 1 \\ d^l(\mathbf{p}) & else \ if \ \hat{y}_p = 0 \\ d^n(\mathbf{p}) & else \ if \ \hat{y}_p = 1 \end{cases} \tag{16}$$

3.5 Texture-based Sub-pixel Refinement

The disparity map obtained in Section 3.4 will be refined to improve the accuracy of the stereo matching. The sub-pixel refinement is used to compute the disparities with a floating-point precision. A quadratic curve is fitted through the neighboring costs and the new disparity map is obtained as follows [3]:

$$d^*(\mathbf{p}) = d - \frac{C_{tex+}(\mathbf{p}, d) - C_{tex-}(\mathbf{p}, d)}{2(C_{tex+}(\mathbf{p}, d) - 2C_{tex}(\mathbf{p}, d) + C_{tex-}(\mathbf{p}, d))} \tag{17}$$

where $d = \hat{d}(\mathbf{p})$ and

$$C_{tex}(\mathbf{p}, d) = \alpha_t C_l(\mathbf{p}, d) + (1 - \alpha_t) C_n(\mathbf{p}, d) \tag{18}$$

In this refinement, we also consider the texture information to compute the refined disparity map, where α_t is defined as in Equation (11) and $C_{tex+}(\mathbf{p}, d) = C_{tex}(\mathbf{p}, d + 1)$ and $C_{tex-}(\mathbf{p}, d) = C_{tex}(\mathbf{p}, d - 1)$. The final disparity map of the proposed method is $d^*(\mathbf{p})$, and no other postprocessing refinement is conducted.

4 Experimental Results and Discussion

The MC-CNN-acrt matching cost computation [49] is executed on a desktop personal computer equipped with an Nvidia Titan X graphics card. The other components of our stereo algorithm are executed on a personal computer with an Intel(R) Core(TM) i5-4590 CPU with 3.30 GHz and 16 GB of RAM. The Middlebury [32] training dataset is used. This dataset contains 15 high-resolution image pairs. It contains different types of challenging problems for stereo matching. Different aspects of the proposed method are evaluated in the following sub-sections.

4.1 Robustness of the Proposed Method

In this paper, the proposed method utilizes local and non-local algorithms and combines these two disparity maps to generate a more accurate disparity map. Due to the limited space, we compare the proposed method with state-of-the-art local and non-local algorithms against Middlebury benchmark 3.0 [32]. We implement four individual stereo algorithms that can generate accurate disparity maps, i.e., local ones, Cross-based Cost Aggregation (CBCA) in [18], Guided Filter (GF) in [36], and non-local ones, Non-local cost aggregation (NL) in [44], and Segment-tree-based cost aggregation (ST) in [17]. We collect the abbreviation of these algorithms in Table 1 for clarity. These algorithms can form four combinations in our adaptive disparity computation frame: (1) CBCA and NL, (2) CBCA and ST, (3) GF and NL, and (4) GF and ST. The source code for these four algorithms is provided by the authors. To compare the effect of the adaptive computation, no refinement steps are included in this experiment.

We tabulate the data based on the default criterion "bad 2.0" in non-occluded regions for our evaluation. In this performance measure, the lower the value, the more superior the result. The individual local and non-local algorithms, the adaptive disparity computation based on texture information (Section 3.4.1), and the

Table 1: Abbreviation for different algorithms

CBCA	Cross-based Cost Aggregation
GF	Guide Filter
NL	Non-local Cost Aggregation
ST	Segment-tree-based Cost Aggregation
(T)	Disparity selection via texture information
(F)	Disparity selection via fusion move

adaptive disparity computation based on the fusion move (Section 3.4.2) are implemented and tested, and the results are collected in Table 2. For instance, CBCA+NL(T) denotes the adaptive disparity computation that utilizes the combination of CBCA and NL, and selects disparities based on the texture information, while CBCA+NL(F) denotes the adaptive disparity computation that selects disparities based on the fusion move. It is clear that regardless of which disparity selection strategy is used, the adaptive disparity computation can always generate more accurate results than the individual local or non-local algorithms. The adaptive disparity computation based on texture information generates slightly less accurate results than that based on the fusion move, but this strategy is simple and efficient. The selection strategy based on the fusion move can generate a more accurate disparity map because this strategy minimizes an energy function defined on the entire image, providing a more sound result.

