IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 30, 2020, accepted March 29, 2020, date of publication April 2, 2020, date of current version April 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985106

CrossPatch-Based Rolling Label Expansion for Dense Stereo Matching

HUAIYUAN XU, XIAODONG CHEN, HAITAO LIANG, SIYU REN, YI WANG, AND HUAIYU CAI

Key Laboratory of Opto-Electronics Information Technology, Tianjin University, Tianjin 300072, China

Corresponding author: Xiaodong Chen (xdchen@tju.edu.cn)

This work was supported in part by the National Major Project of Scientific and Technical Supporting Programs of China during the 13th Five-Year Plan Period under Grant 2017YFC0109702, Grant 2017YFC0109901, and Grant 2018YFC0116202.

ABSTRACT We present a novel algorithm called crosspatch-based rolling label expansion for accurate stereo matching. This optimization-based approach can effectively estimate the 3D label of each pixel from huge and infinite label space and then generate a continuous disparity map. The algorithm has two obvious characteristics when compared with the traditional label expansion algorithms. The first feature is the cross-based multilayer structure, where each layer contains a series of cross patches with adaptive shapes, reflecting the edge structure of objects on the image. Besides, such cross patches are non-overlapping and independent, satisfying the submodular property for employing graph cuts. The second feature is the rolling optimization, that firstly generates new label proposal by expanding candidate labels within cross patches, then globally updates labels for the whole image using a proposed rolling move. The experimental results show the high matching accuracy of our method, both in pixel level and subpixel level. According to the latest ranking list of Middlebury 3.0 benchmark, our method is one of the best stereo matching algorithms.

INDEX TERMS Stereo matching, label expansion, PatchMatch, rolling optimization, cross-based multilayer structure.

I. INTRODUCTION

Stereo matching is one basic task of computer vision, whose goal is to estimate disparity when inputting an image pair [1], which is widely applied in navigation [2], 3D construction [3] and virtual viewpoint imaging [4]. Estimating accurate and continuously varying disparities is the key to stereo matching, and many related algorithms have been proposed and they can output dense and continuous disparity map [5]–[7]. These methods estimate successive disparities by assigning continuous 3D labels [5] to neighboring pixels, and then mapping the labels to disparities. Compared with traditional discrete 1D label [8], using the 3D label can free from the fronto-parallel bias [5], leading to a higher matching accuracy.

Given the 3D label $f_p = \left(a_p, b_p, c_p\right)$ of each pixel p, the disparity d_p can be uniquely determined by the ternary primary function

$$d_p = a_p u + b_p v + c_p. \tag{1}$$

Here (u, v) are the coordinates in the image domain. However, since the 3D label space ($\mathbf{R}^3$) is huge and infinite, two difficulties occur during label assignment, namely, how to reduce the search space of candidate labels to decrease the computational complexity, and how to assign labels accurately.

In order to reduce the search space, a useful technique is spatial nearby propagation [5], that is inspired by approximate nearest-neighbor field in PatchMatch [9], [10] and based on the fact that spatial neighboring pixels are likely to have similar labels in stereo matching. This technique updates the label of each pixel successively, and constrains the search region in a limited label space centered on the label of the previous pixel. Although this raster-scan order search mode can reduce the search space to some degree, the amount of computation is still very huge when dealing with high-resolution images. To this end, an effective solution is to introduce segmentation information [7], [11], [12]. For example, when updating labels, the pixels in the same superpixel [13] share the same candidate label space, which is based on the assumption that the pixels in the same superpixel belong to a same continuous 3D surface and have similar 3D labels.

The associate editor coordinating the review of this manuscript and approving it for publication was Ivan Lee.

In this work, we also consider the segmentation information and adopt a simpler way than superpixel, which can effectively cooperate with the proposed label optimization process.

In order to assign 3D labels accurately, one successful way is to combine second-order disparity smoothness [14], [15] and PatchMatch Stereo [5] to find the optimal label proposal (consisting of labels of the interest pixels) via constructing pairwise Markov random field (MRF) [16]. To solve the above optimization problem, there are several effective optimizers, such as graph cuts (GC) [17], [18] and belief propagation (BP) [19], [20]. BP, as a sequential optimizer, keeps the labels of other pixels fixed when estimating the label of target pixel, that sometimes falls into the local minimum problem. On the contrary, GC estimates the labels of all pixels at the same time, so it has a global property and is more suitable and accurate for label assignment.

In this paper, we propose a crosspatch-based rolling label expansion algorithm. It belongs to global optimization algorithms [21] of stereo matching and it combines the advantages of PatchMatch Stereo [5], segmentation [22] and GC [17]. Firstly, we construct a cross-based multilayer structure that contains cross patches [22]–[25] with segmentation information. The pixels in each cross patch share a candidate label space, so their 3D labels can be updated at the same time. Secondly, we use a rolling move to optimize the label proposal as well as transfer the updated labels to neighboring pixels. Through the rolling move, the globally optimized label proposal can be obtained. Our approach has the following advantages. 1) On each layer of the cross-based multilayer structure, all cross patches are independent, that not only ensures the submodular requirement [26], [27] for graph cuts [17], but also makes use of the multi-core parallel operation of CPU to accelerate the algorithm. 2) During the rolling move, the algorithm regards cross patch as a mask to discard the updated label results near the edge of the object, which strengthens the updating constraint and improves the accuracy of label estimation. 3) Due to the good global properties of graph cuts and rolling move, our method can effectively suppress the local minimum errors when optimizing labels.

Overall, our contributions are mainly threefold. 1) We propose a cross-based multilayer structure, which contains the object edge information. The cross patches of each layer of the structure are independent and do not overlap each other. 2) We design a rolling optimization strategy to update labels globally. The optimization process is divided into two stages: coarse optimization and fine-grained stage. And the cross-based multilayer structure is used to guide the optimization. 3) According to the disparity map accuracy, our algorithm is one of the best stereo matching methods in new Middlebury 3.0 benchmark; especially, its performances rank first in pixel and subpixel levels among published methods of the benchmark.

In the rest of this paper, Section II reviews the related work. Section III describes the proposed label expansion algorithm in detail from formulation, cross patch construction, cross-based multilayer structure and optimization procedure. Section IV shows the performance of the proposed method, comparative experiments results as well as qualitative and quantitative analysis. Section V concludes the whole paper.

II. RELATED WORK

A. GLOBAL STEREO MATCHING FOR HIGH-RESOLUTION IMAGES

Global stereo matching transforms the disparity estimation task into an optimization problem based on a pairwise MRF [16]. Specifically, it looks for the label mapping f that minimizes the following energy functions [14], [17], [28],

$$E(f) = E_{data}(f) + \lambda E_{smooth}(f). \tag{2}$$

Here, data term $E_{data}(f)$ measures the photo-difference between matching pixels, smoothness term $E_{smooth}(f)$ punishes the disparity discontinuity of adjacent pixels and a coefficient λ is used to balance data term and smoothness term. The traditional global stereo matching methods [18], [20] are very time-consuming when dealing with high-resolution images. To solve this problem, two effective techniques have been used in recent years. one is the fusion move [29], that combines label proposals generated from cheap-to-compute solutions and then gets a better proposal by binary fusion. However, the candidate label proposals are not accurate enough, which unavoidably affects the accuracy of the final fusion result [7]. The second is spatial propagation [5], [30], [31], that first optimizes some labels of pixels and then propagates the updated labels to adjacent pixels. This technique, also utilized in our algorithm, can effectively reduce the label search space of adjacent pixels and speed up the algorithm. In addition, it can combine with the global optimizer (e.g., GC) to avoid some local minimum errors and obtain the accurate label proposal.

B. GLOBAL PATCHMATCH-BASED METHODS

Global stereo matching methods, based on spatial propagation, are also known as global PatchMatch-based methods [6], [7], [32]. The basic method of them, PatchMatch Stereo [5], can't deal well with the noise and low texture areas in the image. That is because PatchMatch Stereo is a local stereo method and it lacks the attention to the smoothness constraints of adjacent pixel labels. To overcome this deficiency, our algorithm, as well as other global PatchMatch-based methods, adds a smoothness term constraint to the energy function to keep the neighboring labels and disparities continuous in the image.

