ISSN 2348 – 9928
IJAICT Volume 3, Issue 2, February 2016 Doi:01.0401/ijaict.2014.05.04 Published on 05 (2) 2016

MODIFIED GUIDED IMAGE FILTER

B. Soorya
PG Scholar,
Electronics and Communication Engineering,
Easwari Engineering College,
Chennai, Tamilnadu, India

Dr. S. Dhanalakhsmi
Assistant professor,
Electronics and Communication Engineering,
Easwari Engineering College,
Chennai, Tamilnadu, India

Abstract— Filtering is widely used in image and video processing for various applications. In this paper a explicit image filter called guided filter is proposed to remove noise in images smoothing and sharpening of images. The guided filter is derived from the local linear model compute the filtering output by considering the content of guided image which can be the input image or another different image. The guided filter can be used as an edge preserving smoothing operator. Transfer the structure of guided image to the filtering output enabling and filtering application like dehazing and guided feathering. Computer vision and computer graphics applications, including edge-aware smoothing, detail enhancement, HDR compression, image matting.

Keyword— Edge, Preserving Filtering, Linear Time Filtering.

I INTRODUCTION

Guided image filtering, the task of matching the images taken from a stereo camera and extracting the depth of objects in a scene [1], is commonly employed in embedded vision applications such as intelligence surveillance, autonomous vehicles and mobile robots [1]. Such applications need to satisfy real -time processing speed, high matching accuracy and low-power consumption constraints. The matching algorithm and the implementation platform are both factors that play a significant role in satisfying the requirements of an embedded stereo matching system. Global stereo matching algorithms produce very accurate results [2], but rely on the high-end hardware resources of multi-core CPUs and/or GPU platforms to achieve real-time processing.

Such platforms therefore appear to be unsuitable for the realization of stand-alone stereo matching systems, and also consume excessive power, which is not desirable in battery, powered mobile and embedded devices. In contrast, local algorithms can be greatly benefited by the use of parallel structures implemented on either FPGAs or ASICs, providing the necessary computational power and energy efficiency for embedded vision applications [3] To this end, several real-time stereo matching hardware systems have been developed during the past decade. However, the majority of them have implemented local algorithms that rely on standard block-based aggregation with a fixed support window [3]. While these algorithms can achieve very high frame rates when implemented in hardware [3], they lead to low matching accuracy [4]. As such, a few attempts have been made recently to implement dedicated hardware architectures of more accurate algorithms, such as Semi Global Matching (SGM) [5], [6] and Adaptive Support Weight (ADSW) [7], [8],[9]. For the past few years, hardware implementations based on SGM and ADSW algorithms have become the preferred solution towards higher matching accuracy in embedded vision applications. However, existing implementations have made several modifications and simplifications to adapt the algorithms for real-time processing, resulting in noticeable quality reduction compared to the original software algorithms. In addition, their high memory and hardware demands might limit their scalability to higher resolution images. Recently, the idea to utilize the Guided Image Filter (GIF) [10] in local ADSW stereo matching algorithms has been proposed to reduce the complexity of the cost aggregation step. Such software implementations have yielded promising results [11].

Motivated by the results of the software implementation presented in [11], this paper proposes a fully pipelined, parallel and scalable stereo matching hardware architecture that integrates the recently proposed GIF. There are two main novel contributions in this paper. First, it presents a new and efficient design of the GIF (that can be potentially adopted in other uses of the filter). The Guided Filter has also been applied in the optical flow estimation and image segmentation. In the following: Sections II & III provide background and related work, while Section IV proposed guided image filter Section V shows results and comparison with related work. Finally, Section VI concludes the paper VII future work.

II. BACKGROUND ON STEREO MATCHING

2.1 Overview & Classification of Stereo Matching Algorithms

Stereo matching is a technique aimed at inferring depth information of a scene from a pair of stereo images (usually called reference and target images) [3]. The depth is determined by locating corresponding pixels in the stereo images. Given that the input images are rectified [12], the correspondence of a pixel at coordinate (x, y) of the reference image, can only be found at the same vertical coordinate y, and within a maximum horizontal bound, called disparity range D (dm – dM), in the target image. The disparity is then computed as the location difference of corresponding pixels in both images. The disparities of all pixels form a disparity image, or disparity map, from which depth information can be extracted.