To intuitively show the results achieved by the proposed approach, some visualized results are shown in Fig. 3. Here the results of the GF local method, the NL non-local method, and the combination of GF+NL(F) are compared for instance. As shown in Fig. 3, the local method does not handle well in the regions which are with weak textures, and the non-local method tends to compute disparities wrongly in the regions which have abundant local details. The proposed method gives more reasonable disparities in these regions.

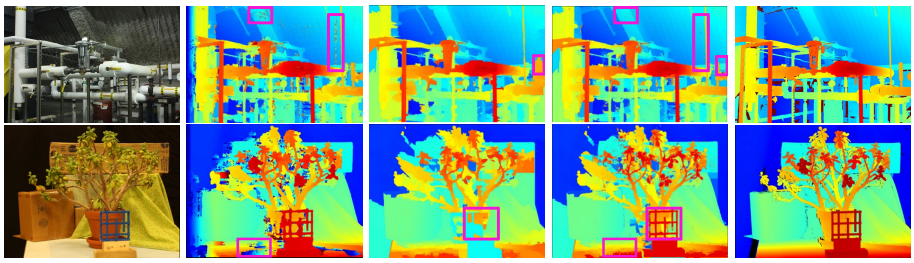

Fig. 3 Some qualitative results of Middlebury dataset 2014. From left to right: left images, disparity maps of the local method, disparity maps of the non-local method, disparity maps of the proposed method, disparity maps of ground truth.

Table 2: Comparison of the local algorithm, the non-local algorithm, and the related combinations in the proposed method against the Middlebury benchmark 3.0. The best performance is boldfaced, where the error criterion of "bad 2.0" is adopted

	Andiron	ArtL	Jadepl	Motor	MotorE	Piano	PianoL	Pipes
CBCA	4.83	6.90	16.4	4.23	4.62	13.8	19.4	4.67
GF	4.57	6.86	16.3	4.18	4.55	13.7	19.1	4.48
NL	4.78	6.90	16.4	4.22	4.58	13.9	19.3	4.57
ST	4.81	6.89	16.4	4.23	4.58	13.9	19.3	4.56
CBCA+NL(T)	4.57	6.87	16.4	4.17	4.54	13.7	19.1	4.60
GF+NL(T)	4.41	6.84	16.3	4.14	4.50	13.7	19.0	4.44
CBCA+ST(T)	4.51	6.88	16.3	4.14	4.50	13.7	19.1	4.57
GF+ST(T)	4.38	6.87	16.3	4.12	4.47	13.7	19.0	4.42
CBCA+NL(F)	4.52	6.77	16.3	4.12	4.50	13.7	19.1	4.51
GF+NL(F)	4.35	6.98	16.3	4.14	4.50	13.7	**18.9**	4.38
CBCA+ST(F)	4.41	**6.69**	**16.2**	**4.05**	**4.40**	13.7	19.0	4.39
GF+ST(F)	**4.33**	6.84	16.3	4.06	4.41	**13.6**	**18.9**	**4.32**

	Playrm	Playt	PlaytP	Recyc	Shelvs	Teddy	Vintge	Average
CBCA	16.4	16.5	14.8	8.04	32.1	3.67	25.8	11.0
GF	16.2	16.3	14.5	7.85	31.9	3.67	25.4	10.8
NL	16.4	16.5	14.8	7.97	31.9	3.66	26.6	11.0
ST	16.4	16.5	14.8	7.99	31.9	3.66	26.9	11.0
CBCA+NL(T)	16.2	16.3	14.6	7.79	31.8	3.65	25.7	10.8
GF+NL(T)	**16.1**	16.2	14.4	7.74	31.7	3.66	**25.3**	**10.7**
CBCA+ST(T)	**16.1**	16.3	14.6	7.74	31.6	3.63	26.3	10.8
GF+ST(T)	**16.1**	16.2	14.4	7.69	**31.5**	3.64	25.9	**10.7**
CBCA+NL(F)	**16.1**	16.3	14.5	7.76	31.8	3.64	25.7	10.8
GF+NL(F)	**16.1**	**16.1**	**14.3**	7.70	31.6	3.70	**25.3**	**10.7**
CBCA+ST(F)	**16.1**	16.3	14.5	7.69	**31.5**	**3.59**	26.4	**10.7**
GF+ST(F)	**16.1**	16.2	14.4	**7.66**	**31.5**	3.62	26.0	**10.7**