For further improvement of stereo matching accuracy, some global PatchMatch-based methods propose to employ image segmentation information, such as superpixel [13]. Compared with the fixed window [30], superpixel has a free contour retaining the edge structure of the object. It means that a superpixel has less noise and it is easier to obtain accurate labels during optimization. A specific example is PMSC [7], whose idea is close to our algorithm. It firstly uses simple linear iterative clustering (SLIC) [33] to construct a multilayer superpixel structure, then generates a series of

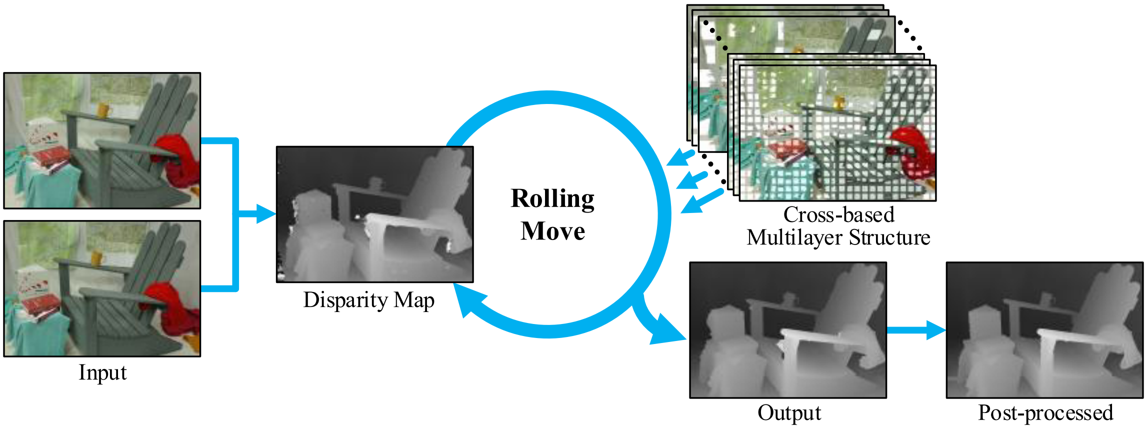

FIGURE 1. The pipeline of the proposed method. Our algorithm inputs an image pair, circularly updates the label of each pixel according to the cross-based multilayer structure, and outputs the post-processed disparity map.

label proposals by random sampling, and finally fuses them. However, in PMSC, the process of generating label proposals is separate from the process of fusion, which results in that the label information cannot be updated in time during fusion. By contrast, our method employs the iterative idea and every time we generate a new label proposal, it is based on the previous one. Under this mechanism, accurate labels can be quickly and thoroughly transferred on the image by our rolling move.

III. PROPOSED METHOD

The proposed method uses rolling optimization to iteratively optimize label proposal. During each iteration, we take one layer and its cross patches from the cross-based multilayer structure, and then update the label proposal within cross patches. After a series of iterations, the label proposal is globally updated. And the disparity map can be calculated by the final label proposal in terms of the relationship between label and disparity (Eq.1). Fig.1 shows the pipeline of the proposed method, that inputs an image pair and outputs its disparity map with post process. Section III-A to section III-D are going to describe the steps of the proposed method in detail.

A. FORMULATION

Our goal is to estimate the optimal 3D label proposal f by minimizing the pairwise MRF energy function, which is described as Eq.2 in Section II and composed of data term and smoothness term (that are also known as unary term and pairwise term in some other works of literature [14], [27]).

1) DATA TERM

To evaluate the photo-difference between matching pixels, we define the data term $E_{data}(f)$ as the sum of matching costs of all pixels in image domain Ω,

$$E_{data}(f) = \sum_{p \in \Omega} \Phi_p(f_p). \tag{3}$$

The cost function $\Phi_p(f_p)$ is defined as Eq.4 using a mainstream and effective manner [34], [35], that combines the advantages of the similarity prediction via CNN [25], [36] and the slanted patch matching [5]: CNN has good robustness for different situations, and slanted patch matching has subpixel accuracy.

$$\Phi_p(f_p) = \sum_{s \in W_p} \omega_{ps} \min\left(C_{CNN}(s, s'), \tau_{CNN}\right). \tag{4}$$

Here, W_p is a square window centered on pixel p. Function $C_{CNN}(s, s')$ calculates the dissimilarity between support pixel $s = (u_s, v_s)$ of W_p in the left image and its matching pixel $s' = s - (a_p u_s + b_p v_s + c_p, 0)$ in the right image. τ_{CNN} is a truncation coefficient to prevent the cost value from being too large when pixels are on some irregular and unsmooth surfaces. The aggregation weight ω_{ps} is the guided image filtering weight [8], [37], [38] with edge-awareness property,

$$\omega_{ps} = \frac{1}{|W'|} \sum_{k:(p,s) \in W'_k} \left(1 + (I_p - \mu_k)^T (\Sigma_k + \varepsilon)^{-1} (I_s - \mu_k)\right). \tag{5}$$

Here, I_p and I_s are normalized 3×1 color vectors. μ_k and Σ_k are the mean and the variance of I_p in the squared support window W'_k, centered on pixel k, where the number of pixels in this window is $|W'|$. ε is a regularization term with a small positive value to avoid ω_{ps} being too large. The guide filter weight has a small computation complexity with a value of $O(1)$, which means it is independent of the size of support window, and it helps to speed up the algorithm.

2) SMOOTHNESS TERM

To evaluate the discontinuity of the disparity map, we define the smoothness term $E_{smooth}(f)$ as a sum of the discontinuity penalties $\Psi_{pq}(f_p, f_q)$ of all adjacent pixel pairs (p, q) in image domain Ω, where the smaller value the smoothness term has,

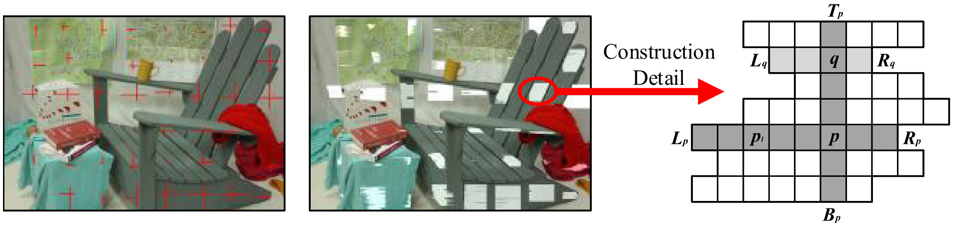

FIGURE 2. The left part shows cross skeletons with randomly selected anchor pixels as the centers. The white regions in the middle image are constructed cross patches. The right image shows one cross patch construction detail: the cross patch is made up of the union of horizontal arms (L_q and R_q), whose center pixel q is on the vertical arms (T_p and B_p) of pixel p.

the more continuous the disparity map is.

$$E_{smooth}(f) = \sum_{p\in\Omega} \sum_{q\in N(p)} \Psi_{pq}\left(f_p, f_q\right), \tag{6}$$

where $N(p)$ represents the neighborhood pixels of pixel p. Discontinuity penalty term $\Psi_{pq}\left(f_p, f_q\right)$ uses a curvature-based second-order smooth regularization, following [7], [14] and defined as

$$\Psi_{pq}\left(f_p, f_q\right) = \max\left(\omega_{pq}, e\right) \min\left(\widetilde{\Psi}_{pq}\left(f_p, f_q\right), \tau_{dis}\right), \tag{7}$$

where the weight ω_{pq} is defined as

$$\omega_{pq} = \exp\left(-\left\|I_L(p) - I_L(q)\right\|_1 / \gamma\right) \tag{8}$$

to describe the similarity of pixels p and q in color space. Here, $\|I_L(p) - I_L(q)\|_1$ calculates the l_1 norm of the difference between p and q in RGB space. γ is a user-defined parameter. Besides, the truncation threshold e limits the lower bound of the weight for increasing robustness. Function $\widetilde{\Psi}_{pq}\left(f_p, f_q\right)$ is defined as

$$\widetilde{\Psi}_{pq}\left(f_p, f_q\right) = \left|d_p\left(f_p\right) - d_p\left(f_q\right)\right| + \left|d_q\left(f_q\right) - d_q\left(f_p\right)\right|, \tag{9}$$

where disparity $d_p\left(f_q\right) = a_q u_p + b_q v_p + c_q$. Function $\widetilde{\Psi}_{pq}\left(f_p, f_q\right)$ calculates the disparity differences corresponding to two different 3D labels $\left(f_p, f_q\right)$ at pixel p and pixel q, that in other words reflects the second-order discontinuity of f_p and f_q. The truncation parameter τ_{dis} limits the upper bound of the weight for reducing the influence of sharp disparity jumps in edge areas.