According to [2], stereo matching algorithms mostly follow four steps: 1) matching cost computation, 2) cost aggregation, 3) disparity computation/optimization and, 4) disparity refinement. Moreover, [2] classifies stereo matching algorithms into two broad categories: global and local. Global algorithms are usually formulated as an energy minimization problem, which is solved with techniques such as Dynamic Programming, Graph Cuts and Belief Propagation. Such methods produce very accurate results at the expense of high computational complexity and memory needs. The Semi-Global Matching (SGM) [5], [6] methods renounce part of the accuracy by approximating a global 2D function using a sum of 1D optimizations from all directions through the image. SGM methods are therefore more affordable for dedicated hardware implementation, but they still consume excessive memory to store the temporary cost of different aggregation paths.

In contrast, local algorithms determine the disparity associated with a minimum cost function (see [2] for a review) at each pixel by performing block matching and winner-takes-all optimization. Hence, they have lower computational complexity and memory requirements compared to global and SGM methods. Early local algorithms relied on simple aggregation strategies that perform block matching by using either a fixed (typically square) window, or multiple windows with different sizes. However, these approaches are prone to matching errors at depth discontinuity regions; they blindly aggregate pixels belonging to different disparities due to the use of a fixed window (shape and/or size) [4]. Among local algorithms, the most recent adaptive support weight (ADSW) methods are currently the most accurate. They work by assigning different weights to the pixels in the support window based on their colour/ proximity distances to the central pixel. In this way, they aggregate only those pixels that lie at the same disparity, leading to very good quality at depth borders [4]. Despite their good quality results, ADSW algorithms cannot take advantage of the “integral image” or “sliding window” techniques, as the adaptive weights have to be recomputed at every pixel. This makes the cost aggregation’s hardware complexity directly dependent on the support window size.

2.2 Stereo Matching using Guided Image Filtering

The recently proposed Guided Image Filter (GIF) [10] has been employed in [11] to reduce the complexity of the cost aggregation step in ADSW methods, leading to a high-quality fast and simple algorithm (Fig. 1), with the following steps:

2.3 Cost Volume Construction

This step calculates a matching cost for each pixel p at all possible disparities. The output is a three-dimensional structure consisting of D cost images (Stereo Cost Volume - SCV). Each cost is computed as the truncated absolute difference of colors and gradients, a metric that exhibits good robustness to illumination changes

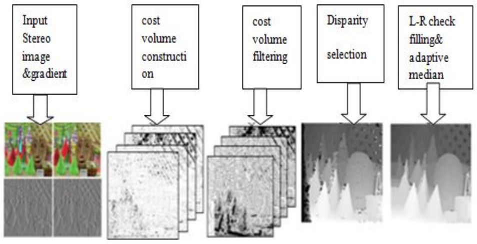


Fig 1: Major Steps of the GIF-based Stereo Matching Algorithm.

The overall cost function C(p,d) is computed with (1)-(3), where a is used to balance the influence of the colour and gradients terms, and Tc and Tg are truncation thresholds.

2.4 Cost Volume Filtering

This step utilizes the GIF to smooth each slice of the SCV. Due to its edge-preserving property, the GIF leads to good accuracy at depth discontinuities. Typically, the filtered cost value at p and disparity d is a weighted average of the pixels in the same slice of the SCV, and is expressed as in (4).

The GIF generally uses a guidance image I to filter a guided image f. In our case, the guidance image is the greyscale reference image, while the guided image is a slice (x, y) of the SCV. The filter weights are defined as in (5), where μ k and σk are the mean and the variance of I in a squared window ωk with dimensions r×r, cantered at pixel k. $|\omega|$ is the number of pixels in the window and ε is a smoothness parameter. An inherent advantage of the GIF is that the weights can be computed with some linear operations (see [10]), which can be decomposed into a series of mean filters with windows radius r. This leads to an efficient algorithm.

