



HAL
open science

An FPGA Correlation-Edge Distance approach for disparity map

Abiel Aguilar-González, Madain Perez-Patricio, Miguel Arias-Estrada,
Jorge-Luis Camas-Anzueto, Héctor-Ricardo Hernández-de León, Avisáí
Sánchez-Alegría

► **To cite this version:**

Abiel Aguilar-González, Madain Perez-Patricio, Miguel Arias-Estrada, Jorge-Luis Camas-Anzueto, Héctor-Ricardo Hernández-de León, et al.. An FPGA Correlation-Edge Distance approach for disparity map. 2015 International Conference on Electronics, Communications and Computers (CONIELECOMP), Feb 2015, Cholula, Mexico. pp.21 - 28, 10.1109/CONIELECOMP.2015.7086952 . hal-01627287

HAL Id: hal-01627287

<https://hal.science/hal-01627287>

Submitted on 31 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An FPGA Correlation-Edge Distance Approach for Disparity Map

Abiel Aguilar-González*, Madaín Pérez-Patricio*, Miguel Arias-Estrada**,
Jorge-Luis Camas-Anzueto*, Héctor-Ricardo Hernández-de León*, Avisai Sánchez-Alegría*

*Instituto Tecnológico de Tuxtla Gutiérrez, Tuxtla Gutiérrez, Chiapas, México

13270869@ittuxtlagutierrez.edu.mx, {mperez, jcamas, hhernandezd}@ittg.edu.mx, 08270538@ittuxtlagutierrez.edu.mx

**Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonantzintla, Puebla, México
ariasmo@inaoep.mx

Abstract—This paper describes an FPGA Correlation-Edge Distance approach for real time disparity map generation in stereo-vision. The proposed method calculates the disparity map for the input and disparity map for Edge Distance images of a stereopair. In both cases the approximation algorithm of disparity map SAD (Sum of Absolute Differences) is used. The final disparity map is determined from the previously generated maps, considering a homogeneity parameter defined for each point in the scene. Due to low complexity when implementing stereo-vision algorithms in FPGA devices, the proposed method was implemented in a Cyclone II EP2C35F672C6 FPGA assembled in an Altera DE2 breadboard. The developed module can process stereo-pairs of 1280×1024 pixel resolution at a rate of 75 frames/s and produces 8-bit dense disparity maps within a range of disparities up to 63 pixels. The presented architecture provides a significant improvement in regions with uniformed texture over correlation based stereo-vision algorithms in the reported literature and an accelerated processing rate.

Keywords—FPGA, Verilog, Disparity map.

I. INTRODUCTION

Perception of depth values of points contained in a scene is one important task of computer vision systems. Recently depth perception has been used in diverse applications such as navigation systems for mobile robots, object recognition and 3D reconstruction [1]-[4]. Extracting information about the depth from images using a stereo configuration is the most used technique. In this technique the correspondence between images and the geometric configuration of the same allows to obtain depth images called disparity maps [5].

To define a disparity map it is necessary to measure the similarity of content points in the stereo image. Techniques to determine these similarities are divided into two categories: area-based algorithms [6]-[7], and feature-based algorithms [8]-[9]. In area-based algorithms the gray level of pixels round the interest pixel are used as similarity measure to produce dense disparity maps, i.e., the disparity is calculated for all the points in the scene. On the other hand, feature-based algorithms are based on specific points of interest. These points are selected in concordance with appropriate feature detectors. Feature-based algorithms are more stable against changes in lighting environment and contrast, because they represent the geometric properties of a scene. The main characteristic of using feature-based algorithms is that they do not generate dense disparity maps. Therefore, these algorithms must be applied in conjunction with other techniques and require an

additional step for characteristics extraction, which increases the computational costs and runtime.

Due to benefits regarding to the management of large amounts of data at high speed of FPGAs devices, there is currently a wide variety of algorithms for estimating disparity maps implemented in FPGAs in the reported literature. In [10], an array of four FPGAs is used to estimate the Cross Correlation for 256×256 pixel resolution images at a rate of 7 frames per second. In [11], a hybrid system that uses digital signal processors with programmable logic devices PLD is presented. The authors of this paper generated disparity maps for 256×256 pixel resolution images at a rate of 30 frames per second. In [12], is proposed a four Xilinx Virtex 2000E structure, on which is possible to obtain real-time dense disparity-maps for 256×360 pixel resolution images at a rate of 40 frames per second. In [13] the use of a single FPGA is proposed. The developed system processes images at 30 frames per second using 640×480 pixel resolution images. An adaptive window technique in conjunction with SAD is used in [14]. The presented method processes images up to 1024×1024 pixel resolution at a rate of 47 frames/s and produces 8-bit dense disparity maps within a range of disparities up to 32 pixels. The architecture presented in [15] uses four FPGAs to make a real time correction, in on the same paper a leftright consistency check is done to improve the quality of the disparity map produced. This method processes images up to 640×480 pixel resolution at a rate of 30 frames/s and produces 8-bit dense disparity maps within a range of disparities up to 128 pixels. In [16] a module for calculating the real-time disparity map is proposed. The module was implemented in a single FPGA of Altera Stratix IV family. The autors of this paper processed images up to 640×480 pixel resolution at a rate of 320 frames/s and produces 8-bit dense disparity maps within a range of disparities up to 80 pixels.

