Real-time Disparity Map Reconstruction with On-board FPGA by Semi-global Matching and Weighted Least Square Filtering

Pengfei Wang[1],Zhi Gao[1,*],Hailong Qin[2], Myo Tun Aung[1],
Xudong Cheng[3], Mingjie Lao[1], Feng Lin[1], Swee Huat Rodney Teo[1]
1. Temasek Laboratories, National University of Singapore
2. Roadstar.ai
3. Carnegie Mellon University

Abstract

In this paper, we propose a real-time disparity map estimation framework on FPGA, combined with an effective post-processing method. Given the input stream of stereo image pairs, a semi-global matching based framework is implemented on the FPGA to estimate the disparity in real-time. The generated disparity map is refined with a weighted least square (WLS) filtering method. The experiments show that the disparity map can be reconstructed in real-time. In addition, the weighted least square filtering based post-processing can significantly improve the accuracy of the disparity map and remove large estimation errors.

1 Introduction

Stereo Vision, resulting in the knowledge of depth information in a scene, is of great importance in the field of robotics vision. Generally speaking, if the two images are rectified, matching pixels reside on corresponding horizontal lines. For each pixel $p_l(x, y)$ in the left image, its possible matching pixel is $p_r(x-d, y)$, where d is the disparity for the matched pixels. The searching range of disparity is [0, d_{max}], where d_{max} is the maximum disparity, as shown in Fig. 1. The disparity map is composed by the disparity value. Once we get the disparity map, the depth of scenery can be calculated by a simple equation $z = fb/d$, where b is the baseline, f is the focal length. The algorithms for disparity estimation can be generally classified into three categories according to the matching cost, local method, global method and semi-global method. Block matching is one of the typical kind of local method, the matching cost is a summation of the difference within a small neighbourhood of each pixel in the left and right image. Local methods are often sensitive to noise. Global methods treat disparity estimation as a multi-label problem and construct a 2D graph optimized by algorithms of graph cut [1] or belief propagation [2, 3, 4]. Semi-global methods [5] are also used to approximate the NP-hard 2D graph as independent scan-lines and leverages dynamic programming to aggregate the matching cost. Global algorithms typically do not perform an aggregation step, but rather seek a disparity assignment (step 3) that minimizes a global cost function that combines data (step 1) and smoothness terms. While the 2D-optimization can be shown to be NP-hard for common classes of smoothness functions [6], dynamic programming can find the global minimum for independent scan-lines in polynomial time. Semi-global method is our choice for disparity estimation as it has both high computation efficiency and accuracy, compared to local methods and global methods.

Nowadays, there exists many algorithms to build accurate correspondence between a pair of images. Besides, dedicated hardware platforms such as FPGAs and GPUs should be utilized to realize real-time processing through speeding up stereo vision systems. Some researches focus on stereo matching algorithms which use local methods or semi-global methods accelerated by GPUs [7]. [5] reached 12 fps at 450 $\times$ 375 resolution with 64 disparities and [8] achieved 8 fps for 320 $\times$ 240 pixel images with 64 disparity levels. FPGA costs less power and its memory hierarchies as well as processing units could be configured according to user needs [9], thus FPGA outperforms GPU to some extents. [10] achieved 60 fps at 1024 $\times$ 768 pixel stereo images by merging cross-based cost aggregation and mini-census transform. In [11], mini-census adaptive support region stereo matching algorithm was applied and the experiments in this paper shown that 47.6 fps for 1920 $\times$ 1080 with a disparity range of 256 and 129 fps for 1024 $\times$ 768 with 128 disparity could be attained respectively. [9] combined cost aggregation and fast locally consistent dense stereo methods. By combining the two aforementioned methods and testing on Xilinx Virtex-6 FPGAs, it reached 507.9 fps for 640 $\times$ 480 pixel images. Although this paper got low error rate with high frequency, it could not obtain good results with high-definition image. 1600 $\times$ 1200 pixel images with 128 disparity levels at 42 fps was achieved in [12], whose test results were evaluated on Altera Stratix-V FPGAs.