B. CROSS PATCH CONSTRUCTION

We construct patches with adaptive shapes, which have depth discontinuity awareness. In other words, the pixels in the same patch are approximately considered from the same physical plane, and they have similar 3D labels. Compared with the fixed patch which assumes constant depth within a square region, the adaptive patch is more reasonable and has less interference noise when optimizing the label proposal, helping to obtain a more accurate optimization result.

Here, we use the cross patch as above adaptive patch, because of its good edge awareness and small computation [22]. Such cross patches are centered on a set of anchor pixels on the image. Each anchor pixel is randomly selected in a constrained region (the center region described in Section III-C). The construction method of cross patch is simple. Generally speaking, it begins by forming an adaptive pixel-wise cross that consists of two orthogonal line segments, then uses this cross to construct a support patch. In detail, given a central pixel p as shown in the right part of Fig.2, we construct the left arm L_p of pixel p by a set of pixels, in which $p_l = \left(u_p - l, v_p\right) \in L_p$ satisfies the following rules simultaneously,

$$\left|I(p) - I\left(p_l\right)\right|_i < \eta_{color}, \tag{10}$$

$$\left\|p - p_l\right\|_2 < \eta_{distance}, \tag{11}$$

where $\left|I(p) - I\left(p_l\right)\right|_i$ is the difference between pixels p and p_l of the ith color component, $\|\cdot\|_2$ is the L2 norm representing the distance on the image, and $\left(\eta_{color}, \eta_{distance}\right)$ are user-defined threshold values. Similarly, we construct right arm R_p, bottom arm B_p and top arm T_p of pixel p, constituting a cross skeleton shown in the right part of Fig.2. We extend the cross skeleton to the cross patch $U(p)$ through a union of all horizontal arms

$$U(p) = \left\{L_q \cup R_q | q \in \left(T_p \cup B_p\right)\right\}. \tag{12}$$

These horizontal arms L_q and R_q are constructed around the pixels in top arm T_p and bottom arm B_p. It is obvious that the cross patch contains the segmentation information of the image, that is, the cross patch with non fixed shape can effectively capture the local texture information on the image. Cross patch technology is proved to be effective in the cost aggregation of local stereo matching [23], [25], to optimize the cost value of each pixel. Here, we use this technique for label expansion for the first time. In the following content, we will construct a multilayer structure containing cross patches, and use this structure to globally optimize 3D label assignment.

C. CROSS-BASED MULTILAYER STRUCTURE

In this section, we describe the construction of proposed cross-based multilayer structure. Benefiting from the careful design of the structure, our label expansion method of stereo matching has the advantage of small candidate label space. Because the pixels in the same cross patch have similar labels and they share the same candidate label space. This is more efficient than many other pixel-wise stereo

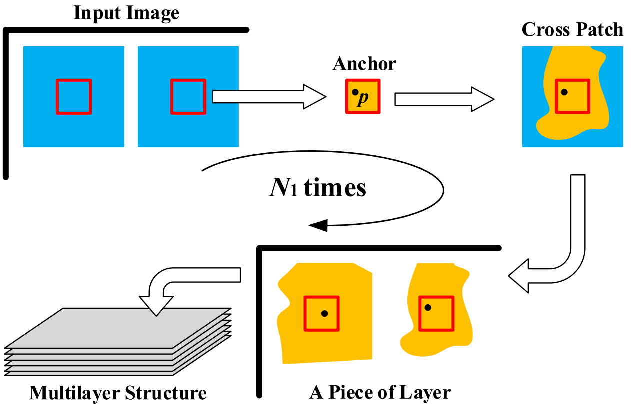

FIGURE 3. The process of cross-based multilayer structure construction. The areas within red boxes are the center regions. Anchor pixel *p* is randomly selected. A cross patch is constructed such as the orange area with *p* as the center, and its size shall not exceed the border region with a blue color. After N_1 times operations on the input image, a cross-based multilayer structure of N_1 layers is output.

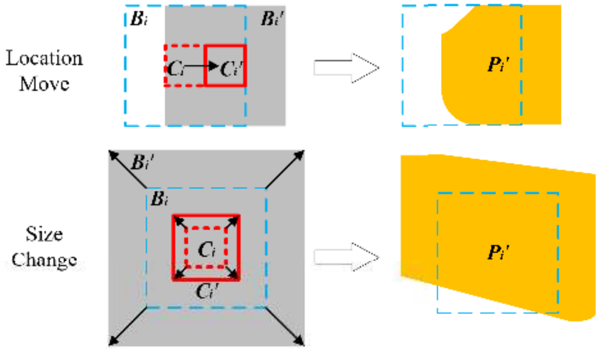

FIGURE 4. Two modes of the changes of center and border regions, namely, location move and size change. The dotted frames correspond to the positions of old central and boundary regions. Areas within red solid boxes and areas in gray are new regions. Cross patches are constructed as orange areas.

Algorithm 1 Construction of Multilayer Structure

```
  Input: image $I$.
  Output: cross-based multilayer structure.
1 for $i = 1, 2, \ldots, N_3$ ($N_3$ is the number of size changes) do
2     for $j = 1, 2, \ldots, N_2$ ($N_2$ is the number of location moves) do
3         Define the center region set $\{\mathbf{C}\}$ and the border region set $\{\mathbf{B}\}$.
4         for $k = 1, 2, \ldots, N_1$ ($N_1$ is the number of the rounds of selecting anchor pixels) do
5             for each $C_i \in \{\mathbf{C}\}$ do
6                 Select an anchor pixel $p_i$.
7                 Construct a cross patch $P_i$
8             Output a cross-based layer.
```

methods [5], [6], [17], which struggle with a large number of label spaces, because each pixel in these methods has a huge and independent candidate label space.

Fig.3 shows the construction detail of our cross-based multilayer structure. Firstly, we define some center regions on the image. Then we define the border regions that do not overlap each other. A border region, composed of a center region and its eight neighboring areas, is to limit the size of cross patch. We randomly select anchor pixels in the center regions, and generate cross patches with anchor pixels as the centers. After such an operation, a piece of layer with non intersecting cross patches is formed. After N_1 times operations on the input image, we obtain a cross-based multilayer structure with N_1 layers.

To increase the types and quantity of the layers in the cross-based multilayer structure, we use two modes to change the center and border regions in each layer. These two modes are namely location move and size change. They are compatible with the spatial propagation technology [5], thus the multilayer structure can be applied in the global label expansion process. In the location move mode, we translate the position of the center region from dotted box to solid box as shown in the first line of Fig.4. Then a new border region B_i' is constructed based on the new central region C_i'. The size change mode is to change the sizes of C_i and B_i as shown in the second line of Fig.4. Correspondingly, the new cross patch P_i' can break through the border limitation set by the previous border region so that updated labels can be expanded much further on the image during optimization process.

The complete construction steps of multilayer structure is shown as Algorithm1. It contains the two change modes described above (see line 1 and line 2 in the algorithm). The input of the algorithm is a color image, and its output is a multilayer structure with $N_1 \times N_2 \times N_3$ layers containing cross patches. Similar to PMSC algorithm [7], our constructed multilayer structure will be also used to guide the label optimization process. In addition, this carefully designed structure meets the submodularity [26], which ensures that α expansion move [17], [18] can be carried out simultaneously in different cross patches of one layer. In other words, α expansion move can reach the global minimum energy in each cross patch only if the pairwise penal term Ψ_{pq} of the energy function Eq.2 meets the following condition:

$$\Psi_{pq}\left(c_p, c_q\right) + \Psi_{pq}\left(f_p, f_q\right) \leq \Psi_{pq}\left(f_p, c_q\right) + \Psi_{pq}\left(c_p, f_q\right), \tag{13}$$

where label c_p is set to α_i which is a candidate label of the cross patch P_i when pixel $p \in P_i$. In another situation where p is located between cross patches and it does not belong to any patch, we assign an infinite cost value to the data term $\Phi_p\left(f_p\right)$ for forcing the pixel to fix its original label, that is, $c_p = f_p$. Since it is easy to prove that Eq.13 relationship is valid for our multilayer structure by introducing Eq.7 and Eq.9 into Eq.13, so we don't present the detailed process of proof here.