In this paper an FPGA module to calculate real-time dense disparity maps is presented. The novelty is the architecture design and FPGA implementation for the proposed method. The disparity maps are calculated at a rate of 75 frames/s for 1280×1024 pixel resolution images and generate 8-bit dense disparity maps within a range of disparities up to 63 pixels. The developed module allows simple and systematically scalability to different range of disparities, therefore the resulting hardware could be applied to a wide range of real-time stereo-vision applications such as high-speed tracking, tracking paths, high-speed objects recognition and mobile robot navigation.

II. THE PROPOSED METHOD

An overview of the proposed method can be seen in Fig. 1. This method consists in two steps. First, disparity maps for the input images and edge-distance images are computed, simultaneously an homogeneity parameter ψ is calculated for each point in the scene. Followed, based on the disparity maps previously generated and using the ψ parameter, a final disparity map is generated.

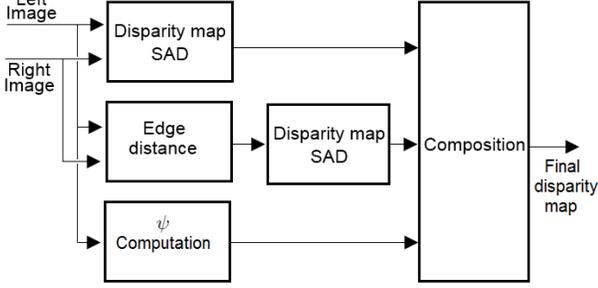


Fig. 1: Block diagram of the proposed method.

A. Sum of absolute differences (SAD)

The Sum of Absolute Differences (SAD) is a correlation-based method mostly used due to its high computational efficiency. The general behavior can be described as following, given (x, y) coordinates of a pixel in left image and maximum value of expected disparity d_{\max} , a correlation index $\text{Crl}(x, y, s)$ is calculated for each displacement S of the correlation window in right image. To calculate the correlation the following equation is used:

$$\text{Crl}(x, y, s) = \sum_{u=-w, v=-w}^{u=w, v=w} |I_l(x+u, y+v) - I_r(x+u+s, y+v)| \quad (1)$$

as $2w+1$ is the window size centered on the pixel with position (x, y) . I_l, I_r are the gray values of the pixels in the left and right images respectively and S ranges from 0 to d_{\max} . The disparity $d(x, y)$ is defined as S displacement that minimizes the correlation index:

$$d(x, y) = \arg \min_s \text{Crl}(x, y, s) \quad (2)$$

The main problem with this method is to select the correlation window size. High window size values allow to determine the true correlation values in areas with uniform texture, however this window size values imply a high computational demand and erroneous values at certain points due to the blurring edges and small features eliminated (see Fig. 3). On the other hand, small window size values imply low computational demand but the correlation coefficient measurement is very sensitive to noise and erroneous values at uniform texture regions are generated as seen in Fig. 4. Fig. 2 shows the true disparity map of a Tsukuba scene.



Fig. 2: Tsukuba scene, true disparity map.



Fig. 3: Tsukuba scene, disparity map SAD $w = 15$.



Fig. 4: Tsukuba scene, disparity map SAD $w = 1$.

B. Edge distance

The euclidean distance between each pixel with $I_l(x, y), I_r(x, y)$ coordinates, and the nearest left edge is calculated as following:

$$k(x, y) = |I_\delta(x, y) - I_\delta(x-1, y)| \quad (3)$$

$$\text{distance}(x, y) = \begin{cases} l = 0, & k(x, y) < \beta \\ l = l + 1, & k(x, y) > \beta \end{cases} \quad (4)$$

as β is the threshold value that defines an edge and δ is l or r for left or right images, respectively.

Fig. 5 shows the nearest left edge distance for each point of the left image of Tsukuba scene. The darker values represent low value distances while light values represent high value distances. Fig. 6 shows the disparity map obtained using images of edge distance as input images for SAD. In this figure a significant improvement in regions with uniform texture is perceived, on the other hand, an increase of errors in regions with uneven texture is generated.



Fig. 5: Left image of Tsukuba scene, distance to nearest left edge.



Fig. 6: Tsukuba edge distance scene, edge disparity map SAD $w = 1$.

