A Hardware-Oriented Dynamically Adaptive Disparity Estimation Algorithm and its Real-Time Hardware

Abdulkadir Akin, Ipek Baz
STI, EPFL, 1015
Lausanne, Switzerland
{abdulkadir.akin,
ipek.baz}@epfl.ch

Baris Atakan
EEE, METU,
Ankara, Turkey
el174054@
metu.edu.tr

Irem Boybat
EE, SU,
Istanbul, Turkey
iremboybat@
sabanciuniv.edu

Alexandre Schmid, Yusuf Leblebici
STI, EPFL, 1015,
Lausanne, Switzerland
{alexandre.schmid,
yusuf.leblebici}@epfl.ch

ABSTRACT

The computational complexity of disparity estimation algorithms and the need of large size and bandwidth for the external and internal memory make the real-time processing of disparity estimation challenging, especially for High Resolution (HR) images. This paper proposes a hardware-oriented adaptive window size disparity estimation (AWDE) algorithm and its real-time reconfigurable hardware implementation that targets HR video with high quality disparity results. The proposed algorithm is a hybrid solution involving the Sum of Absolute Differences and the Census cost computation methods to vote and select the best suitable disparity candidates. It utilizes a pixel intensity based refinement step to remove faulty disparity computations. The AWDE algorithm dynamically adapts the window size considering the local texture of the image to increase the disparity estimation quality. The proposed reconfigurable hardware of the AWDE algorithm enables handling 60 frames per second on Virtex-5 FPGA at a 1024×768 XGA video resolution for a 120 pixel disparity range.[1]

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application Based Systems**]: Real-time and embedded systems, signal processing systems

General Terms

Algorithms, Design.

Keywords

Disparity Estimation, Census Transform, Sum of Absolute Differences, High Resolution, Real-Time Implementation, FPGA.

1. INTRODUCTION

Depth estimation is an algorithmic step in a variety of applications such as autonomous navigation, robot and driving systems, 3D geographic information systems, object detection and tracking, medical imaging, computer games, 3D television, stereoscopic video compression, and disparity-based rendering.

Many Disparity Estimation (DE) algorithms have been developed with the goal to provide high-quality depth map results. These are ranked with respect to their performance in the evaluation tool for the Middlebury benchmarks [1]. Although top performer algorithms provide impressive visual and quantitative results [2-3], their implementations in real-time High Resolution (HR) stereo video are challenging due to their complex multi-step refinement processes or their global processing requirements that demand huge memory size and bandwidth.

Various hardware architectures that are presented in literature provide real-time DE [4-9]. Some implemented hardware architectures only target CIF or VGA video [4-6]. The hardware proposed in [4] only claims real time for CIF video. It uses the Census transform [10] and currently provides the highest quality disparity results compared to real time hardware implementations in ASICs and FPGAs. The hardware presented in [4] uses low complexity Mini Census method to determine the matching cost, and aggregates the Hamming costs following the method in [2]. Due to high complexity cost aggregation, the hardware proposed in [4] requires high memory bandwidth and intense hardware resource utilization, even for Low Resolution (LR) video.

Real-time DE for HR images offers some crucial advantages compared to low resolution DE. Processing HR stereo images increases the disparity map resolution which improves the quality of the object definition. In addition, DE for HR stereo images is able to define the disparity with sub-pixel efficiency compared to the DE for LR image. Therefore, the DE for HR provides more precise depth measurement than the DE for LR. However, the use of HR stereo images brings some challenges. Pixel-wise stereo matching operations cause a sharp increase in computational complexity when DE for HR is targeted. Moreover, DE for HR stereo images requires stereo matching checks with larger number of candidate pixels than the disparity estimation for LR images.

The systems proposed in [7-9] claim to reach real time for HR video. Still, their quality results in terms of the HR benchmarks given in [1] are not provided. [7] claims to reach 550 fps for 80 pixel disparity range at a 800×600 video resolution, but it requires extremely large hardware resources. A simple edge-directed method presented in [8] reaches 50 fps at a 1280×1024 video resolution and 120 pixel disparity range, but does not provide satisfactory DE results due to a low-complexity architecture. In [9], a hierarchical structure with respect to image resolution is presented to reach 30 fps at a 1920×1080 video resolution and 256 pixel disparity range, but it does not provide high quality DE for HR.

In this paper, we present a hardware-oriented adaptive window size disparity estimation (AWDE) algorithm and its real-time reconfigurable hardware implementation to process HR stereo video with high-quality disparity estimation results. The proposed

algorithm combines the strengths of the Binary Window SAD (BW-SAD) [11] and Census Transform methods thus enables an efficient hybrid solution for the hardware implementation.