Algorithm 2 Optimization Procedure

```
   Input: stereo image pair, cross-based multilayer
          structure.
   Output: optimal label mapping $f$.
1  Initialize the mapping $f$ randomly.
2  // The first-stage rolling optimization $T_1$.
3  for updating round $n = 1, 2, \ldots, T_1$ do
4      Initialize the perturbation $\Delta_i$ randomly.
5      for each layer in multilayer structure do
6          for each $B_i \in \{\mathbf{B}\}$ do
7              $\alpha_i \leftarrow l_{anchor} + \Delta_i$.
8              Do label expansion move $\left(f^B, \alpha_i\right)$.
9          $\Delta_d^{max} \leftarrow \left(\Delta_d^{max}/2\right)$, $\Delta_n^{max} \leftarrow \left(\Delta_n^{max}/2\right)$.
10         if size or location of $\{\mathbf{B}\}$ is changed then
11             $\Delta_d^{max} = maxdisp/2$, $\Delta_n^{max} = 1$.
12 // The second-stage rolling optimization $T_2$.
13 for updating round m= 1,2,...,$T_2$ do
14     Initialize the perturbation $\Delta_i$ randomly.
15     for each layer in multilayer structure do
16         for each $B_i \in \{\mathbf{B}\}$ do
17             $\alpha_i \leftarrow l_{anchor} + \Delta_i$.
18             Do $\alpha$ expansion move $\left(f', \alpha_i\right)$.
19             $f^B \leftarrow M\left(f^B, f', P_i\right)$.
20         $\Delta_d^{max} \leftarrow \left(\Delta_d^{max}/2\right)$, $\Delta_n^{max} \leftarrow \left(\Delta_n^{max}/2\right)$.
21         if size or location of $\{\mathbf{B}\}$ is changed then
22             $\Delta_d^{max} = maxdisp/2$, $\Delta_n^{max} = 1$.
```

D. OPTIMIZATION PROCEDURE

We propose a rolling optimization strategy to optimize the label proposal globally, using the cross-based multilayer structure as well as α expansion move. We present the optimization details in this section and summarize the optimization procedure as Algorithm 2. Rolling optimization is like a roller repeatedly updating the label mapping f. Then the disparity map can be generated by the label mapping. The optimization process is divided into two parts: first-stage rolling optimization (lines 2-11) and second-stage rolling optimization (lines 12-22). The first stage aims at fast rough matching. In this stage, label expansion is carried out in the border region (line 8), and labels are updated multiple times (lines 6-8). At the same time, the perturbation component of the candidate label is gradually reduced due to its gradually decreased selection range (line 9), so that labels in the border region can be optimized more and more finely. When the size or location of $\{\mathbf{B}\}$ is changed, the boundary of the selection range is reinitialized (line 11). In the second stage, the mapping refinement is carried out, which is different from the first stage in that α expansion move (line 18) and mask operation (line 19) are used. After the second stage, the label mapping and the disparity map are smoother than before, because this stage optimization is based on minimizing the MRF energy function [14], [17] that considers the label and disparity continuity of neighboring pixels.

Perturbation term Δ_i consists of a disparity increment Δ_d and a normal vector increment $\vec{\Delta}_n$. Correspondingly, the label of anchor pixel l_{anchor} can also be converted to the form of a disparity d and a normal vector $\vec{n}$ [5]. Δ_d randomly takes value from $\left[-\Delta_d^{max}, \Delta_d^{max}\right]$, and the three elements of $\vec{\Delta}_n$ all randomly take values from $\left[-\Delta_n^{max}, \Delta_n^{max}\right]$. At line 7 in Algorithm 2, we add permutations Δ_d and $\vec{\Delta}_n$ to d and $\vec{n}$, and normalize the vector as $\vec{n} = u\left(\vec{n} + \vec{\Delta}_n\right)$. Then the new disparity and normal vector are translated to a candidate label α_i. It should be noted that we start Δ_d^{max} and Δ_n^{max} by setting $\Delta_d^{max} = maxdisp/2$ and $\Delta_n^{max} = 1$, and after one time expansion move, Δ_d^{max} and Δ_n^{max} are reduced to half of the original as at line 9 of Algorithm 2.

At line 8 in Algorithm 2, label expansion move only focuses on the minimization of data term, avoiding the complex computation of optimization process that considers data term and smoothness term simultaneously. Specifically, the local label mapping f^B in each border region B_i is updated by

$$f^B = \text{argmin} E_{data}\left(f^B | f_p^B \in \left\{f_p^B, \alpha_i\right\}, p \in B_i\right), \tag{14}$$

where the candidate label α_i is equal to the label of the anchor pixel plus the disturbance component, that is, $\alpha_i = l_{anchor} + \Delta$. Whether the candidate label can replace the original label of pixel p depends on whether the candidate label can reduce the value of the data term.

However, the label expansion move does not consider the smoothness of the labels of neighboring pixels, it is easy to fall into local minimum errors. The second-stage optimization can solve this problem better. We first calculates a mapping f' using α expansion [17] (line 18 in Algorithm 2),

$$f' = \text{argmin} E\left(f' | f_p' \in \left\{f_p', \alpha_i\right\}, p \in B_i\right). \tag{15}$$

The above formula can be solved via min-cut/max-flow algorithms [18], which can effectively deal with the situation where multiple labels become α_i at the same time. Then we perform mask operation as line 19 in Algorithm 2 which updates each member f_p^B of f^B by

$$f_p^B = \begin{cases} f_p', p \in P_i \\ f_p^B, p \notin P_i, \end{cases} \tag{16}$$

where P_i is the cross patch contained in border region B_i. In the process of α expansion move, the edge pixels have less smoothness term information. And the reliability of their optimized labels is not as high as that of the pixels close to the center region. Based on this, mask operation can discard some incorrect labels and improve the whole accuracy of label proposal.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we first introduce the running environment and core parameter settings of our algorithm. Then we use

TABLE 1. Snapshot of ranks on the test set of Middlebury 3.0 benchmark under the criterion "bad 2.0" by the time of submission (December 2019). Eight top-performance algorithms with published papers are listed here. The best results are in bold. Blue numbers represent rankings and black numbers represent error rates.

Name	Avg	Austr	AustrP	Bicyc2	Class	ClassE	Compu	Crusa	CrusaP	Djemb	DjembL	Hoops	Livgrm	Nkuba	Plants	Stairs
LocalExp [30]	**5.43 1**	3.65 3	2.87 3	**2.98 1**	**1.99 1**	5.59 2	3.37 2	3.48 3	3.35 2	2.05 2	10.3 5	9.75 4	8.57 5	14.4 14	5.40 5	9.55 5
Ours	5.75 2	3.66 4	3.11 4	5.92 12	2.14 2	6.01 3	3.39 3	3.49 4	3.68 3	2.34 3	10.2 4	9.63 3	8.04 3	14.9 17	5.45 6	9.26 4
3DMST [35]	5.92 3	3.71 5	2.78 2	4.75 3	2.72 5	7.36 8	4.28 4	**3.44 1**	3.76 4	2.35 4	12.6 10	11.5 7	8.56 4	14.0 12	5.35 4	8.87 3
MC-CNN+TDSR [39]	6.35 4	5.45 15	4.45 18	6.80 20	3.46 15	10.7 17	6.05 12	5.01 10	5.19 11	2.62 8	10.8 6	9.62 2	**6.59 1**	**11.4 1**	6.01 9	**7.04 1**
PMSC [7]	6.71 5	**3.46 1**	**2.68 1**	6.19 16	2.54 3	6.92 6	4.54 5	3.96 5	4.04 7	2.37 5	13.1 11	12.3 8	12.2 8	16.2 24	5.88 8	10.8 9
LW-CNN [40]	7.04 6	4.65 8	3.95 9	5.30 8	2.63 4	11.2 21	5.41 8	4.32 7	4.22 8	2.43 7	12.2 9	13.4 11	13.6 18	14.8 15	**4.72 1**	12.0 16
FEN-D2DRR [41]	7.23 7	4.68 9	4.11 12	5.03 6	3.03 10	8.42 9	6.05 12	4.90 9	5.32 12	3.20 19	11.5 8	14.1 13	13.4 17	13.9 10	5.06 2	14.3 23
APAP-Stereo [42]	7.26 8	5.43 14	4.91 25	5.11 7	5.17 22	21.6 37	6.99 18	4.31 6	4.23 9	3.24 21	14.3 14	9.78 5	7.32 2	13.4 6	6.30 10	8.46 2

TABLE 2. Snapshot of ranks on the test set of Middlebury 3.0 benchmark under the criterion "bad 1.0" by the time of submission (December 2019). Eight top-performance algorithms with published papers are listed here. The best results are in bold. Blue numbers represent rankings and black numbers represent error rates.

