Proceedings of the World Congress on Electrical Engineering and Computer Systems and Science (EECSS 2015)
Barcelona, Spain, July 13-14, 2015
Paper No. 341

Efficient stereo matching by dropping disparity levels for FPGA implementation

Jiho Chang, Jae-chan Jeong
Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu, Daejeon, Republic of Korea
changjh@etri.re.kr; channij80@etri.re.kr

Abstract- Stereo matching is a traditional method used to obtain three-dimensional depth information and has been studied for decades. However, it is still difficult to apply stereo matching algorithms to real-time systems because of its heavy computation requirements. A stereo matching implementation of an FPGA system with high-resolution images uses a significant amount of logic and memory. When implementing stereo matching in FPGA, factors that determine the size of logic and memory required are the resolution of the images and the disparity levels. In this paper, we present a spare cost computation method to implement a stereo matching system on FPGA by dropping disparity levels. In addition, using a subpixel estimation and filtering method to calculate the dropped disparity levels, we present an effective method to regenerate the costs. In addition, the performance and resource usage of the proposed method is compared with that of conventional methods.

Keywords: Stereo matching, FPGA implementation, reducing disparity level

1 Introduction

A high resolution stereo matching algorithm is not easy to implement in real time because of the amount of calculation required (Jeong et al., 2013). Recently, real-time implementations using a variety of hardware have been proposed that use recent hardware technologies (e.g., Field Programmable Gate Arrays(FPGA) and Graphic Processing Units(GPU)) and simple and efficient algorithms (Ding et al., 2011; Mattoccia, 2013; Cuong and Jeon, 2013). The computable distance in stereo matching is determined by specifications such as the focal length of lenses, size of image sensors, and baseline between cameras. As the disparity levels increase, it is possible to estimate a greater range of distances. Moreover, in order to increase the resolution of the depth map in a system with the same specification, it is necessary to calculate large disparity levels using input images with a high resolution. For example, if the highest disparity level is 64 in a system that uses 640x480 VGA images, this disparity level must be 128 to obtain the same depth map in a system that uses 1280x960 images.

However, because memory transfer and computational complexity increase as the disparity levels increase, execution time grows in a system using hardware accelerators such as GPUs. In FPGA, in order to increase the parallelism, it is necessary to design more computational units proportional to the increase in disparity levels. After all, this is the cause of the increase in the consumption of logic and memory. In particular, if the system uses a cost aggregation system based on the adaptive weight kernel that is commonly used in recent local optimization methods, it requires a saving of cost volume, although there are differences in size according to the algorithm used. The adaptive support weight (ASW) method requires only the number of lines corresponding to the size of the window to be saved to memory. On the other hand, for information permeability filtering (PF), the whole cost volume should be stored. In the case of modified PF (MPF) (Chang et al., 2014), the version that has been simplified for FPGAs needs memory for saving the cost volume of two lines. For these reasons, many studies use small disparity levels or a simplified version of the original algorithm. This also leads to a decrease in performance. Alternatively, some researchers use powerful hardware (i.e., large, costly FPGAs) to solve this problem while maintaining performance (Chang et al., 2014). Therefore, as disparity levels reduce, it is possible to implement additional algorithms and use affordable hardware.

This paper is structured as follows. In Section 2, we introduce the stereo matching algorithm that is used to evaluate the performance of our approach. In Section 3, we introduce our main idea to reduce disparity levels and propose how to address the errors caused by skipped disparity levels. In Chapter 4, the proposed method is analyzed using the Middlebury images and compared with existing fully implemented algorithms. The results show that the proposed method effectively reduces the logic and memory required in the FPGA.

2 Stereo matching algorithm

In this section, we introduce an appropriate stereo matching algorithm for evaluating methods that use sparse disparity levels. According to Scharstein's taxonomy, stereo matching algorithms are distinguished by global and local matching and their disparity computation method (Scharstein et al., 2001). Many implementations of global matching were proposed in order to obtain high performance before 2006 (Hosni et al., 2013). However, since Yoon and Kwon's ASW method was proposed (Yoon and Kwon, 2006), most proposals in recent years have used local matching. Recently, even when using a GPU or FPGA for real-time processing, stereo matching systems implement a local matching algorithm based on a simplified ASW because of its computational complexity(Ding et al., 2011). Fig. 1. shows an overall block diagram of the proposed stereo matching algorithm.