*Email address(es): contact_gaozhinus@gmail.com

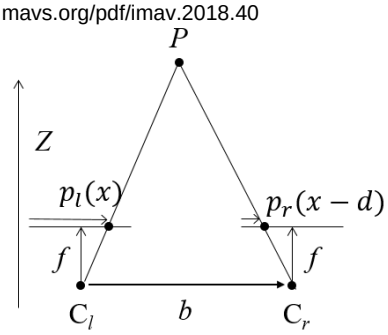


Figure 1: Disparity

2 Proposed Method

Considering the balance between accuracy and efficiency, we take semi-global method [5] as our framework for disparity estimation. The problem of disparity estimation can be modelled as an energy minimization problem, as shown in Equation 1.

$$E(D) = \sum_{p}(C(\mathbf{p}, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1]) \quad (1)$$

where, N_p is the neighbour of p, and P_1 and P_2 is the penalty for disparity variation for neighbouring pixels. The energy consists of pixel-wise matching cost, smoothness cost, and edge-preserving cost. This is an NP-hard problem, and it can be approximated by minimization along 4 individual scan-lines with dynamic programming. We also add a WLS filter as the post-processing to improve the performance after the disparity map is estimated. The whole algorithm consists of the following steps:

- Median filtering is applied to filter the speckle noise such as salt and pepper noise.
- Matching cost for each pixel is calculated by census transform and hamming distance.
- The cost of the pixels are aggregated along scan-lines with dynamic programming.
- For each pixel, the disparity with the smallest matching cost is selected with a winner-take-all strategy.
- Weighted least square filter based post-processing is applied to improve the result of disparity matching.

All the steps except the post-processing are implemented on FPGA.

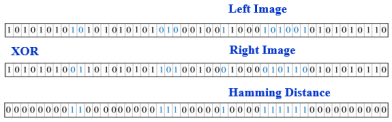


Figure 2: Hamming distance

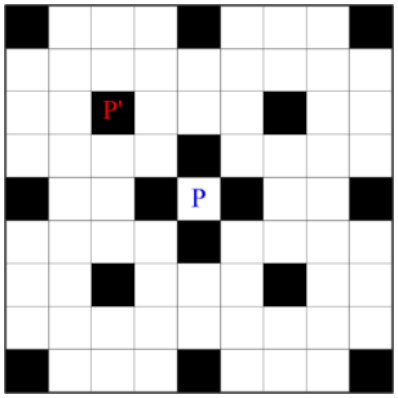


Figure 3: Sparse census transform

2.1 Algorithm for Disparity Estimation

The input left and right images are rectified and fed into a 3×3 median filter. Census transform is applied on both the left and right images, as in Equation 2. The matching cost between $p_l(x, y)$ and $p_r(x-d, y)$ is determined as the hamming distance (Fig. 2) between the two binary vectors obtained by census transform, as in Equation 3. In order to enlarge the size of the neighbour without obviously increasing computation time, we implement a sparse census transform pattern with a window of size 9×9, in which only 16 pixels are counted in the census transform (Fig. 3). The matching cost is aggregated as Equation 4. After cost aggregation, the disparity with the minimal cost is selected for each pixel.

$$\xi(p, p') = \left\{ \begin{matrix} 1, if I(p') < I(p) \\ 0, otherwise \end{matrix} \right\} \quad (2)$$

$$C(p, d_p) = \bigotimes_{p' \in N(p)} \xi(p, p') \quad (3)$$

$$\begin{aligned} L_r(p, d) = & C(\mathbf{p}, d) + min(L_r(\mathbf{p} - \mathbf{r}, d), \\ & L_r(\mathbf{p} - \mathbf{r}, d-1) + P_1, \\ & L_r(p - r, d+1) + P_1, \\ & min_i L_r(\mathbf{p} - \mathbf{r}, i) + P_2) \\ & - min_k L_r(\mathbf{p} - \mathbf{r}, k) \end{aligned} \quad (4)$$

2.2 FPGA Implementation for Disaprity Estimation

The algorithm of the disparity estimation was implemented on a Xilinx ZYNQ XC7Z045 FPGA. The window size of median filter is 3×3. The window size for the census transform is 9×9. Pixels slide through 9×9 shift register

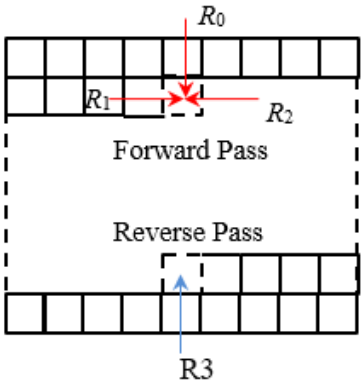


Figure 4: Forward and reverse path of cost aggregation

representing the neighborhood and loaded into census computation block. The Hamming Distance is calculated through an XOR operation followed by count of ones, the summation of set bits. The census transform of left image is stored in a d_{max} 16-bit shift register and XOR with the census transform of the right image. At each clock cycle, cost for all disparity levels are generated and stored in output BRAM.