4.2 Comparison with Conventional Stereo Methods

The proposed method is also compared with some state-of-the-art stereo methods listed in the Middlebury benchmark 3.0. The results are shown in Table 3. Here, we use the combination GF+NL(F) as the representative algorithm of the proposed frame, and utilize the disparity refinement in Section 3.5 to obtain the final disparity map. The criteria "bad 0.5", "bad 1.0", "bad 2.0", and "bad 4.0" are adopted for evaluations. The proposed method achieves a state-of-the-art result and currently ranks tenth in the criterion "bad 4.0", ninth in "bad 2.0" , eighth in "bad 1.0", and seventh in "bad 0.5" among 67 algorithms. When the error tolerance between the disparity obtained by the stereo algorithm and the ground truth is low, i.e., the performance measure of the stereo algorithm is strict, the proposed method ranks higher, indicating the proposed method is compatible with accurate stereo matching. Fig. 4 illustrates some qualitative results of the proposed method under the error of "bad 2.0". The result of MotorE ranks second among all the algorithms, demonstrating that the proposed method can generate robust stereo matching in the presence of large exposure variations. The result of Pipes ranks fourth among all the algorithms, demonstrating that the method is able to successfully process depth discontinuities and slim foreground objects.

Table 3: Comparison of the proposed method with the state-of-the-art stereo methods against the Middlebury benchmark 3.0

	bad 0.5	bad 1.0	bad 2.0	bad 4.0
Proposed method	40.0	18.4	9.89	6.10
LocalExp [38]	36.8	13.7	6.52	4.07
3DMST [15]	38.0	15.1	7.08	4.43
APAP-Stereo [26]	49.5	20.9	7.53	4.50
FEN-D2DRR [45]	40.1	16.7	7.89	3.98
PMSC [16]	39.5	16.4	8.20	5.15
LW-CNN [25]	40.0	16.6	8.31	4.89
MeshStereoExt [51]	41.5	18.4	9.32	5.53
MCCNN-Layout	39.1	18.0	9.34	5.21
NTDE [10]	41.7	18.1	9.94	6.13
MC-CNN-acrt [49]	39.8	18.4	10.1	6.34
MC-CNN+TDSR [5]	42.1	19.8	10.2	5.99

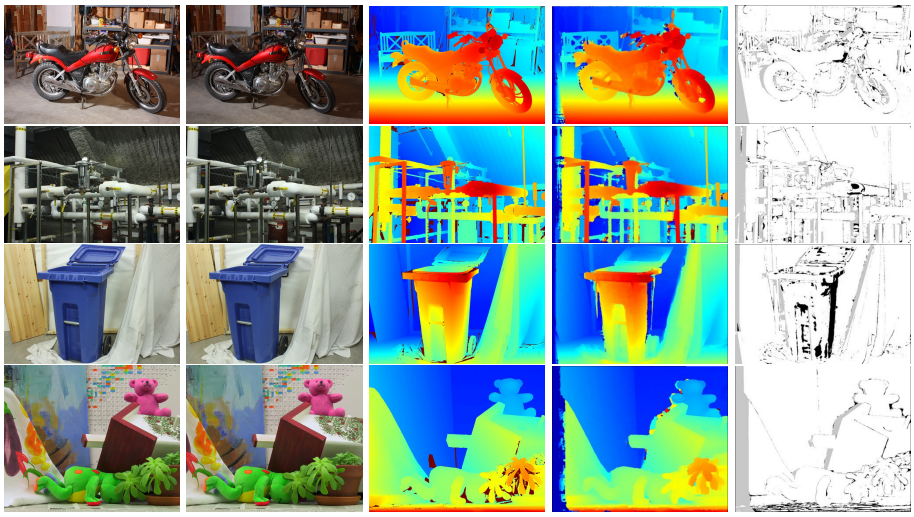

Fig. 4 Some implementation results against the Middlebury benchmark 3.0. From left to right: left images, right images, disparity maps of ground truth, disparity maps of the proposed method, and "bad 2.0" error maps of the proposed method. From top to bottom: MotorcycleE, Pipes, Recycle, and Teddy

4.3 Efficiency of Fusion Moves

In this sub-section, we evaluate the results of different numbers of fusions. We use four disparity maps generated by CBCA, GF, NL, and ST as the inputs. The first two disparity maps are chosen randomly, but we require that one be a local disparity map and the other a non-local disparity map. The unary term is computed in the first fusion step and is kept invariant in the following fusion steps. Next, the remaining disparity maps are visited in a random order, and each of them is fused with the current disparity map in sequence. Finally, several fusion strategies are obtained and their accuracies are tested. The criterion "bad 2.0" is used to compare the results of different fusion strategies. As illustrated in Fig. 5, it

is clear that the accuracy of fusing two disparity maps and fusing more disparity maps is similar. This result shows that the proposed method is efficient in the fusion move step since it can generate a reasonable disparity map without fusing many disparity maps.