Name	Avg	Austr	AustrP	Bicyc2	Class	ClassE	Compu	Crusa	CrusaP	Djemb	DjembL	Hoops	Livgrm	Nkuba	Plants	Stairs
Ours	**13.4 1**	9.01 6	7.60 9	9.79 5	9.09 10	23.4 10	**8.49 1**	12.5 3	13.4 3	7.69 11	21.6 3	19.7 2	**14.0 1**	26.6 18	13.3 3	15.7 4
LocalExp [30]	13.9 2	9.80 8	7.92 10	**7.41 1**	9.12 11	24.4 12	9.13 3	**11.7 1**	**12.8 1**	7.74 12	22.2 6	20.6 3	16.1 3	29.1 23	16.7 11	15.5 2
3DMST [35]	14.5 3	8.97 5	7.03 7	10.0 7	9.18 12	23.0 7	11.1 4	13.7 4	14.9 7	8.53 20	24.2 10	23.2 6	17.1 4	28.0 21	14.1 4	15.7 3
PMSC [7]	14.8 4	7.92 2	5.88 2	10.6 11	8.99 8	22.6 4	12.1 5	13.8 5	14.6 6	8.17 15	23.9 9	24.0 7	19.8 7	27.6 20	15.2 7	18.6 7
LW-CNN [40]	14.9 5	8.47 3	6.89 4	9.47 2	8.15 4	27.6 23	12.8 9	16.6 7	16.0 8	5.88 2	22.0 4	25.9 8	21.4 12	22.7 5	12.8 2	24.4 19
CBMV_ROB [43]	15.6 6	**7.34 1**	7.35 8	11.7 16	11.3 19	17.0 2	12.1 6	14.5 6	13.4 5	6.95 8	34.3 27	21.2 4	20.1 8	31.4 27	16.8 13	19.3 8
FEN-D2DRR [41]	15.9 7	8.80 4	6.91 5	9.55 4	8.39 5	26.2 19	15.2 19	18.4 9	16.4 9	6.77 4	22.1 5	26.4 11	21.8 15	23.9 10	15.0 6	29.4 25
MC-CNN+TDSR [39]	16.1 8	10.9 14	9.40 20	11.8 17	9.08 9	24.5 14	14.8 16	18.4 8	18.9 11	8.51 18	22.9 7	29.8 17	17.2 5	22.5 2	19.6 18	**14.6 1**

TABLE 3. Snapshot of ranks on the test set of Middlebury 3.0 benchmark under the criterion "bad 0.5" by the time of submission (December 2019). Eight top-performance algorithms with published papers are listed here. The best results are in bold. Blue numbers represent rankings and black numbers represent error rates.

Name	Avg	Austr	AustrP	Bicyc2	Class	ClassE	Compu	Crusa	CrusaP	Djemb	DjembL	Hoops	Livgrm	Nkuba	Plants	Stairs
Ours	**38.1 1**	28.5 12	28.0 20	20.8 3	32.1 14	61.7 12	**50.3 1**	44.9 3	45.3 3	30.0 21	45.2 6	40.2 4	**31.3 1**	52.4 16	**40.3 1**	**25.4 1**
LocalExp [30]	38.7 2	30.0 14	28.9 25	**19.6 1**	32.0 13	62.0 14	52.8 5	**44.0 1**	44.9 2	30.1 22	46.1 13	39.5 2	33.1 3	54.9 23	41.0 3	28.5 2
PMSC [7]	39.1 3	**23.2 1**	22.6 9	22.5 7	26.9 5	63.1 18	54.7 14	46.0 5	46.0 6	32.7 29	47.6 16	45.4 6	34.7 10	52.9 17	42.9 5	34.9 7
CBMV_ROB [43]	39.7 4	25.9 6	26.1 16	21.9 4	36.8 24	53.2 4	52.6 3	47.4 6	45.3 4	28.9 18	51.3 22	40.1 3	34.4 7	57.3 31	45.5 10	28.7 3
LW-CNN [40]	39.7 5	23.8 2	22.7 10	23.6 10	29.9 11	67.1 32	55.2 18	48.4 7	48.5 10	27.9 15	44.7 5	47.8 10	34.7 10	46.0 4	43.2 6	49.1 28
3DMST [35]	39.9 6	26.9 9	28.8 24	22.2 5	28.8 9	62.7 17	54.1 10	45.6 4	46.3 8	34.0 30	49.6 19	45.4 7	34.9 13	53.8 21	43.4 7	29.8 6
OVOD [44]	40.0 7	24.2 3	21.3 6	20.8 2	33.3 15	64.2 23	55.7 20	53.2 12	53.9 18	25.5 4	45.6 9	44.8 5	32.7 2	50.8 12	41.8 4	44.5 21
MC-CNN+TDSR [39]	40.1 8	28.2 11	28.1 22	25.7 20	25.3 2	64.1 22	53.9 9	48.8 8	51.3 12	30.3 23	44.4 4	55.0 24	34.3 6	46.2 5	46.8 12	28.9 5

the 2014 datasets, the latest Middlebury stereo datasets generated by the technique of [45], to test our algorithm, and we compare our method with other state-of-the-art algorithms. In addition, in both qualitative and quantitative aspects, we compare the proposed algorithm with PatchMatch Stereo [5] and other global PatchMatch-based methods such as PMSC [7], LocalExp [30]. Finally, we verify the effectiveness of the core modules of the algorithm, namely first-stage rolling optimization, second-stage rolling optimization and cross-based multilayer structure, through the ablation analysis.

A. PARAMETER SETTINGS

Our algorithm is conducted on a PC equipped with an i5-7400 3.0GHz CPU. During the test, 4 cores of CPU are used for parallel acceleration. When we test the performance of our method on stereo datasets, the parameters of the algorithm are required to remain constant for all experiments. The following are the settings of the parameters mentioned in our algorithm. With reference to [30], the coefficient λ, used to balance data term and smoothness term, is set to 0.5. The reason for this setting is that the smoothness term has a less impact on matching accuracy than data term, so we give it a relatively lower attention degree. The parameter γ for pixel similarity weight ω_{pq} is set to 10, referring to [5], [30], [31]. The truncation thresholds τ_{CNN}, τ_{dis}, e and ε are set to 0.5, 1, 0.01, 0.01^2 respectively. To construct the adaptive cross patch, the threshold of color difference η_{color} is set to 100, and the threshold of distance $\eta_{distance}$ is set to the width of border region. For the multilayer structure, the central regions have three different sizes with widths of $(14, 14 \times 3, 14 \times 9)$ pixels. And the number of the total layers of multilayer structure $(N_1 \times N_2 \times N_3)$ is set to $(7 \times 16 \times 3)$. For the rolling optimization process, the iteration numbers of the first stage and the second stage (T_1, T_2) are set to 10 and 20 respectively.

