ADAPTIVE STEREO MATCHING VIA LOOP-ERASED RANDOM WALK

Xuejiao Bai, Xuan Luo, Shuo Li, Hongtao Lu

MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems,
Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

This paper proposes an adaptive tree-based cost aggregation strategy for stereo matching. The previous tree-based algorithms [1, 2], hindered by the greediness of minimum spanning tree (MST), provide poorly adaptive support windows and have bad performance on curved and slanted surfaces. The proposed method incorporates randomness and overcomes these drawbacks by introducing loop-erased random walk (LERW) into tree construction. Experimental results over Middlebury dataset [3, 4] demonstrate that our LERW-based strategy outperforms other tree-based state-of-the-art strategies in most of the high resolution test cases. Three contributations are included: 1) a LERW-based cost aggregation strategy; 2) a LERW-based refinement method; 3) mathematical analysis of the adaptability of our support windows.

Index Terms— Loop-erased random walk, adaptive support window, tree-based cost aggregation, stereo matching

1. INTRODUCTION

Depth estimation is one of the foundational tasks in stereo vision. It recovers depth information between corresponding pixels in a pair of stereo images based on their visual disparities. Many binocular stereo matching algorithms have been proposed to achieve accurate disparity maps among which the cost aggregation based methods are the most widely-used ones [1, 2, 5, 6, 7, 8, 9, 10]. They generally consist of four steps: 1) matching cost computation, where the similarity of corresponding pixels is assigned to each pixel for all possible disparities; 2) cost aggregation, where the matching cost is aggregated over a support window around each pixel; 3) disparity computation, where an optimal disparity with the lowest aggregated cost for each pixel is selected; 4) refinement, which further improves the accuracy of inaccurate disparities.

According to recent researches, the performance of the aggregation-based methods is highly dependent on the support window. Regular windows of fixed or variable shape [5, 11] are demonstrated to be very efficient. Alternatively, shiftable support windows [11] can also be used. Zhang proposes a cross-based method [6, 9] that can construct aggregation windows in arbitrary shapes. However, these support windows are not adaptive enough to fit sharp depth discontinuities while staying large in ambiguous regions. Yang adopts a non-local aggregation strategy [1] by aggregating matching cost on a MST. Based on the MST structure, segment-tree (ST) [2] further incorporates the segmentation technique. These tree-based algorithms can freely extend the support window by joining pixels of similar colors. However, the extreme greediness of MST causes poor performance in large areas with similar colors but various disparities since pixels are joined too close to discern their disparity differences.

The drawbacks of such greediness in the tree-based strategies motivate us to incorporate randomness in the tree structure. The loop-erased random walk (LERW) based algorithm, proposed in this paper, manages to generate an even more adaptive support window with finer discrimination. This new support window is closer to the ideal one because it can deal with both depth discontinuities and ambiguous regions by modifying itself according to the local image content. The support window is even smaller near the boundary while remaining large inside the less-textured regions.

The rest of the paper is organized as follows. Section 2 details the proposed LERW-based algorithm, followed by the mathematical analysis in Section 3. Section 4 evaluates the experimental results. Finally, section 5 gives conclusions.

2. LERW-BASED ALGORITHM

2.1. LERW-based cost aggregation

This section details the proposed LERW-based cost aggregation strategy. Similar to other tree-based algorithms [1, 2, 12], the reference image is represented as an undirected graph G of the standard 4-connected grid. The aggregation phase consists of two steps: **tree construction** and **cost aggregation**.

Tree construction: Loop-erased random walk is a random simple path which erases all the loops of the random walk in chronological order. In this step, we adopt Wilson's Algorithm [13] to connect all the vertices in graph G with LERWs, as summarized in Algorithm 1. It repeatedly adds new LERW and output a uniform spanning tree (UST), which is a random spanning tree chosen among all the possible spanning trees of graph G with equal probability.

