A SoPC design of a real-time high-definition stereo matching algorithm based on SAD

Xiang Zhang*, Huaixiang Zhang, Yifan Wu

School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China

Received 1 January 2014, www.tsi.lv

Abstract

The System-on-Programmable-Chip (SoPC) architecture to implement a stereo matching algorithm based on the sum of absolute differences (SAD) in a FPGA chip is proposed. The hardware implementation involves a 32-bit Nios II microprocessor, memory interfaces and stereo matching algorithm circuit module. The Nios II microprocessor is a configurable soft IP core in charge of managing the buffer of the stereo images and users' configuration data. The system can process any different sizes of stereo pair images through a configuration interface. The maximum horizon resolution of stereo images is 2048. When the algorithm core works under 60MHz, the 1396×1110 disparity map can be achieved at 30 fps speed.

Keywords: Stereo matching, System-on-programmable-chip, FPGA, Disparity map, SAD

1 Introduction

The Stereo vision has been one of the most active research topics in computer vision and widely used in many application areas including intelligent robots, automated guided vehicle, human-computer interface, and so on [1- 3]. Stereo matching algorithms have played an important role in stereo vision. They can be classified into either local or global methods of correspondence. Local methods match one window region centred at a pixel of interest in one image with a similar window region in the other image by searching along epipolar lines. The performance of local stereo matching algorithms depends to a large extent on what similarity metric is selected. Typical similarity metrics are cross-correlation (CC), the sum of absolute differences (SAD), the sum of squared differences (SSD), *etc.* SSD and SAD find correspondences by minimizing the sum of squared or that of absolute differences in $W \times W$ windows. The computational complexity for $N \times N$ resolution image pair, $W \times W$ window size and D disparity level is $O(N^2W^2D)$. It can be decreased to $O(N^2D)$ by some kind of optimization tips [4]. Therefore, the stereo vision has limitations for real-time applications due to its computational expense.

Many researchers have proposed their FPGA implementations of stereo vision algorithms in literature. The circuit [5] is a stereovision system based on a Xilinx Virtex II using the SAD algorithm. The system can process images with a size of 270 × 270 at a frame rate of 30 fps. Paper [6] presents a FPGA-based stereo matching system that operates on 512 × 512 stereo images with a maximum disparity of 255 and achieves a frame rate of 25.6 fps running under a frequency of 286 MHz. In [7], a development system based on four Xilinx XCV2000E chips is used to implement a dense, phase correlation-based stereo system that runs at a frame rate of 30 fps for 256 × 360 pixels stereo pairs. Gardel *et al* introduce in [8] their design, which can obtain 30,000 depth points from images of 2 Mpix at a frame rate of 50 frames per second under a 100 MHz working frequency. A real-time fuzzy hardware module based on a colour SAD window-based technique is proposed in [9]. This module can theoretically provide accurate disparity map computation at a rate of nearly 440 frames per second without considering the memory delay and other factors of time consumption, thus giving a stereo image pair with a disparity range of 80 pixels and 640 × 480 pixels resolution. The design in [10] is a 7 × 7 binary adaptive SAD based real-time stereo vision architecture with a depth range of 80, which is implemented on the Altera Cyclone II EP2C70 FPGA chip based on 800 × 600 colour images and operates in real-time at a frame of 56 Hz. The architecture captures the 90 Megapixel/sec 12 bit signals of two cameras in real-time and does not require memories external to the FPGA. However, these designs rarely attain the target of producing above 720P resolution disparity map at real-time speed.

In this paper, we propose the SoPC architecture to implement a stereo matching algorithm, which can process HD stereo, images in real-time by using the SAD stereo matching algorithm. The stereo matching process, including cost calculation, cost aggregation and L-R checking, are designed and parallelized within a pipelined architecture. Based on efficient hardware-oriented optimizations, our design achieves 30 frames per second when it matches 1396 × 1110 high-definition stereo images under 60MHz working frequency.