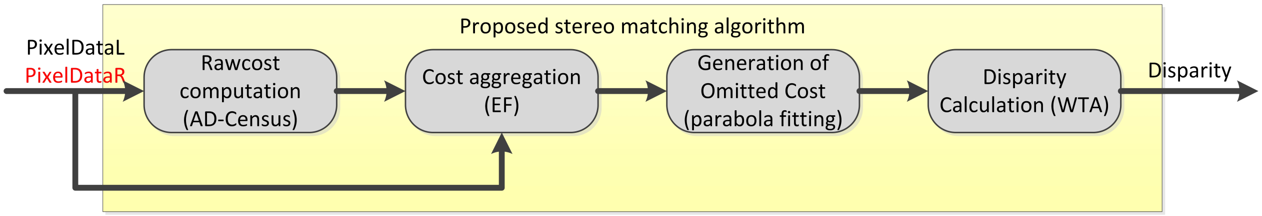

Proposed stereo matching algorithm
PixelDataL
PixelDataR
Rawcost computation (AD-Census)
Cost aggregation (EF)
Generation of Omitted Cost (parabola fitting)
Disparity Calculation (WTA)
Disparity

Fig. 1: Overall block diagram of the proposed stereo matching algorithm

2.1 Cost computation

The matching cost computation is the initial cost computation of the stereo matching algorithm. In this paper, we calculate the raw cost volume using the absolute difference (AD)-Census. The reason for combining the AD and Census Transform (CT) cost measures is that AD-Census provides better matching accuracy than either the AD or CT measures individually (Zabih and Woodfill, 1994). In addition, this combined method is robust in actual environments because of using both parametric and non-parametric methods. In this study, we use cost combining with alpha-blending for the AD-Census.

After Yoon et al. used the ASW approach for cost aggregation, many studies have used a similar cost aggregation method. Cigla and Alatan (2011) proposed PF as an approach to ASW. PF has simple parameters and calculates adaptive-weighted aggregation of cost values in constant operational time. However, because there is no proximity weight term, PF can encounter problems with images that contain large untextured regions. Hence, MPF, which includes a proximity weight term, was proposed by Chang et al. (2014).

2.2 Disparity selection and sub-pixel estimation

Disparity computation involves the calculation of disparity that has been properly matched with the results of the raw cost calculation. Typically, a window with a support weight uses the winner-take-all (WTA) method to select the disparity in minimum aggregated cost when calculating cost volume. Our proposed system uses WTA as the disparity computation because it is a very simple algorithm.

In this paper, sub-pixel estimation is done via local minimum finding using quadratic fitting. This is a classic approach to estimating sub-pixel disparity, although applying it directly to disparities can lead to severe biases (pixel-locking). However, quadratic fitting is simple to compute. In addition, it is very easy to apply to the point unit results in the FPGA.

3 FPGA reduction

This section describes a method for reducing FPGA resources (e.g., logic and memory) for the algorithm described in Section 2. In general, FPGA resource usage is affected by the disparity levels. For this reason, a method to reduce the disparity levels is described. In addition, we discuss how to estimate the omitted disparity levels.

3.1 Disparity and depth relation

As shown in Fig. 2., disparity values are inversely proportional to depth. Hence, when a disparity level increases by one, the depth increases at a different rate that depends on the disparity value. For example, an increment of one disparity level at a disparity of 64 is equal to about 5.5 cm in depth in a stereo matching system (specifications baseline = 15 cm, focal length = 8 mm and pixel pitch of CMOS = 5.3 um). On the other hand, a change of one disparity level at a disparity of 123 is equal to about 1.5 cm in depth in the same system.

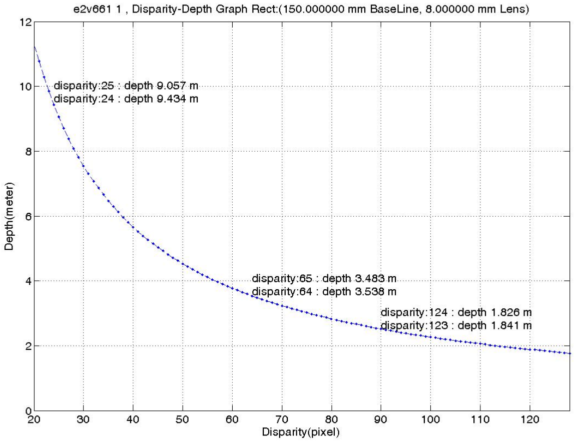
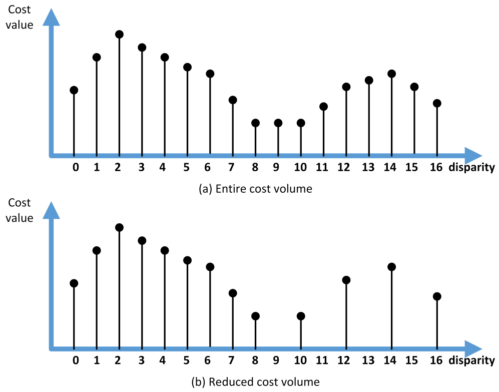


