Pattern Recognition 44 (2011) 2701–2710

Contents lists available at ScienceDirect

Pattern Recognition

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

# Near-real-time stereo matching with slanted surface modeling and sub-pixel accuracy

Minglun Gong [a,*], Yilei Zhang [b], Yee-Hong Yang [b]

[a] Memorial University of Newfoundland, St. John's, Newfoundland, Canada A1B 3X5
[b] University of Alberta, Edmonton, Alberta, Canada T6G 2E8

ARTICLE INFO

*Article history:*
Received 24 August 2010
Received in revised form
12 March 2011
Accepted 23 March 2011
Available online 29 March 2011

*Keywords:*
Real-time stereo
Adaptive-weight cost aggregation
GPU computing

ABSTRACT

This paper presents a new stereo matching algorithm which takes into consideration surface orientation at the per-pixel level. Two disparity calculation passes are used. The first pass assumes that surfaces in the scene are fronto-parallel and generates an initial disparity map, from which the disparity plane orientations of all pixels are estimated and refined. In the second pass, the matching costs for different pixels are aggregated along the estimated disparity plane orientations using adaptive support weights, where the support weights of neighboring pixels are calculated using a combination of four terms: a spatial proximity term, a color similarity term, a disparity similarity term, and an occlusion handling term. The disparity search space is quantized at sub-pixel level to improve the accuracy of the disparity results. The algorithm is designed for parallel execution on Graphics Processing Units (GPUs) for near-real-time processing speed. The evaluation using Middlebury benchmark shows that the presented approach outperforms existing real-time and near-real-time algorithms in terms of subpixel level accuracy.



## 1. Introduction

The binocular stereo matching problem has been extensively studied in the past few decades because of its many applications. As well, the taxonomy and evaluation method proposed by Scharstein and Szeliski [17] also contribute to the increase in attention to this problem. According to their taxonomy, stereo matching algorithms perform all or some of the four steps: matching costs initialization, cost aggregation, optimization, and refinement.

Different stereo algorithms can be classified into global and local approaches based on the optimization techniques used. Global optimization methods, such as graph cuts [3] and belief optimization [19], in general give more accurate results. Local (winner-take-all [21]) and scanline (dynamic programming [23]) optimization approaches, on the other hand, have lower computational cost and are the popular choice in real-time applications.

The selection of cost aggregation approach also has important impact on result accuracy. Previous study [4] shows that, among the cost aggregation approaches evaluated under the real-time stereo setting, the one based on adaptive-weight [28] performs the best. Our recent research suggests that the adaptive-weight cost aggregation can be further improved through aggregating the costs along surface orientation [31]. It is therefore worth noting to investigate whether it is possible to incorporate this improvement into real-time stereo applications.

As a follow up to our preliminary work [31], we hereby propose a novel two-pass real-time stereo approach, which introduces per-pixel non-fronto-parallel disparity plane modeling and performs adaptive-weight cost aggregation in 3D cost volume along slanted planes. In addition, the following improvements are made over the previous approach [31]: (1) the new adaptive weight defined not only considers the spatial proximity and color similarity, but also includes an occlusion term to better handle areas that are occluded in the second view; (2) a dynamic programming based optimization approach is used, instead of the local winner-take-all, for better result accuracy; and (3) the whole algorithm is re-designed for parallel processing and implemented on GPU for real-time performance.

The remaining of the paper is organized as following: Related work is discussed in Section 2. Section 3 gives the details of the proposed two-pass algorithm. The experimental results are presented in Section 4. Finally, the paper concludes in Section 5.

## 2. Related work

### 2.1. Cost aggregation

Due to image capturing noise and non-Lambertian reflection, selecting the best disparity hypotheses based on matching costs

* Corresponding author.
*E-mail address:* gongml@gmail.com (M. Gong).

0031-3203/$ - see front matter 
doi:10.1016/j.patcog.2011.03.028

calculated using individual pixel pairs is often prone to error. Cost aggregation is therefore an important step, which replaces an initial matching cost with the (weighted) average cost within a local support window [17]. Ideally, the support window should be large enough to capture sufficient intensity variation for handling weakly textured areas, while at the same time, small enough to exclude pixels of different disparities to preserve object boundaries. Furthermore, when there are horizontally slanted surfaces in the scene, different window sizes shall be used for the two stereo views to match the unequal projection sizes of slanted surfaces [14].

Unlike the adaptive window approaches [2,10], which focus on varying the size, shape, and position of the support window, the adaptive-weight method gives promising results using a large fix-sized window with varying support weight for each pixel in the window [28]. The weight is calculated based on Gestalt Principles, which state that the grouping of pixels should be based on spatial proximity and chromatic similarity.