* ***Corresponding author*** e-mail: gavin@hdu.edu.cn

2 SoPC architecture for SAD matching algorithm

The SoPC architecture proposed herein is divided into the following main modules as shown in Figure 1:

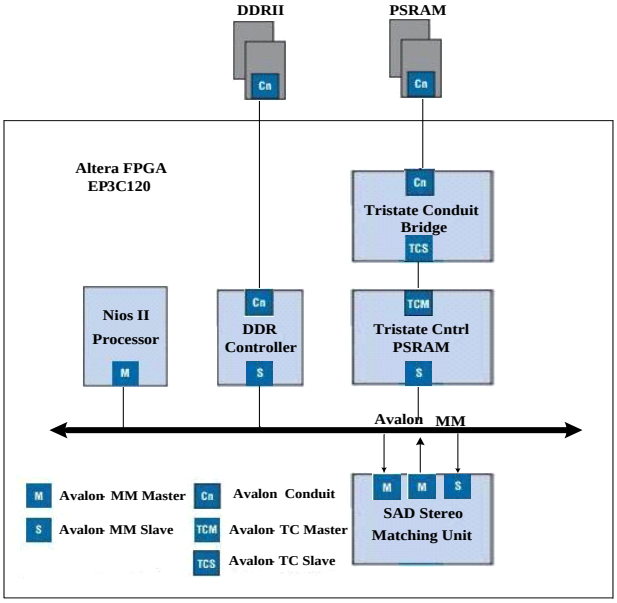


FIGURE 1 Proposed SoPC architecture

(a) Nios II processor system: It consists of a 32-bit Nios II processor core, a set of on-chip peripherals, on-chip memory, and interfaces to off-chip memory.

(b) SAD Stereo Matching Unit (SSMU): This unit computes sum of the absolute difference as similarity metrics to seek disparities from 64 candidates 5×5 windows. The disparities for right and left image are computed concurrently and sent to L-R checking module to take valid detection. The unit has three Avalon-MM interfaces. One is slave interface to communicate with the Nios II CPU. The other two are read and write master interfaces. The read master is in charge of reading raw data of stereo images from the off-chip PSRAM acting as frame buffer in system. The write master takes charge of write final disparities to the DDRII.

(c) DDR Controller and PSRAM Controller: These two IPs are provided by Altera. They enable the system to access the DDRII and PSRAM memory out of the FPGA chip.

The modules listed above are all synthesized in one EP3C120 FPGA chip produced by Altera Company. They connect with each other by Avalon bus interface.

3 Implementation of the SAD stereo matching unit

3.1 THE SAD STEREO MATCHING ALGORITHM

The SAD algorithm has the advantage of computational efficiency. The SAD equation used for 5×5 windows with a maximum disparity of 64 can be seen below:

$$\mathrm{SAD}(i, j, disp) = \sum_{h=-2}^{2} \sum_{k=-2}^{2} \left| P_R(i+h, j+k) - P_L(i+h, j+k+disp) \right|, \quad (1)$$

where ***disp*** is the disparity value ranging from 0 to 63, $P_R(i, j)$ serves as the reference pixel in the right image and $P_L(i, j+disp)$ as the currently analysed candidate pixel in the left image. The reference 5×5 window centred at $P_R(i, j)$ is compared to 64 possible candidate windows in left image to calculate 64 SAD values. There are 25 bytes of data in the right image and 340 bytes of data in the left image involved in the calculation of disparity(i, j), where disparity(i, j) means the disparity value of the pixel(i, j).

3.2 IMPLEMENTATION OF THE SAD STEREO MATCHING UNIT

The layout plan of the SSMU is showed in Figure 2. At the first stage of the SSMU is a custom DMA engine. It is in charge of transferring all raw image data from the PSRAM to the dual-clock FIFOs.

