SoC Oriented Real-time High-quality Stereo Vision System

Yanzhe Li, Kai Huang
Institute of VLSI Design
Zhejiang University
Hangzhou, China
Email: {liyz,huangk}@vlsi.zju.edu.cn

Luc Claesen
Engineering Technology - Electronics-ICT Dept.
Hasselt University
3590 Diepenbeek, Belgium
Email: luc.claesen@uhasselt.be

***Abstract*—Stereo matching is a crucial step to extract depth information from stereo images. However, it is still challenging to achieve good performance in both speed and accuracy for various stereo vision applications. In this paper, a hardware-compatible stereo matching algorithm is proposed and its associated hardware implementation is also presented. The proposed algorithm can produce high-quality disparity maps with the combined use of the mini-census transform, segmentation-based adaptive support weight and effective refinement. Moreover, the proposed architecture is optimized as a fully pipelined and scalable hardware system. Implemented on an Altera Stratix-IV FPGA board, it can achieve 65 frames per second (fps) for 1024 $\times$ 768 stereo images and a 64 pixel disparity range. The proposed architecture is evaluated based on the Middlebury benchmarks and the average error rate is 6.56%. The experimental results indicate that the accuracy is competitive with some state-of-the-art software implementations.**

I. Introduction

Stereo vision is one of the most active research topics in computer vision and it is widely used in many applications. Stereo matching, which is treated as the key role in a stereo vision system, takes a pair of rectified images, estimates the movement of each pixel between two images and displays the associated movement in a disparity map. As a result, stereo matching is a complicated and time-consuming procedure. Considering that many applications often require high performance and real-time processing speed, it is difficult for software implementations of stereo matching algorithms on a CPU to meet these constraints.

In this condition, hardware acceleration of stereo matching algorithms is inevitable and it has been done extensively using DSPs, GPUs and dedicated hardware. However, DSPs are limited by the computational ability and fail to support real-time processing; while GPUs always result in excessive power consumption. In contrary, the dedicated hardware approaches using FPGAs and ASICs can provide a balance between computational power and energy efficiency.

A segmentation-based design with adaptive support weight (ADSW) has been implemented on FPGAs [1]. Their proposed design can achieve 30 fps for 640×480 images using a disparity range of 64 pixels. This design is inspired by the algorithm in [2], which used to be the best local method on the Middlebury benchmarks [3]. However, the performance of the design is restricted by the fixed small window size. In [4], a hardware solution provided high-quality disparity results in ASICs based on the mini-census adaptive support weight (MCADSW) method. But this solution only targets low resolution images for real-time. Its performance drops to 6 fps for 1024×768 images and a 64 pixel disparity range. In [5], an algorithm was proposed to achieve high accuracy based on mini-census and variable-cross methods and a fully pipelined architecture was presented for real-time processing. The design can process 1024×768 images with a disparity range of 64 pixels in 60 fps. A. Akin proposed a hardware-oriented adaptive window size disparity estimation (AWDE) algorithm and its real-time hardware implementation [6]. It can handle 60 fps at a 1024×768 resolution for a 128 pixel disparity range. Although the results in [5] and [6] are outstanding among hardware implementations, the accuracy is not comparable to the state-of-the-art software implementations.

In this paper, the mini-census and segmentation-based ADSW algorithms are combined to achieve a high matching accuracy. Different from many other hardware designs that are lack of refinement, we present a disparity refinement with segmentation information. This refinement step can significantly improve the quality of initial disparity maps. Moreover, a fully pipelined and scalable architecture is implemented based on the proposed algorithm. In order to make a tradeoff between accuracy and speed, some techniques such as simplified weight function and adaptive window size are applied. A prototype of the proposed hardware system is built on an Altera FPGA, which achieves 65 fps for 1024×768 stereo images and a 64 pixel disparity range. The design is evaluated with the Middlebury benchmarks and visual satisfactory results are provided. The experimental results indicate that the accuracy is competitive with the state-of-the-art software implementations.