Cost aggregation: Define the weight of an edge (s, r) as:

$$w(s, r) = |I(s) - I(r)|, \tag{1}$$

input : A connected undirected graph G
output: A uniform spanning tree Tree

1 Randomly pick a vertex x, put x in Tree;
2 **repeat**
3 Randomly pick a vertex y that is not in Tree;
4 Path $\leftarrow$ random walk from y until the walk hits a vertex in Tree;
5 LERW $\leftarrow$ `EraseLoop(`Path`)`;
6 Add LERW to Tree;
7 **until** all vertices are in Tree;

Algorithm 1: LERW based Tree Construction

where $I(s)$ and $I(r)$ are the intensity values of vertex s and r in the graph. Therefore, the weight of the path from pixel p to q along UST is $D(p,q) = \sum_{(s,r)\in path(p,q)} w(s,r)$. Like other tree-based algorithms [1, 2], the similarity $S(p,q)$ between pixel p and q is defined as:

$$S(p,q) = e^{-D(p,q)/\delta}, \tag{2}$$

where δ is a constant to adjust the similarity. As is explained in [1], the joint bilateral filter, which has been proved in the prior work [10] to be reasonably accurate in cost aggregation, can be directly extended to the tree structure: the aggregated cost $C(p,d)$ for p at disparity label d is defined as:

$$C(p,d) = \sum_{q\in G} S(p,q)\cdot M(q,d), \tag{3}$$

where $M(q,d)$ represents the pixel-wise matching cost for q at disparity label d, as defined in [7]. Then a linear time exact algorithm in [1] is implemented to compute the overall aggregated matching cost C for each pixel over the UST.

Remark. According to Eq. 2 and Eq. 3, $S(p,q)$ will become extremely low if p and q are far away from each other on the UST or they have large color difference. It means that for each pixel p, only pixels inside a neighboring region, where $D(p,q)$ is relatively small, provide supports to it. This specific region is denoted as the "support window" of p.

Finally, a winner-takes-all (WTA) strategy is applied to select the disparity label d that minimizes the overall aggregated cost $C(p,d)$ for each pixel as its disparity:

$$D(p) = \underset{d}{\operatorname{argmin}}\, C(p,d). \tag{4}$$

(a) raw disparity map (b) refine via original UST (c) refine via reorganized UST

Fig. 1. Partial enlarged results of *Venus*[11]

2.2. LERW-based refinement

This section proposes a new refinement method via LERW. The LERW-based cost aggregation is initially performed within the left and right images and generates raw left and right disparity maps D_l and D_r. The stable pixels are then found by left-to-right consistency check [14]. Figure 1a presents the partial enlarged raw disparity map of *Venus* in [11] where we can clearly find jagged edges near boundary. Our observation is that it is caused by the large-weighted edges from the unstable pixels in the original UST which is built in tree construction. It motivates us to reorganize the UST to weaken the influence by those edges.

Reorganize UST: To rebuild the UST, we first remove those undesirable edges from the original graph G. let $\{r, r'\}$ be the set of the two vertices right to and beneath the vertex s. An edge (s,r) is undesirable if the vertex s is unstable and $w(s,r) \geq w(s,r')$. After removing undesirable edges, the new graph G' is likely to become a disconnected graph with several connected components. Therefore, we apply Algorithm 1 to rebuild a UST in each connected component independently and obtain a forest.

Refine along reorganized UST: A similar technique as other tree-based refinements is implemented [1, 2] along the reorganized UST. As proposed in [1], the new matching cost,

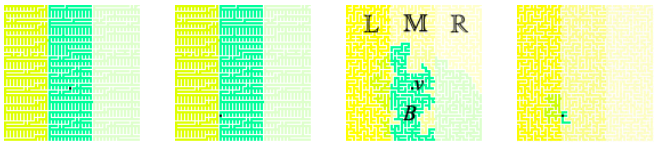
$$M'(p,d) = \begin{cases} d - D_l(p) & p \text{ is stable} \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

is updated for each pixel at all disparities. The same approach introduced in Sec.2.1 is applied to aggregate the new matching cost to propagate disparities from stable pixels to unstable ones along the reorganized UST. Compared with tree-based refinements [1, 2], the aggregation in LERW-based refinement runs in each connected component indepdently. Finally, the disparity that minimizes the aggregated matching cost is selected as the final disparity for each pixel.