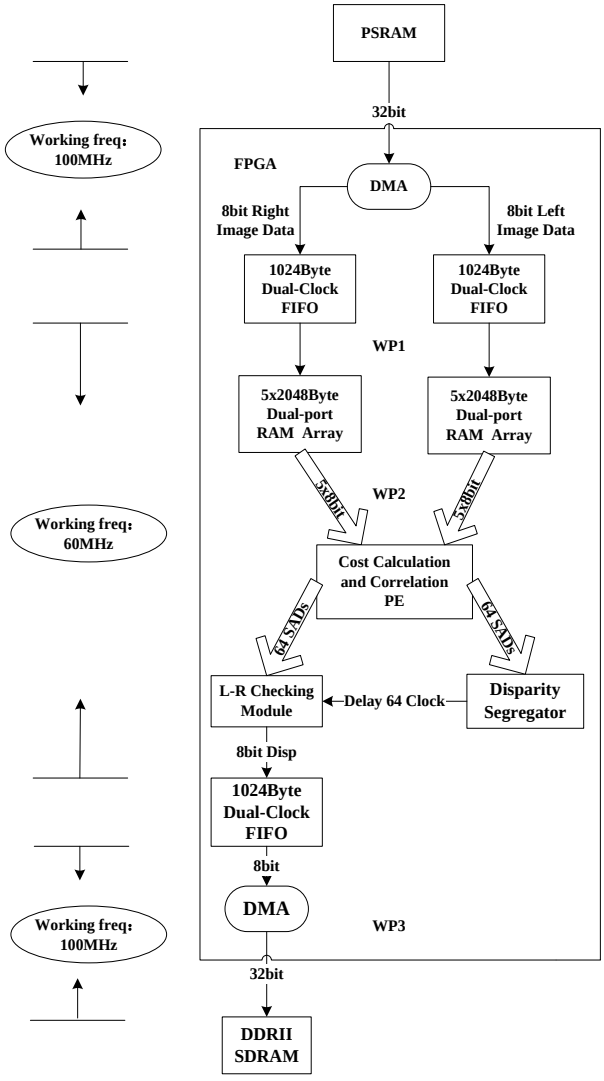


FIGURE 2 Layout plan of the SSMU module

Every time the DMA engine invokes 8 times of word size pipelined Avalon-MM interface reading to get 32 bytes data from the right or left image in turn and it almost consumes 27 clocks. Therefore, under the working frequency of 100MHz, the DMA engine can offer about 113MB per second data bandwidth. It is enough for 2 × 1396 × 1110@30fps needs.

The two dual-clock FIFOs (DCFIFO), followed by the DMA engine, has two functions. One is a temporary storage for the raw pixel data. The other is separating the workspace into two regions, which work under different working frequency. At the writing and reading side of the FIFO, the working frequency is 100MHz and 60MHz respectively.

Beside the read side of the DCFIFO is a dual-port RAM (DPRAM) array, which is constituted by 5 DPRAMs, each with 2048 byte of memory space. They are used as line buffers for cost calculation and correlation processing element (CCCPE). Every DPRAM has data bus connected with CCCPE; therefore, every DPRAM array can output 5 bytes of image data to the CCCPE in every clock cycle.

The CCCPE is the most complex hardware circuits in the SSMU. As shown in Figure 3, there are 64 SAD computers and 2 shift-tap devices in the CCCPE.

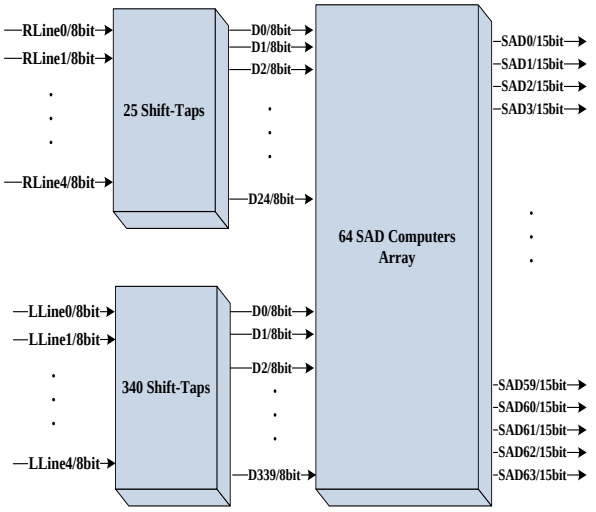


FIGURE 3 Block diagram of the CCCPE

The 2 shift-tap devices have 25 and 340 shift registers respectively to store the pixel data participated in computation of the SADs. One SAD computer can sum 25 absolute differences up produced by two 5 × 5 matching windows in a clock-period. Therefore, the CCCPE module can achieve 64 SADs from a template window in the right census image compared with 64 candidate windows in the left census image in a single clock. Figure 4 is the block diagram of the SAD computers. The SAD computer is constituted with 25 absolute difference calculators (AD) and a parallel adder with 25 inputs. The parallel adder has 4 pipelined stages architecture for improving the f_{max}.

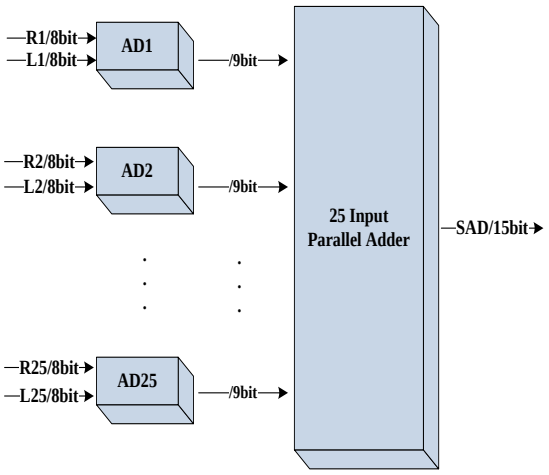


FIGURE 4 Block diagram of the SAD computer

The 64 SADs produced by CCCPE are sent to the disparity segregator (DS) and L-R checking module simultaneously. The DS calculates the minimum SAD using parallel comparators from 64 SADs and outputs the index number as the disparity of the right image, and the cost time is one clock period. Figure 5 is the block diagram of the disparity segregator. The L-R checking module produces the disparity of the left image and checks the right and left disparity to get the final valid one.