The benefit of using different window sizes for different texture features on the image is observed from the DE results in [11]. The hardware presented in [11] is not able to dynamically change the window size, since it requires to re-synthesize the hardware for using different window sizes. The hardware presented in this paper provides dynamic configurability to have satisfactory disparity estimation quality for the images with different contents. It provides dynamic reconfigurability to switch between window sizes of 7×7, 13×13 and 25×25 pixels in run-time to adapt to the texture of the image.

The proposed dynamic reconfigurability provides better DE results than existing real-time DE hardware implementations for HR images [7-9] for the tested HR benchmarks. The proposed hardware can reach 60 frames per second on Virtex-5 FPGA at a 1024×768 XGA video resolution and 120 pixel disparity range.

2. HARDWARE-ORIENTED AWDE ALGORITHM

The main focus of the AWDE algorithm is its compatibility with real-time hardware implementation while providing high quality DE results for HR. The algorithm consists of three main parts: window size determination, disparity voting, and disparity refinement.

As a terminology, we use the term "block" to define the 49 pixels in the left image that are processed in parallel. The term "window" is used to define the 49 sampled neighboring pixels of any pixel in the right or left images with variable sizes of 7×7, 13×13 or 25×25. The pixels in the window are used to calculate the Census and BW-SAD cost metrics during the search process. The parameters that are used in the AWDE algorithm are given in Section 4.

2.1 Window Size Determination

The window size of the 49 pixels in each block is adaptively determined according to the Mean Absolute Deviation (MAD) of the pixel in the center of the block with its neighbors. The formula of the MAD is presented in (1), where $\mathbf{c}$ is the center pixel of the block and $\mathbf{q}$ is the pixel in the neighborhood, N_c, of $\mathbf{c}$. The center of the block is the pixel located at block(4, 4) in Fig. 1. (a). Three different window sizes are used. As expressed in (2), a 7×7 window is used if the MAD of the center pixel is high. A very small deviation is the sign of a region with low texture content, and a 25×25 window is used for these regions of the image.

$$MAD(\boldsymbol{c}) = \frac{1}{48} \times \sum_{\boldsymbol{q} \in N_c} |I_L(\boldsymbol{q}) - I_L(\boldsymbol{c})| \qquad (1)$$

$$\begin{aligned} window \\ size \end{aligned} = \begin{cases} 7 \times 7 & if\ \text{MAD}(\boldsymbol{c}) > tr_{7 \times 7} \\ 13 \times 13\ else\ if\ \text{MAD}(\boldsymbol{c}) > tr_{13 \times 13} \\ 25 \times 25\ else \end{cases} \qquad (2)$$

As a general rule, increasing the window size increases the algorithm and hardware complexity [11]. As shown in Fig. 1. (b), in our proposed algorithm, in order to provide constant hardware complexity over the three different window sizes, 49 neighbors are constantly sampled for different window sizes. "1", "2" and "3" indicate the 49 pixels used for the different window sizes 7×7, 13×13 and 25×25, respectively.

2.2 Disparity voting

In this work, the BW-SAD is used as cost metrics instead of SAD. The use of BW-SAD provides better results than using the SAD when there is disparity discontinuity since it combines shape information with the SAD [11]. However, the computational complexity of the BW-SAD is high, thus result of this metric is provided for nine of the 49 pixels in a block and they are linearly interpolated to find the BW-SAD values for the remaining 40 pixels in a block. The selected nine pixels for the computation of BW-SAD are shown in Fig. 1 (a). The low complexity Census metric is computed for all of the 49 pixels of a block.

The formula expressing the BW-SAD for a pixel $\mathbf{p}$ is shown in (3) and (4). The BW-SAD is calculated over all pixels $\mathbf{q}$ of a neighborhood Np, where the notation d is used to denote the disparity. The binary window, $\mathbf{w}$, is used to accumulate absolute differences of the pixels, if they have an intensity value which is similar to the intensity value of the center of the window. The multiplication with $\mathbf{w}$ in (4) is implemented as reset signal for the resulting absolute differences (AD). In the rest of the paper, the term, "*Shape*" is indicated by $\mathbf{w}$.

