Pattern Recognition 48 (2015) 2269–2278

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Cross-trees, edge and superpixel priors-based cost aggregation for stereo matching

Feiyang Cheng [a], Hong Zhang [a], Mingui Sun [b], Ding Yuan [a,*]

[a] *Image Research Center, Beihang University, Xueyuan Rd., Haiidan Dist., 100191, Beijing, China*

[b] *Department of Neurosurgery, University of Pittsburgh, Suite B-400, 200 Lothrop Street, PA15213, Pittsburgh, USA*

ARTICLE INFO

Article history:
Received 9 April 2014
Received in revised form
6 November 2014
Accepted 5 January 2015
Available online 14 January 2015

Keywords:
Stereo matching
Cost aggregation
Image filtering
Spanning trees

ABSTRACT

In this paper, we propose a novel *cross-trees* structure to perform the non-local cost aggregation strategy, and the *cross-trees* structure consists of a *horizontal-tree* and a *vertical-tree*. Compared to other spanning trees, the significant superiorities of the *cross-trees* are that the trees' constructions are efficient and the trees are exactly unique since the constructions are independent on any local or global property of the image itself. Additionally, two different priors: *edge prior* and *superpixel prior*, are proposed to tackle the false cost aggregations which cross the depth boundaries. Hence, our method contains two different algorithms in terms of *cross-trees+prior*. By traversing the two crossed trees successively, a fast non-local cost aggregation algorithm is performed twice to compute the aggregated cost volume. Performance evaluation on the 27 Middlebury data sets shows that both our algorithms outperform the other two tree-based non-local methods, namely *minimum spanning tree* (MST) and *segment-tree* (ST).

1. Introduction

Dense two-frame stereo matching has been extensively investigated for decades as a traditional low-level vision task, since it is crucial for many applications such as 3D reconstruction [1,2], image-based rendering [3,4] and anonymous driving [5]. According to the analysis and taxonomy scheme proposed in [6], stereo matching algorithms can be categorized into two groups: local algorithms and global algorithms. Stereo matching algorithms are often implemented following a subset of the four steps or all:

1. Cost function/cost volume estimation.
2. Cost aggregation within a support region.
3. Disparity computation/optimization.
4. Disparity refinement.

Global algorithms usually make explicit smoothness assumptions, and minimize a predefined energy function to obtain optimal results [7–9]. Despite the reliable matching results obtained, global algorithms are often time-consuming. All local algorithms compute the matching cost (step 1) firstly and then perform the cost aggregation (step 2) to get a locally optimized cost volume [6,10–14]. We mainly focus on efficient and effective local and non-local methods in this paper, and the readers are referred to a recent study for a comprehensive study of the global methods [15].

To find a correspondence (x, x'), the problem of the local methods can be concluded as a comparison of the similarity of two local patches which around x and x' respectively. The similarity of the two patches is computed by aggregating the costs of the pixels within the patches. Hence, the cost aggregation (step 2) procedure has important impacts on the accuracy and the efficiency of a local algorithm. The cost aggregation of a pixel in traditional local algorithms is usually performed by averaging the costs of the pixel itself and all its neighboring pixels. Here, the implicit assumption is that all the pixels which lie in a special local support region have similar disparities, as shown in Fig. 1(a). Such local methods suffer from well known ”edge fatten” effect once the local support regions cover the depth boundaries. The problem can be explained in the context of image filtering. For instance, the box filter always blurs the edges of an image during the image denoising procedure. Hence, the problem of the cost aggregation step is how to choose optimal local support regions for each pixel. Various researches have been conducted to estimate optimal support regions for the cost aggregation, such as *various window-based* methods [10,11] and *adaptive support weights* (ASW) methods (also known as *local filtering-based* methods) [12–14] which have state of the art performance in the last years. However, the selected support

* Corresponding author. Tel.:/fax: +86 10 82338991.
E-mail addresses: chfybuaa@gmail.com (F. Cheng), dmrzhang@buaa.edu.cn (H. Zhang), drsun@pitt.edu (M. Sun), dyuan@buaa.edu.cn (D. Yuan).

http://dx.doi.org/10.1016/j.patcog.2015.01.002
0031-3203/

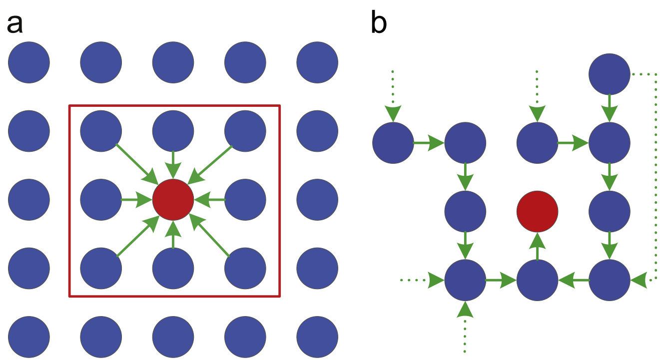

Fig. 1. Cost aggregation of the center pixel. (a) Local support region within the square frame: the center pixel gets support only from its neighboring pixels. (b) Tree: an unique path can be found between the center pixel and each pixel of the image. The dot line denotes that many pixels on the path are not shown here.

regions of the ASW methods are often limited in a pre-defined window of fixed size. Due to this reason, this kind of methods cannot work well for the stereo images with large planar surfaces.