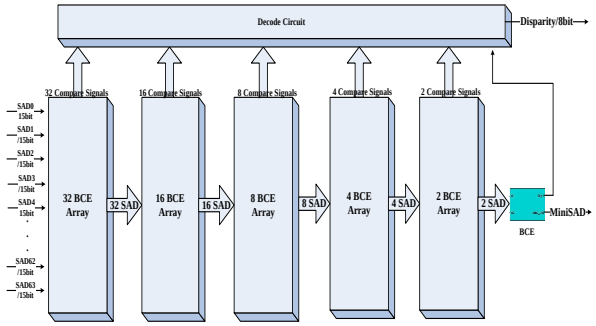


FIGURE 5 Block diagram of the disparity segregator

The final disparities are pushed into the DCFIFO connected with the L-R checking module. The DCFIFO's function in here is similar with the one mentioned in previous. A custom DMA engine followed with read port of the DCFIFO is responsible for writing the final disparities to the disparity table stored in the off-chip DDRII SDRAM.

The work procedure of L-R checking module can be separated into two parts. The one is calculating the disparity map, which takes the left image as reference, and the other is comparing the right and left disparity map with the metric listed in (2):

$$d_{p'}(x - d_p(x,y), y) = d_p(x,y), \quad (2)$$

where $d_p(x,y)$ and $d_{p'}(x,y)$ are the disparities of the p and p', which are pixels in the left and right image separately. If the $d_p(x,y)$ and $d_{p'}(x,y)$ make the equation (2) true, the $d_p(x,y)$ is a valid disparity,

otherwise we should replace it with the valid disparity near to it.

The principle of producing the left image disparity in L-R checking module is showed in Figure 6. The number pair (L/R) displayed in the block in Figure 6 means that the window L in the left image matches with the window R in the right image at the same horizon line. We can get a conclusion that the 64 blocks at the vertical direction of Figure 6 are the results produced by matching the right image as the reference with the left image. From the every column, we can get a disparity of the right image. Along the diagonal of Figure 6, we can get the disparity of the left image. Therefore, we can get both right and left disparities by establishing only one stereo matching algorithm circuit.

63/0	**64/1**	**65/2**	·····	**126/63**
62/0	**63/1**	**64/2**	·····	**125/63**
61/0	**62/1**	**63/2**	·····	**124/63**
⋮	⋮	⋮	·····	⋮
0/0	**1/1**	**2/2**	·····	**63/63**

FIGURE 6 The principle of producing the left image disparity

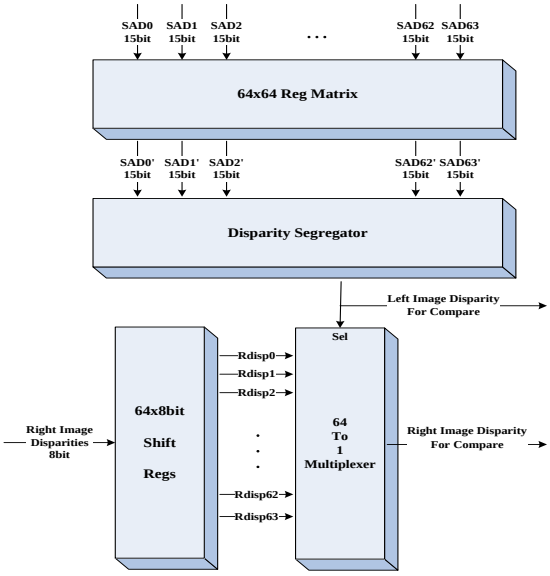


FIGURE 7 Diagram of the L-R checking module

The hardware diagram of the L-R checking module is showed in Figure 7. At the beginning of the L-R checking module is a 64×64 register matrix in unit of byte. It consists of 64 shift-taps with 64 taps each, and accepts 64 SADs produced by CCCPE in every clock. The register matrix is fully initiated with 4096 SADs, and then outputs buffered SADs along the diagonal of the matrix to the disparity segregator to calculate the left image disparity. So when first disparity of the left image is calculated, there were 64 right image disparities have been calculated, and they are buffered into a shift-tap device with 64 taps. The shift-tap device outputs the all buffered right image disparities to a 64 to 1 multiplexer, which takes the left image disparity as the select signal. The output of the multiplexer is the right image disparity, which will compare with the left image disparity as listed in (2).