Depending on the texture of the image, a hybrid selection method is used to combine Census and the BW-SAD. As shown in (5) and (6), an adaptive penalty (ap) that depends on the texture observed in the image is applied to the cost of the Hamming differences between the Census values. Subsequently, the disparity with the minimum *Hybrid Cost* (HC) is selected as the disparity of a searched pixel. 2's order penalty values are used to turn the multiplication operation into a shift operation. If there is a texture on the block, the BW-SAD difference between the candidate disparities needs to be more convincing to change the decision of Census, thus a higher penalty value is applied. If there is no texture on the block, a small penalty value is applied since the BW-SAD metric is more reliable than the decision of Census.
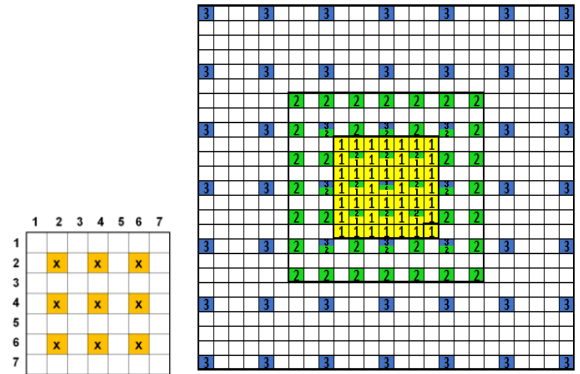

Figure 1. (a) 9 selected pixels in a block for BW-SAD calculation. (b) 49 selected pixels of adaptive windows (yellow (1): 7×7, green (2): 13×13 and blue (3):25×25).

$$w = \begin{cases} 0 & if\ |I_L(\boldsymbol{q}) - I_L(\boldsymbol{p})| > threshold_w,\ \boldsymbol{q} \in N_p \\ 1 & else \end{cases} \qquad (3)$$

$$BW\text{-}SAD(\boldsymbol{p}, d) = \sum_{\boldsymbol{q} \in N_p} |I_L(\boldsymbol{q}) - I_R(\boldsymbol{q} - d)| * \boldsymbol{w} \qquad (4)$$

$$HC(\boldsymbol{p}, d) = BW\text{-}SAD(\boldsymbol{p}, d) + hamming(\boldsymbol{p}, d) \times ap \qquad (5)$$

$$ap = \begin{cases} ap_{7x7} & if & window\ size(\boldsymbol{p}) == 7 \times 7 \\ ap_{13x13}\ else\ if\ window\ size(\boldsymbol{p}) == 13 \times 13 \\ ap_{25x25}\ else\ if\ window\ size(\boldsymbol{p}) == 25 \times 25 \end{cases} \qquad (6)$$

2.3 Disparity Refinement

The proposed Disparity Refinement (DR) process assumes that neighboring pixels within the same *Shape* needs to have an identical disparity value, since they may belong to one unique object. In order to remove the faulty computations, the most frequent disparity value within the *Shape* is used.

The DR process of each pixel is complemented with the disparities of 16 neighbor pixels and its own disparity value. Finally, the most frequent disparity in the selected 17 contributors is replaced with the disparity of that pixel. The disparity of the processed pixel and the disparity of its four adjacent pixels always contribute to the selection of the most frequent disparity.

In Fig. 2, examples of the selection of contributing pixel locations are shown for three different window sizes. Since the proposed hardware processes seven rows in parallel during the search process of a block, the DR process only takes the disparity of pixels in the processed seven rows. Considering the proposed contributor selection scheme, the pixels in the same row with the same window size have identical masks. The masks for the seven rows of a block and three window sizes are different. Therefore, 21 different masks are applied in the refinement process. These masks turn out to simple wiring in hardware.
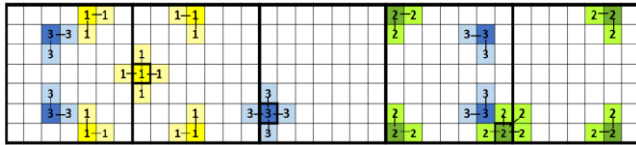

Figure 2. Examples for selecting 17 contributing pixels for 7×7, 13×13 and 25×25 window sizes during the disparity refinement process (yellow (1): 7×7, green (2): 13×13 and blue (3): 25×25).

3. HARDWARE IMPLEMENTATION

3.1 System Overview

The top-level block diagram of the proposed reconfigurable disparity estimation hardware and the required embedded system components for the realization of the full system are shown in Fig. 3. Since the main improvement of the proposed system relates to the Reconfigurable Disparity Map Estimation module, it is further explained in detail.

The Reconfigurable Disparity Map Estimation module involves 5 sub-modules and 62 dual port BRAMs. These five sub-modules are the Control Unit, Reconfigurable Data Allocation, Reconfigurable Computation of Metrics (RCM), Adaptive Disparity Selection (ADS) and Disparity Refinement. 31 of the 62 BRAMs are used to store 31 consecutive rows of the right image, and the remaining 31 BRAMs are used to store 31 rows of the left image.

External memory bandwidth is an important limitation for disparity estimation of HR images. Our proposed memory organization and data allocation scheme require reading each pixel only one time from the external memory during the search process.