Figure 1b-1c compare the disparity maps refined by aggregating M' along the original UST with the one by LERW-based refinement along the reorganized UST. As shown in Fig.1c, LERW-based refinement significantly improves the accuracy near the boundary and removes jagged edges.

3. ANALYSIS OF LERW-BASED AGGREGATION

This section quantitatively evaluates the adaptability in our LERW-based cost aggregation. In the MST-based algorithms [1, 2], after building a spanning tree from the reference image, each monochromatic region will be partitioned into several connected components, denoted as "support blocks", see areas of the darkest green in Fig.2. As remarked in Sec.2.1, the support blocks that provide support to a pixel form its support window, see the green areas in Fig.2 where darker green indicates larger support. The MST-based algorithms [1, 2], greedily select the edge with the minimum weight to connect



(a) MST[1] in the middle (b) MST[1] boundary (c) LERW in the middle (d) LERW boundary

Fig. 2. Support blocks of a toy example via MST and LERW. The maze-like lines show how the trees are connected.

the blocks. It results in a large support window containing excessive monochromatic regions that may vary in disparities.

On the other hand, the LERW-based strategy incorporates randomness. It weakens the greediness near depth discontinuities while keeping large support windows inside less-textured regions. Consider the toy example in Fig.2, an $h \times (3w)$ image partitioned into three $h \times w$ monochromatic regions, $\mathbb{L}, \mathbb{M}, \mathbb{R}$, where $h = 40$ and $w = 13$. The support block of MST is always the entire monochromatic region whether or not the vertex (the black pixel) is on the boundary, see Fig.2a-2b. The support blocks of the LERW-based strategy, however, vary in size, see Fig.2c-2d. The following part analyzes the expected size of support blocks by estimating a lower bound to it.

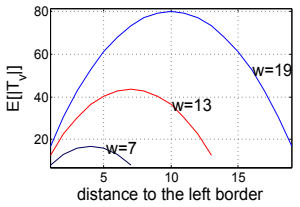
Support window adaptability: Let B be the support block containing a vertex v in the middle region $\mathbb{M}$ of the toy example. $|B|$ is the size of B. We'll show that a lower bound to the expected size of B, $E[|B|]$, is as plotted in [fig].

To simplify the problem, we adopt the model of Aldous-Broder algorithm[15], which can also output a UST. It runs a random walk (R_n) until it covers all the vertices, where R_n is the vertex hit by the random walk at the n^{th} step. An edge (R_{n-1}, R_n) is put into the tree if the vertex R_n is hit for the first time at step n. $E[|B|]$ keeps the same whatever the choice of R_1 since the distribution of the output tree remains unchanged. Therefore, to evaluate the support block of v, we instead consider a random walk (R_n) starting at v $(R_1 = v)$ and stopping once it escapes from $\mathbb{M}$. Let T_v be the subtree constructed by running Aldous-Broder on (R_n) and $|T_v|$ be the size of T_v, i.e. $|T_v|$ is incremented by one every time (R_n) hits a new vertex. So

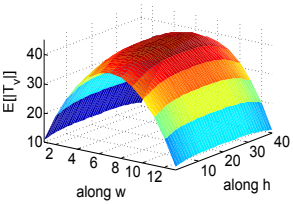
$$|B| \geq |T_v| \Rightarrow E[|B|] \geq E[|T_v|]. \tag{6}$$

For each vertex u, we define: $X_u^{(v)} = 1$ if $u \in T_v$, and $X_u^{(v)} = 0$ otherwise. Then $|T_v| = \sum_{u \in \mathbb{M}} X_u^{(v)}$. Let c_L be the rightmost column of $\mathbb{L}$ and c_R be the leftmost column of $\mathbb{R}$. Then $E[X_u^{(v)}] = \Pr[u \in T_v] = \Pr[u \text{ is hit by } (R_n)] = \Pr[(R_n) \text{ hits } u \text{ before hitting } c_L \text{ or } c_R]$. [16] relates the voltage of an electric network with such probability:

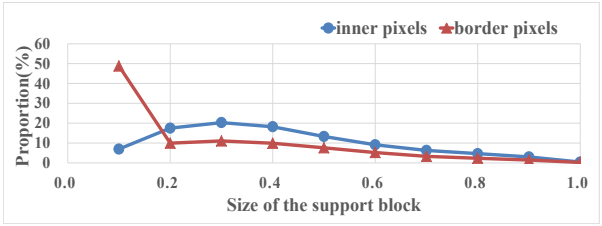
Consider an electric network with conductance C_{xy} on each edge (x, y), $(C_{xy} = C_{yx})$, and a random walk on the same underlying graph with the probability for a step from x to y $p_{xy} = C_{xy}/C_x$, where $C_x = \sum_{y \sim x} C_{xy}$ ($y \sim x$ if y is adjacent to x). Let v_x be the voltage of vertex x. Choose two



(a) $E[|T_v|]$ under various w, $v \in$ middle row of $\mathbb{M}$



(b) $E[|T_v|], v \in \mathbb{M}$



(c) proportions of pixels in support blocks of different sizes: border pixels vs. inner pixels.

Fig. 3. Mathematical and statistical results of the toy example.

vertices a and b from the electric network and assign $v_a = 1$ and $v_b = 0$. Then we have

$$\forall x, \Pr[\text{the random walk from } x \text{ hits } a \text{ before } b] = v_x.$$

So, to evaluate $E[X_u^{(v)}]$, we assign the voltage of vertices in c_L and c_R to be 0, the voltage of u to be 1 and C_{xy} to be 1 $\forall$ edge (x, y). Define the voltage of vertex v in this circumstance as $v_v^{(u)}$. We have $E[X_u^{(v)}] = v_v^{(u)}$ and

$$E[|T_v|] = \sum_{u \in \mathbb{M}} E[X_u^{(v)}] = \sum_{u \in \mathbb{M}} v_v^{(u)}. \tag{7}$$

We've obtained the numeric results of $E[|T_v|]$ by solving the voltages. Fig.3a depicts the $E[|T_v|]$'s along the middle row of $\mathbb{M}$ ($h = 40, w \in \{7, 13, 19\}$). Fig.3b displays the $E[|T_v|]$'s for all vertices $v \in \mathbb{M}$ of the toy example. The line chart in Fig.3c summarizes the statistical results over 10,000 experimental sets for pixels in the two columns in the middle and near the boundary of $\mathbb{M}$. Each point in the line chart indicates the proportion of pixels in support blocks of different sizes (scaled to $[0, 1]$). In summary, Fig.3a-3c all imply that our support blocks are small near the boundary while large inside the monochromatic regions, which well exhibits the adaptability. This finer discrimination in support blocks not only results in better accuracy and stronger adaptability, but also overcomes the greediness of MST-based methods [1, 2] especially over curved/slanted surfaces, see results in Sec.4.

4. EXPERIMANTAL RESULTS

This section demonstrates that our strategy outperforms other tree-based methods especially over high resolution images in the Middlebury [3, 4], which is the most widely-used dataset