2.5 Disparity Selection

Once the SCV slices are filtered, the best disparity for pixel p is chosen through a simple Winner-Takes-All (WTA) minimization approach using (6).Disparity Refinement. A left/right consistency check (L-R check) is performed. Hence, the disparity map, DR, using the right image as reference is also computed. The L-R check marks disparities as invalid if the disparity DL (p) and its corresponding disparity of DR (p) differ by more than 1 pixel. Invalid pixels are then filled with the minimum disparity between their closest consistent pixels in the left and right direction. Weighted and typical median filtering is applied next to smooth the filled regions and remove spikes.

Algorithm: Guided Filter. Input: I, C(p,d). Parameters: r, ε.

```
1: $mean_{I} = f_{mean}(I)$
   $mean_{p} = f_{mean}(p)$
   $corr_{I} = f_{mean}(I.*I)$
   $corr_{Ip} = f_{mean}(I.*P)$

2: $var_{I} = corr_{I} - mean_{I.} * mean_{I}$
   $cov_{I p} = corr_{Ip} - mean_{I}. * mean_{p}$
```

III. RELATED WORK

In recent years, a fair amount of work has been carried out on real-time hardware implementations of local stereo matching algorithms (e.g. [13]); a thorough review is presented in [3]. The majority of these implementations have adopted simple fixed support and multiple window methods, therefore trading accuracy for speed. High matching accuracy though is of foremost importance in many of today's embedded vision applications. As such, a few attempts have been made recently directed towards improving the matching accuracy, either by combining different stereo algorithms together, or by implementing modified versions of SGM and ADSW algorithms. The hardware implementation in [14] performs a modified version of the Census transform in both the intensity and gradient images, in combination with the SAD correlation metric. An FPGA implementation of a stereo algorithm based on the neural network and Disparity Space Image (DSI) data structure is introduced in [15]. The real-time FPGA-based stereo matching design presented in [16] combines the mini-Census transform and the Cross-based cost aggregation. SGM-based stereo matching systems have been introduced in [5], [6] and implemented on FPGAs and a hybrid FPGA/RISC architecture, respectively. The technical details/parameters of the different implementations are summarized

The works that are closely related to ours in terms of the matching algorithm are the works in [7], [8], [9], which implement ADSW-based algorithms. [7] Proposed the VLSI design of a hardware-friendly ADSW algorithm that adopted the mini-Census transform to improve the accuracy and robustness to radiometric distortion. [8] Proposed the implementation of a complete stereo vision system, which incorporates an ADSW algorithm and also integrates pre- and post- processing units. Finally, a hardware-oriented stereo matching system based on the adaptive Census transform is presented in [9]. The aforementioned high-quality ADSW-based systems follow a similar algorithm-to-hardware mapping methodology. That is, a complex, but accurate, algorithm is adapted for dedicated hardware implementation through a series of algorithmic modifications In most cases, however, these implementations scarify part of the accuracy; quality reduction compared to the original implementation of the ADSW approach in [4] is ~ 4-5%.

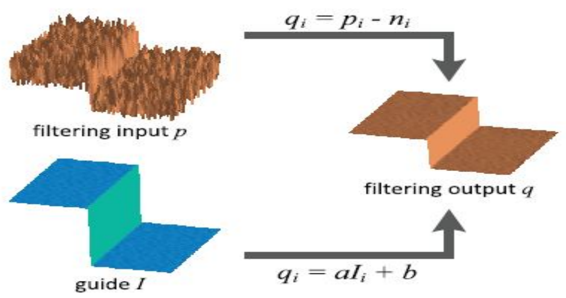
In contrast, the proposed stereo matching architecture implements the ADSW aggregation step in a different way; by smoothing the SCV with an edge-preserving filter, the GIF. Due to this type of filter, and its optimized hardware design presented in this work, the proposed architecture pushes further the accuracy limits of hardware -based stereo matching systems, while it also achieves real-time frame-rates for HD images.

