Digital Signal Processing 53 (2016) 51–61

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Real-time stereo matching using extended binary weighted aggregation

Sanghun Kim [a], Suk-Ju Kang [b], Young Hwan Kim [a,*]

[a] Department of Electronics and Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, Republic of Korea
[b] Department of Electronic Engineering, Sogang University, Seoul 04107, Republic of Korea

ARTICLE INFO

Article history:
Available online 4 February 2016

Keywords:
Real-time stereo matching
Binary weighted aggregation
Sparse support region
Outlier-excluded aggregation

ABSTRACT

This paper presents an accurate real-time stereo matching method, which is based on the extended binary weighted aggregation. The accuracy of the proposed stereo matching method was significantly enhanced by extending its binary weighted aggregation so that remote connections of support regions can be allowed for aggregation. The extended binary weighted aggregation is based on the following two new ideas. First, the extended binary weighted aggregation connects distant regions over color boundaries, making them one large support region for a given pixel. This approach induces more aggregation targets, and, thus, makes the aggregation step more robust. Second, it excludes cost outliers in the support region to prevent them from being propagated during the aggregation step, making a quality support region. With the extended binary weighted aggregation, the proposed stereo matching method obtains more accurate disparity maps than existing stereo matching methods using binary weighted aggregation methods, while maintaining the speed advantage of binary weighted aggregation. Experimental results illustrated that the proposed stereo matching method outperformed all existing real-time stereo matching methods in terms of accuracy, providing the average bad pixel rate of 5.12%, for the Middlebury stereo test images. The proposed stereo matching method was implemented on a CUDA platform with a high-end GPU. The implemented system operated at up to 300 fps for the stereo images with 320×240 pixel resolution and a disparity range of 32 pixels.

1. Introduction

Stereo matching is the process of computing a disparity map from a given stereo image pair. An image pair indicates left and right images taken by a horizontally linked stereo camera. The disparity is the positional difference between corresponding points in the two images. Stereo matching is used for various applications, for example, recognizing surrounding geographical features for robot navigation [1], generating virtual reality for an immersive conference [2], and making more abundant and robust 3D user interfaces [3]. It is notable that many stereo matching applications require real-time processing, despite the fact that stereo matching suffers from massive computation. Fortunately, parallel processing technology has been remarkably advanced in terms of both hardware and software. Therefore, real-time stereo matching methods have been extensively studied in recent years.

Stereo matching can be categorized into two types: global and local methods. Global methods formulate an energy function that represents the stereo matching problem to be resolved, and then find the optimal solution by minimizing this energy function [4,5]. On the other hand, local methods compute the per-pixel dissimilarities for disparity candidates, and then determine the optimal disparity for each pixel. Local methods are inherently easy to compute and appropriate for parallelization. Therefore, most recent real-time stereo matching methods have adopted local methods [6–12]. Generally, local stereo matching methods consist of the following four steps although some steps can be omitted or performed implicitly [13]: *cost computation*, *cost aggregation*, *disparity computation*, and *disparity refinement*.

Cost computation – A cost represents the dissimilarity between two pixels. Each pixel is compared with all pixels in another image that are in the disparity range. The sum of absolute differences (SAD) is the most intuitive cost, and it has been widely used. However, SAD has some limitations. For example, if two cameras are placed under different lighting conditions, it provides inferior results. Hamming distance (HD) of census-transformed data [14] and zero-mean normalized cross-correlation (ZNCC) [15] are also widely utilized because they efficiently consider the relationship between the center pixel and neighboring pixels. Thus, they are relatively robust to different lighting conditions. Based on the tra-

* Corresponding author. Fax: +82 54 279 5933.
E-mail address: youngk@postech.ac.kr (Y.H. Kim).

http://dx.doi.org/10.1016/j.dsp.2015.12.019
1051-2004/

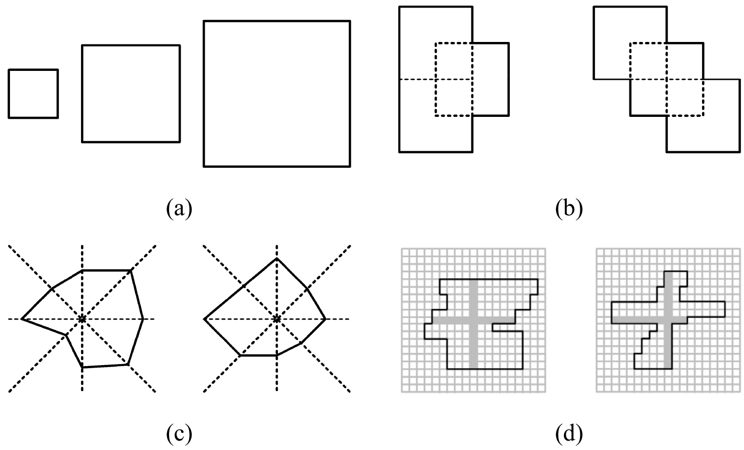

Fig. 1. Graphical representations of various kinds of binary weight: (a) variable window, (b) multiple window, (c) adaptive polygon, and (d) cross-based support regions.

