FPGA based Memory Efficient High Resolution Stereo Vision System for Video Tolling

Yi Shan [#1], Zilong Wang [#], Wenqiang Wang [#], Yuchen Hao [#]

Yu Wang [#], Kuenhung Tsoi [*], Wayne Luk [*], Huazhong Yang [#]

[#] *Tsinghua National Laboratory for Information Science and Technology*

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

[1] shany08@mails.tsinghua.edu.cn

[*] *Computing Department, Imperial College London*

***Abstract*—This paper presents an FPGA based stereo vision system for future video tolling, which can achieve real-time processing for high resolution video streams. The key component for the system is SAD (Sum of Absolute Differences) based stereo matching. Although simple and effective, this method usually needs much computation power to satisfy real-time requirement. We propose a Hybrid-D Box-Filtering algorithm in hardware to explore disparity-level and row-level parallelism for SAD computation. This method enables processing of high resolution images with limited on-chip memory resources. The experimental results show that the system can process 46 fps (frames per second) for video of 1280*1024 resolution with a large disparity range of 256, and 400 fps for a video of 640*480 resolution with a disparity range of 128. Our results are up to 3 times better than previous work in the metric of points times disparity per second (PDS).**

I. INTRODUCTION

Video tolling, a technique for toll collection using video or still images to identify and classify vehicles for payment, has potential to provide more powerful solutions for charging vehicles with lower cost. However, current video tolling system can only recognize the number plate and then charge based on the number plate. This is often inadequate for a vehicle tolling system and even wrong when there is a trailer or the plate number is changed on purpose. Stereo vision systems can provide a supplementary accuracy in these circumstances which can automatically extract 3D shape information.

Due to the high frame rate, high resolution and large disparity range of the video for future tolling systems, a high throughput platform should be used to achieve real-time processing requirement. At the same time, the large deployment and frequent updating cost should be carefully considered. We decide to develop and implement this system on FPGAs (Field Programmable Gate Arrays), which can meet these two conflicting requirements. However, the on-chip computation resource of FPGAs is insufficient for a complete video tolling system with high resolution requirement. How to optimize the most resource-consuming module of the system is an emerging problem.

978-1-4673-2845-6/12/$31.00 ©2012 IEEE

There is work on accelerating stereo vision algorithms [1] on FPGAs. However, little work covers high-resolution video while meeting real-time processing requirement due to on-chip memory or off-chip memory bandwidth limitation. This paper proposes a fully pipelined FPGA based stereo vision system for high resolution video and uses a novel stereo matching method to reduce the on-chip memory cost. Our main contributions are:

(1) a real-time stereo vision system for video tolling based on FPGA technology capable of stereo matching and size extraction. Experiments show that it can achieve 46 fps for video of 1280*1024 resolution with a large disparity range of 256;

(2) a modified Hybrid-D Box-Filtering stereo matching algorithm enabling high resolution images computation with limited on-chip memory resources.

II. RELATED WORK

Stereo vision aims to acquire 3D information and there are many algorithms for establishing accurate correspondence between images. Masanori et al. [2] has proposed a SAD computation based on recursive computation and achieved 200 fps at 640*480 resolution with a disparity of 64. Jin et al. [3] proposed a pipelined Census transform and achieved 230 fps at 640*480 with a disparity of 64. Zhang et al. [4] used Mini-Census transform and the Cross-based cost aggregation to achieved 60 fps at 1024*768 with a disparity of 64. Some work[5, 6] tried to improve both the accuracy and the speed and achieved 32 fps and 30 fps for 640*480 respectively.

The resolution and the disparity range of the previous work are small and the amount of on-chip block memory and logic resources limits its potential for higher resolution. Furthermore, designs reported in previous work often use up all FPGA resources for the stereo matching module alone. However nowadays more modules are expected to be integrated in an embedded system and they compete for chip resources. So it is challenging to extend the maximum resolution and disparity for advanced vision systems. This paper focuses on optimizations for the stereo matching module to reduce resource utilization.