Non-occluded error rate (%) (error $\geq$ 1.0)				
Test Case	MST[1]	ST[2]	LERW-1	LERW-2
Aloe	12.11	10.31	9.42±0.10	**7.90±0.09**
Baby1	18.96	11.94	9.06±0.24	**7.65±0.20**
Baby2	35.44	34.48	31.07±0.39	**24.67±0.28**
Baby3	17.86	12.41	11.14±0.32	**10.72±0.20**
Bowling1	41.02	39.64	36.76±0.62	**30.60±0.44**
Bowling2	30.27	28.78	24.38±0.31	**18.66±0.24**
Cloth1	4.17	2.97	1.99±0.03	**1.14±0.03**
Cloth2	17.69	13.27	11.81±0.22	**7.62±0.09**
Cloth3	8.03	6.42	4.87±0.09	**2.98±0.07**
Cloth4	7.74	5.61	4.36±0.11	**2.78±0.05**
Flowerpots	45.72	38.71	35.65±0.34	**32.11±0.37**
Lampshade1	25.99	26.21	19.14±0.49	**18.13±0.55**
Lampshade2	31.14	34.76	28.61±0.83	**26.69±0.79**
Midd1	**40.44**	44.67	47.18±1.18	52.17±0.74
Midd2	**40.85**	45.36	42.22±1.79	45.81±0.71
Monopoly	51.99	**36.81**	47.16±2.63	46.53±1.01
Plastic	58.06	55.29	**51.33±1.75**	52.03±1.22
Rocks1	23.51	22.26	21.34±0.25	**20.83±0.11**
Rocks2	12.05	9.79	8.20±0.12	**6.45±0.11**
Wood1	23.02	14.58	15.84±0.22	**12.41±0.16**
Wood2	11.94	18.26	16.46±0.48	**11.71±0.28**
Avg.	26.57	24.41	22.76±0.20	**20.93±0.10**

Table 1. Quantitative evaluation of the three algorithms on large resolution images from Middlebury [3, 4]. LERW-1: LERW-based aggregation + refinement in MST [1]; LERW-2: LERW-based aggregation + LERW-based refinement.

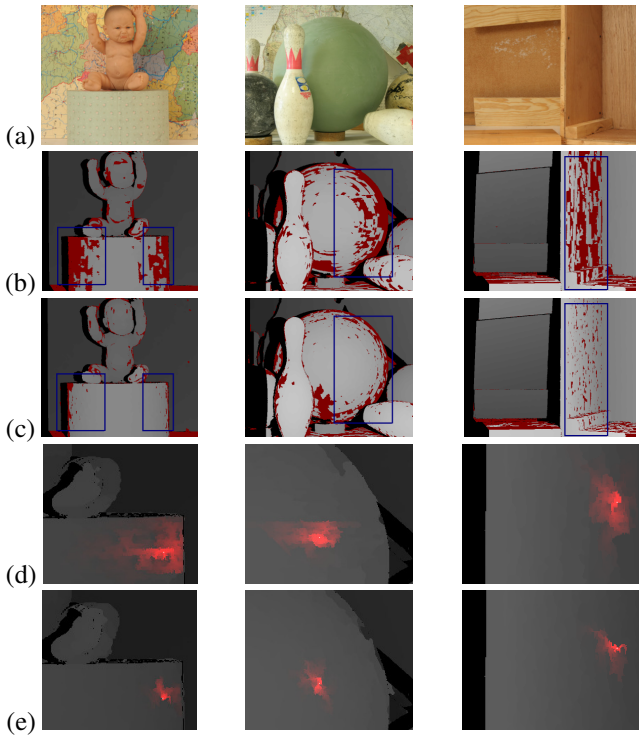


Fig. 4. Results on *Baby1, Bowling2* and *Wood1*. (a) Left images; (b) disparity maps of MST [1]; (c) disparity maps of LERW; (d) partial enlarged support windows (in red) of MST [1]; (e) partial enlarged support windows (in red) of LERW.

for stereo matching. We compare our approach against the following state-of-the-art algorithms: the MST aggregation (denoted as MST [1] and the enhanced ST aggregation (denoted as ST) [2]. They both include the refinements proposed in their papers. Throughout all experiments we set $\delta = 0.1$.