3.2 Data Allocation and Disparity Voting

The block diagram of the Reconfigurable Data Allocation module is shown in Fig. 4. The data allocation module reads pixels from BRAMs, and depending on the processed rows, it rotates the rows using the Vertical Rotator to maintain the consecutive order.

The search process starts with reading the 31×31 size window of searched block from the BRAMs of the left image. Therefore, the Control Unit sends the *image select* signal to the multiplexers that are shown in Fig. 4 to select the BRAMs of the left image. While the window of searched block are loaded to the D flip flop (DFF) Array, the RCM computes and stores the 49 Census transforms, 49 *Shapes* and 9 windows pertaining to the pixels in the block for the computation of BW-SAD.

The Census transforms and windows of the candidate pixels in the right image are also needed for the matching process. After the computation of metrics for the 7×7 block, the Control Unit selects the pixels in the right image by changing the *image select* signal, and starts to read the pixels in the right image from the highest level of disparity by sending the address signals of the candidate pixels to the BRAMs. The disparity range can be configured by the user depending on the expected distance to the objects.

The detailed block diagram of the DFF Array and the Weaver are shown in Fig. 5. They are the units of the system that provide the configurability of the adaptive window size. As a terminology, we used the term "weaving" to mean "selecting 49 contributor pixels in different window sizes 7×7, 13×13 and 25×25 by skipping 1, 2 and 4 pixels respectively". Seven rows and one column are processed in parallel, and the processed pixels flow inside the DFF Array from the left to the right. Additionally, the weaving process is applied to the location (15, 8) of the DFF Array at the beginning of the search process only, to select the window size by computing the deviation of the center of the block from its neighbors for 7×7 and 13×13 windows.
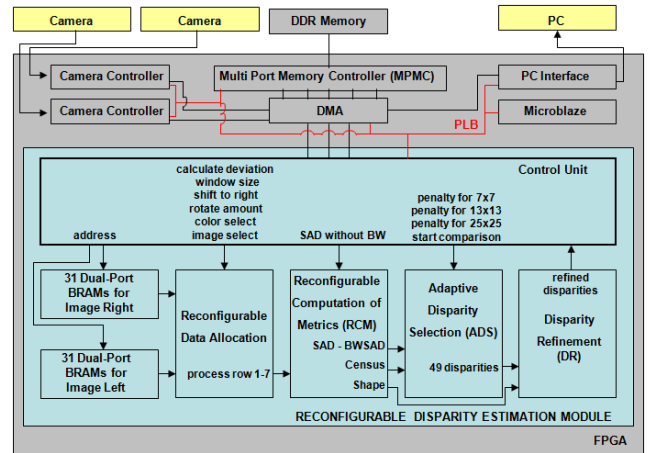

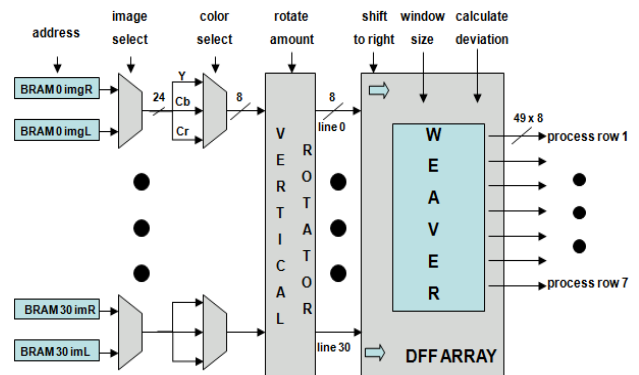**Figure 3. Top-Level Block Diagram of the System Architecture.**

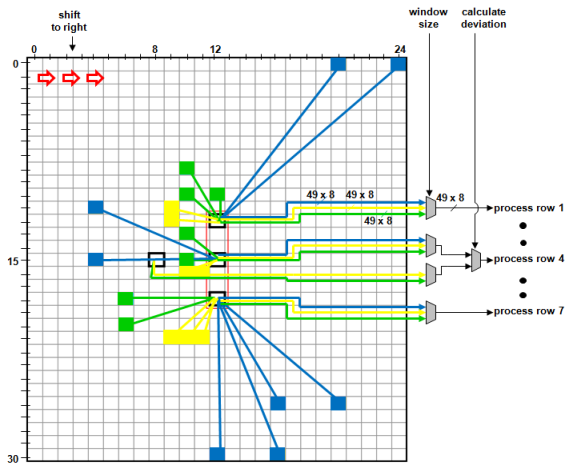**Figure 4. Reconfigurable Data Allocation Module.**

Figure 5. DFF Array and the Weaver (yellow: 7×7, green: 13×13 and blue: 25×25).