Fig. 2: Relationship between disparity and depth


Cost value
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 disparity
Cost value
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 disparity

(a) Entire cost volume

(b) Reduced cost volume

Fig. 3: Conventional and proposed disparity spaces

Generally, for a required measurement range and accuracy, a stereo system will be designed with specifications that satisfy these requirements. However, shorter distances will exceed the required accuracy and the accuracy of the system reduces for larger distances. Given this phenomenon, we calculate stereo at different intervals of disparity levels as they increase instead of calculating all possible disparity levels. In addition, we generate the other disparity levels by parabola fitting or edge-aware filtering that is used for subpixel estimation. Such methods use fewer FPGA resources than the calculation of all disparity levels.

3.2 Disparity reduction and excluded cost estimation

Fig. 3 illustrates the main idea of this paper. Given a stereo matching system with a disparity range of 0-63, the system computes the cost for all disparities in the 0-31 range. In addition, it computes the cost for every other disparity in the 32-47 range. Finally, it computes the cost for every fourth disparity in the 48-63 range.

There are two methods to reduce the disparity levels. One is to reduce the levels only for the rawcost computation and not the cost aggregation. In this method, the system needs to interpolate the cost of the omitted disparities. In addition, this does not significantly reduce the memory required for cost aggregation because only the rawcost calculation block is reduced. This method obtains similar performance results to the calculation of the entire disparity. The other method is to reduce the levels for the entire cost computation including cost aggregation. In this case, the memory and logic required are greatly reduced. The method can then perform at higher resolutions using an improved timing margin and it is possible to implement additional algorithms. However, computation is required to estimate the cost value of the omitted disparity levels. Depending on these results, the performance of the final disparity method is determined.

Originally, quadratic fitting subpixel estimation was used to determine the disparity of the decimal point unit. When the disparity has been adequately sampled, this is a useful method for estimating the analytic minimum from the sampled disparity space. We use three neighboring cost values to locate the local minimum point via parabola fitting. For example, if the disparity interval is 2, we estimate the disparity with the lowest cost value in the disparity range [d-2, d+2] using the cost values of each disparity: d-2, d, and d+2.

4 Performance comparison and Resource usage

In this section, by applying the algorithm of Section 2 to the Middlebury Evaluation v2 image dataset, we compare the results of reducing the disparity range with those of the original disparity range. We evaluated the quantitative performance for the Teddy and Cones images, which have a higher resolution than the other images in the Middlebury dataset. Figure 4 shows the disparity results calculated by the proposed and conventional methods. Pixels indicating the difference between the conventional and proposed methods demonstrate the errors on regions of significant change in disparity.