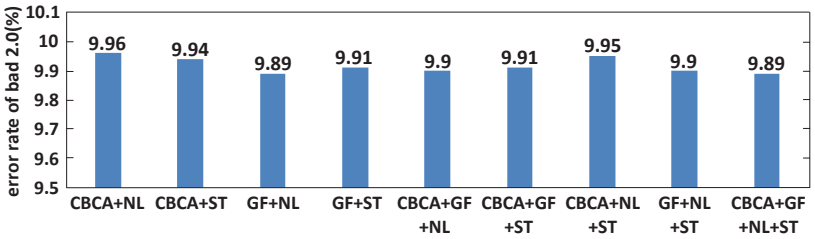

Fig. 5 Comparison of different fusion strategies. The proposed method can achieve satisfactory results with only one fusion step

4.4 Effect of Texture-based Sub-pixel Refinement

In Section 3.5, the costs with textured information are used in the sub-pixel refinement step. The effect of texture-based sub-pixel refinement is tested in this sub-section. We replace C_{tex} in Equation (18) with the costs obtained via the local and non-local algorithms. Let FL denote sub-pixel refinement with the costs obtained by the local algorithm and FN denote sub-pixel refinement with the costs obtained by the non-local algorithm. The average error rates "bad 2.0" of the three sub-pixel refinement methods are illustrated in Fig. 6. The texture-based sub-pixel refinement always achieves superior performances among the three approaches since the costs with texture information are more robust, which is crucial in the sub-pixel refinement step.

4.5 Runtime

The runtime of each section of our stereo algorithm is also tested. The average runtime of the 15 training datasets of the Middlebury 2014 dataset is 103.22s for the CBCA, 146.40s for the GF, 15.0s for the MST, 15.5s for the ST, and 10.34s for the sub-pixel refinement. The average runtime for selection via texture information is 0.0029s, and 5.27s for selection via the fusion move.

5 Conclusion

An adaptive disparity computation algorithm based on local and non-local cost aggregations is proposed and shown to enhance the performance of stereo matching. Experimental results show that our algorithm achieves superior results to those of

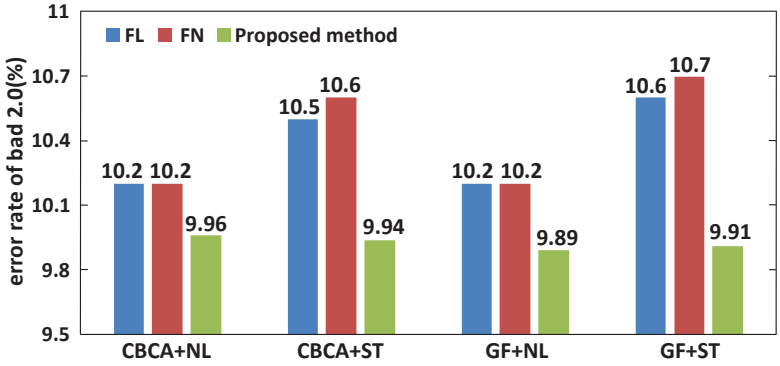

Fig. 6 Effect of the texture-based sub-pixel refinement. The proposed method always achieves the most accurate results with different stereo algorithms

local or non-local algorithms individually. Moreover, our algorithm is a framework for local and non-local algorithms, and the experimental results show that the proposed method can enhance performance independent of the cost aggregation functions. For different local and non-local algorithms, the proposed method is always able to combine them in a sound manner and select the more reasonable disparity to achieve accurate stereo matching. The questions of how to select the superior disparity and how to balance the accuracy and number of initial proposals for fusion moves remain unanswered questions, and we will study these aspects more deeply in the future.

6 Acknowledgments

The authors would like to thank Qing Ran for her instructive discussion of this paper. This work was supported by the National Natural Science Foundation of China under Grants Nos. 61472349 and 61732015.