ditional costs, summed normalized cross-correlation (SNCC) [16] and mean sum of relative pixel intensity differences (DIFFCensus) [17] were proposed, and they achieved enhanced matching quality by improving robustness to radiometric distortions.

Cost aggregation – Costs of the pixels in the support window are combined to enhance the matching accuracy. When the costs are mixed, they are weighted based on the relationship between the corresponding pixel and the center pixel, such as the spatial and color distances. The most important task in the cost aggregation step is to set the weights properly. In Section 2, the motivations and effects of existing aggregation methods are described in detail.

Disparity computation – The disparity of a pixel corresponding to the minimum per-pixel cost is selected.

Disparity refinement – The final disparity map can be obtained through a refinement step. There are various refinement approaches. For example, the outlier detection and correction strategy was utilized to remove unstable disparity [12], and region voting method was used to propagate reliable disparity [11]. In addition, median filtering and bilateral filtering [18] are extensively used to reduce the error in the final step.

In this paper, we propose a highly accurate real-time stereo matching method. To accomplish this, the proposed method employs a binary weight, which is easy to compute, for the aggregation step, and it removes the connectivity constraint of the existing binary weighted methods. In other words, the proposed method performs the aggregation step using sparsely distributed binary weights, whereas all binary weights should be connected with the center pixel in existing binary weighted methods. The proposed extended binary weighted method leads to an elaborate aggregation, and thus shows a high accuracy. In addition, the whole proposed stereo matching system including the extended binary weighted aggregation step is implemented on the graphics processing unit (GPU) using the compute unified device architecture (CUDA) platform.

The rest of this paper is organized as follows. In Section 2, a review of existing aggregation methods is introduced. In Section 3, the proposed real-time stereo matching method is then presented. In Section 4, the experimental results are compared with those of existing real-time stereo matching methods. Finally, Section 5 provides some concluding remarks regarding this research.

2. Background

In this section, existing aggregation methods are briefly described. The aggregation step significantly influences the performance of local stereo matching methods in terms of both accuracy and operational speed.

According to the type of weight value, aggregation methods can be classified into two types: binary weighted aggregation and real-valued weighted aggregation. Although binary weighted aggregation generally provides low accuracy, it requires low computational complexity. On the contrary, real-valued weighted aggregation results in high accuracy but requires high computational complexity.

2.1. Binary weighted aggregation

Initially, a square window with a fixed size is employed in the aggregation step, allowing all costs in the window to be averaged. However, depth-discontinuous regions become more blurred as the window size increases, and errors in weakly-textured regions grow as the window size decreases.

To overcome the drawbacks of a fixed sized window, both variable window [19] and multiple window [20] have been proposed. A variable window assigns a square window with an adaptive size to each pixel based on the cost function and its variance, as shown in Fig. 1(a). The multiple window approach divides a fixed size window into sub-windows, and then selects some of the sub-windows as a support region that means the set of pixels to be aggregated, as shown in Fig. 1(b).

A more flexible shape, i.e., a polygon, as shown in Fig. 1(c), is presented to overcome the limitations of a rectangle-shaped support region [21,22]. Four or eight vertices move away from the center pixel in each direction until they meet a pixel that has a very different color from the center pixel. The polygon-shaped support region is then simply generated by connecting the vertices.

The most recent binary weighted aggregation uses a cross-based support region, which assigns the binary weight in a pixel unit [23]. The term “cross-based” is used because each pixel has a cross-shaped element. To generate the element of each pixel, four arms are expanded until they meet a pixel that has a very different color as the center pixel. The support region is dynamically synthesized by gathering the horizontal crosses of the pixels in the vertical cross of the center pixel, as shown in Fig. 1(d). The accuracy is increased because the shapes of the assigned binary weights are more suitable for each pixel.

Existing binary weighted aggregations have a connectivity constraint in that all pixels assigned binary weights should be connected with the center pixel. This constraint is the key reason for the fast operational speed. However, the connectivity constraint is also one of the reasons for the relatively poor accuracy. There are many pixels that cannot belong to the aggregation targets owing to the connectivity constraint.