III. STEREO MATCHING

The main purpose of stereo matching is to establish reliable correspondence between stereo images and then give depth information of the images. The SAD-based method is commonly used due to its regularity. In this section, traditionnal 1D and 2D Box Filtering methods for SAD-based method are introduced and the disadvantages of the two methods, large logic and memory requirements, are discussed. To solve these problems, a Hybrid-D Box Filtering algorithm is proposed and the key paramters are carefully analyzed to achieve high performance.

A. Traditional Box Filtering Algorithm

We measure the similarity of two pixels, centric in the window with size of *2n+1*, between stereo images, left image and right image, with the disparity of *d* by calculating the *SAD (sum of absolute difference) Cost*:

$$C(x,y,d) = \sum_{i=-n}^{n}\sum_{j=-n}^{n}|L(x+i,y+j) - R(x+d+i,y+j)| \quad (1)$$

L(x,y) and *R(x,y)* are the luminance of the pixel *(x,y)* in left image and right image respectively. The pixel pair with the minimum SAD Cost among different disparities is the matched one and this disparity can be used to construct the depth map. The computation for different disparities has no data dependency and can be executed in parallel, and this is called the **disparity-level parallelism**. For the same disparity, the neighbor costs, *C(x, y, d)* and *C(x, y+1, d)*, have the common computations, so the Box-Filtering algorithm [7] can be used to reduce the computation cost based on proper data reuse.

$$C(x+1,y,d) = C(x,y,d) + U(x+1,y,d) \quad (2)$$

$$U(x+1,y,d) = \sum_{i=-n}^{n}|L(x+1+n,y+i) - R(x+1+d+n,y+i)| - \sum_{i=-n}^{n}|L(x-n,y+i) - R(x+d-n,y+i)|$$

For the 1D Box-Filtering algorithm as in Equation (2), vector absolute subtraction and adder tree logic is needed to compute $U(x+1,y,d)$ of the support window. If we wish to utilize disparity-level parallelism, then *P_dis* copies of this kind of vector computation logic are needed.

For the 2D Box-Filtering algorithm, the *Cost* computation is based on the *Costs* of both the left and the top pixel. We can get the equation:

$$C(x+1,y,d) = C(x,y,d) + U(x+1,y-1,d) + \Delta A + \Delta C - \Delta B - \Delta D \quad (3)$$

ΔA, ΔB, ΔC and ΔD are the corresponding differences of the pixels from the left and the right images. In the 2D algorithm which is also used in [2], the logic is reduced because only 8 numbers' addition/subtraction is needed. However one row of *U(:, y-1, d)* should be recorded for each disparity. To achieve high frame rate processing, the memory cost is *IMG_width*U_bitwidth*Disparity_num*, and *U_bitwidth* is the bitwidth of $U(x,y,d)$.

Considering our application, for a video with a high resolution of 1280*1024 with a disparity of 256 and a window size of 25*25, the memory cost for the 2D algorithm will be nearly 5 Mb on-chip block RAM. This is unaffordable when there are many other modules on the FPGA in current and near-future technologies.

B. Hybrid-D Box Filtering Algorithm

To maintain parallelism while reducing logic and memory costs, we propose a hybrid parallel algorithm, Hybrid-D, which combines the 1D and 2D algorithms. **In this method, 1D method is used to compute one row of *U(x+1, y, d)*. Then the neighbor rows, such as *U(x+1, y+1, d)* and *U(x+1, y+2, d)*, will be computed based on *U(x+1, y, d)* according to the 2D algorithm using much less computation logic than 1D one.** One group of these rows is called a row batch and the image is partitioned into many row batches. The *U* value of the first row in a row batch is computed by 1D algorithm and the other rows in this row batch are computed by 2D algorithm.