Several approaches have been proposed to further improve the matching accuracy upon the original adaptive-weight approach. Instead of computing the weights based the Gestalt Principles, Min and Sohn treat the cost aggregation as an optimization problem and compute the optimal weights via energy minimization [13]. Hosni et al., on the other hand, set the weight of each neighboring pixel based on the geodesic distance between the neighboring pixel and the center pixel [8]. They show that, since the connectivity between the two pixels is considered, more accurate results can be obtained with the same local optimization.

The above approaches assume all surfaces in the scene are fronto-parallel and aggregate matching costs within the 2D constant disparity planes. This assumption is often violated due to the large support window used — even when the slant is very small, the big neighborhood span can still go through multiple disparity levels. To address this problem, our preliminary work [31] models the slanted surfaces through estimating the disparity plane orientation at each pixel location. The cost aggregation is then performed in 3D disparity space along the estimated disparity plane orientation. As a result, smoother and more accurate disparity maps can be generated using local optimization. Aggregating per-pixel matching costs in 3D disparity space also implicitly uses different window sizes for the two views, without the need for explicitly stretching the one of the images as in [14].

To compute the aggregated cost for a give pixel under a given disparity hypothesis, the above adaptive-weight methods need to calculate the support weights for all pixels within the support window. Assuming the support window size is $W \times W$, the computation needed for aggregating a single cost is $O(W^2)$. Since the window size $W$ must be big enough for the aggregation to be effective, these adaptive-weight techniques are too computationally intensive for real-time application, even when used with the simplest local optimization approach.

### 2.2. Segmentation-based stereo

Instead of performing all computations on individual pixels, segmentation-based stereo algorithms first over-segment the input image into small homogeneously colored regions. The segmentation information is then used as *a priori* knowledge for further stereo calculation.

The disparity plane-fitting approaches [11,20] model the scene structure using a set of planar surface patches. Assuming that the pixels from the same color segment belong to the same patch, these approaches apply plane-fitting technique to find candidate disparity planes for each segment. The optimal disparity plane assignment can be determined using either local [20] or global [11] optimization. Since the fitted disparity planes naturally provide sub-pixel disparity values, the scene can be reconstructed at a much finer level. Techniques for handling slanted surfaces using segmentation results are also proposed [1].

Attempts have been made to combine segmentation information with adaptive-weight cost aggregation. Tomabari et al. propose to assign full support weights to pixels in the same segment with the pixel of interest, whereas the weights for pixels outside of the segment are calculated using the original adaptive weight approach [22]. Yang et al. integrate techniques such as color segmentation, plane fitting, adaptive weight, and belief propagation in an iterative process [25]. Their approach can generate highly accurate disparity maps, but is rather slow since the global optimization process is performed multiple times.

While utilizing segmentation information generally helps to obtain more accurate results, generating the segmentation itself is a challenging problem and consumes additional processing time. As a result, these techniques are not suitable for real-time applications. Our approach does not require *a priori* image segmentation and is designed for parallel processing on GPUs.

### 2.3. Real-time stereo matching

Due to the processing speed requirement, most real-time stereo systems employ per-pixel [21] or per-scanline [23] optimization. Recent advances in computing power, however, make it possible to perform more complex operations in real-time. For example, real-time belief propagation based techniques has been developed [24,26].

Several techniques have been proposed for incorporating adaptive-weight cost aggregation in real-time systems. Among them, Wang et al. use the adaptive-weight calculation in a dynamic programming (DP) framework, where the matching costs are aggregated within 1D vertical scanlines and the dynamic programming is performed along horizontal scanlines [23]. As a result, smoothness is enforced along both directions, reducing the "streaking" artifacts associated with traditional DP algorithms.

With two simplifications, it is also possible to perform adaptive-weight aggregation within 2D squared window in real-time [4,23]. The first simplification is to compute the weights using the central image only and omit the weight term obtained using the reference image. The second one is to approximate the weighted average using a two-pass approach, with one pass aggregates along the horizontal scanlines and the other computes along the vertical ones. Hence, the computation needed for aggregating a single cost is cut down to $O(W)$, where $W$ is the width of the support window. The simplified approach, though yields more matching errors than the original adaptive-weight approach, still outperforms other real-time aggregation techniques evaluated in [4].

The exponential step size adaptive weight (ESAW) algorithm [29] proposed by Yu et al. further simplifies the above real-time adaptive-weight implementation. Instead of directly calculating the weight of a distant pixel on the same scanline, ESAW approximates it with the production of the weights calculated for using multiple pixel pairs. This simplification further cuts down the computational cost for a single pixel to $O(\log W)$.

The process of adaptive-weight aggregation can also be formulated as cross-bilateral filtering, where the filtering of the matching cost volume is guided by the color variation in the input image. Inspired by acceleration techniques for bilateral filtering, Richardt et al. recently propose a real-time implementation using dual-cross-bilateral (DCB) grid [15]. To generate disparity maps at subpixel accuracy, they also incorporate Yang et al.'s sub-pixel refinement technique [27].

The approach presented in this paper uses the same two simplifications proposed in [4,23] to reduce the computational