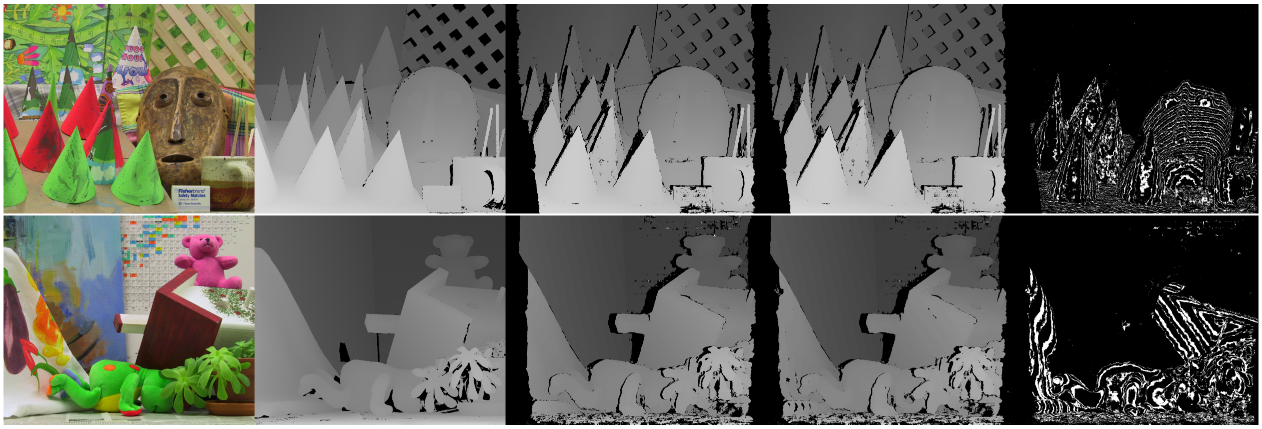


Fig. 4: Comparison of disparity results for entire cost and reducing cost volumes: (left to right) original image, groundtruth image, entire cost volume computation, reduced cost volume computation, and difference between the results

Table 1 shows the resource utilization in a Virtex7 2000T. The parameters of the entire cost computation are: maximum census size = 11x11, maximum disparity = 256, and image resolution = 1280x720. The parameters of the reduced cost computation are all the same except for the disparity range = [0-127, and even numbers between 128 to 255]. The results confirm that the proposed method significantly reduces the resource usage compared to that of the conventional method.

Table 1: Resource utillization in Virtex7 2000T

Resource	Entire cost computation	Reducing cost computation	Total resource Available
Slice LUTs	431938 (35%)	298336 (21%)	1221600
Slice Registers	755406 (31%)	518214 (24%)	2443200
Memory	679 (53%)	529 (40%)	1292
DSP	768 (36%)	576 (27%)	2160

5 Conclusion and Future work

In this paper, we presented a spare cost computation method to implement a stereo matching system on an FPGA by dropping disparity levels. In addition, using subpixel estimation and filtering to calculate the dropped disparity levels, we presented an effective method to regenerate the cost. However, because of pixel-locking and local minimum methods, the proposed algorithm has a significant number of errors. In future, the ordering and resolution (from integer to point number) to calculate the subpixel will be investigated. After this, we will investigate using edge-preserving filtering to eliminate errors. By changing the operation order and resolution, we expect to better estimate the omitted cost values. After this, we plan to perform edge-preserving filtering to eliminate errors and noise.

Acknowledgments

This work was supported by the ETRI R&D Program (15ZC1400, The Development of a Realistic Surgery Rehearsal System based on Patient Specific Surgical Planning) funded by the Government of Korea.

References

Chang, J., Jeong, J. & Hwang, D. (2014), Real-time hybrid stereo vision system for hd resolution disparity map, *in* 'Proceedings of the British Machine Vision Conference', BMVA Press.

Cigla, C. & Alatan, A. (2011), 'Efficient edge-preserving stereo matching', *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on* pp. 696–699.

Ding, J., Liu, J., Zhou, W., Yu, H., Wang, Y. & Gong, X. (2011), 'Real-time stereo vision system using adaptive weight cost aggregation approach', *EURASIP Journal on Image and Video Processing* **2011**(1), 1–19.

Hosni, A., Bleyer, M. & Gelautz., M. (2013), 'Secrets of adaptive support weight techniques for local stereo matching', *Computer Vision and Image Understanding* **117**(6), 620–632.

Jeong, J., Shin, H., Chang, J., Lim, E., Choi, S., Yoon, K. & Cho, J. (2013), 'High-quality stereo depth map generation using infrared pattern projection', *ETRI Journal* **35**(6), 1011–1020.

Mattoccia, S. (2013), Stereo vision algorithms for fpgas, *in* 'Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on', pp. 636–641.

Pham, C. C. & Jeon, J. W. (2013), 'Domain transformation-based efficient cost aggregation for local stereo matching', *Circuits and Systems for Video Technology, IEEE Transactions on* **23**(7), 1119–1130.

Scharstein, D., Szeliski, R. & Zabih, R. (2001), A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *in* 'Stereo and Multi-Baseline Vision, 2001. (SMBV 2001). Proceedings. IEEE Workshop on', pp. 131–140.

Yoon, K.-J. & Kweon, I.-S. (2006), 'Adaptive support-weight approach for correspondence search', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **28**(4), 650–656.

Zabih, R. & Woodfill, J. (1994), Non-parametric local transforms for computing visual correspondence, *in* J.-O. Eklundh, ed., 'Computer Vision ECCV '94', Vol. 801 of *Lecture Notes in Computer Science*, pp. 151–158.