In the rest of this paper, Section II presents the proposed algorithm. The hardware implementation based on the algorithm is described in Section III. Experimental results are shown in Section IV and the paper is concluded in Section V.

II. Stereo Matching Algorithm

A. Algorithm Overview

Cost calculation, cost aggregation, disparity selection and disparity refinement are four well-defined steps in stereo matching algorithms [7]. Here we utilize the mini-census

978-1-5090-3561-8/16/$31.00

transform in the cost calculation step and the segmentation-based ADSW algorithm in the cost aggregation step. The refinement step consists of three stages: consistency check, disparity voting and invalid disparity inpainting. Finally, disparity maps of both images are finished simultaneously. The proposed algorithm is suggested in Fig. 1.

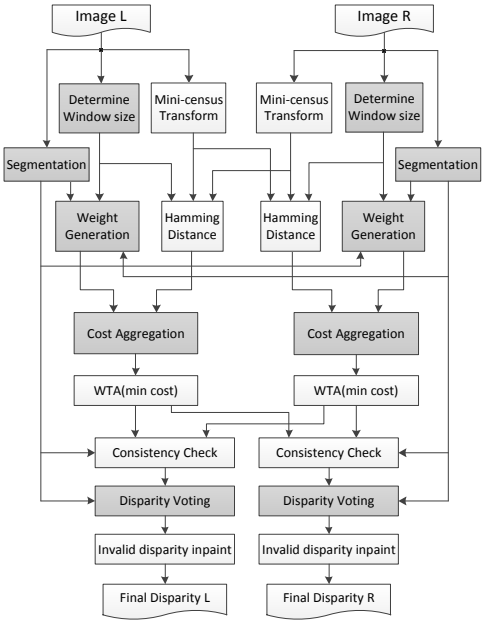

Fig. 1. Overview of the proposed algorithm.

The mini-census transform extracts a 6-pixel neighborhood information of the center pixel and encodes the information in a vector [4]. As a result, each pixel can be represented by only 6 bits. It reduces the memory utilization due to the lower amount of storage bits. Then the matching cost is defined as the Hamming distance among the output vectors.

The segmentation-based ADSW algorithm employs the segmentation information within the weight cost function to increase the robustness of the matching process. The weight cost function generates the weight coefficients w_r and w_t as shown in (1).

$$w_{r,t} = \begin{cases} 1.0 & p_i \in S_c \\ \exp(-\frac{d_c(I_{r,t}(p_i), I_{r,t}(p_c))}{\gamma_c}) & \text{otherwise} \end{cases} \quad (1)$$

S_c is the segment where the central point of p_c (or q_c) lies, d_c is the Euclidean distance between two triplets in the CIELAB color space, and γ_c is a parameter of the algorithm. The final aggregated cost is given by summing up all the weighted matching costs in the support windows W_r and W_t then normalizing with the weights sum as shown in (2).

$$C(p_c, q_c) = \frac{\sum_{p_i \in W_r, q_i \in W_t} w_r(p_i, p_c) w_t(q_i, q_c) MC(p_i, q_i)}{\sum_{p_i \in W_r, q_i \in W_t} w_r(p_i, p_c) w_t(q_i, q_c)} \quad (2)$$

A tree-structure winner-takes-all (WTA) method is used in the disparity selection step. Then disparity refinement operates on the initial disparity maps. The consistency check is to verify whether the disparity maps satisfy (3).

$$d'_p(x - d_p(x,y), y) == d_p(x,y) \cap S_p == S'_p \quad (3)$$

The disparities of pixel p and p' are in the target and reference images respectively. S_p and S'_p are the corresponding segmentation information. If they are different, the consistency check fails and the disparity is marked as invalid. The disparity voting updates the center disparity with the most present valid disparity in its local support window. It removes many invalid disparities, but it will fail if the window does not contain any valid values. So the disparity inpainting replaces the invalid disparity with the closest valid disparity on its scanline to get the final disparity maps. The median filtering is not used in the design for its complicated hardware implementation with negligible quality improvement.