C. Homogeneity parameter

A ψ parameter corresponding to the homogeneity degree for each point of the left image of stereo pair is determined as following:

$$h(x, y) = \sum_{u=-w, v=-w}^{u=w, v=w} |I_l(x + u, y + v)| \quad (5)$$

$$\psi(x, y) = \begin{cases} 0, & h(x, y)/(w + 1)^2 < \lambda \\ 1, & h(x, y)/(w + 1)^2 > \lambda \end{cases} \quad (6)$$

as λ is threshold value that determines the homogeneity of a point with respect to its corresponding correlation window and $2w + 1$ is the window size centered on the pixel with position (x, y) . Fig. 7 shows ψ values for all points contained in the Tsukuba scene considering $\lambda = 1$.



Fig. 7: Tsukuba scene, ψ parameter.

D. Composition

Using the ψ parameter it is possible to determine a final disparity map by assigning the values obtained from the edge disparity map (Fig. 6) for points with uniform texture and the values obtained from the disparity map (Fig. 4) for points with uneven texture as following:

$$\text{disparity}(x, y) = \begin{cases} \text{distance}(x, y), & \psi(x, y) == 0, \sigma == 1 \\ d(x, y), & \psi(x, y) == 1 \end{cases} \quad (7)$$

as σ is determined as following: $\sigma : 0 < \text{disparity}(x, y) > d_{\max}$.

Fig. 8 shown the disparity maps generated by the proposed method for the Tsukuba scene where w, β and λ parameters were configured with values equal to $\{1, 32, 1\}$ respectively. In this figure a high improvement in regions of uniform texture and a low improvement in the points near the edges is obtained.

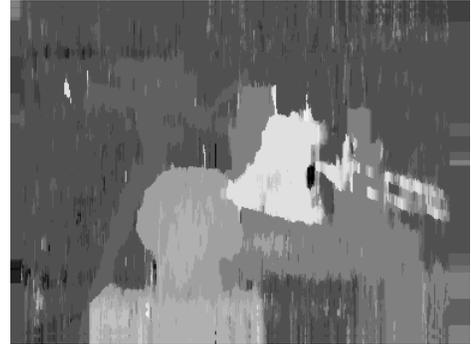


Fig. 8: Tsukuba scene, disparity map using the proposed method $w = 1$.

The proposed method requires less computational load in contrast to various methods in the reported literature [7]-[8], [15]-[16], however compute a disparity map for a stereo pair of 384×288 pixel resolution (Tsukuba scene resolution) implies a runtime close to 1 second. These runtime values are not acceptable for real-time applications. This was the main motivation to search efficient ways to implement the developed method, opted for an FPGA implementation. In Section III the detailed implementation in hardware is presented while in section IV the experimental results generated by this implementation are shown.

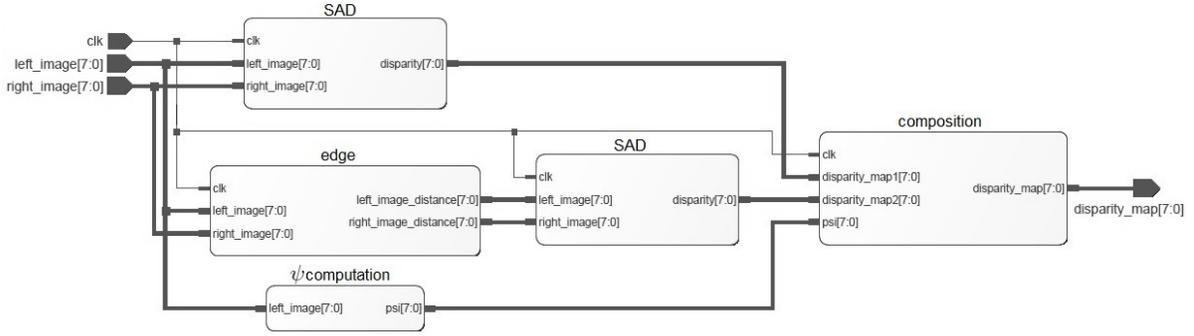


Fig. 9: General diagram of the developed architecture.

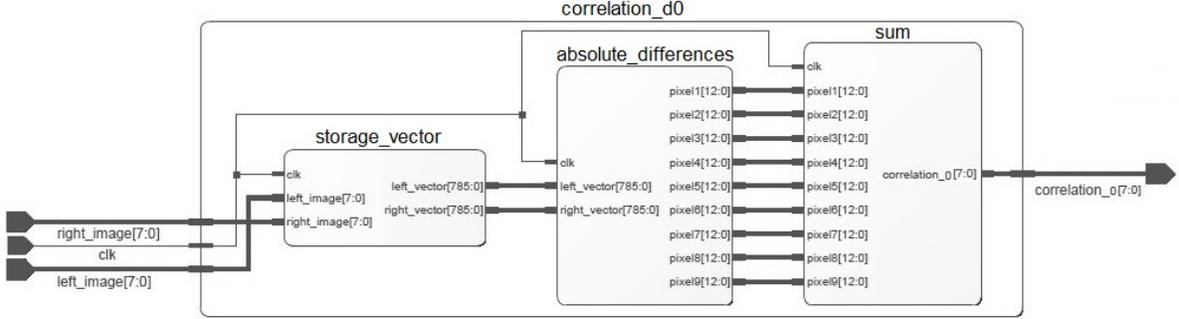


Fig. 10: Correlation_computation module architecture, disparity=0.