The disparities produced after L-R valid checking are the last results we need. They are pushed into a dual-clock FIFO and a DMA engine is responsible for writing them to the disparity table stored in the off-chip DDRII SDRAM. The dual-clock FIFO's function in here is similar with the one mentioned in CTU module.

3.3 CONTROL PRINCIPLES OF THE SSMU

The main management work of the SSMU module is listed below:

(a) Initialization and updating data of the DPRAM array;
(b) The CCCPE work process control;
(c) Getting the finial disparities and writing them back to the buffer memory.

There are three Finite State Machines, which are in charge of the process management of the SSMU. They take effect at the work positions "WP1", "WP2" and "WP3" which are marked in Figure 2. The FSMs are custom IPs and designed in Verilog HDL. The hardware modules modelled by DSP Builder have interfaces for controlled by the FSMs to work together. Figure 8 shows the FSM for data updating management of the DPRAM array.

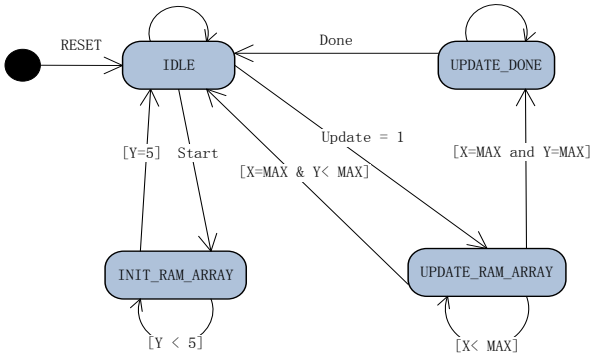


FIGURE 8 Finite state machine of the DPRAM array updating management

Figure 9 indicates the FSM for the computation control of the CCCPE.

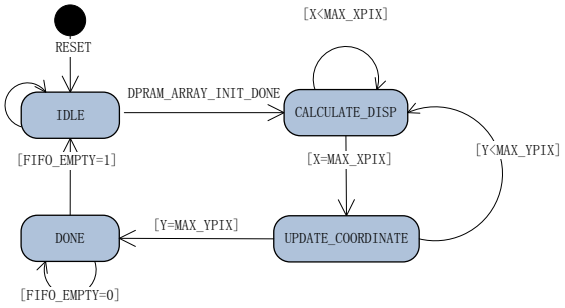


FIGURE 9 Finite state machine of the CCCPE control

The data updating in DPRAM array is activated by the start signal of the system. After the set action of the start signal, the FSM begins to read the raw data from the dual-clock FIFO if it is not empty, and send them to the DPRAM array according to the pixels' coordinate order. This state is not finished until 5 DPRAMs are fully initialized. Then the FSM comes into idle state. An update signal activates the FSM into the UPDATE_RAM_ARRAY state. The FSM continuously updates data using a unit in bytes to the DPRAM array. The FSM will not change into idle state until the X coordinate reaches maximum. The detail description about state machine of Figure 8 is listed as follows:

(a) After system reset, the FSM enters into the IDLE state automatically. In this state, the FSM waits for two signals to invoke the state transition. The one is a "Start" signal sent by the program executed in Nios II microprocessor to make the state change to INIT_RAM_ARRAY. The other is a "Update" signal sent from the FSM at "WP2" to make the state transit to UPDATE_RAM_ARRAY state.

(b) The task of the INIT_RAM_ARRY state is to initiate the DPRAM array with first 5 lines data of the stereo images, then return to the IDLE state to wait for update signal.

(c) Every time the FSM enters into the UPDATE_RAM_ARRAY state, a line of new pixel data will be transferred into the DPRAM array. After finishing the update work, the Y coordinate will be increased by one and checked if it reaches maximum. If Y equals to maximum, the state transits to UPDATE_DONE state, otherwise the state moves to IDLE state to wait for the next update signal.

(d) In UPDATE_DONE state, the FSM checks the "Done" signal of the system. If the signal is set, the FSM backs to IDLE state to wait for a new "Start" signal.

The work process control of the CCCPE is performed by the FSM working at "WP2" which is described in Figure 9. In the following list, the details of the tasks performed in each of the states are described:

(a) After system reset, the FSM enters into the IDLE state automatically. In this state, all variables are initialized.

(b) In IDLE state, the FSM waits for a DPRAM_ARRAY_INIT _DONE signal to transit into CALCULATE_DISP state. This signal is sent by the FSM at "WP1".