The DFF Array is a 31×25 array of 8-bit registers as shown in Fig. 5. While the pixels are shifting to the right, the Weaver is able to select the 49 components of the 7×7, 13×13 and 25×25 window sizes from the DFF Array with simple wiring and multiplexing architecture. Some of the contributor pixels of the windows for different window sizes are shown in Fig. 5 in different colors. The Weaver sends seven windows to be processed by RCM as *process row 1 – process row 7*, and each *process row* consists of 49 selected pixels.

A large window size normally involves high amount of pixels and thus requires more hardware resource and computational cost to support the matching process [11]. By using the proposed weaving architecture, even if the window size is changed, always 49 pixels are selected for the window. Therefore, the proposed hardware architecture is able to reach the largest window size (25×25) among the hardware architectures implemented for DE [4-9].

During the weaving process of the 49 pixels in the block and the candidate pixels in the right image, the RCM computes the Census and *Shape* of these pixels in a pipeline architecture. The block diagram of the RCM is shown in Fig. 6. In Fig. 6, the registers are named as "Shape $_{row\_column}$" and "Census $_{row\_column}$". Since the BW-SAD is only applied for 9 of the 49 pixels, the BW-SAD computation sub-modules are only implemented in process rows 2, 4 and 6. The computation of the Hamming distance requires significantly less hardware area than the BW-SAD. Therefore, the Hamming computation is used for all of the 49 pixels in a block.

As shown in Fig. 7, the proposed hardware searches 49 pixels in a block in parallel. While the proposed architecture computes the Hamming distance for the left-most pixels of the block, the Hamming for disparity **d**, rightmost pixels of the block computes their Hamming for disparity **d**+6. Therefore, the resulting Hamming costs are delayed in the ADS to synchronize the costs. This delay is also an issue of the BW-SAD results and they are also synchronized in the ADS.

The ADS module shifts the Hamming results of the candidate pixels depending on the 2's order adaptive penalty for the multiplication process as shown in formula (5). The ADS module adds the resulting Hamming penalty on the BW-SADs to compute *Hybrid Costs*. 49 comparators are used to select 49 disparity results that point minimum *Hybrid Costs*.
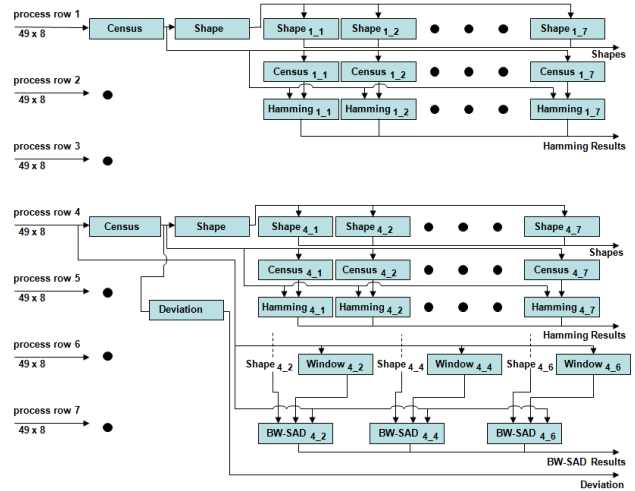

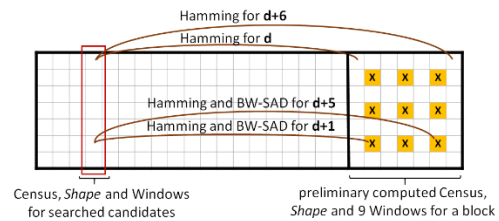**Figure 6. Reconfigurable Computation of Metrics.**

Figure 7. Processing Scheme ("x" indicates 9 selected pixels in a block for BW-SAD calculations).

3.3 Disparity Refinement

The DR module receives the 49 disparity results from the ADS and the *Shapes* of the 49 pixels of a block from the RCM and determines the final refined disparity values.

As presented in Fig. 8, after the ADS module has computed 49 disparity values in parallel, it loads this data with the respective *Shape* information in to the DFF Array of DR module (DR-Array). The DR-Array has a size of five blocks for the refinement process. DR-Array is designed to shift the disparity and *Shape* values from right to left to allocate data for the refinement processes.