Support windows on curved/slanted surfaces: We select several test images from Middlebury [3, 4] that have large curved or slanted surfaces for visual comparison, which are *Baby1, Bowling2* and *Wood1* (resolution: $620 \sim 686 \times 555$). Fig.4b-4c display the disparity maps using MST [1] and LERW-based strategies. Pixels with erroneous disparities (error > 1.0) are marked in red and pixels in occluded regions are marked in black. The blue boxes highlight where LERW obtains considerably higher accuracy over MST [1] due to the gradual color change on the slanted and curved surfaces. Fig.4d-4e depict the support window of a selected pixel in the blue box using different strategies where darker red indicates larger support. The support window of MST [1] is too large, see Fig.4d, and therefore, the disparity calculated is quite flat. In contrast, LERW can produce a reasonably large support window which well detects the disparity change.

Performance on Middlebury dataset: To quantitively evaluate the performance, we test three tree-based methods on the 21 high resolution test cases (resolution: $620 \sim 698 \times 555$) from Middlebury 2006 dataset [3, 4]. Table 1 compares the performance of MST [1], ST [2], LERW-based aggregation with the refinement in [1] and with the LERW-based refinement. It displays the percentages of erroneous pixels in non-occluded regions. First, LERW-based aggregation achieves the lowest average error rate, which is about 5.6% lower than MST [1] and 3.5% lower than ST [2]. Second, among 21 test cases, LERW-based aggregation achieves the highest accuracy in 17 test cases. In particular, its error rate is over 10% lower than MST [1] or ST [2] aggregation strategies after refinement in the 7 test cases in bold font in Table 1. Third, the accuracy is improved by using the LERW-based refinement instead of the refinement in [1].

5. CONCLUSIONS

An adaptive tree-based cost aggregation strategy as well as a novel refinement method via LERW are proposed. The new algorithm incorporates randomness into tree construction and is demonstrated to provide more adaptive support window than other tree-based methods [1, 2] according to the mathematical analysis. Moreover, the proposed cost aggregation strategy shows leading performance in a large number of Middlebury test cases [3, 4] and it is demonstrated to be highly effective in the curved and slanted surfaces.

6. REFERENCES

[1] Q. Yang, "A non-local cost aggregation method for stereomatching," in *IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1402–1409.

[2] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang, "Segment-tree based cost aggregation for stereo matching," in *IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 313–320.

[3] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2007.

[4] H. Hirschmller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 2007.

[5] K. Oda H. Kano T. Kanade, A. Yoshida and M. Tanaka, "A stereo matching for video-rate dense depth mapping and its new application," in *IEEE Conf. on Computer Vision and Pattern Recognition*. IEEE, 1996, pp. 196–202.

[6] J. Lu K. Zhang and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," in *IEEE Trans. on Circuits and Systems for Video Technology*. IEEE, July 2009, pp. 1073–1079.

[7] M. Bleyer C. Rother C. Rhemann, A. Hosni and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3017–3024.

[8] S. Giardino S. Mattoccia and A. Gambini, "Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering," in *Asian Conference of Computer Vision*, 2009, vol. II, pp. 371–380.

[9] F K. Zhang, G. Lafruit and Catthoor, "Real-time stereo matching: A cross-based local approach," in *International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 733–736.

[10] K. Yoon and I. Kweon, "Adaptive support-weight approach for correspondence search," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*. IEEE, 2006, vol. 28, pp. 650–656.

[11] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *International Journal of Computer Vision*, 2002.

[12] O. Veksler, "Stereo correspondence by dynamic programming on a tree," in *CVPR*, 2005, vol. 2, pp. 384–390.

[13] D. B. Wilson, "Generating random spanning trees more quickly than the cover time," in *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing*. ACM, 1996, pp. 296–303.

[14] P. Fua, "A parallel stereo algorithm that produces dense depth maps and preserves image features," in *Machine Vision and Application*, 1993, vol. 6, pp. 35–49.

[15] D. J. Aldous, "The random walk construction of uniform spanning trees and uniform labelled trees," *SIAM Journal on Discrete Mathematics*, vol. 3(4), pp. 450–465, 1990.

[16] L. Lovasz, "Random walks on graphs: a survey," *Combinatorics: Paul Erdos is Eighty*, vol. 2, pp. 1–46, 1993.

[17] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2003, vol. 1, pp. 195–202.