For the step of cost aggregation, the 4 directions are split into two paths, forward pass and reverse path, as shown in Fig. 4. The cost calculation for each direction is performed using Equation 4. It concurrently computes the aggregation cost for all disparity levels and three paths. The implementation of the elementary block of the aggregation cost computation is shown in Fig. 5. The block is replicated d_{max} times for each 3 paths, resulting $3 \times d_{max}$ blocks to process concurrently. The bottom multiplexer selects the output between the initial cost for the beginning of a path, and the currently computed value along the path.

After the cost aggregation, the disparity with the minimal cost for each pixel is to be selected. A serial tree structure comparator is applied for disparity selection. The implementation uses $d_{max}/2$ comparators with computation complexity of $O(d_{max})$, as shown in Fig. 6.

Fig. 7 shows the time schedule for each module in the processing of one frame. Data load, census transform, and hamming distance are pipe-lined. The generated cost by hamming distance is buffered in external DDR memory. The forward pass starts after DDR buffering. The reverse pass and depth map generation starts after forward pass completed.

2.3 Post-processing by Weighted Least Square Filtering

The accuracy of the disparity estimation is often suffered from extreme scenario, such as texture-less region, overexposure, repetitive structure, etc. In order to improve the accuracy of the disparity, post-processing is always necessary. We take the weighted least square filtering [13] for post-processing, for its good performance of edge preserving smoothing. The objective of filtering the disparity can be

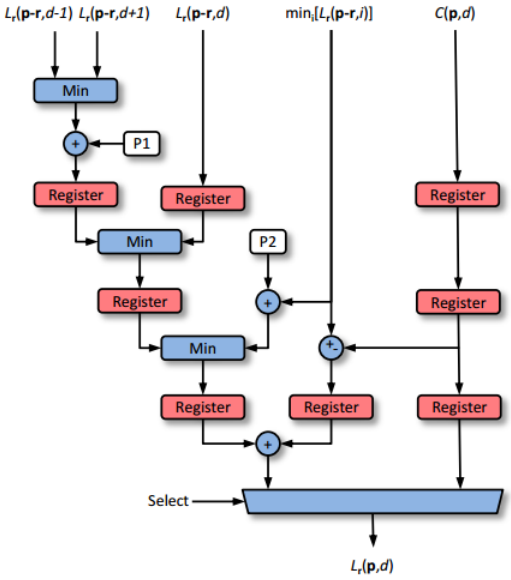


Figure 5: Cost aggregation structure on FPGA

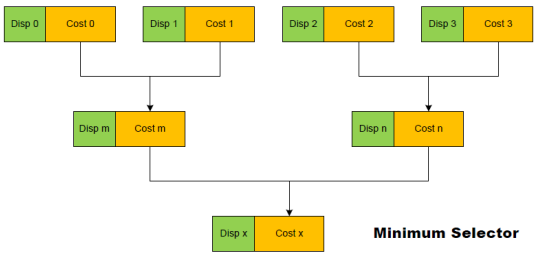


Figure 6: Tree structure comparator for disparity selection

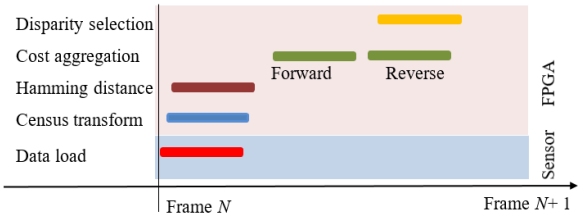


Figure 7: Timing diagram of the system

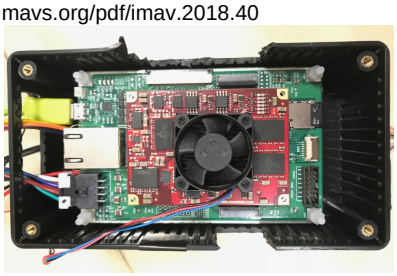


Figure 8: The whole system

expressed as minimizing Equation 5.

$$\sum_p ((D'_p - D_p)^2 + \lambda(a_{x,p}(I)(\frac{\partial D}{\partial x})_p^2 + a_{y,p}(I)(\frac{\partial D}{\partial y})_p^2)) \quad (5)$$

$a_{x,p}(I)$ and $a_{y,p}(I)$ are the smoothness weights as defined in Equation 6 as [14]:

$$\begin{aligned} a_{x,p}(I) &= (\left|\frac{\partial l}{\partial x}(p)\right|^{\alpha} + \epsilon)^{-1} \\ a_{y,p}(I) &= (\left|\frac{\partial l}{\partial y}(p)\right|^{\alpha} + \epsilon)^{-1} \end{aligned} \quad (6)$$

where l is the log-luminance channel of the guidance image I, the parameter α determines the edge sharpness, while ϵ is a small constant, for example, 0.0001.

