Computer Vision and Image Understanding 117 (2013) 620–632

Contents lists available at SciVerse ScienceDirect

Computer Vision and Image Understanding

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

Secrets of adaptive support weight techniques for local stereo matching ☆

Asmaa Hosni *, Michael Bleyer, Margrit Gelautz

Institute for Software Technology and Interactive Systems, Vienna University of Technology, Favoritenstr. 9-11, Vienna, Austria

ARTICLE INFO

Article history:
Received 24 July 2012
Accepted 19 January 2013
Available online 4 February 2013

Keywords:
Local stereo matching
Adaptive support weights
Evaluation study

ABSTRACT

In recent years, local stereo matching algorithms have again become very popular in the stereo community. This is mainly due to the introduction of adaptive support weight algorithms that can for the first time produce results that are on par with global stereo methods. The crux in these adaptive support weight methods is to assign an individual weight to each pixel within the support window. Adaptive support weight algorithms differ mainly in the manner in which this weight computation is carried out.

In this paper we present an extensive evaluation study. We evaluate the performance of various methods for computing adaptive support weights including the original bilateral filter-based weights, as well as more recent approaches based on geodesic distances or on the guided filter. To obtain reliable findings, we test these different weight functions on a large set of 35 ground truth disparity pairs. We have implemented all approaches on the GPU, which allows for a fair comparison of run time on modern hardware platforms. Apart from the standard local matching using fronto-parallel windows, we also embed the competing weight functions into the recent PatchMatch Stereo approach, which uses slanted sub-pixel windows and represents a state-of-the-art local algorithm. In the final part of the paper, we aim at shedding light on general points of adaptive support weight matching, which, for example, includes a discussion about symmetric versus asymmetric support weight approaches.

1. Introduction

Local algorithms have a long tradition in stereo matching. They typically use squared match windows that are displaced in the second image to find a correspondence. The use of a support window leads to an implicit smoothness assumption, i.e., all pixels within the window are assumed to have the same constant disparity. The inherent problem of standard local algorithms is that this smoothness assumption is broken at depth discontinuities where the window contains pixels of the background as well as of the foreground disparity. This leads to the well-known foreground fattening effect.[1] Almost all early papers on local stereo matching have concentrated on overcoming this edge fattening effect (e.g., [1,2] to cite a few), but have not been convincingly successful in this attempt.

In 2005, Yoon and Kweon [3] introduced a surprisingly simple strategy that can clearly outperform previous local algorithms, i.e., adaptive support weight (ASW) stereo matching. This has led to the first local method that is able to compete with global algorithms in terms of quality of results (e.g., see results in the Middlebury benchmark [4]). The key idea to overcome edge-fattening is to assign an individual weight to each pixel of the match window. This weight determines the pixel's influence in the matching process. Ideally, pixels lying on the same disparity as the window's center pixel should obtain high weights (high influence) and pixels of a different disparity should obtain low weights (low influence). Defining these weights leads to a chicken and egg problem given that the disparity map is not known in advance. Yoon and Kweon's algorithm [3] as well as all subsequent ASW papers (e.g., [5–7]) exploit the appearance (color) cue to estimate the weights. The idea is that pixels whose colors are similar to the center pixel's color are also likely to be similar in disparity. We review different methods to compute these weights in the following. Note that ASW methods basically differ in the way they calculate these weights. In the first part of this paper, we present an evaluation study that aims at identifying the best weighting function, i.e., the one that leads to the lowest percentage of disparity errors on our benchmark test set.

In the original ASW paper [3], a pixel's weight inside a support window is computed from: (1) its color similarity to the window's center pixel and (2) its spatial distance from the center pixel. Note that this is equivalent to the way weights are computed in bilateral filtering, and it is known that the aggregation step of Yoon and Kweon's method [3] can be understood as filtering the cost volume

☆ This paper has been recommended for acceptance by Shahriar Negahdaripour, Ph.D.

* Corresponding author.
E-mail addresses: asmaa@ims.tuwien.ac.at (A. Hosni), bleyer@ims.tuwien.ac.at (M. Bleyer), gelautz@ims.tuwien.ac.at (M. Gelautz).

[1] Note that the assumption of constant disparity within the window is also broken for slanted surfaces where the match window contains pixels of many slightly different disparities. We will address this problem later.

1077-3142/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2013.01.007

(disparity space image (DSI)) with a joint bilateral filter [8,7]. (This equivalence forms the foundation for various fast ASW algorithms discussed later.)

In [9] an alternative weighting function that relies on a precomputed mean-shift color segmentation is proposed. All pixels that lie in the same color segment as the center pixel are given a weight of 1, while all pixels outside the segment are given weight 0. The work of [5] refines that strategy by treating pixels that lie outside of the center pixel's segment differently. The authors use the weighting function of [3] for these pixels. (Since better results can be expected from the latter strategy, we only include [5] in our benchmark study.) Note that a problem of both algorithms is the computational overhead due to the computation of the mean-shift segmentation.

Hosni et al. [6] define the weights within a window by computing the geodesic distance to the center pixel. The authors claim that better segmentations can be obtained by enforcing connectivity, i.e., to obtain a high weight a pixel needs to have a path of approximately constant color to the center pixel. For example, consider a plant with green leaves, each at a different disparity. For a support window that overlaps multiple leaves, the weighting function of [3] erroneously assigns high weights to all pixels regardless whether they lie on the center pixel's leaf or not, because all of them are green. Hosni et al. [6] find that there is a color edge between the center pixel's leaf and all other leaves so that pixels outside of the center pixel's leaf cannot have a path of constant color and correctly derive low weights.