B. Hardware-oriented Optimization

To reduce the computation complexity and improve the hardware compatibility of the algorithm, some optimizations are proposed as shown in the shaded blocks of Fig. 1. YUV color representation is adopted instead of CIELAB color representation so that unsigned integers will be used instead of signed floating-point integers. Moreover, only the luminance channel (Y) is used in the design to reduce the potential bandwidth and storage requirements. Manhattan distance is used rather than Euclidean distance to avoid square and square root computations. These modifications affect the accuracy of the disparity maps slightly. A very simple method which partitions the image into segments using thresholding [1] is adopted instead of the mean shift method. In this way, luminance distance can be largely represented by segment distance.

In weight generation, the weight allows the pixel with luminance similar to the center pixel to have more influence on the final matching cost. A scale-and-truncate approximation of the weight function is proposed for the purpose, and the curve is shown in Fig. 2. So the multiplication of the weight coefficients is reduced to a left shift operation.

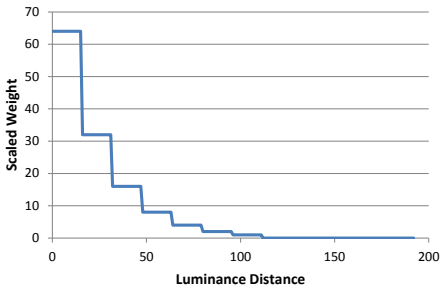

Fig. 2. Weight function.

To make a tradeoff between accuracy and speed, a method called AWDE [6] is introduced. It uses three different window

sizes for different textures on the image and determines the window size by the mean absolute deviation (MAD) of the pixel in the center of a 7×7 block, as expressed in (4).

$$MAD(c) = \frac{\sum_{q \in N_c} |I_t(q) - I_t(c)|}{48} \quad (4)$$

The high MAD value is a sign of high texture content and the low MAD value is a sign of low texture content. 49 pixels are constantly sampled with different intervals so that low computation cost is utilized for large support window sizes.

When disparity voting is implemented, a 25×25 support window is applied to achieve a reliable result. To determine the most frequent disparity value in the window efficiently, a vertical-horizontal approach is adopted. It firstly searches for the majority disparity in each column vertically, then picks out the majority one horizontally as the final disparity. In addition to the computation complexity reduction, the approach also reduces the internal bandwidth. And it shows that this modification provides a slightly different disparity maps compared to the original method.

III. Hardware Implementation

A hardware architecture is designed for the proposed algorithm, as shown in Fig. 3. The key to improve throughput is to design a fully pipelined architecture, which can be decomposed into three stages: pre-processing, stereo-matching and post-processing. The pre-processing stage does not deal with any disparity information and performs pixel-based operations on each pixel for both images independently. The stereo-matching stage operates on the transformed data and computes disparity maps. Here the computing structure proposed in [8] is adopted for cost aggregation, which combines the disparity-level parallelism with the row-level parallelism. In the proposed system, P_{row} neighboring pixels are processed in parallel and P_{dis} disparities are processed in parallel for each pixel. The post-processing stage is implemented to refine the initial disparity maps when they are ready.

For both images, source image pixels are fetched in scanline order and disparity maps are generated through the pipeline. Each pixel is read only once from the external memory during the whole processing flow, so that memory bandwidth is not a limitation. The design can be scaled with image resolution, disparity range, parallelism degree and segmentation level to achieve maximum flexibility.