(c) In CALCULATEDISP state, the FSM reads the data from the DPRAM array to the CCCPE uninterruptedly till the pixel's horizontal coordinate of current line reaches maximum. An updated signal is also sent to notify the FSM at "WP1" to start updating a new line data to the DPRAM array. The update signal of the right DPRAM array is sent after 68 pixels of the left image were sent to the CCCPE for avoiding collision. The update signal of the left DPRAM array is sent after 5 pixels of the left image were sent to the CCCPE for avoiding collision. When entire 5 lines of data are sent, the state comes into the UPDATE_COORDINATE state.

(d) In the UPDATE_COORDINATE state, the pixel's vertical coordinate is updated. There are two exits: (1) if the vertical coordinate is smaller than the maximum, the FSM transits to CALCULATE_DISP state; (2) if the vertical coordinate is equal to the maximum, the DONE state is the next state.

(e) In DONE state, the FSM scans the disparity FIFO's empty signal. If all disparities are written back to the DDRII SDRAM, the FSM becomes idle again.

4 Results and discussion

The stereo matching circuit has been realized in Altera Cyclone III EP3C120f789 FPGA board shown in Figure 10. Table 1 and Table 2 show the resources required from the FPGA device in order to implement the designs presented in this paper.

FIGURE 10 The Cyclone III development board

TABLE 1 Resources needed for the implementation of the algorithm (with L-R checking module)

Device :Altera EP3C120F780C7N (Cyclone III device family)		
Resource		
Total logic elements	98598 / 119088	(83%)
Total combinational functions	89145 / 119088	(75%)
Dedicated logic registers	51099 / 119088	(43%)
Total registers	51328	
Total pins	151 / 532	(28%)
Total memory bits	530216 / 3981312	(3.3%)
Embedded Multiplier 9-bit elements	12 / 576	(2%)
Total PLLs	2 / 4	(50%)

TABLE 2 Resources needed for the implementation of the algorithm (without L-R checking module)

Device :Altera EP3C120F780C7N (Cyclone III device family)		
Resource		
Total logic elements	93242 / 119088	(78%)
Total combinational functions	86012/ 119088	(72%)
Dedicated logic registers	46765 / 119088	(39%)
Total registers	46994	
Total pins	151 / 532	(28%)
Total memory bits	448296 / 3981312	(12%)
Embedded Multiplier 9-bit elements	12 / 576	(2%)
Total PLLs	2 / 4	(50%)

The quality measures proposed by [11] are based on known ground truth data $d_T(x,y)$ offered by Middlebury Stereo datasets were used for evaluation. The percentage of bad matching pixels is computed with respect to some error tolerance δ_d:

$$M = \frac{1}{N}\sum_{x,y}(|\, d_c(x,y) - d_T(x,y) \,| > \delta_d)\,, \qquad (3)$$

where $d_c(x,y)$ is a disparity map produced by the proposed hardware .

The ground truth image has a disparity range from 0 to 59. The disparity range of our design is 0 to 63. So a disparity error tolerance δ_d = 1 to 4 is used. The measures are computed over the whole disparity map, excluding image borders, where part of the image is totally occluded.

Several tests have been performed. In the Table 3, the disparity maps are produced without L-R checking module. In the Table 4, the examples of disparity maps obtained with the L-R Checking module. Disparities are all encoded using a scale factor of 4 for grey levels 0 to 252.

TABLE 3 Evaluation result of the proposed system using Middlebury stereo images (without L-R checking module)