B. EVALUATION ON MIDDLEBURY BENCHMARK

We test our algorithm on Middlebury stereo 3.0 benchmark [45]. It contains 30 pairs of high-resolution image pairs, 15 pairs for training and 15 pairs for test. The datasets of this benchmark provide various challenges, such as rich and complex scenes, large field of view (disparity range from 200 to 800 pixels), incomplete correction as well as different illumination and exposure in left and right images. Consistent with other algorithms [7], [30], we use half-resolution image pairs to estimate disparity, then interpolate the disparity map

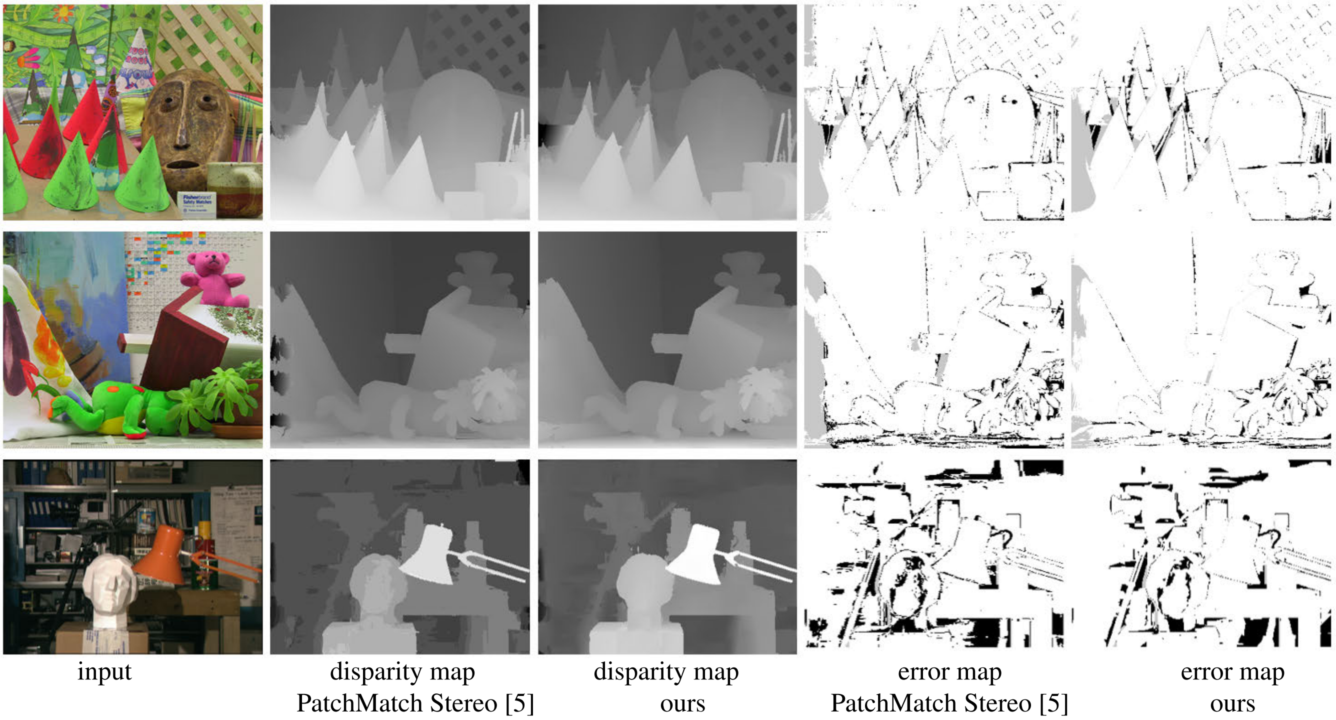

FIGURE 5. Qualitative comparison between PatchMatch Stereo and our algorithm. From row 1 to row 3 are Cones, Teddy and Tsukuba respectively. The pictures in columns 2 and 3 are the disparity maps corresponding to the input images (column 1). The pictures in columns 4 and 5 are error maps, where black areas indicate the pixels with wrong disparities.

to restore the full resolution. Three evaluation criteria are used to evaluate the accuracy of disparity map, which are bad 2.0, bad 1.0 and bad 0.5, respectively. They are defined as the proportion of the error pixels with mismatches > 2.0, 1.0 and 0.5 pixels in the whole map. They reflect the matching performances at different pixel levels, where the smaller proportion represents the better matching accuracy.

We generate the disparity maps of test images via the proposed algorithm, submit them to Middlebury benchmark to calculate the matching error rates under different criteria online, and publish the performance evaluation results of our algorithm. Table 1, Table 2 and Table 3 are the snapshots of ranks on Middlebury 3.0 benchmark under bad 2.0, bad 1.0 and bad 0.5 criteria respectively, in which the overall weighted error rates of our algorithm are 5.75%, 13.4% and 38.1% respectively. Correspondingly, our algorithm ranks the second, the first and the first among all published algorithms under above three criteria. This demonstrates the excellent performance of our algorithm. Especially, our algorithm outperforms current state-of-the-art methods at the pixel level and sub-pixel level, that benefits from our rolling optimization and adaptive update region constraint.

C. COMPARISON WITH PATCHMATCH STEREO

We compare our algorithm with PatchMatch Stereo [5], from which many stereo matching algorithms, including ours, are derived [6], [7], [30], [31]. PatchMatch Stereo is designed for low-resolution images of about 0.1 Mpixel, and the performance of this algorithm can not be evaluated using the datasets from Middlebury 3.0 benchmark, but from the old Middlebury 2.0 benchmark [1]. Therefore, we select "Cones", "Teddy" and "Tsukuba" from Middlebury 2.0 benchmark to compare our algorithm with PatchMatch Stereo.

Table 4 shows the matching error rates of the two algorithms. The error threshold is 0.5 pixels. The error rates of our algorithm are (3.71%, 3.62%, 10.3%) respectively, which

TABLE 4. The quantitative comparison between PatchMatch Stereo and our algorithm. The table shows the matching error rates. The best results are in bold.

Algorithms	PatchMatch Stereo [5]	**Ours**
Cones	3.80	**3.71**
Teddy	5.66	**3.62**
Tsukuba	15.0	**10.3**

TABLE 5. The quantitative comparison of global PatchMatch-based algorithms. The table shows the matching error rates and disparity error quantiles under different criteria. The best results are in bold.

Algorithms	LocalExp [30]	PMSC [7]	**Ours**
bad 0.5 nonocc	38.7	39.1	**38.1**
bad 0.5 all	44.2	45.4	**43.5**
bad 1.0 nonocc	13.9	14.8	**13.4**
bad 1.0 all	21.0	22.8	**20.3**
A50 nonocc	0.43	0.43	**0.42**
A50 all	**0.47**	0.49	**0.47**
A95 nonocc	**4.81**	5.27	4.95
A95 all	31.6	31.4	**30.7**

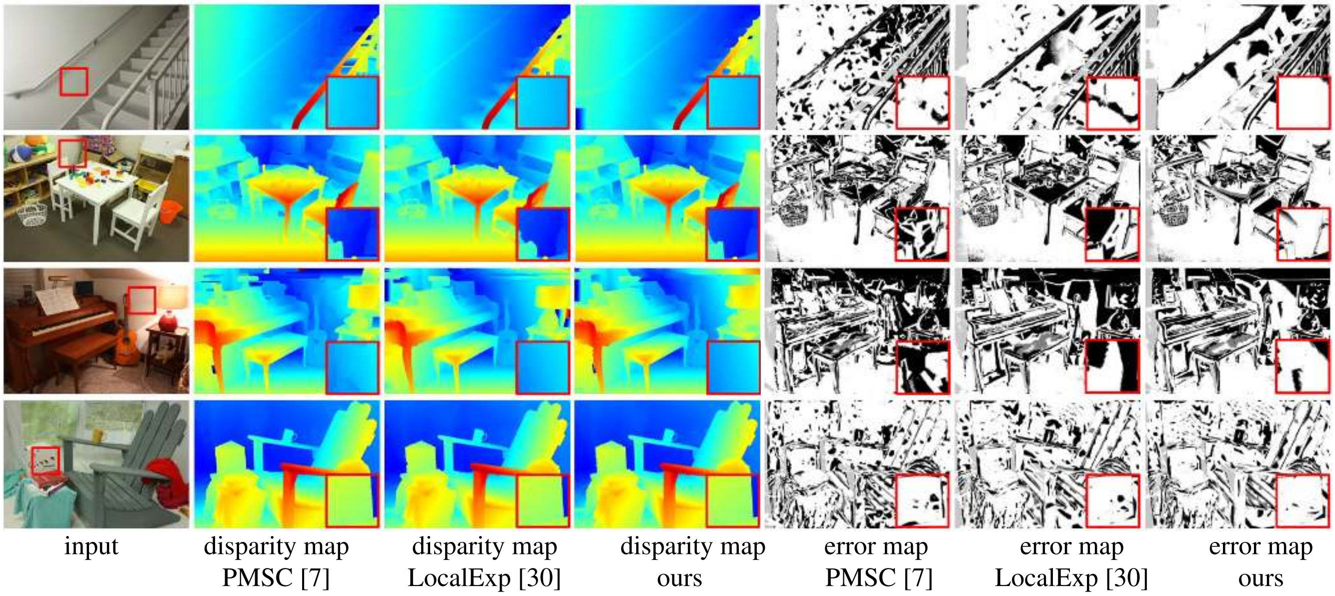

FIGURE 6. Qualitative comparison of global PatchMatch algorithms. The pictures in columns 2, 3 and 4 are the disparity maps with rainbow colors. The pictures in columns 5, 6 and 7 are error maps, where black areas indicate the pixels with wrong disparities.