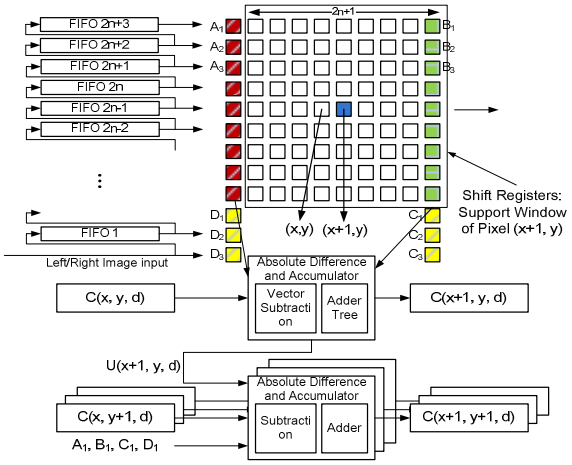

Figure 1. Hybrid-D algorithm for SAD

As shown in Figure 1, when $U(x+1,y,d)$ is calculated, *C(x+1, y, d)*, *C(x+1, y+1, d)* and *C(x+1, y+2, d)* can be produced by:

$$C(x+1,y,d) = C(x,y,d) + U(x+1,y,d)$$

$$C(x+1,y+1,d) = C(x,y+1,d) + U(x+1,y,d) + \Delta A_1 + \Delta C_1 - \Delta B_1 - \Delta D_1$$

$$C(x+1,y+2,d) = C(x,y+2,d) + U(x+1,y,d) + \Delta A_1 + \Delta A_2 + \Delta C_1 + \Delta C_2 - \Delta B_1 - \Delta B_2 - \Delta D_1 - \Delta D_2$$

Using the hybrid algorithm, *U* values are not needed to be recorded, so memory cost will be largely reduced compared with the 2D algorithm. At the same time, with the same logic cost as the 1D algorithm, results of several rows can be achieved at the same time, which is call the **row-level parallelism** (*P_row*). To achieve the same parallel degree, only *1/P_row* of original disparity-level parallelism is needed with the help of row-level parallelism. So the computation logic is *P_row* times reduced with the same parallel degree as the 1D algorithm.

The challenge is to find a proper degree of row-level parallelism to reduce computation logic with limited memory resources. The synthesis results of resource utilization are shown in Figure 2. Our scenario deals with 1280*1024 resolution with the disparity of 256 with a window size of 25*25. There are 128 degrees of parallelism in our design due to the resource limitation, so we need two cycles for stereo matching computation of one pixel.

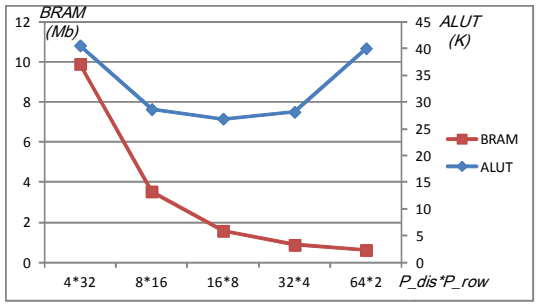

Figure 2. Resource utilization of different parallelism

To achieve 128 degrees of parallelism, many kinds of parallelism configurations are tested to find the most resource efficient one. Figure 2 shows the BRAM and ALUT utilization of designs with different parallelism configurations. When *P_row* increases from 2 to 8, the ALUT utilization decreases due to the reduction of adder logic. However, when *P_row* further increases, the ALUT utilization increases because the logic for ΔA computation and *Cost* comparison will be the main factors of resource utilization. BRAM utilization will increase due to the large amount of buffers in the *Cost* comparison stage. Because of resource availability, 32 degrees of disparity-level parallelism and 4 degrees of row-level parallelism are used in our design.

IV. FPGA IMPLEMENTATION

Based on the proposed Hybrid-D algorithm in Section III, there is parallelism both at row-level and at disparity-level. According to FPGA synthesis results, 32 PEs can be used for disparity-level parallelism with 4 degrees of row-level parallelism.