The DR module involves seven identical Processing Elements (DR-PE). As presented in Fig. 8, DR-PEs are positioned to refine seven disparities in 15th column of DR Array in parallel while the disparity and shape values shift through the DR-Array. The hardware architecture of a single DR-PE is presented in Fig. 9.
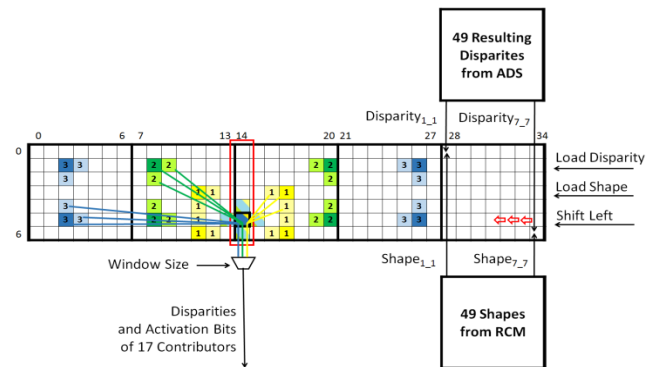

Figure 8. DR-Array of the Disparity Refinement Module (yellow (1): 7×7, green (2): 13×13 and blue (3):25×25).
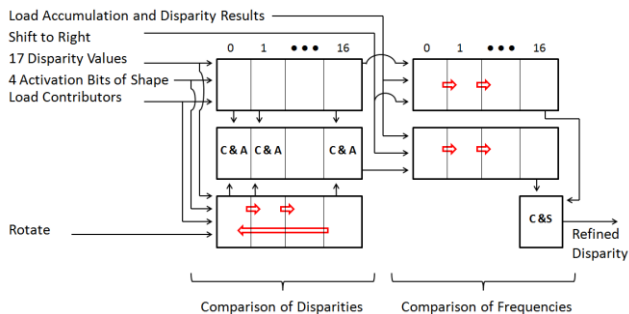
Figure 9. Processing Element of the Disparity Refinement Module.

In Fig. 8, while 17 disparity values are selected by the multiplexers, the *Shape* information corresponding to the four corners are also selected from the 48-bit shape information of the processed pixel. The selected 4-bits inform the DR-PE which of these 17 disparity values will be used while computing the highest frequency disparity. These 4 bits of the *Shape* are called *activation bits* in Fig. 9. Each activation bit activates itself together with its two adjacent disparities. The DR-PE uses shift arrays, 17 Compare and Accumulate (C&A) and 17 Compare and Select (C&S) sub-modules to select the disparity with the highest frequency as refined disparity.

4. IMPLEMENTATION RESULTS

The reconfigurable hardware architecture of the proposed AWDE algorithm is implemented using Verilog HDL, verified using Modelsim 6.6c. The Verilog RTL models are mapped to a Virtex-5 XCUVP-110T FPGA comprising 69k Look-Up Tables (LUT), 69k DFFs and 144 Block RAMs (BRAM). The proposed hardware consumes 59% of the LUTs, 51% of the DFF resources and 42% of the BRAM resources of the Virtex-5 FPGA. The proposed hardware operates at 190 MHz after place & route and computes the disparities of 49 pixels in 195 clock cycles for 120 pixel disparity range. Therefore, it can process 60 fps at a 768×1024 XGA video resolution.

The parameters of the AWDE algorithm are shown in Table 1. Parameters are selected by sweeping to obtain high quality DE of HR images considering different features.

Table 2 and Table 3 compare the disparity estimation performance and hardware implementation results of the AWDE architecture with other existing hardware implementations that targets HR [7-9] and currently the highest quality DE hardware that targets LR [4]. These papers do not provide the disparity estimation quality results for the HR benchmarks of the Middlebury data-set. Thus, we implemented [4], [7], and [9] in software, and the software implementation of [8] is obtained from the authors. The DE results for the Census and the BW-SAD metrics for different window sizes are also presented in Table 2. The comparisons of the resulting disparities with the ground-truths are done as prescribed by the Middlebury evaluation module. If the estimated disparity value is not within a ∓1 range of the ground truth, the disparity estimation of the respective pixel is considered as erroneous. 18 pixels located on the borders are neglected in the evaluation of LR benchmarks, and a disparity range of 30 is applied for all algorithms. 30 pixels located on the borders are neglected in the evaluation of HR benchmarks, and a disparity range of 120 is applied for all algorithms.

The Census and BW-SAD results that are shown in Table 2 are provided by sampling 49 pixels in a window. Although the

Census and the BW-SAD algorithms do not provide individually very efficient results, the combination of these algorithms into a reconfigurable hardware provides an efficient hybrid solution, as demonstrated from the AWDE results. If the sampling is not applied and all the pixels in a window are used during the matching process, the complexity of the AWDE algorithm increases by 12 times. The result of the high complexity version of the AWDE algorithm (AWDE-HC) is also provided in Table 2 for comparison. The AWDE-HC provides almost same quality results as the AWDE. Considering the hardware overhead, the low complexity version of the algorithm, AWDE, is selected for hardware implementation, and its efficient reconfigurable hardware is presented.