TABLE 6. Ablation analysis of different modules. ✓ indicates that the corresponding module is included, and × indicates that the module has been removed. We test the matching error rates when using different variants of the algorithm. The last column is the average error rate.

Modules			Matching error rates (bad 1.0 metric)					
First-stage optimization	Second-stage optimization	Cross-based multilayer	Adiron	Jadepl	Pipes	Playrm	Teddy	Average
×	✓	✓	5.66	23.1	11.2	21.3	6.16	13.48
✓	×	✓	10.2	22.0	11.3	30.5	6.96	16.19
✓	✓	×	5.18	20.4	**10.6**	20.9	6.30	12.80
✓	✓	✓	**5.48**	**20.2**	**10.6**	**20.6**	**6.15**	**12.60**

are less than those of PatchMatch Stereo (3.80%, 5.66%, 15.0%). Fig. 5 shows disparity maps generated by the two algorithms and corresponding error maps. The error maps of our algorithm have fewer and smaller black areas than those of PatchMatch Stereo, where black area marks the pixels with wrong disparities. The scene of the last row in Fig. 5 is complex. In this situation, PatchMatch Stereo has many discretely distributed disparity errors (see its black areas in error map), reflecting that it has many local minimum errors. This is an unavoidable problem for local stereo matching algorithm, because it only focuses on neighborhood information and ignores global information. On the contrary, our algorithm adopts the pairwise MRF model to ensure the global smoothness of disparity map, guaranteeing the effective suppression of local minimum errors.

D. COMPARISON WITH GLOBAL PATCHMATCH-BASED METHODS

We compare our algorithm with LocalExp [30] and PMSC [7]. The same thing is that both of them and our algorithm are global PatchMatch-based methods, and all are designed to process high-resolution image pairs larger than 1 Mpixel. The differences are that we construct cross-based patches to adaptively constrain the label updating region, instead of using fixed rectangular windows like LocalExp; besides, we use rolling optimization strategy to generate more abundant and continuously changing candidate labels than PMSC. When evaluating the algorithms, we use 15 high-resolution test image pairs from Middlebury 3.0 benchmark for test targets, employ bad 2.0, bad 1.0 and bad 0.5 error rates to evaluate matching accuracy, and employ error quantiles A50 (Media error) and A95 to evaluate robustness. Besides, two kinds of masks are used, namely, ‘‘nonocc’’ (non-occluded pixels visible in both views) and ‘‘all’’ (all pixels).

The quantitative comparison is shown in Table 5. On the one hand, the average error rates (38.1%, 43.5%, 13.4%, 20.3%) of the proposed algorithm are smaller than those of LocalExp (38.7%, 44.2%, 13.9%, 21.0%) and PMSC (39.1%, 45.4%, 14.8%, 22.8%), that demonstrates more accuracy of our method at pixel level and sub-pixel level than PMSC and LocalExp. On the other hand, our algorithm has error quantiles which are similar to LocalExp but smaller than PMSC, in other words, our method has smaller values of big disparity errors than PMSC, reflecting less abnormal matching and better robustness than PMSC. The qualitative comparison is shown in Fig. 6, where rainbow-color disparity maps and corresponding error maps are presented. It is obvious that in

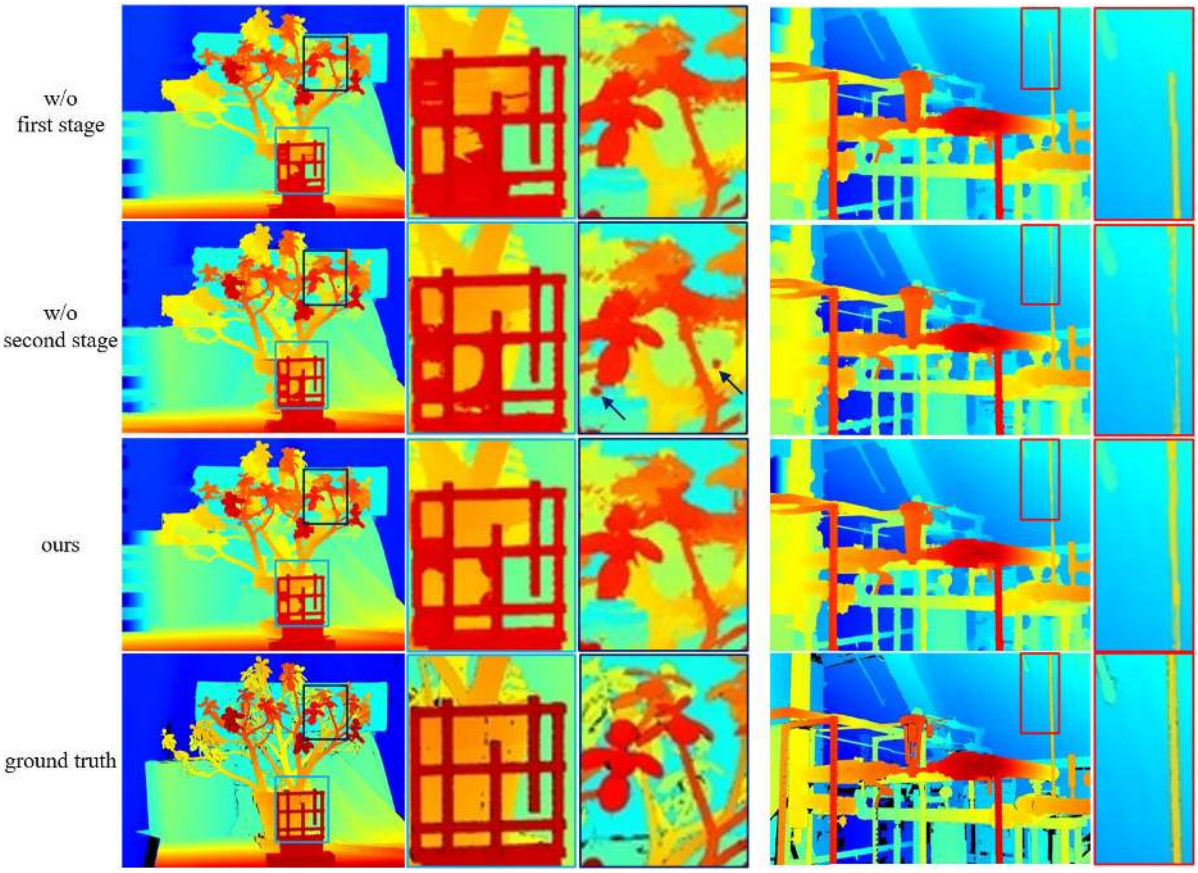

FIGURE 7. Qualitative analysis of the function of different modules. We present the disparity maps obtained by using different algorithm variants, as well as the ground truth, shown in columns 1 and 4. Columns 2, 3 and 5 are the local magnifications of the disparity maps.

weak texture regions (red boxes in Fig. 6), our algorithm has fewer mismatches than LocalExp and PMSC (comparing the black area sizes within the red boxes of error maps), indicating that our algorithm is able to solve the matching ambiguity of weak texture, and has better matching performance than other two algorithms.

E. EFFECTIVENESS OF EACH MODULE

To demonstrate the effectiveness of first-stage optimization, second-stage optimization, and cross-based multilayer structure, we employ ablation experiments to analyze the significance of each core module of our algorithm. We observe the change of matching accuracy after the removal of each module. It should be noted that when cross-based multilayer structure is removed, a fixed size multilayer structure is used instead; specifically, border regions are used to replace cross patches during the construction of multilayer structure (in Sec. III-C). We use the error rate of threshold = 1.0 pixel to evaluate matching accuracy. This criterion is commonly concerned in practical application, and reflects pixel level matching accuracy. Six image pairs from the training datasets of Middlebury 3.0 benchmark are used to test algorithm variants. Unlike the image pairs in test datasets, these image pairs have the corresponding published ground truth of disparity maps, so that users can calculate matching error rates flexibly.

Quantitative experimental results are listed in Table 6. Compared with other variants, the full version algorithm (last row in Table 6), including first-stage optimization, second-stage optimization and cross-based multilayer, has a minimum average error rate of 12.60%. In contrast, without any core module, the error rate will increase to a certain extent; in the absence of second-stage optimization, the average error rate increases the most, with an increase of 3.59%. This illustrates that first-stage optimization, second-stage optimization and cross-based multilayer structure all have positive effects to the improvement of matching accuracy, and second-stage optimization, considering the disparity smoothness of neighborhood pixels, contributes the most.