IV. PROPOSED GUIDED IMAGE FILTER

In this section, we first describe how the GIF, the core element of the proposed stereo matching architecture, is efficiently in a way that its logic resources are independent of the kernel radius r.

4.1 Simulation of the Guided Image Filter

Besides the high quality obtained by smoothing the SCV with the GIF, this type of filter can have an efficient dedicated simulation, as the basic operation involved is the mean filter with windows of radius r. The mean intensity of pixels over rectangular windows in the image can be simulated in a fast way using the integral image technique. However, this technique requires huge amount of memory, especially for high-resolution images. Therefore, we instead followed a variant of the approach in [17], to implement a custom mean filter design that consumes compact hardware resources. The main idea is to maintain a sum for each column in the image to be filtered. Each column sum accumulates 2r+1 pixel, while the window sum is computed by adding 2r+1 adjacent column sums. When the window is moved to the right from one pixel to the next, the column sum to the right of the window is yet to be computed for the current row, so it is cantered one row above. Therefore, the first step consists of updating the column sum to the right of the window, by subtracting its topmost old pixel and adding one new pixel below it. The second step moves the window to the right and updates the window sum by subtracting its leftmost column sum (old column sum), and adding the updated column sum computed in step 1 (new column sum).The final mean value is computed by multiplying the window sum with 1/(2r+1)2.



The architecture of the GIF is depicted in it receives two pixels from the reference image (used as guidance image) and two from the slice of the SCV to be smoothed. The architecture consists of four mean filters that compute the values of meanI, meanp, corrI and corrIp. The remaining values of the algorithm shown in Fig. 2 are computed using a set of arithmetic units (fixed-point multipliers, adders/ sub tractors).The key assumption of the guided filter is a Local linear model between the guidance I and filtering output q.

4.2 Proposed GIF-based Stereo Matcher (GIF-SM)

The simplicity of the GIF architecture makes the stereo matching process independent of the match window size. Therefore, stereo matching can now rely on pixel-based operations. The Edge preserving smoothing operator is like the bilateral filter. Its avoid the gradient setback artifacts that may be appear in the HDR Compression. The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image.

The higher the PSNR, the better the quality of the compressed or reconstructed image. The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error. To compute the PSNR, the block first calculates the mean-squared error using the following equation:

V. RESULT AND DISCUSSION

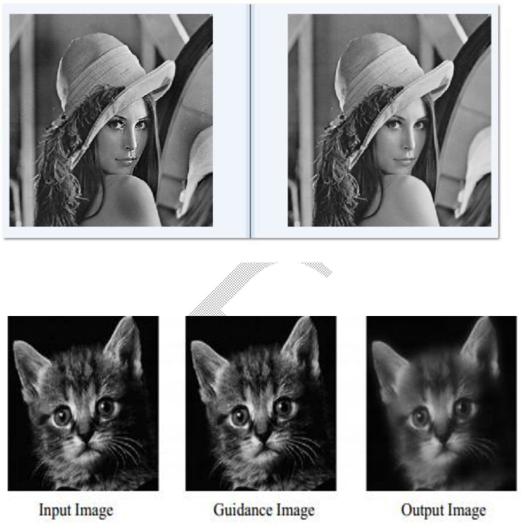


Fig 2 : Results from MATLAB simulation of Edge- Preserving filtering.

	r=2	r=4	r=8
$\varepsilon=0.1^2$	39.37	38.25	37.25
$\varepsilon=0.2^2$	38.11	37.75	37.05

Guided filter is an edge-preserving smoothing operator like the bilateral filter, it avoids the gradient setback artefacts that may appear in detail enhancement and HDR compression. Given the input signal p, its edge-preserving smoothed output is used as a base layer q. The difference between the input signal and the base layer is the detail layer d = p −q. The guided filter does not lose thin structures because the guided filter has a patch-wise model

VI. CONCLUSION

In this paper we propose a novel explicit guided image filter. Originated from a local linear model, this filter can be used to replace the soft matting step and lead total real-time performance. Show the edge-aware and gradient-preserving properties of this filter. The guided filter is a faster and better technique than previous filter. Edge-aware techniques have more applications in computer vision/graphics than what we have introduced in this section. In many applications, we assign each pixel an estimated value, which can be a cost, confidence, a vote or any other data.