3 EXPERIMENTS

The disparity map estimation algorithm is implemented on Xilinx ZYNQ ZC7045 FPGA, and the post-processing is implemented on ARM processor. The whole system is shown in Fig. 8.

In order to test the performance of the algorithm, we have carried out experiments on Middlebury dataset and self-collected data. The sample result of the image "Teddy" from the dataset is shown in Fig. 9. The result shows that the WLS filter can improve the accuracy of the estimated disparity.

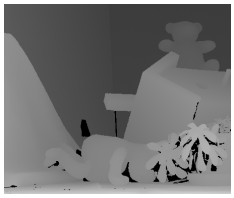
For the self-collected data (Fig. 10), the result of the estimated disparity by semi-global matching suffers stride effect, especially at the right-top corner with overexposure, the disparity is totally wrong as it is a large texture-less region. The weighted least square filter can remove the stride effect, makes the disparity level more clear, and reduce the error caused by the overexposure.

4 CONCLUSION

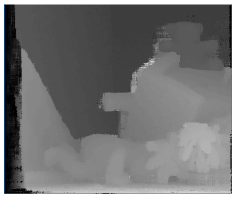
In this paper, we have implemented a Semi-global disparity estimation algorithm on FPGA, and improve the disparity map through weighted least square filtering. In the disparity estimation, the sparse census transform has two folds, large window makes the descriptor more robust to noise, and the computation load increases not too much. The WLS filtering based can improve the accuracy of the estimated disparity

(a) Left image

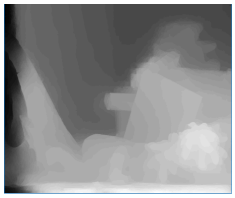
(b) Right image



(c) Disparity groundtruth



(d) Disparity by SGM, error rate=8.0%

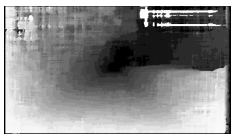


(e) Disparity after WLS, error rate=6.7%

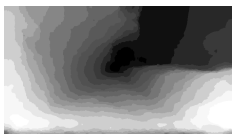
Figure 9: Estimated disparity by SGM and WLS for Middlebury "Teddy"

(a) Left image

(b) Right image



(c) Disparity by SGM



(d) Disparity after WLS

Figure 10: Estimated disparity by SGM and WLS for self-collected data

map, and reduces some defects, such as error caused by over-exposure.

REFERENCES

[1] Michael Bleyer and Margrit Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Signal Processing: Image Communication*, 22(2):127–143, 2007.

[2] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54, 2006.

[3] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, 25(7):787–800, 2003.

[4] Qingxiong Yang, Liang Wang, and Narendra Ahuja. A constant-space belief propagation algorithm for stereo matching. In *Computer vision and pattern recognition (CVPR), 2010 IEEE Conference on*, pages 1458–1465. IEEE, 2010.

[5] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008.

[6] Olga Veksler and Ramin Zabih. Efficient graph-based energy minimization methods in computer vision. 1999.

[7] Ruigang Yang, Greg Welch, and Gary Bishop. Real-time consensus-based scene reconstruction using commodity graphics hardware. In *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on*, pages 225–234. IEEE, 2002.

[8] Ilya D Rosenberg, Philip L Davidson, Casey MR Muller, and Jefferson Y Han. Real-time stereo vision using semi-global matching on programmable graphics hardware. In *ACM SIGGRAPH 2006 Sketches*, page 89. ACM, 2006.

[9] Minxi Jin and Tsutomu Maruyama. Fast and accurate stereo vision system on fpga. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 7(1):3, 2014.

[10] Lu Zhang, Ke Zhang, Tian Sheuan Chang, Gauthier Lafruit, Georgi Krasimirov Kuzmanov, and Diederik Verkest. Real-time high-definition stereo matching on fpga. In *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*, pages 55–64. ACM, 2011.

[11] Yi Shan, Yuchen Hao, Wenqiang Wang, Yu Wang, Xu Chen, Huazhong Yang, and Wayne Luk. Hardware acceleration for an accurate stereo vision system using mini-census adaptive support region. *ACM Transactions on Embedded Computing Systems (TECS)*, 13(4s):132, 2014.

[12] Wenqiang Wang, Jing Yan, Ningyi Xu, Yu Wang, and Feng-Hsiung Hsu. Real-time high-quality stereo vision system in fpga. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(10):1696–1708, 2015.

[13] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM Transactions on Graphics (TOG)*, volume 27, page 67. ACM, 2008.

[14] Dani Lischinski, Zeev Farbman, Matt Uyttendaele, and Richard Szeliski. Interactive local adjustment of tonal values. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 646–653. ACM, 2006.