While the above papers focus on improving the quality of disparity maps, there is also a different branch of ASW algorithms that focus on computational speed. The main disadvantage of the methods above is that their computational complexity directly depends on the size of the support windows. Unfortunately, these support windows have to be large (e.g., 31×31 pixels) in order to handle untextured regions. In this case, computing the disparity map for a Middlebury pair can easily consume 1 min in a CPU-based implementation.

As stated above, the aggregation step of [3] is equivalent to filtering the cost volume with a joint bilateral filter. Hence, being able to implement the original ASW approach with a runtime independent of the window size boils down to the question whether it is possible to implement joint bilateral filtering with a runtime complexity independent of the filter kernel size. According to the current state of research, an O(1) implementation only works for *approximations* of the joint bilateral filter. Several authors [8,10–12] have used such approximations to derive fast implementations of the original ASW algorithm [3]. In [10,11] the joint bilateral filter is approximated by using integral histograms as described in [13]. Richardt et al. [8] uses an approximation based on the bilateral grid of [14]. Finally, the authors of [12] have presented a cost aggregation strategy that is also based on joint bilateral filtering and applies an incremental calculation scheme similar to [15]. In our benchmark, we include [8,16] as representatives of these approximate methods. The downside of these methods is that they sacrifice quality for high computational speed, and we will demonstrate this in our evaluation study.

A different strategy to derive an O(1) implementation of ASW matching is to replace the joint bilateral filter with a different filter that shares the joint bilateral filter's edge-preserving property, but can innately be implemented with a runtime independent of the filter kernel size. In this line of research, Zhang et al. [17,18] use a cross-shaped filter. Due to using a cross-shaped support region, the algorithm fails at fine structures that are neither horizontal nor vertical. Another approach is presented in [19]. The authors of that paper propose a new filtering technique, i.e., "*information permeability*" filtering. This filter can be interpreted as a hybrid approach between the cross-based filter [17] and the geodesic one [6].

A better alternative (that we include in our benchmark) is to use the recently proposed guided filter [20] for smoothing the cost volume, as has been done in [7,21]. Note that an interesting aspect of [7] is that the concept of smoothing the cost volume with an edge-preserving filter can be generalized to other computer vision problems that are typically formulated as Markov Random Fields.[2] Hence, we expect that the results of the benchmark presented in our paper are also valuable for researchers outside of stereo matching.

Moreover, it is worth mentioning the fast stereo matching method presented in [22]. The authors of that work have achieved efficient performance by: (1) reducing the computational redundancy that occurs when the aggregation is repeated at every disparity hypothesis; (2) implementing the cost aggregation step from a histogram perspective using an efficient sampling strategy.

The main contribution of this paper lies in a systematic evaluation study on ASW local stereo matching methods. We thereby focus on the role of the weights used in the cost aggregation step. Standard benchmarks such as the Middlebury online table [4] already compare a relatively large number of different ASW approaches. However, we believe that the differences in Middlebury rankings do not necessarily originate from the different weight computation methods, but rather from other ingredients such as the use of different match measures, different postprocessing procedures and the amount of parameter tuning that authors apply to optimize their ranking on the four Middlebury images. In this paper, we opt for an improved evaluation strategy by testing the weight computation algorithms under constant surrounding conditions (i.e., same match measures and postprocessing scheme) and by using a large number of 35 test images (in contrast to the four Middlebury pairs). Our benchmark shall aid other researchers in making their decisions on which weight computation method they should choose. We concentrate on two questions: Firstly, what is the best support weights computation method to achieve maximum quality? Secondly, how fast can we go in ASW matching?

While we focus on standard local matching using fronto-parallel support windows in the first part of our experiments, we embed the ASW methods in an improved algorithm in the second part. This improved algorithm, i.e., PatchMatch Stereo [23], uses slanted planar support windows matched at continuous sub-pixel disparities. By moving away from the traditionally applied fronto-parallel assumption, PatchMatch Stereo achieves impressive results for slanted and rounded surfaces and shows excellent sub-pixel performance. The second series of experiments shall provide an answer to the question on how far we can go in local matching in order to maximize matching quality. Moreover, it is interesting to compare the performance of ASW strategies, if a fundamental weakness of traditional local approaches, i.e., that of using only fronto-parallel windows, is eliminated.

Another contribution of this paper is that we try to reveal several secrets of ASW stereo matching that we believe to be interesting for other researchers working in this domain. In particular, we focus on the following questions: (1) Does it make sense to compute support weights in a symmetrical manner from both images (as e.g., proposed in [3]) or is it sufficient to only use the left image in support weight computation? (2) Some researchers (e.g., in [23]) simplify the original way of ASW computation [3] by removing the spatial term to get rid of one parameter and claim that this does not considerably worsen disparity results. We investigate if this claim is true. (3) Does it make sense to apply preprocessing on the color image that is used in the weight computation (e.g., by median filtering) in order to get less noisy support weight masks?

[2] The authors demonstrate this for optical flow estimation and interactive image segmentation.