VII. FUTURE WORK

Future work would be to extend the capability of process to improve a hardware implementation.

References

[1] B. Cyganek and J. P. Siebert, Introduction to 3D Computer Vision Techniques and Algorithms. Wiley, John & Sons, 2009.

[2] D. Szeliski and R. Scharstein, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," Inter. J. of Comput. Vision, vol. 47, pp. 7-42, 2002.

[3] L. Nalpantidis, G. C. Sirakoulis, and A. Gasteratos, "Review of stereo vision algorithms: from software to hardware," Inter. J. of Optomechatronics, vol. 2, no. 4, pp. 435-462, 2008.

[4] K.-J. Yoon and I.-S. Kweon, "Adaptive support-weight approach for correspondence search," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 650-656, Apr. 2006.

[5] S. K. Gehrig, F. Eberli, and T. Meyer, "A Real-Time Low-Power Stereo Vision Engine Using Semi-Global Matching," Computer Vision Systems, Lecture Notes in Computer Science, vol. 5815, pp. 134-143, 2009.

[6] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch, "Real-time stereo vision system using semi-global matching disparity estimation: Architecture and fpga-implementation," in ICSAMOS, 2010, pp. 93-101.

[7] N. Chang, T. Tsai, B. Hsu, Y. Chen, and T. Chang, "Algorithm and Architecture of Disparity Estimation With Mini-Census Adaptive Support Weight," IEEE Trans. on CSVT, vol.20, no. 6, pp. 792-805, 2010.

[8] J. Ding et al., "Real-time stereo vision system using adaptive weight cost aggregation approach," EURASIP JIVP, 2011.

[9] S. Perri, P. Corsonello, and G. Cocorullo, "Adaptive Census Transform: A novel hardware-oriented stereovision algorithm," Computer Vision and Image Understanding, vol. 117, no. 1, pp. 29-41, January 2013.

[10] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided Image Filtering," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 6, pp. 1397-1409, 2013.

[11] A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, and C. Rother, "Real-time local stereo matching using guided image filtering," in IEEE Int. Conf. on Multimedia and Expo (ICME), 2011, pp. 1-6.

[12] H.-S. Son, K.-r. Bae, S.-H. Ok, Y.-H. Lee, and B. Moon, "A Rectification Hardware Architecture for an Adaptive Multiple-Baseline Stereo Vision System," in Communication and Networking.: Springer Berlin Heidelberg, 2012, pp. 147-155.

[13] S. Jin et al., "FPGA design and implementation of a real-time stereo vision system," IEEE Trans. on CSVT, vol. 20, no. 1, pp. 15-26, 2010.

[14] K. Ambrosch and W. Kubinger, "Accurate hardware-based stereo vision," Comput. Vis. Image Und., vol. 114, no. 11, pp1303-1316, Nov2010.

[15] N. Baha and S. Larabi, "Accurate real-time neural disparity MAP estimation with FPGA," Pattern Recognition, vol. 45, no. 3, pp. 1195-1204, March 2012.

[16] L. Zhang et al., "Real-Time High-Definition Stereo Matching on FPGA," in FPGA'11, Monterey, CA, USA, 2011, pp. 55-64.

[17] S. Perreault and P. Hebert, "Median Filtering in Constant Time," IEEE Tran. on Image Process., vol. 16, no. 9, pp. 2389-2394, 2007.

[18] S. A. Fahmy, P. Y. K. Cheung, and W. Luk, "Novel FPGA-Based Implementation of Median and Weighted Median Filters for Image Processing," in FPL2005, Tampere, Finland, August 2005, pp. 142–147.