Table 1. Parameters of the AWDE

tr_{7x7}	tr_{13x13}	ap_{7x7}	ap_{13x13}	ap_{25x25}	$threshold_w$
5	2	32	16	4	8

Table 2. Disparity estimation performance comparisons

Algorithm	Error Rate (%)				
	Tsukuba 288 x384	Venus 383 x484	Aloe 1110 x1282	Art 1110 x1390	Clothes 1110 x1300
Chang [4]	4.15	0.56	3.75	12.80	2.97
Ttofis [8]	13.21	4.56	8.88	32.18	7.67
Greisen [9]	12.42	4.14	8.65	23.46	5.30
Georgoulas [7]	12.38	15.20	6.97	23.75	9.15
Census7	26.05	30.80	20.36	45.39	21.80
Census13	18.19	18.83	11.21	31.65	9.36
Census25	15.94	15.38	10.41	29.66	7.16
BWSAD7	12.19	19.45	8.31	34.03	13.33
BWSAD13	11.23	15.16	7.13	28.57	9.27
BWSAD25	10.43	11.12	6.74	24.74	6.28
AWDE	7.64	5.33	4.94	16.33	2.89
AWDE-HC	7.47	4.73	4.92	16.17	2.95

Table 3. Hardware performance comparisons

Hardware	Technology	Image Resolution	Disparity Range	fps	Clock Speed (MHz)
Chang[4]	ASIC-90nm	352×288	64	42	95
Ttofis[8]	Virtex-5	1280×1024	120	50	100
Greis.[9]	Stratix-III	1920×1080	256	30	130
Geor.[7]	Stratix-IV	800×600	80	550	511
Proposed (AWDE)	Virtex-5	1024×768	120	60	190
		640×480	60	224	
		352×288	60	680	

The algorithm presented in [4] uses the Census algorithm with the cost aggregation method, and provides the best results for both LR and HR stereo images except the HR benchmark Clothes. As shown in Table 3, due to the high-complexity of cost aggregation, it only reaches 42 fps for CIF images, thereby consuming a large amount of hardware resource. If the performance of [4] is scaled to 1024×768 for disparity range of 120, less than 3 fps can be achieved.

None of the compared algorithms that have a real-time HR hardware implementation [7-9] is able to exceed the DE quality of AWDE for HR images. The overall best results following the results of AWDE are obtained from [9]. The hardware presented in [9] consumes 20% of the 270k Adaptive LUT (ALUT) resources of a Stratix-III. It provides high disparity range due to its hierarchical structure. However, this structure easily causes faulty computations when the disparity selection finds wrong matches in low resolution.

The hardware implementation of [7] provides the highest speed performance in our comparison. However this hardware applies 480 SAD computations for a 7×7 window in parallel. The hardware presented in [7] consumes %60 of the 244k ALUT resources of a Stratix-IV FPGA. In our hardware implementation we only use 9 SAD computations in parallel for the same size window and this module consumes 16% of the resources of Virtex-5 FPGA on its own. Therefore, the hardware proposed in [7] may not fit in to 3 Virtex-5 FPGAs.

The visual results of the AWDE algorithm for the HR benchmarks Clothes, Art and Aloe are shown in Fig. 10 (a-f). The 1024×768 resolution disparity map result of the AWDE algorithm for the pictures taken by our stereo camera system is shown in Fig. 10 (g-h). Our hardware architecture provides both quantitative and visual satisfactory results and reaches real-time for HR.
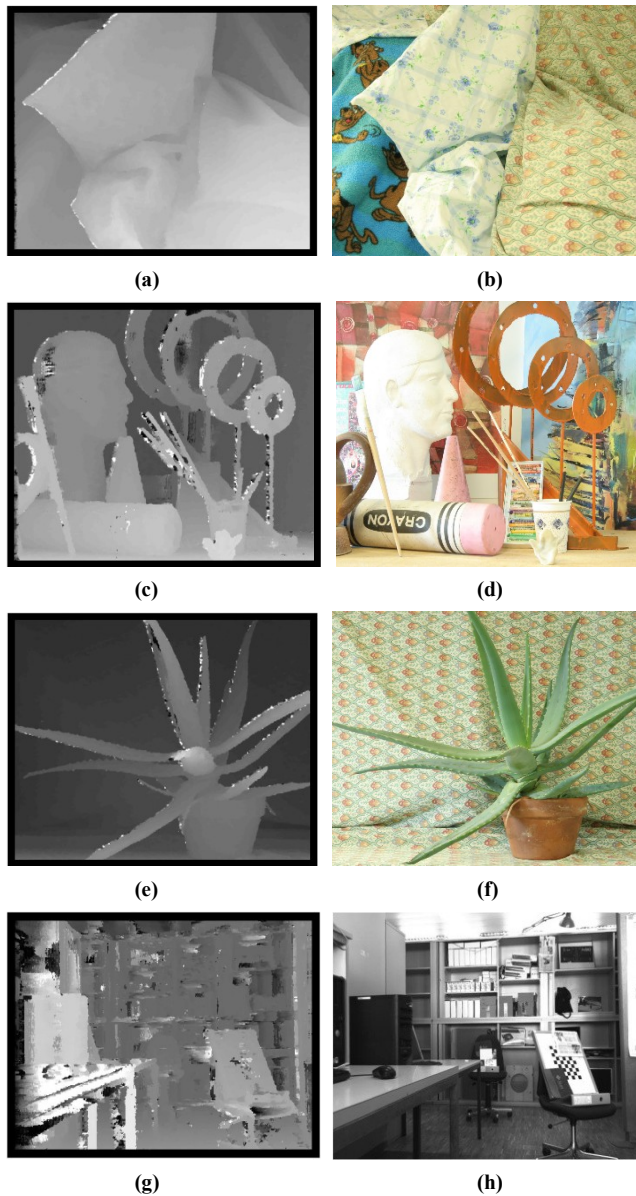