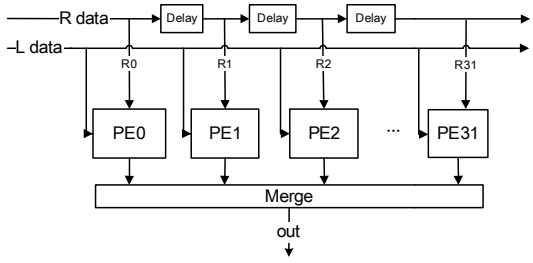

Figure 3. The framework of SAD based stereo matching

Figure 3 shows disparity-level parallelism. There are 32 PEs to compute matching cost of 32 disparities for one pixel and the logic in each PE can produce results of another 3 pixels in neighbor rows by row level computation reuse. The data from the left image is shared by multiple PEs and the data from the right images is shifted to each PE to compute the matching cost of different disparities. Using stream processing through the datapath, the data do not need to be buffered and on-chip memory cost is reduced. After one-row scan, matching costs of 32 disparities for 4 rows are achieved. Therefore, 8 rounds of one-row scan will be needed to compute the 256 disparities for 4 rows. Then a merge stage is used to find the minimum of the matching cost to represent the matched disparity.

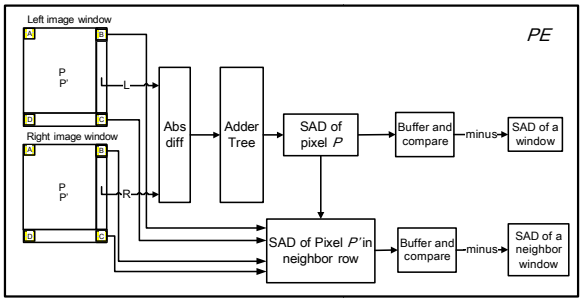

Figure 4. The PE with row-level parallelism

Figure 4 shows the detailed logic of each PE. Firstly, the new input column is subtracted to get the absolute difference vector. Secondly, the elements of the absolute difference vector are accumulated by an adder tree. Based on equation (2), the SAD of one pixel with one disparity is achieved. The SAD of a row can be reused for several neighbor rows. In our implementation, we use the results of one row to compute 3 neighbor rows with little increase in resource consumption. The structure in Figure 4 shows how to compute two neighboring rows in parallel.

V. EXPERIMENTAL RESULTS

The whole system benefits from the architecture design and algorithm tuning in the previous sections and has been successfully implemented on a platform with an Altera Stratix IV EP4SGX110HF35C2 device. In this section, the accuracy and the performance of the system will be shown.

A. Accuracy

We check the accuracy of our stereo matching design by simulation based on a standard benchmark [8]. The results are shown in Figure 5. We check that the hardware has similar results as the software.

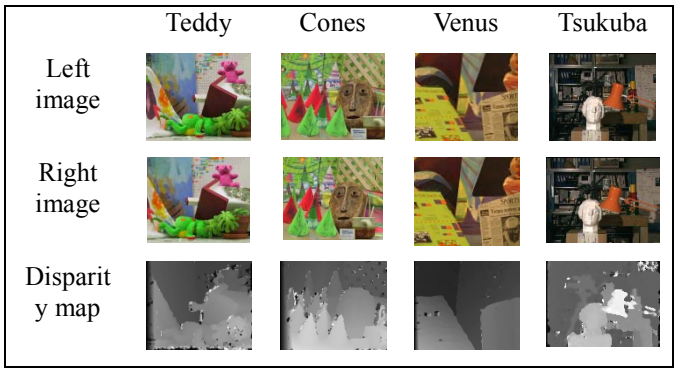

Figure 5. Our stereo matching results

B. Speedup and Resource Utilization

Based on simulation by ModelSim SE 6.5 and synthesis by Quartus II 10.0, our hardware implementation on Altera Stratix IV FPGA can reach 125MHz. Our hardware implementation could achieve about 46 fps for video of 1280*1024 resolution with a large disparity range of 256, and 400 fps for a video of 640*480 resolution with a disparity range of 128. The software results are tested on an i7 930 2.8GHz CPU based on OpenCV library. The results show that our FPGA implementation of the system can be 72.48 times faster than software, and 63.91 times faster for stereo matching.