A. Pre-processing Stage

In the pre-processing stage, 24-bit source pixels of RGB are fetched, hereafter the color space converters are operated. 8-Bit grayscale values are used for the segmentation module. The number of segments k is given as input and a label is computed by a simple method that multiplies the grayscale value by the value of $k/256$. Here k is defined as a power of 2 so that the left shift operation is exploited. 8-Bit Y values are used for the mini-census transform and the window size determination. As the operations are all window-based, a register matrix is employed to provide pipelined window

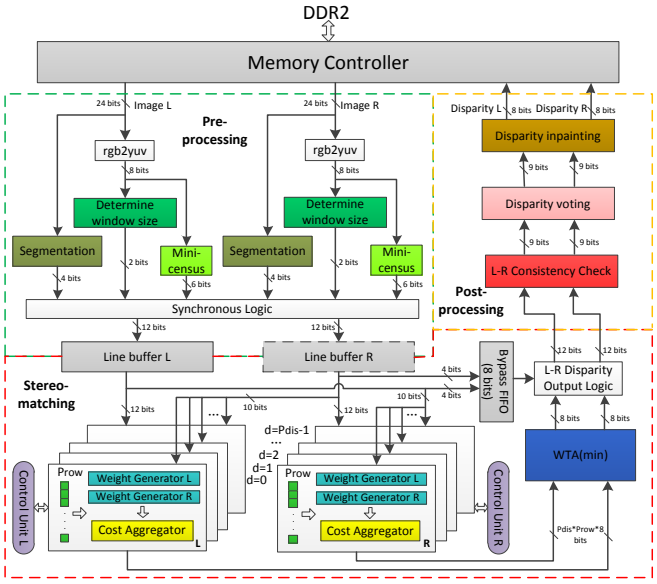

Fig. 3. Block diagram of the proposed architecture.

content in Fig. 4. The whole register matrix of 7×7 is used for window size determination. Meanwhile the six green ones are used for the mini-census transform and the center column of red ones are used for segmentation. Finally, a result of 12 (2+6+4) bits is written into the line buffer.

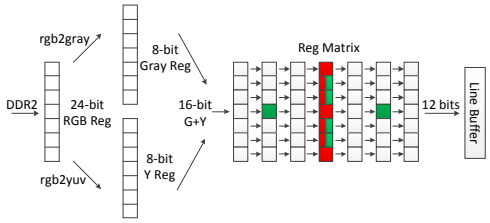

Fig. 4. Pipeline architecture in the pre-processing stage.

To satisfy the row-level parallelism, P_{row} pixels along the column direction are processed in parallel. So the line buffer is composed of $(P_{row} + 6)$ dual port BRAMs to build a wide throughout, and the size of the register matrix is extended to $(P_{row}+6) \times 7$. Source data and temporary results in each column can be reused to reduce the computational requirements because a column is usually a part of multiple horizontally overlapping windows.

B. Stereo-matching Stage

The control unit begins to fetch the transformed data from the line buffers and allocate them to aggregation modules. The line buffers are exploited to replace the processed pixels with the new required pixels during the process. A total of P_{dis} aggregation modules are generated to deal with different disparities. In each module, P_{row} pixels are processed in parallel. Here the generate statement in Verilog is used for these two parameters so that the hardware can be scalable. The weight generator receives segmentation information from both

images depending on the window size of the center pixel and computes weight coefficients. The circuit that generates the weight coefficient of a single pixel is shown in Fig. 5a. The cost aggregator calculates the Hamming distances as matching costs and shifts them with the according weights. The final aggregated cost is computed by summing the weighted scores using a tree adder, then normalizing it. The architecture of the cost aggregator is in Fig. 5b.

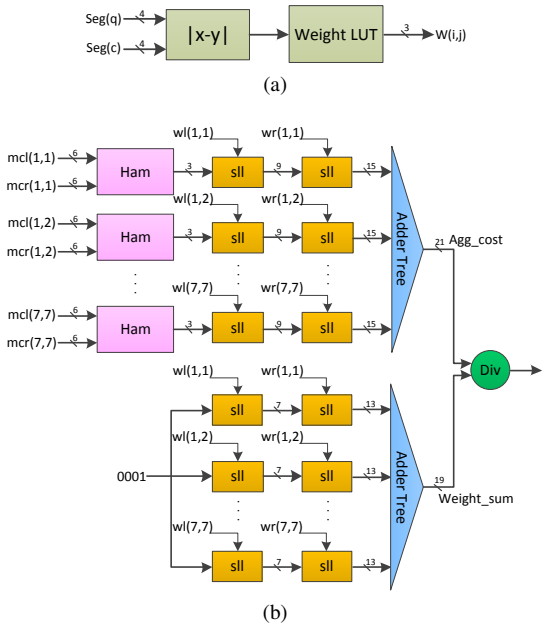

Fig. 5. (a) Circuit that generates weight coefficient, (b) Architecture of the cost aggregator.