(a) **(b)**

(c) **(d)**

(e) **(f)**

(g) **(h)**

Figure 10. Visual disparity estimation results of AWDE algorithm for HR benchmarks (a-b) Clothes, (c-d) Art, (e-f) Aloe, and (g-h) LSM lab.

5. CONCLUSION

In this paper, a hardware-oriented adaptive window size disparity estimation algorithm and its real-time reconfigurable hardware implementation are presented. The proposed AWDE algorithm dynamically adapts the window size considering the local texture of the image to increase the disparity estimation quality. Currently, the AWDE algorithm and its real-time hardware implementation reach the highest DE quality compared to existing real-time DE hardware implementations for HR images. The proposed reconfigurable hardware can process 60 fps on Virtex-5 FPGA at a 1024×768 XGA video resolution for 120 pixel disparity range. The AWDE algorithm and its reconfigurable hardware can be used in consumer electronic products where high-quality real-time disparity estimation is needed for HR video.

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] D. Scharstein, and R. Szeliski. 2002. A Taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, (Apr. 2002), 7–42.

[2] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. 2011. On building an accurate stereo matching system on graphics hardware. *In Proc. of GPUCV*, (Nov. 2011).

[3] A. Klaus, M. Sormann, and K. Kamer. 2006. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. *In Proc. ICPR*, 3 (2006).

[4] N. C. Chang, T. H. Tsai, B. H. Hsu, Y. C. Chen, and T. S. Chang. 2010. Algorithm and architecture of disparity estimation with mini-census adaptive support weight. *IEEE Trans. on Circuits and Systems for Video Technology*, 20, (Jun. 2010), 792-805.

[5] S. Jin, J. Cho, X. D. Pham, K. M. Lee, S. K. Park, M. Kim, and J. W. Jeon. 2010. FPGA design and implementation of a real-time stereo vision system. *IEEE Trans. on CSVT*, (Jan. 2010), 15 - 26.

[6] S. H. Lee, and S. Sharma. 2011. Real-time disparity estimation algorithm for stereo camera systems. *IEEE Trans. on Consumer Electronics*, 57, (Aug. 2011).

[7] C. Georgoulas, and I. Andreadis. 2009. A Real-Time Occlusion Aware Hardware Structure for Disparity Map Computation. *ICIAP*, (2009), 721-730.

[8] C. Ttofis, S. Hadjitheophanous, A. S. Georghiades, and T. Theocharides. 2012. Edge-directed hardware architecture for realtime disparity map computation. *In Proceedings of the IEEE Transactions on Computers*, (Jan. 2012).

[9] P. Greisen, S. Heinzle, M. Gross, and A. P. Burg. 2011. An FPGA-based Processing Pipeline for High-Definition Stereo Video. *EURASIP Journal on Image and Video Processing*, (Nov. 2011).

[10] R. Zabih, and J. Woodfill. 1994. Non-parametric local transforms for computing visual correspondence. *In Proc. 3rd Eur. Conf. Comput. Vision*, 2, (1994), 151–158.

[11] A. Motten, and L. Claesen. 2010. A binary adaptable window SoC architecture for a stereo based depth field processor. *In Proceedings of the IEEE VLSISOC*, (Sep. 2010), 25 - 30.