III. FPGA IMPLEMENTATION

The main advantages of using FPGA devices are: easiness to re-designed architectures based on specifications without incurring in non-recurring engineering and parallel processing ability, allowing real-time processing. Due to these advantages the proposed method was implement in a single Cyclone II EP2C35F672C6 FPGA assembled in an Altera DE2 bread-board. A 50 MHz clock frequency was used. In order to get input stereo pairs, TRDB_DC2 plate connected in 1st expansion port of DE2 plate is used and provides stereo pairs to 1280×1024 pixel resolution in RGB scale. On the other hand, to determine the grayscale value of input stereo pairs the green channel value are used as grayscale value. The developed architecture is described in detail in the following sub-sections. A general diagram of the designed architecture can be seen in Fig. 9. In TABLE I a full description of the physical ports used by proposed architecture is shown.

TABLE I: PORTS DESCRIPTION OF THE DEVELOPED ARCHITECTURE

Name	Type	Description
Clk	Input	Pixel clock
Left image	Input	Logical vector [7 : 0]
Right image	Input	Logical vector [7 : 0]
Disparity map	Output	Logical vector [7 : 0]

A. Sum of absolute differences (SAD)

To calculate the disparity map by SAD it is necessary to have stored fragments of images to be processed. In order to manage the FPGA memory two vectors with capacity to store three rows of left image and three rows of right

image respectively were used. The behavior of these vectors is similar to a shift register unit, however, due simplicity of these vectors hardware resources demand is reduced. In general terms, during time zero the pixel value (1, 1) is stored in index vector 0, a clock cycle later this value is crossed to index 1 and the pixel (1, 2) is stored in index 0. A similar process is repeated for all pixels making up the image.

To calculate the SAD, a 3×3 correlation window and maximum value of expected disparity $d_{max} = 63$ is used. An architecture of pixel-parallel window-parallel was designed. The inputs are obtained from the vectors previously configured, using appropriate indexes is possible to process the image at video rate. SAD calculation starts after storing the first three rows of each image. This results in a $(x - w) * (y - w)$ disparity map resolution, as x, y is the resolution values of input image and $2w + 1$ is the correlation window size. Fig. 10 shows the architecture for correlation calculation to a disparity value equal to zero. To calculate the correlation values for the remaining disparities similar architectures implemented in parallel were used.

To determine the appropriate disparity value, the **min** module is used (Fig. 11). First, this module uses the previously calculated correlation values and using an array of multiplexers activated by comparators as $Sel : I1 > I2$ determines the correlation value which minimizes the correlation window displacement. After, a CASE structure is used to assigned the corresponding correlation index CrI_{index} , to determining the disparity value an equalizer is applied to CrI_{index} as follows:

$$disparity = CrI_{index} * (255/d_{max}) \quad (8)$$

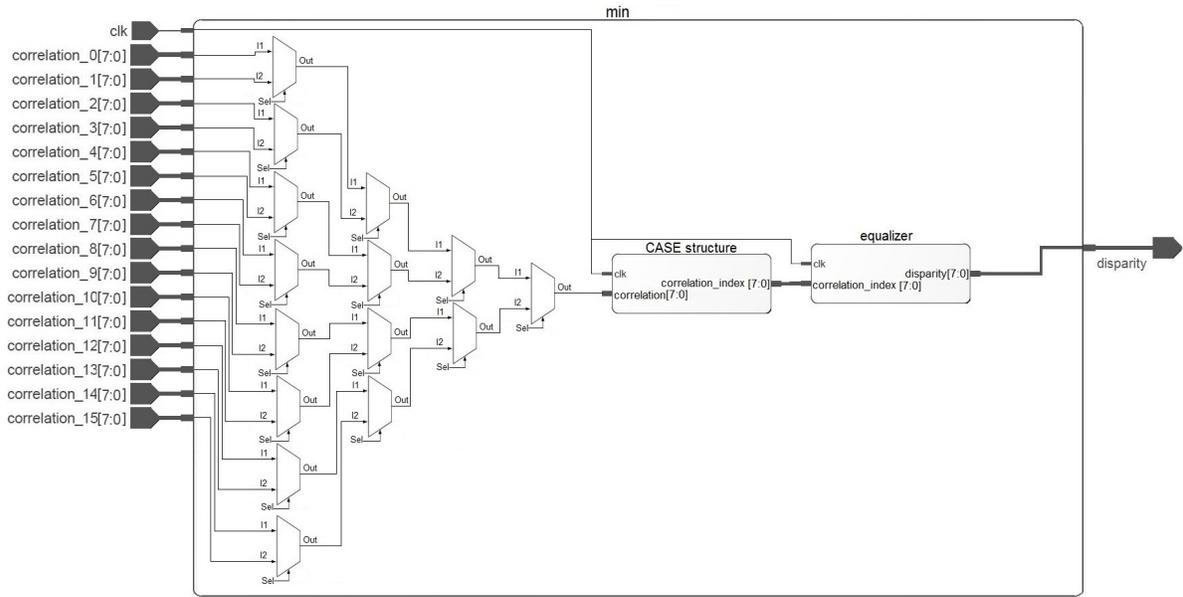


Fig. 11: **min** module architecture.

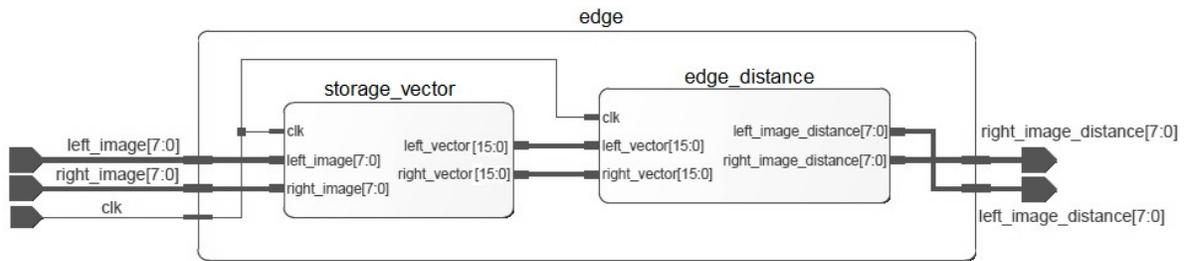


Fig. 12: **edge** module architecture.

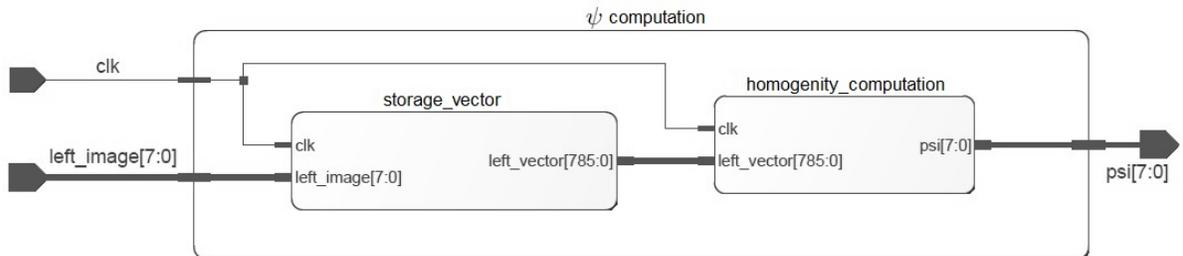


Fig. 13: ψ **computation** module architecture.

B. Edge distance

Fig. 12 shows the architecture for nearest left edge distance calculation. First, the **storage_vector** module stores pixels with $(x - 1, y)$ and (x, y) coordinates for the left image and right images, then, applying equations 3 and 4 nearest left edge distance is calculated in **edge_distance** module.

C. Homogeneity parameter

Fig. 13 shows the architecture for calculating the ψ parameter. The **vector_storage** module stores three lines of the left image of the estero pair. Then, using the stored lines and

applying equations 5 and 6, the **homogeneity_computation** module determines the correlation homogeneity window centered in the pixel of interest and assigns the corresponding ψ value.

D. Composition

The composition module operates on the principle of a conventional multiplexer assigning for each point (x, y) the value of one of its two inputs determined by the **Sel** parameter see (Fig. 14).

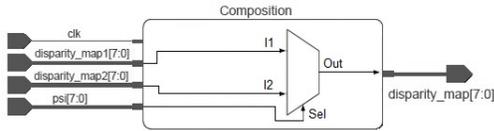


Fig. 14: **Composition** module architecture.

Finally, generated disparity maps are shown on a 4.3” LCD screen of terasIC with 800×480 pixels resolution connected to the 2nd expansion port of the DE2 board.