Fig.7 qualitatively explains the functions of the two optimization stages. First-stage optimization estimates the label and disparity of each pixel separately, and suppresses the matching errors that are easy to occur at the edge of the object (see local magnified disparity maps in column 2, rows 1 and 3 of Fig.7). Second-stage optimization is helpful to solve the local minimum problem of energy function by introducing the disparity smoothness constraint, so as to suppress the occurrence of local mismatching (the local mismatches, indicated by the arrows in row 2 and column 3, no longer exist in the disparity map of row 3 and column 3). Besides, it also effectively guarantees the continuity of the object contour (see the pipes in column 5, rows 2 and 3).

V. CONCLUSION

In this paper, we have presented a novel global stereo matching approach for the accurate estimation of 3D label and sub-pixel disparity. To adaptively constrain the region of 3D label expansion, we proposed a cross-based multilayer structure which includes a series of cross patches with adaptive shapes. We proposed a rolling optimization strategy, using the constructed cross-based multilayer to guide label updating, so that the randomly initialized label proposal converges to the optimal solution through the rolling optimization. Experimental results showed that our method can obtain a highly accurate disparity map, and it is one of the best stereo matching algorithms on Middlebury 3.0 benchmark. Whether compared with similar methods or other approaches with published papers on the benchmark, our algorithm has higher matching accuracy at pixel level and subpixel level.

REFERENCES

[1] 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, 2002.
[2] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss, "Stereo vision-based obstacle avoidance for micro air vehicles using disparity space," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2014, pp. 3242–3249.
[3] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, and M. Dou, "Holoportation: Virtual 3D teleportation in real-time," in *Proc. 29th Annu. Symp. User Interface Softw. Technol.*, 2016, pp. 741–754.
[4] X. Chen, H. Liang, H. Xu, S. Ren, H. Cai, and Y. Wang, "Artifact handling based on depth image for view synthesis," *Appl. Sci.*, vol. 9, no. 9, p. 1834, 2019.
[5] M. Bleyer, C. Rhemann, and C. Rother, "PatchMatch stereo-stereo matching with slanted support windows," in *Proc. BMVC*, vol. 11, 2011, pp. 1–11.
[6] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, "PMBP: PatchMatch belief propagation for correspondence field estimation," *Int. J. Comput. Vis.*, vol. 110, no. 1, pp. 2–13, Oct. 2014.
[7] L. Li, S. Zhang, X. Yu, and L. Zhang, "PMSC: PatchMatch-based superpixel cut for accurate stereo matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 679–692, Mar. 2018.
[8] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 504–511, Feb. 2013.
[9] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.
[10] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 29–43.
[11] M. Veldandi, S. Ukil, and K. Govindarao, "Robust segment-based stereo using cost aggregation," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.
[12] M. Ble, C. Rother, and P. Kohli, "Surface stereo with soft segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1570–1577.
[13] D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An evaluation of the state-of-the-art," *Comput. Vis. Image Understand.*, vol. 166, pp. 1–27, Jan. 2018.
[14] C. Olsson, J. Ulén, and Y. Boykov, "In defense of 3D-label stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1730–1737.
[15] J. Ni, Q. Li, Y. Liu, and Y. Zhou, "Second-order semi-global stereo matching algorithm based on slanted plane iterative optimization," *IEEE Access*, vol. 6, pp. 61735–61747, 2018.
[16] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.
[17] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[18] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.
[19] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 689–695.
[20] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, Oct. 2006.
[21] H. Tao, H. S. Sawhney, and R. Kumar, "A global matching framework for stereo computation," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Jul. 2001, pp. 532–539.
[22] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1073–1079, Jul. 2009.
[23] Y. Xu, Y. Zhao, and M. Ji, "Local stereo matching with adaptive shape support window based cost aggregation," *Appl. Opt.*, vol. 53, no. 29, pp. 6885–6892, Oct. 2014.
[24] K.-R. Kim, Y. J. Koh, and C.-S. Kim, "Multiscale feature extractors for stereo matching cost computation," *IEEE Access*, vol. 6, pp. 27971–27983, 2018.
[25] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, nos. 1–32, p. 2, 2016.
[26] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.
[27] V. Kolmogorov and C. Rother, "Minimizing nonsubmodular functions with graph cuts—A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1274–1279, Jul. 2007.
[28] L. Wang and R. Yang, "Global stereo matching leveraged by sparse ground control points," in *Proc. CVPR*, Jun. 2011, pp. 3033–3040.
[29] V. Lempitsky, C. Rother, S. Roth, and A. Blake, "Fusion moves for Markov random field optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1392–1405, Aug. 2010.
[30] T. Taniai, Y. Matsushita, Y. Sato, and T. Naemura, "Continuous 3D label stereo matching using local expansion moves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2725–2739, Nov. 2018.
[31] T. Taniai, Y. Matsushita, and T. Naemura, "Graph cut based continuous stereo matching using locally shared labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1613–1620.
[32] P. Heise, S. Klose, B. Jensen, and A. Knoll, "PM-huber: PatchMatch with huber regularization for stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2360–2367.
[33] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[34] K.-R. Kim and C.-S. Kim, "Adaptive smoothness constraints for efficient stereo matching using texture and edge information," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3429–3433.
[35] L. Li, X. Yu, S. Zhang, X. Zhao, and L. Zhang, "3D cost aggregation with multiple minimum spanning trees for stereo matching," *Appl. Opt.*, vol. 56, no. 12, pp. 3411–3420, Apr. 2017.
[36] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1592–1599.
[37] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
[38] P. Tan and P. Monasse, "Stereo disparity through cost aggregation with guided filter," *Image Process. Line*, vol. 4, pp. 252–275, Oct. 2014.
[39] S. Drouyer, S. Beucher, M. Bilodeau, M. Moreaud, and L. Sorbier, "Sparse stereo disparity map densification using hierarchical image segmentation," in *Proc. Int. Symp. Math. Morphol. Appl. Signal Image Process.* Cham, Switzerland: Springer, 2017, pp. 172–184.
[40] H. Park and K. M. Lee, "Look wider to match image patches with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1788–1792, Dec. 2017.
[41] X. Ye, J. Li, H. Wang, H. Huang, and X. Zhang, "Efficient stereo matching leveraging deep local and context information," *IEEE Access*, vol. 5, pp. 18745–18755, 2017.

[42] M.-G. Park and K.-J. Yoon, "As-planar-as-possible depth map estimation," *Comput. Vis. Image Understand.*, vol. 181, pp. 50–59, Apr. 2019.

[43] K. Batsos, C. Cai, and P. Mordohai, "CBMV: A coalesced bidirectional matching volume for disparity estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2060–2069.

[44] M. G. Mozerov and J. van de Weijer, "One-view occlusion detection for stereo matching with a fully connected CRF model," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2936–2947, Jun. 2019.

[45] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. German Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2014, pp. 31–42.

HUAIYUAN XU received the B.S. and M.S. degrees from Tianjin University, Tianjin, China, where he is currently pursuing the Ph.D. degree in optical engineering. His research interests include computer vision and deep learning.

XIAODONG CHEN received the Ph.D. degree in optical engineering from Tianjin University. He is currently a Professor with the School of Precision Instruments and Opto-Electronic Engineering, Tianjin University. He is the author of two books, more than 180 articles, and more than seven inventions. His research interests include photoelectric detection, medical image processing, computer vision, and computer graphics.

HAITAO LIANG received the B.S. degree from Tianjin University, Tianjin, China, where he is currently pursuing the Ph.D. degree. His research topics include image processing and computer graphics.

SIYU REN received the B.S. degree from Tianjin University, Tianjin, China, in 2018, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and deep learning.

YI WANG received the Ph.D. degree in optical engineering from Tianjin University. She works in the Key Laboratory of Opto-Electronics Information Technology, Ministry of Education. Her research interests include optical coherence tomography and medical image processing.

HUAIYU CAI received the Ph.D. degree in optical engineering from Tianjin University. She is a Professor with the School of Precision Instruments and Opto-Electronic Engineering, Tianjin University. She is the author of one book and more than 70 articles. Her research interests include photoelectric detection, information optics, and image processing.