Recently, Yang proposed a non-local cost aggregation method based on a MST [16]. As shown in Fig. 1(b), a pixel is able to get support from all the other pixels of the image through unique paths on the tree structure. Different from aforementioned *various window-based* methods and ASW methods, the cost aggregation was performed over the whole image for each pixel to establish non-local optimized results. Xing proposed a *segment-tree* structure to perform the non-local cost aggregation strategy [17]. These work proved that the non-local cost aggregation methods outperform the local ones much more.

Hence, this paper mainly focuses on the non-local cost aggregation procedure by comparing different tree construction techniques [16,17]. Section 2 is an overview of the previous work of the cost aggregation procedure. We briefly review the workflow of the non-local framework and the non-local cost aggregation algorithm in Section 3.1 and then introduce the *cross-trees* and the two priors in Section 3.2. A discussion of the strategies for constructing different tree structures is also provided in Section 3.2. Experimental results and performance evaluations are shown in Section 4. A detailed analysis of the short points of the tree-based non-local cost aggregation is given in Section 5. Finally, we draw the conclusions and discuss the future work in Section 6.

2. Previous work

The cost aggregation procedure can be considered as two subproblems: (1) how to estimate the optimal support regions; (2) how to aggregate the matching costs of the pixels within the estimated support regions (usually, in terms of support weights). We review the related work in this section according to the two subproblems above.

2.1. Various window-based methods

Most early local methods aimed at estimating various windows for different pixels, from adaptive [10,11] to shiftable windows [18]. The optimal windows were often selected based on certain local properties to avoid covering disparity discontinuities. Fusiello et al. developed a multiple window approach by performing cost aggregation in nine different window models and chose the window with the smallest aggregated cost as the optimal window. However, limited window models are not sufficient to represent support regions with arbitrary shapes and sizes. Some researchers proposed to use *cross-based structures* to represent various support regions and developed several competitive algorithms [19,20].

Most of these methods considered the sub-problem (1) only, and gave all the pixels the same weight. It means that these methods were developed based on a simple smoothness assumption that all the pixels within the same support region have a constant disparity. Such methods may result in over-smooth disparity slices within smooth curved surfaces.

2.2. Adaptive support weights methods

One main resolution is the adaptive support weights (ASW) strategy, in which weighted supports decide whether the neighboring pixels contribute more or less to the center pixel [12,13]. The support weights that adapt according to similarity and proximity to the central pixel of the predefined large support window actually control the real aggregation region and power. However, computing support weights iteratively for each central pixel is a time-consuming task. Many researchers indicated that such strategy can be approximately re-implemented by using local filter such as bilateral filter [21] and guided filter [14,22]. In this way, both the accuracy and the efficiency of the ASW algorithms have been improved. Hereto, the idea that the edge-preserving filters can be employed to aggregate the matching costs and to preserve the depth edges simultaneously becomes clear gradually. However, all the *local filtering-based* methods still establish locally optimized results within predefined windows which have a fixed size. Detailed comparisons and discussions of the *local filtering-based* methods can be found in two recent reviews [23,24].

2.3. Tree-based non-local methods

As mentioned above, a MST-based non-local cost aggregation method was proposed recently [16]. In the same non-local framework, Mei et al. proposed to employ a ST instead of a MST to optimize the non-local cost aggregation procedure [17]. Conceptually, they segmented the image at first and then constructed a sub-tree (i.e., a sub-MST) for each segment. Finally, a ST was constructed by linking these sub-trees for the non-local cost aggregation. The main idea, using the segment prior to avoid connecting two pixels which locate at the different sides of a segment boundary (i.e., potential depth boundaries), is similar with other segment-based stereo matching methods [25,26]. Both the MST method and the ST method outperform the local methods in aggregation accuracy [14,21].

Actually, constructing a tree or a graph to improve the optimization procedure is not new in many global methods such as *Dynamic Programming* (DP) and *Loop Belief Propagation* (LBP). Veksler firstly employed DP on a tree instead of a scanline to enforce vertical consistency to establish truly global optimization results [27]. Cheng Lei et al. improved the tree-based DP method by using a novel tree structure which they called region-tree [28]. Zitnick et al. formulated a new MRF for global optimization by over-segmenting the images [4].

In conclusion, for both the non-local cost aggregation and the global optimization which are based on a tree structure, the critical problem is the construction of the tree. In this paper, we focus on the non-local cost aggregation methods only. Hence, we describe the problem in the context of the non-local cost aggregation. To construct a tree from a graph, the important edges need to be preserved and the edges crossing the depth boundaries must be removed. By an important edge, we mean an edge whose two nodes are more likely to have the same disparity. Local criterions (i.e., color difference [16], distance [27], etc.), non-local properties (i.e., segmentation [17], over-segmentation [4], etc.) or a joint version of the two can be used to decide whether an edge should