IV. EXPERIMENTAL RESULTS

The architecture presented in Section III was implemented using a top-down approach. All modules were coded in Verilog and were simulated using ModelSim-Altera 6.6c to verify its functionality. Quartus II Web Edition SP1 version 10.1 was used for the synthesis and download in a Cyclone II EP2C35F672C6 FPGA assembled in Altera DE2 breadboard. Resource consumption of the developed architecture is shown in TABLE II.

TABLE II: DEVELOPED ARCHITECTURE SPECIFICATIONS

Resource	Demand
Logic elements (LEs)	16,240
Memory bits (ALUTs)	33,080

In order to evaluate the proposed method performance, the developed architecture was tested using different values for β and λ parameters. Tests were performed using the Tsukuba scene as test images. Whereas hardware implementation operates using binary strings and color depth equal to 8 bits, values with base 2 were tested. Fig. 15 shows the behavior of the error obtained in the final disparity map for $\beta = \{1, 8, 16, 32, 64, 128, 255\}$ and $\lambda = 1$. Fig. 16 shows the behavior of the error obtained in the final disparity map for $\lambda = \{1, 8, 16, 32, 64, 128, 255\}$ and $\beta = 1$. To determine the number of erroneous pixels, the generated disparity maps were evaluated using the Middlebury stereo vision system web site [21]

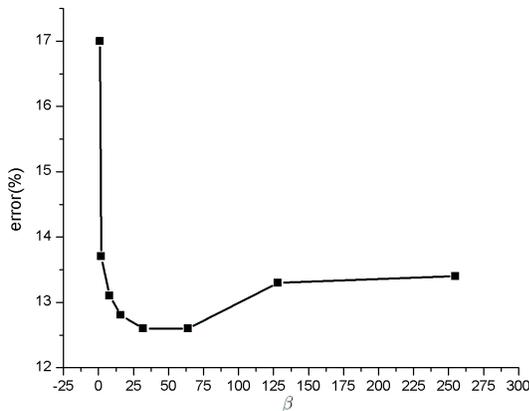


Fig. 15: β parameter behavior.

In TABLE III, quantitative results for the proposed method with a 3×3 correlation window compared with other methods

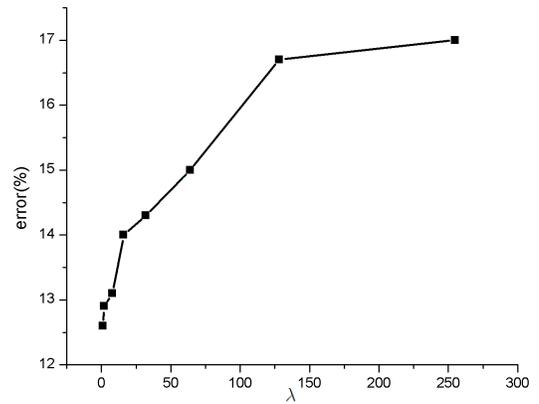


Fig. 16: λ parameter behavior.

in the literature [17] are presented, **occ** is the error for points of occlusion in the input images, **disc** is the error for points with discontinuities and **all** is the error for total points in the scene. The shown values were obtained from the Middlebury stereo vision system evaluation [21] with following Image Settings: Tsukuba (384×288 resolution, $d_{\max} = 15$), Venus (434×383 resolution, $d_{\max} = 19$), Teddy (450×375 resolution, $d_{\max} = 59$) Cones (450×375 resolution, $d_{\max} = 59$) and performing re-designs in developed architecture to operate with d_{\max} values equal to $\{15, 31, 63, 63\}$ respectively. For all test images w , β and λ parameters were configured values equal to $\{1, 32, 1\}$ respectively, in Fig. 17 disparity maps generated by the proposed method for the same test scenes is shown. In TABLE IV the speed of the developed architecture compared with other FPGAs implementations reported in the literature are presented. In TABLE V hardware resources demand compared with other FPGAs implementations reported in the literature are shown.

TABLE III: QUANTITATIVE RESULTS OF THE PROPOSED METHOD COMPARED TO OTHER METHODS IN THE LITERATURE

Method	Scene	occ	disc	all
Sum of Absolute Differences (SAD)	Tsukuba	21.8%	23.6%	36.9%
	Venus	26.2%	26.8%	42.1%
	Teddy	52.6%	54.0%	68.5%
	Cones	62.5%	63.3%	76.9%
Sum of Squared Differences (SSD)	Tsukuba	21.3%	23.2%	36.5%
	Venus	25.0%	25.6%	41.8%
	Teddy	56.2%	53.2%	67.8%
Normalized Cross-Correlation (NCC)	Tsukuba	29.9%	31.1%	42.5%
	Venus	30.6%	31.1%	44.1%
	Teddy	55.8%	57.5%	69.2%
Census	Tsukuba	34.9%	336.3%	46.5%
	Venus	44.4%	44.8%	53.0%
	Teddy	69.7%	70.2%	69.2%
	Cones	77.5%	77.6%	84.7%
Proposed method	Tsukuba	8.85%	34.0%	10.9%
	Venus	6.99%	35.9%	6.93%
	Teddy	30.2%	47.8%	34.1%
	Cones	43.4%	55.9%	46.5%