TABLE 1. SW AND HW RESULTS FOR ONE 1280*1024 IMAGE

Module	Software(ms)	Hardware(ms)	Speedup
Stereo	1380.40	21.60	63.91
Total	1565.50	21.60	72.48

We compare our work with other publications; the results are listed in Table 2. Our approach achieves the highest value of PDS (Points times Disparity per second), the metric of computational capability for this application. Note that the rectification step is included in our design.

TABLE 2. COMPARISON WITH OTHER PUBLICATIONS

(MPPS is mega points per second processing ability and the PDS is MPPS*disparity range)

Work	Rectify	MPPS	PDS/disparity	method
Our 1280*1024	Yes	60.3	15437/256	SAD
Our 640*480	Yes	122.8	15718/128	SAD
Jin et al.[3]	Yes	70.6	4521/64	Census
Tomasi et al.[1]	Yes	17.6	4505/256	Phase-based
Hariyama et al.[2]	No	80	5120/64	SAD
Diaz et al.[9]	No	63.89	1885/29	Phase-based
Li[10]	Yes	9.85	2875/300	Spherical
Zhang et al.[4]	No	47.19	3019/64	Adaptive SAD
Jin et al.[5]	No	9.83	590/60	Dynamic programming
Ttofis et al.[6]	No	5.07	649/128	Adaptive SAD

Our implementation achieves the highest computation capability, and its resource utilization is carefully optimized. For the stereo matching module, only 25K logic elements and 0.95Mbits on-chip memory are used. The proposed method requires 5 times less on-chip memory than the 2-D box filtering method, and it uses less resources to achieve 5 times higher computation capability than previous work [4]. The Hybrid-D algorithm largely reduces the resource occupation of stereo matching module and makes it possible to include the other important modules to the video tolling system.

VI. CONCLUSION

The real-time processing requirements of video tolling systems are challenging. This paper proposes an FPGA based stereo vision system for video tolling which can extract the size of vehicles. Based on careful analysis and algorithm tuning, the processing elements are all fully pipelined. For the critical module in stereo matching, a Hybrid-D Box-Filtering algorithm is proposed to reduce the computation and memory costs. Our design is 72.48 times faster than the software version and achieves higher performance than designs in other publications. The optimization described in this paper can easily be used in other adaptive SAD based stereo matching algorithms, which are also constrained by computational resources. Future work will extend our Hybrid-D method to cross-based stereo matching to achieve more accurate results.

ACKNOWLEDGMENT

This work was supported by the Royal Academy of Engineering, National Science and Technology Major Project (2010ZX01030-001), National Natural Science Foundation of China (No.61076035, 61028006), and Tsinghua University Initiative Scientific Research Program.

REFERENCES

[1] M. Tomasi et al, Real-Time Architecture for a Robust Multi-Scale Stereo Engine on FPGA. IEEE Transactions on Very Large Scale Integration Systems, to appear.

[2] M. Hariyama et al, FPGA Implementation of a High-Speed Stereo Matching Processor Based on Recursive Computation. The International Conference on Engineering of Reconfigurable Systems and Algorithms, 2009, pp. 263-266.

[3] S. Jin et al, FPGA Design and Implementation of a Real-Time Stereo Vision System, IEEE Transactions on Circuits and Systems for Video Technology, Jan. 2010, 15-26

[4] Lu Zhang et al, Real-time High-definition Stereo Matching on FPGA, pp. 55-64, 2011

[5] Minxi Jin, et al, A real-time stereo vision system using a tree-structured dynamic programming on FPGA, FPGA, pp. 21-24, 2012.

[6] C. Ttofis et al, Towards Accurate Hardware Stereo Correspondence, DATE, pp. 703-708, 2012.

[7] M. McDonnel, Box-filtering techniques. Computer Graphics and Image Processing, 17:65-70, 1981.

[8] http://vision.middlebury.edu/stereo/eval/

[9] J. Díaz et al, Real-time System for High-image Resolution Disparity Estimation," IEEE Trans. Image Process., vol. 16, 1, 280-285, 2007.

[10] S. Li, Binocular Spherical Stereo. IEEE Trans. Intell. Transport. Syst., vol. 9, no. 4, pp. 589-600, Dec. 2008.