Aggregated costs are sent to the tree-structure WTA module to select the disparity with the minimum cost. The Bypass FIFO is used to preserve segmentation information associated with each pixel for the next stage. The output is a data stream that consists of initial disparity maps and segmentation information for each image.

C. Post-processing Stage

In the post-processing stage, the consistency check module compares the initial disparity maps and segmentation information in order to generate one more bit which labels each disparity as valid or not. The disparity voting module updates the center disparity in the 25×25 support window using the vertical-first method. Here, a bitwise fast voting technique [9] is applied to handle each column. It drives each bit of the most frequent disparity independently from the other bits so that the hardware cost depends on the number of disparity bits in binary. While counting bit votes, the valid information must be taken into account. The architecture of bitwise fast voting is shown in Fig. 6. 25 disparities that are the most occurring values in 25 columns are given and the final disparity is picked out horizontally from them.

After disparity voting, disparity maps will be written back into the DDR2 memory. The disparity inpainting module

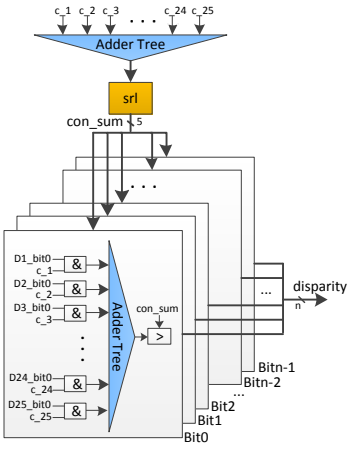

Fig. 6. Architecture of bitwise fast voting.

TABLE I
RESOURCE UTILIZATION REPORT

Altera EP4SGX230	ALUTs Total: 228000	Registers Total: 228000	Memory Bits Total: 17133000
Pre-processing	4170	2286	1015808
Stereo-matching	158496	120292	1638400
Post-processing	18852	9684	28672
Whole System	181518	132262	2682880

checks the valid bit and replaces the invalid disparity with the closest valid disparity when writing it out.

IV. EXPERIMENTAL RESULTS

A prototype of the proposed design was developed on an Altera EP4SGX230 FPGA. The system is evaluated using rectified synthetic stereo images, initially stored in the DDR2 memory. The images are loaded as the input through the Avalon bus. Resource utilization, disparity quality and processing speed are always important criteria when evaluating FPGA-based stereo vision systems.

A. Resource Utilization

Table I gives the overall resource utilization of the FPGA prototype. The hardware system utilizes 80% of the ALUTs, 58% of the registers, and 16% of the memory bits on the FPGA board, and can operate at 120 MHz. The ALUTs and registers are dominated by the cost aggregators and the weight generators in the stereo-matching stage. So the resource utilization is mainly determined by the disparity-level parallelism P_{dis} and the row-level parallelism P_{row}. The two parameters can be scaled to make the system more flexible, just like the image resolution and the disparity range. To make a tradeoff between resource utilization and processing speed, P_{dis} is 8 and P_{row} is 4 in the current system. The memory bits are mostly used as line buffers in the pre-processing and stereo-matching stages to make sure that external memory bandwidth is not a limitation.