By analyzing TABLE III a significant improvement over other methods reported in the literature is observed. This



(a) Tsukuba scene



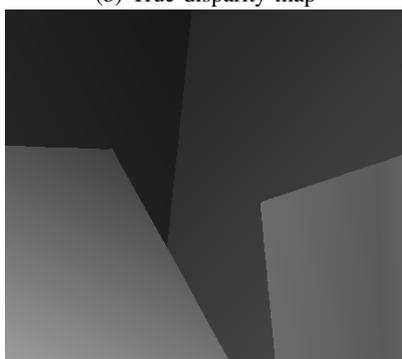
(b) True disparity map



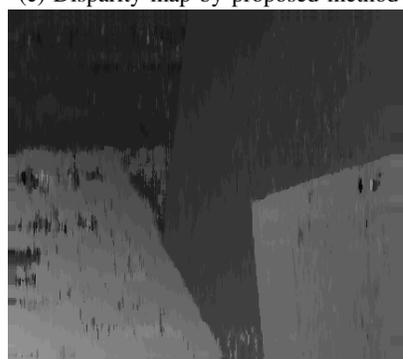
(c) Disparity map by proposed method



(d) Venus scene



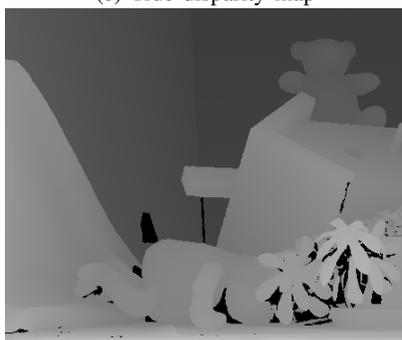
(e) True disparity map



(f) Disparity map by proposed method



(g) Cones scene



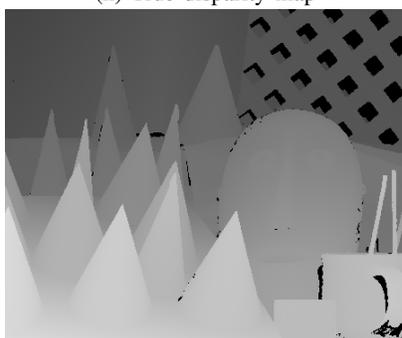
(h) True disparity map



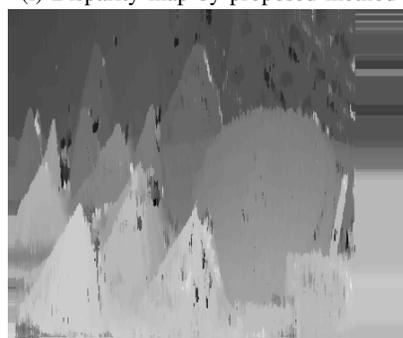
(i) Disparity map by proposed method



(j) Teddy scene



(k) True disparity map



(l) Disparity map by proposed method

Fig. 17: Disparity maps generated by the proposed method for different test images.

TABLE IV: PROCESSING SPEED COMPARED WITH OTHER FPGAs IMPLEMENTATIONS.

Method	Resolution	Frames/s	Pixels/s
[16]	1024X1024	102	106,954,752
[18]	1280×1024	65	85,196,800
[19]	640×480	230	70,656,000
[20]	1280×1024	50	65,536,000
Proposed*	1280×1024	75	98,304,000

*Operating frequency = 50 MHz

TABLE V: HARDWARE RESOURCE DEMAND COMPARED WITH OTHER FPGAs IMPLEMENTATIONS.

Method	Logic elements (LEs)	Memory bits (ALUTs)
[16]	86,252	62,669
[18]	89,459	84,307
[19]	53,616	60,598
[20]	31,863	47,331
Proposed	12,765	24,048

improvement is due to the left-edge distance that like feature-based methods represents the geometries of the input scene.

Due to mathematical simplicity of the proposed method, in the developed architecture a high processing speed is presented. Comparing the processing speed of the developed architecture with other FPGA implementations reported in the literature (TABLE IV) up to 32,768,000 pixels/s increase is observed.

Due to strategies such implementation of storage vectors and small correlation windows, the developed architecture has a low hardware resources demand (TABLE V). When comparing the consumption of logic elements and memory bits with other methods reported in the literature up to 76,694 and 60,259 reduction was observed.