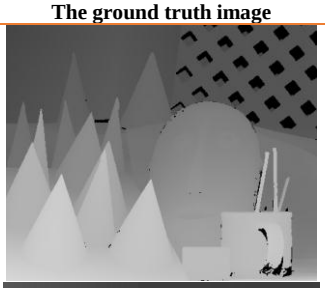
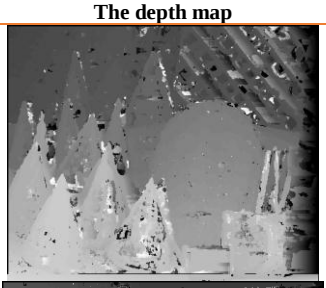
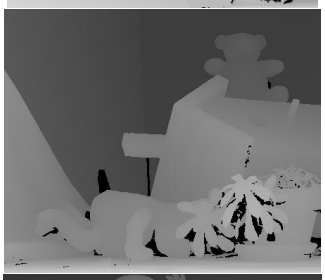
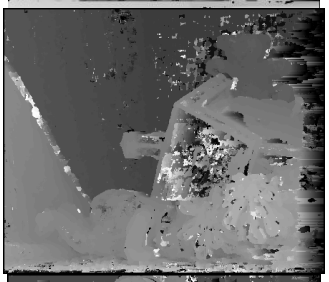
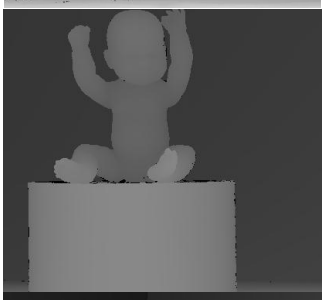
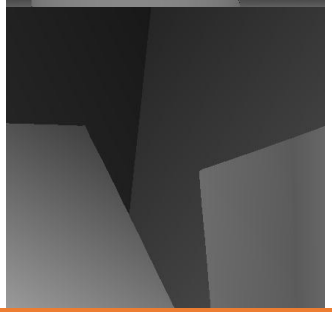
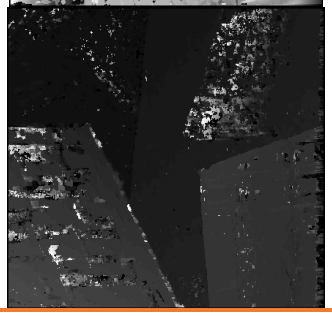
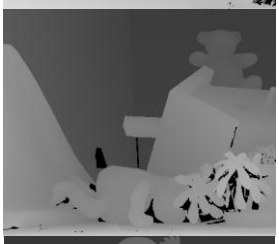
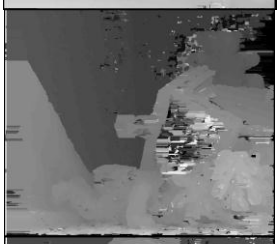
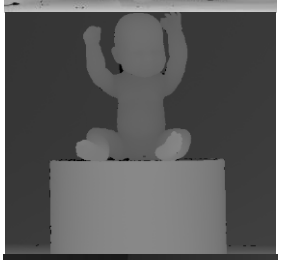
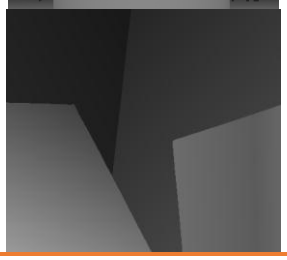
The original right image	The ground truth image	The depth map	Bad pixels
			25.3% (δ_d = 1) 22.15% (δ_d = 2) 19.33% (δ_d = 3) 17.09% (δ_d = 4)
			23.92% (δ_d = 1) 20.98% (δ_d = 2) 18.93% (δ_d = 3) 17.44% (δ_d = 4)
			31.39% (δ_d = 1) 28.25% (δ_d = 2) 26.44% (δ_d = 3) 25% (δ_d = 4)
			16.34% (δ_d = 1) 14.07% (δ_d = 2) 12.31% (δ_d = 3) 10.76% (δ_d = 4)

TABLE 4 Evaluation result of the proposed system using Middlebury stereo images (with L-R checking module)

The original left image	The ground truth image	The depth map	Bad pixels
			22.46% (δ_d = 1) 19.64% (δ_d = 2) 16.74% (δ_d = 3) 14.52% (δ_d = 4)
			23.39% (δ_d = 1) 20.6% (δ_d = 2) 18.75% (δ_d = 3) 17.40% (δ_d = 4)
			28.43% (δ_d = 1) 24.96% (δ_d = 2) 23.33% (δ_d = 3) 22.16% (δ_d = 4)
			14.31% (δ_d = 1) 12.56% (δ_d = 2) 11.28% (δ_d = 3) 9.78% (δ_d = 4)

In Table 5, the proposed system is compared to the existing approaches in terms of speed.

TABLE 5 Speed comparison of the stereo matching implementations

Authors	Frame Rate	Image Size	Max. Disp	Algorithm	Window Size	Platform
Motten *et al.* [10]	56 fps	800 × 600	80	SAD	7 × 7	1FPGA
Proposed impl.	30 fps	1396 × 1110	64	SAD	5 × 5	1 FPGA
Software impl. [12]	2.55 fps	320 × 240	100	SAD	3 × 3	PC
Kalomiros *et al.* [13]	162 fps	640 × 480	64	SAD	3 × 3	1FPGA + PC
Niitsuma *et al.* [14]	30 fps	640 × 480	27	SAD	7 × 7	1 FPGA
Miyajima *et al.* [15]	18.9 fps	640 × 480	80	SAD	7 × 7	1FPGA + PC

5 Conclusions

An efficient hardware implementation of a real-time stereo matching algorithm is proposed for the calculation of disparity maps. It takes full advantage of the convenience of IP reuse based on SoPC architecture. The frame rate could enable real time performance at the resolution of 1396×1110. The architecture of our system is very promising and may get better in the future. The system has been implemented on static image input from C code in the Nios II processor. We plan to incorporate live stereo video streams and combine the algorithm with pre-stage to make it more suitable for the operation in robot auto- navigation and visual servo applications.