TABLE II
ACCURACY COMPARISON ON MIDDLEBURY BENCHAMRK

Data Set	Tsukuba			Venus			Teddy			Cones			Average
Evaluation	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	Error Rate
AD-Census [10]	1.07	1.48	5.73	0.09	0.25	1.15	4.10	6.22	10.9	2.42	7.25	6.95	3.97
Wang et al. [11]	1.93	2.95	7.90	0.61	1.43	2.87	6.44	13.8	16.0	2.37	11.1	6.70	6.17
SegSupport [2]	1.25	1.62	6.68	0.25	0.64	2.59	8.43	14.2	18.2	3.77	9.87	9.77	6.44
MCADSW [4]	-	2.80	-	-	0.64	-	-	13.7	-	-	10.1	-	all=6.81
SemiGlobal [12]	3.26	3.96	12.8	1.00	1.57	11.3	6.02	12.2	16.3	3.06	9.75	8.90	7.50
VariableCross [13]	1.99	2.65	6.77	0.62	0.96	3.20	9.75	15.1	18.2	6.28	12.7	12.9	7.60
MCADSR [14]	3.62	4.15	14.0	0.48	0.87	2.79	7.54	14.7	19.4	3.51	11.1	9.64	7.65
Zhang et al. [5]	3.84	4.34	14.2	1.20	1.68	5.62	7.17	12.6	17.4	5.41	11.0	13.9	8.20
Ttofis et al. [1]	4.48	6.04	12.7	6.01	7.47	18.2	21.5	28.1	28.8	17.1	25.9	25.8	16.8
Proposed without refinement	9.86	11.3	19.3	5.44	7.64	17.9	10.3	19.3	22.4	4.88	15.3	12.3	13.0
Proposed with refinement	3.50	3.98	11.7	0.44	0.71	5.66	4.60	9.25	16.1	3.34	8.64	10.8	6.56

B. Quality Evaluation

To discuss the quality of the proposed design, the disparity maps are evaluated based on the Middlebury benchmarks using the percentage of bad pixels on different regions, a commonly accepted metric [7]. Table II lists the accuracy comparison with some state-of-the-art implementations. The first is the AD-Census algorithm implemented on a GPU. It is challenging to realize it into an FPGA because of its multi disparity enhancement functions. The design in [11] utilizes cross-based regions and semi-global optimization on an FPGA. However, its high accuracy is achieved at the expense of the decreased processing speed. The SegSupport algorithm outperforms the proposed design slightly but fails to reach real-time performance. The algorithms in [12] and [13], which are also software implementations, have a higher error rate than the proposed algorithm. The comparison shows that the accuracy of our design is not only among the best in hardware accelerated stereo systems, but also competitive with the state-of-the-art software implementations.

The final disparity maps are compared with the initial disparity maps to indicate the effect of the refinement step. The comparison results are shown in Fig. 7. Some of them are generated from the left images and the others are generated from the right images so that the proposed system is comprehensively evaluated. The refinement step contributes significantly to the final disparity maps and many visual improvements are obvious, including the elimination of speckle noise, few errors at the image borders and sharply delineated edges. The quantitative results in Table II also prove it.

The resolutions of the data set Tsukuba, Venus, Teddy and Cones are all smaller than that of VGA. To further evaluate the proposed design, some high-definition images in the benchmark are used with a disparity range of 128 pixels. The results of the data set Art and Cloth2 captured at different viewpoints are shown in Fig. 7. The overall error rates are 12.85% and 4.67% respectively. The proposed system provides quite clear and smooth disparity maps and the accuracy is comparable to the low-definition results.