V. CONCLUSIONS

In this paper a module for real-time disparity maps computation has been presented. The developed architecture shows a better performance in regions of uniform texture regarding to other methods mentioned in the literature. The principal advantage with the developed module it's a high processing speed and a low consumption of hardware resources, it allows to implement the proposed method in FPGA devices with relatively few resources which facilitates its application in real-time stereo vision applications.

Besides, one of the main characteristic of the developed architecture is its flexibility to be reconfigured or modified to work with different windows sizes and different maximum values of expected disparity.

REFERENCES

[1] E. Parrilla, J.-R. Torregrosa, J. Riera, and J.-L. Hueso, "Fuzzy control for obstacle detection in stereo video sequences," *Mathematical and Computer Modelling*, vol. 54, pp. 1813–1817, 01, 2011.

[2] X. Rong, J. Huanyu, and Y. Yibin, "Recognition of clustered tomatoes based on binocular stereo vision," *Computers and Electronics in Agriculture*, vol. 106, pp. 75–90, 05, 2014.

[3] R. Correal, G. Pajares, and J. Ruz, "Automatic expert system for 3d terrain reconstruction based on stereo vision and histogram matching," *Expert Systems with Applications*, vol. 106, pp. 75–90, 09, 2013.

[4] J. A. Delmerico, P. Davidb, and J. J. Corso, "Building facade detection, segmentation, and parameter estimation for mobile robot stereo vision," *Image and Vision Computing*, vol. 31, pp. 841–852, 08, 2013.

[5] O. Faugeras, *Three Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA: MIT Press, 1993.

[6] L. D. Stefano, M. Marchionni, and S. Mattocchia, "A fast area-based stereo matching algorithm," *Image and Vision Computing*, vol. 22, pp. 983–1005, 03, 2004.

[7] M. Jin and T. Maruyama, "A fast and high quality stereo matching algorithm on fpga," *Field Programmable Logic and Applications (FPL), International Conference on*, vol. 22, pp. 507–5010, 08, 2012.

[8] FeiyangCheng, HongZhang, DingYuan, and MinguiSun, "Stereo matching by using the global edge constraint," *Neurocomputing*, vol. 131, pp. 217–226, 11, 2013.

[9] H. Y. Jung, H. Park, I. K. Park, K. M. Lee, and S. U. Lee, "Stereo reconstruction using high-order likelihoods," *Computer Vision and Image Understanding*, vol. 125, pp. 223–236, 04, 2014.

[10] O. Faugeras, B. Hotz, H. Mathieu, T. Vieville, P. F. Z. Zhang, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy, "Real time correlation-based stereo: algorithm, implementations and applications," *Technical Report Research Report, INRIA Sophia Antipolis*, 08, 2013.

[11] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, "A stereo machine for videorate dense depth mapping and its new applications," *IEEE Computer Vision & Pattern Recognition Conference*, vol. 15, pp. 196–202, 06, 1996.

[12] A. Darabiha, W. MacLean, and J. Rose, "Reconfigurable hardware implementation of a phase-correlation stereo algorithm," *Journal of Machine Vision and Applications*, vol. 17, pp. 116–132, 03, 2006.

[13] J. Diaz, E. Ros, F. Pelayo, E. Ortigosa, and S. Mota, "Fpga based real-time opticalflow system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 274–279, 02, 2006.

[14] C. Roh, T. Ha, S. Kim, and J. Kim, "Symmetrical dense disparity estimation: algorithms and fpgas implementation," *IEEE International Symposium on Consumer Electronics*, pp. 452–456, 08, 2004.

[15] D. Masrani and W. MacLean, "A real-time large disparity range stereo-system using fpgas," *IEEE International Conference on Computer Vision Systems*, pp. 13–19, 01, 2006.

[16] C. Georgoulas and I. Andreadis, "Fpga based disparity map computation with vergence control," *Microprocessors and Microsystems*, vol. 34, pp. 259–273, 06, 2010.

[17] M.-Z. Brown, D. Burschka and G.-D. Hager, "Advances in Computational Stereo," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 993–1008, 08, 2003.

[18] C. Georgoulas, L. Kotoulas, G. C. Sirakoulis, I. Andreadis, and A. Gasteratos, "Real-time disparity map computation module," *Microprocessors and Microsystems*, vol. 32, pp. 159–170, 10, 2008.

[19] S. Jin, J. Cho, X.-D. Pham, K.-M. Lee, S.-K. Park, M. Kim and J.-W. Jeon, "Real-time disparity map computation module," *Microprocessors and Microsystems*, vol. 32, pp. 159–170, 10, 2008.

[20] C. Ttofis, S. Hadjitheophanous, A.-S. Georghiadis, and T. Theocharides, "FPGA Design and Implementation of a Real-Time Stereo Vision System," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, vol. 20, pp. 15–26, 01, 2010.

[21] Middlebury Stereo Vision Page. <<http://vision.middlebury.edu/stereo/>>.