Acknowledgments

This work has been cooperatively supported by the projects as follows:
(a) National Nature Science Foundation of China (No. 61202093);
(b) National Key Technology R&D Program of China (No. 2014BAF07B01);
(c) 3D Innovation team of Zhejiang provincial.

References

[1] Bertozzi M, Broggi A 1998 *IEEE Transactions on Image Process* **7**(1) 62-81
[2] Uchida N, Shibahara T, Aoki T, Nakajima H, Kobayashi K 2005 3-D face recognition using passive stereo vision *Proceedings of IEEE International Conference on Image Process* 950-3
[3] Bensrhair A, Bertozzi M, Broggi A, Fascioli A, Mousset A, Toulminet G 2002 Stereo vision-based feature extraction for vehicle detection *IEEE Intelligent Vehicle Symposium 2* 465-70
[4] Brown Z M, Burschka D, Hager D G 2003 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(8) 993-1008
[5] Yi J, Kim J, Li L, Morris J, Lee G, Leclercq P 2004 *Lecture Notes in Computer Science* **3189** 309–20
[6] Perri S, Colonna D, Zicari P, Corsonello P 2006 SAD-Based Stereo Matching Circuit for FPGAs *Proceedings of the 13th IEEE International Conference on Electronics, Circuits and Systems* 846–9
[7] Darabiha A, Rose J, MacLean W J 2003 Video-Rate Stereo Depth Measurement on Programmable Hardware *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 203–10
[8] Gardel A, Montejo P, Garcia J, Bravo I, Lázaro J 2012 *Sensors* **12** 1863–84
[9] Georgoulas C, Andreadis I 2011 *J. Real Time Image Process* **6** 257–73
[10] Motten A, Claesen L 2011 Low-Cost Real-Time Stereo Vision Hardware with Binary Confidence Metric and Disparity Refinement *Proceedings of the 2011 International Conference on Multimedia Technology* 3559–62
[11] Scharstein D, Szeliski R 2002 *International Journal of Computer Vision* **47**(1-3) 7-42
[12] Ambrosch K, Humenberger M, Kubinger W, Steininger A 2007 Hardware Implement of an SAD Based Stereo Vision Algorithm *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1–6
[13] Kalomiros J, Lygouras J 2009 Comparative study of local SAD and dynamic programming for stereo processing using dedicated hardware *EURASIP J. Adv. Signal Process* doi:10.1155/2009/914186
[14] Niitsuma H, Maruyama T 2004 *Lecture Notes in Computer Science* **3203** 1155–7
[15] Miyajima Y, Maruyama T 2003 *Lecture Notes in Computer Science* **2778** 448–57

Authors

Xiang Zhang, born in 1979

Current position and grades: MSc, a lecturer at Hangzhou Dianzi University in China.
University studies: BSc and degrees in computer science, Hangzhou Dianzi University, Hangzhou City, China, in 2001 and 2004, respectively. He is currently working toward a Ph.D. degrees with the State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University.
Research interests: computer vision, image parallel processing, embedded systems, and real-time applications

Huaixiang Zhang, born in 1978

Current position and grades: PhD, Department of Computer Science, Hangzhou Dianzi University, since March 2007
University studies: BSc in Mechanical Engineering from North China University of Technology in 2000, MSc in Automatic Control from Beijing Institute of Technology in 2003, and PhD in Automatic Control from Institute of Automation, Chinese Academy of Science in 2007
Research interests: motion control theory in mobile robot, non-linear and adaptive control, electrical motors and artificial intelligence

Yifan Wu

Current position and grades: Assistant Professor at Hangzhou Dianzi University in China
University studies: BSc and BSc degrees in Control Science and Engineering from Zhejiang University in 2003 and 2006, respectively, and the Ph.D. degree in Computer Engineering from Scuola Superiore Sant'Anna of Pisa in January 2010, a visiting student at the Department of Automatic Control, Lund University, 2009.
Research interests: real-time control systems, scheduling algorithms, and resource management