TABLE III
PROCESSING SPEED COMPARISON

Design	Platform	Image size	Disparity range	FPS	MDE/s
Shan et al. [8]	FPGA	1280 × 1024	256	46	15437
MCADSR [14]	FPGA	1024 × 768	128	129	13076
AWDE-IR [6]	FPGA	1024 × 768	128	60	6040
Zhang et al. [5]	FPGA	1024 × 768	64	60	3019
Wang et al. [11]	FPGA	1024 × 768	96	31.8	2400
Ttofis et al. [1]	FPGA	640 × 480	64	30	589
MCADSW [4]	ASIC	352 × 288	64	42	272
AD-Census [10]	GPU	450 × 375	60	10.6	107
Yang et al. [15]	GPU	640 × 360	20	10	46
VariableCross [13]	CPU	450 × 375	60	0.63	13
SemiGlobal [12]	CPU	450 × 375	64	0.55	6
Proposed	FPGA	1024 × 768	64	65	6543

C. Processing Speed

The processing speed is given by million disparity estimations per second (MDE/s), which equals to (image size × disparity range × frame rate). Table III presents a comparison between some exiting implementations and the proposed design. It is obvious that CPU and GPU based implementations can hardly achieve real-time speed with high resolution images. For FPGA implementations, the achievable processing speed is usually limited by the available hardware resources, such as on-chip memories. The proposed system could achieve 65 fps for 1024 × 768 images with a disparity range of 64 pixels. It is a fully pipelined design and generates disparity maps of both images simultaneously. Some systems [8], [14] which are quite outstanding in Table III perform much better than the proposed system. Although the design in [8] has the highest processing speed, it is based on a simple SAD matching method that leads to low accuracy. The system in [14] improves the accuracy with variable support regions, but its accuracy is still worse than that of the proposed system.

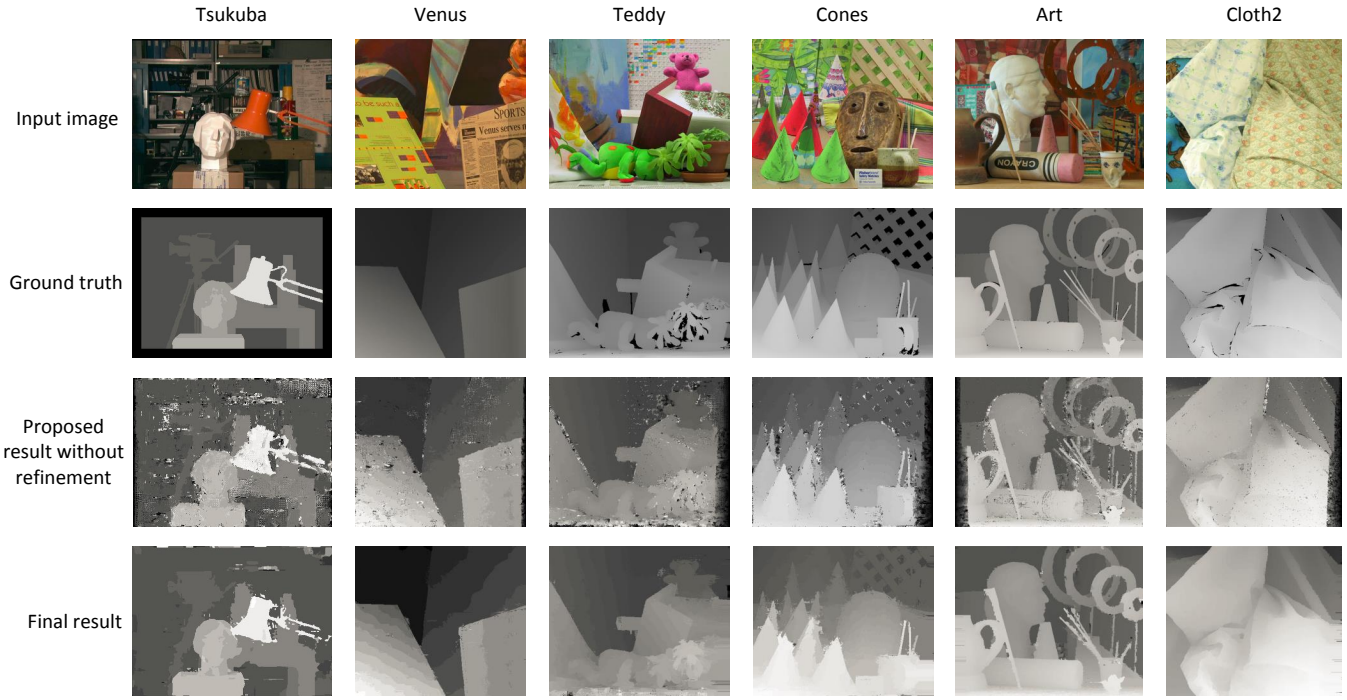

Fig. 7. True disparity maps and experimental results.