References

1. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo - stereo matching with slanted support windows. In: British Machine Vision Conference, pp. 14.1–14.11 (2011)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(11), 1222–1239 (2001)
3. Brockers, R., Hund, M., Mertsching, B.: Stereo vision using cost-relaxation with 3d support regions. Cortex **9**, 11 (2005)
4. Crouzil, A., Massip-Pailhes, L., Castan, S.: A new correlation criterion based on gradient fields similarity. In: International Conference on Pattern Recognition, vol. 1, pp. 632–636. IEEE (1996)
5. Drouyer, S., Beucher, S., Bilodeau, M., Moreaud, M., Sorbier, L.: Sparse stereo disparity map densification using hierarchical image segmentation. In: International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, pp. 172–184 (2017)
6. Ghaleb, F.F., Youness, E.A., Elmezain, M., Dewdar, F.S.: Vision-based hand gesture spotting and recognition using crf and svm. Journal of Software Engineering and Applications **8**(07), 313 (2015)

7. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Computer Vision and Pattern Recognition, vol. 2, pp. 807–814. IEEE (2005)
8. Huang, X., Yuan, C., Zhang, J.: Graph cuts stereo matching based on patch-match and ground control points constraint. In: Pacific Rim Conference on Multimedia, pp. 14–23. Springer (2015)
9. Jiao, J., Wang, R., Wang, W., Dong, S., Wang, Z., Gao, W.: Local stereo matching with improved matching cost and disparity refinement. IEEE MultiMedia **21**(4), 16–27 (2014)
10. Kim, K.R., Kim, C.S.: Adaptive smoothness constraints for efficient stereo matching using texture and edge information. In: IEEE International Conference on Image Processing, pp. 3429–3433 (2016)
11. Kolmogorov, V., Rother, C.: Minimizing nonsubmodular functions with graph cuts-a review. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(7), 1274 (2007)
12. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: IEEE Conference on Computer Vision, vol. 2, pp. 508–515. IEEE (2001)
13. Kong, D., Tao, H.: A method for learning matching errors for stereo computation. In: British Machine Vision Conference, vol. 1, p. 2 (2004)
14. Lempitsky, V., Rother, C., Roth, S., Blake, A.: Fusion moves for markov random field optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(8), 1392 (2010)
15. Li, L., Yu, X., Zhang, S., Zhao, X., Zhang, L.: 3d cost aggregation with multiple minimum spanning trees for stereo matching. Applied Optics (2017)
16. Li, L., Zhang, S., Yu, X., Zhang, L.: Pmsc: Patchmatch-based superpixel cut for accurate stereo matching. IEEE Transactions on Circuits and Systems for Video Technology (2016)
17. Mei, X., Sun, X., Dong, W., Wang, H., Zhang, X.: Segment-tree based cost aggregation for stereo matching. In: Computer Vision and Pattern Recognition, pp. 313–320 (2013)
18. Mei, X., Sun, X., Zhou, M., Jiao, S., Wang, H., Zhang, X.: On building an accurate stereo matching system on graphics hardware. In: IEEE Conference on Computer Vision, pp. 467–474. IEEE (2011)
19. Miyazaki, D., Matsushita, Y., Ikeuchi, K.: Interactive shadow removal from a single image using hierarchical graph cut pp. 234–245 (2009)
20. Mizukami, Y., Okada, K., Nomura, A., Nakanishi, S.: Sub-pixel disparity search for binocular stereo vision. In: International Conference on Pattern Recognition, pp. 364–367 (2012)
21. Narducci, F., Ricciardi, S., Vertucci, R.: Enabling consistent hand-based interaction in mixed reality by occlusions handling. Multimedia Tools and Applications **75**(16), 9549–9562 (2016)
22. Ogawara, K.: Approximate belief propagation by hierarchical averaging of outgoing messages. In: International Conference on Pattern Recognition, pp. 1368–1372 (2010)
23. Olsson, C., Ulen, J., Boykov, Y.: In defense of 3d-label stereo. In: Computer Vision and Pattern Recognition, pp. 1730–1737 (2013)
24. Ošep, A., Hermans, A., Engelmann, F., Klostermann, D., Mathias, M., Leibe, B.: Multi-scale object candidates for generic object tracking in street scenes. In: Robotics and automation (icra), 2016 ieee international conference on, pp. 3180–3187. IEEE (2016)
25. Park, H., Lee, K.M.: Look wider to match image patches with convolutional neural networks. IEEE Signal Processing Letters (2016)
26. Park, M., Yoon, K.: As-planar-as-possible depth map estimation. IEEE Transactions Pattern Anal (2016)
27. Peng, Y., Li, G., Wang, R., Wang, W.: Stereo matching with space-constrained cost aggregation and segmentation-based disparity refinement. In: Three-Dimensional Image Processing, Measurement (3DIPM), and Applications, p. 939309 (2015)
28. Psota, E.T., Kowalczuk, J., Mittek, M., Prez, L.C.: Map disparity estimation using hidden markov trees. In: IEEE International Conference on Computer Vision (2016)
29. Rameau, F., Ha, H., Joo, K., Choi, J., Park, K., Kweon, I.S.: A real-time augmented reality system to see-through cars. IEEE transactions on visualization and computer graphics **22**(11), 2395–2404 (2016)
30. Rhemann, C., Hosni, A., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. In: Computer Vision and Pattern Recognition, pp. 3017–3024 (2011)
31. Scharstein, D.: Matching images by comparing their gradient fields. In: International Conference on Pattern Recognition, vol. 1, pp. 572–575. IEEE (1994)

32. Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., Westling, P.: High-resolution stereo datasets with subpixel-accurate ground truth. In: German Conference on Pattern Recognition, pp. 31–42. Springer (2014)
33. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision **47**(1-3), 7–42 (2002)
34. Shu, X., Qi, G.J., Tang, J., Wang, J.: Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation. In: ACM International Conference on Multimedia, pp. 35–44 (2015)
35. Suarez, J., Murphy, R.R.: Hand gesture recognition with depth images: A review. In: Ro-man, 2012 IEEE, pp. 411–417. IEEE (2012)
36. Tan, P., Monasse, P.: Stereo disparity through cost aggregation with guided filter. Image Processing On Line **4**, 252–275 (2014)
37. Taniai, T., Matsushita, Y., Naemura, T.: Graph cut based continuous stereo matching using locally shared labels. In: Computer Vision and Pattern Recognition, pp. 1613–1620 (2014)
38. Taniai, T., Matsushita, Y., Sato, Y., Naemura, T.: Continuous stereo matching using local expansion moves. Computer Vision and Pattern Recognition (2016)
39. Tian, Y., Long, Y., Xia, D., Yao, H., Zhang, J.: Handling occlusions in augmented reality based on 3d reconstruction method. Neurocomputing **156**, 96–104 (2015)
40. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 5 (2017)
41. Vu, D.T., Chidester, B., Yang, H., Do, M.N., Lu, J.: Efficient hybrid tree-based stereo matching with applications to postcapture image refocusing. IEEE Transactions on Image Processing **23**(8), 3428–3442 (2014)
42. Wang, L., Yang, R., Gong, M., Liao, M.: Real-time stereo using approximated joint bilateral filtering and dynamic programming. Journal of Real-Time Image Processing **9**(3), 447–461 (2014)
43. Woodford, O.J., Torr, P.H.S., Reid, I.D., Fitzgibbon, A.W.: Global stereo reconstruction under second order smoothness priors. In: Computer Vision and Pattern Recognition, pp. 1–8 (2008)
44. Yang, Q.: A non-local cost aggregation method for stereo matching. In: Computer Vision and Pattern Recognition, pp. 1402–1409. IEEE (2012)
45. Ye, X., Li, J., Wang, H., Huang, H., Zhang, X.: Efficient stereo matching leveraging deep local and context information. IEEE Access (2017)
46. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(4), 650–656 (2006)
47. Yu, T., Lin, R.S., Super, B., Tang, B.: Efficient message representations for belief propagation. In: IEEE Conference on Computer Vision, pp. 1–8. IEEE (2007)
48. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: European Conference on Computer Vision, pp. 151–158. Springer (1994)
49. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research **17**, 1–32 (2016)
50. Zhan, Y., Gu, Y., Huang, K., Zhang, C., Hu, K.: Accurate image-guided stereo matching with efficient matching cost and disparity refinement. IEEE Transactions on Circuits and Systems for Video Technology (2015)
51. Zhang, C., Li, Z., Cheng, Y., Cai, R.: Meshstereo: A global stereo model with mesh alignment regularization for view interpolation. In: IEEE International Conference on Computer Vision, pp. 2057–2065 (2015)
52. Zhang, K., Lu, J., Lafruit, G.: Cross-based local stereo matching using orthogonal integral images. IEEE Transactions on Circuits and Systems for Video Technology **19**(7), 1073–1079 (2009)
53. Zhou, C., Zhang, H., Shen, X., Jia, J.: Unsupervised learning of stereo matching. In: IEEE International Conference on Computer Vision, pp. 1576–1584 (2017)
54. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video pp. 6612–6619 (2017)