V. Conclusion

This paper has proposed a stereo matching algorithm based on the mini-census transform and segmentation-based ADSW. The disparity refinement step with segmentation information has been presented and the quality of disparity maps has been improved significantly. Furthermore, a fully pipelined and scalable hardware architecture is designed with hardware-oriented optimizations. A prototype of the hardware system has been built on an Altera Stratix-IV FPGA. The design is evaluated with the Middlebury benchmarks and the average error rate is 6.56%. In order to improve the accuracy, a part of global matching algorithms can be introduced into the algorithm in the future.

Acknowledgment

The research in this paper was sponsored in part by the Belgian FWO (Flemish Research Council) and the Chinese MOST (Ministry of Science and Technology) bilateral cooperation project number G.0524.13.

References

[1] C. Ttofis and T. Theocharides, "Towards accurate hardware stereo correspondence: A real-time fpga implementation of a segmentation-based adaptive support weight algorithm," in *in Proc. Design Autom. Test Eur. Conf. Exhibit. (DATE)*, 2012, pp. 703–708.

[2] F. Tombari, S. Mattoccia, and L. Di Stefano, "Segmentation-based adaptive support for accurate stereo correspondence," in *Proceedings of the 2Nd Pacific Rim Conference on Advances in Image and Video Technology*, 2007, pp. 427–438.

[3] http://vision.middlebury.edu/stereo/.

[4] N. Y.-C. Chang, T.-H. Tsai, B.-H. Hsu, Y.-C. Chen, and T.-S. Chang, "Algorithm and architecture of disparity estimation with mini-census adaptive support weight," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 20, no. 6, pp. 792–805, Jun. 2010.

[5] L. Zhang, K. Zhang, T. S. Chang, G. Lafruit, G. K. Kuzmanov, and D. Verkest, "Real-time high-definition stereo matching on fpga," in *Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays*. ACM, 2011, pp. 55–64.

[6] A. Akin, I. Baz, A. Schmid, and Y. Leblebici, "Dynamically adaptive real-time disparity estimation hardware using iterative refinement," *Integration, the VLSI Journal*, vol. 47, no. 3, pp. 365 – 376, 2014.

[7] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1, pp. 7–42, 2012.

[8] Y. Shan, Z. Wang, W. Wang, Y. Hao, Y. Wang, K. Tsoi, W. Luk, and H. Yang, "Fpga based memory efficient high resolution stereo vision system for video tolling," in *Field-Programmable Technology (FPT), 2012 International Conference on*, Dec 2012, pp. 29–32.

[9] K. Zhang, J. Lu, G. Lafruit, R. Lauwereins, and L. V. Gool, "Real-time accurate stereo with bitwise fast voting on cuda," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, Sept 2009, pp. 794–800.

[10] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 467–474.

[11] W. Wang, J. Yan, N. Xu, Y. Wang, and F. H. Hsu, "Real-time high-quality stereo vision system in fpga," in *Field-Programmable Technology (FPT), 2013 International Conference on*, Dec 2013, pp. 358–361.

[12] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb 2008.

[13] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1073–1079, July 2009.

[14] Y. Shan, Y. Hao, W. Wang, Y. Wang, X. Chen, H. Yang, and W. Luk, "Hardware acceleration for an accurate stereo vision system using mini-census adaptive support region," *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 4s, pp. 132:1–132:24, Apr. 2014.

[15] Q. Yang, D. Li, L. Wang, and M. Zhang, "Fast local stereo matching using two-level adaptive cost filtering," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 1986–1990.