An FPGA-CAPH Stereo Matching Processor Based on the Sum of Hamming Distances

Abiel Aguilar-González, Miguel Arias-Estrada

An FPGA-CAPH stereo matching processor based on the Sum of Hamming Distances

Abiel Aguilar-González and Miguel Arias-Estrada

Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE),
Tonanzintla, Puebla, México
http://www.inaoep.mx/

Abstract. Stereo matching is a useful algorithm to infer depth information from two or more of images and has uses in mobile robotics, three-dimensional building mapping and three-dimensional reconstruction of objects. In area-based algorithms, the similarity between one pixel of an image (key frame) and one pixel of another image is measured using a correlation index computed on neighbors of these pixels (correlation windows). In order to preserve edges, the use of small correlation windows is necessary while for homogeneous areas, large windows are required. In addition, to improve the execution time, stereo matching algorithms often are implemented in dedicated hardware such as FPGA or GPU devices. In this article, we present an FPGA stereo matching processor based on the Sum of Hamming Distances (SHD). We propose a grayscale-based similarity criterion, which allows separating the objects and background from the correlation window. By using the similarity criterion, it is possible to improve the performance of any grayscale-based correlation coefficient and reach high performance for homogeneous areas and edges. The developed FPGA architecture reaches high performance compared to other real-time stereo matching algorithms, up to 10% more accuracy and enables to increase the processing speed near to 20 megapixels per second.

Keywords: Stero matching; FPGA; Sum of Hamming Distances; CAPH

1 Introduction

The perception of depth from images is an important task of computer vision systems and has been used in several applications such as recognition, detection, three-dimensional reconstruction and positioning systems for mobile robots [12,16]. There are several techniques to compute depth from images, such as matching all pixels using correlation windows [21,13], matching interest points or features [24,19] and optimization techniques based on dynamic programming or graph cuts [10,9]. In case of matching all pixels with windows, the correspondence between stereo pairs and the geometrical configuration of the stereo camera allows obtaining dense disparity maps. To obtain a dense disparity map it is necessary to measure the similarity of all points in the stereo pair.

1.1 Related work

In this research, we are interested in dense disparity maps. There are several dense stereo matching algorithms in the literature. To reach real-time processing, stereo matching algorithms are often implemented in FPGA devices. [6] presents an FPGA module for computing dense disparity maps. The developed module enables a hardware-based cellular automata (CA) parallel-pipelined design. The presented algorithm provides high processing speed at the expense of accuracy, with large scalability in terms of disparity levels. In [4] a fuzzy approach for computing dense disparity maps is presented. The FPGA architecture determines the similarity between pixels using a Fuzzy Inference System. Although the proposed algorithm increases the accuracy in the computed disparities, it is sensible to the untextured pixels and their performance in untextured regions is limited. In [5] an FPGA module for computing dense disparity maps using a vergence control is proposed. Different to previous work, the developed module constantly estimates the required range of disparity levels upon a given stereo image set using a vergence control. In [1] a correlation-edge distance approach is described. By using a geometric feature (the Euclidean distance between the selected point and the nearest left edge), the developed FPGA architecture has low utilization of hardware resources, high speed processing and offers high performance for low disparity levels. However, the accuracy decreases for large disparity values. In [14] an adaptive window algorithm based on the SAD algorithm is proposed. The developed FPGA architecture offers more accuracy with respect to other FPGA-based stereo matching algorithms in the literature and allows to increasing the processing speed but large correlation window's sizes are required and hardware resource consumption is high.

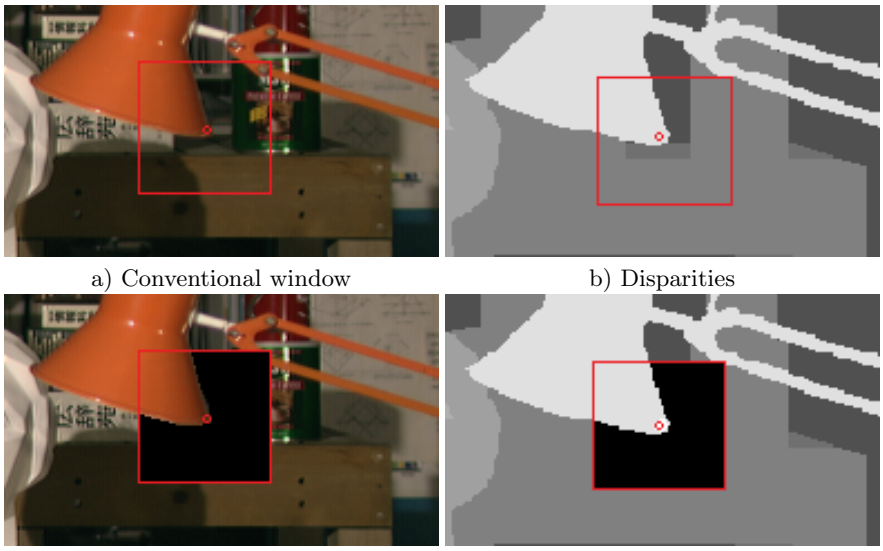
1.2 Motivation and scope

The main disadvantage of matching all pixels with windows is selecting the correlation window's size. Large window size values allow determining the correct correlation values in untextured areas. However, large window sizes imply high computational demand and erroneous values due to the averaging effect of comparing an object and the background. On the other hand, small window sizes imply low computational demand but the correlation coefficient measurement is sensitive to noise hence, erroneous values at untextured regions are generated. If a correlation window large enough to avoid noise is used, high accuracy in untextured areas must be reached. However, edges be slightly blurred and erroneous values at depth discontinuities are generated. If the objects and the background in the correlation window are separated, it is possible to compute the correlation index using pixels of the same object to the reference pixel. This allows maintaining high performance at untextured areas whilst blurring edges are avoided. i.e., it allows retaining the high performance characteristics of the small and large window sizes. To separate the objects and background from a correlation window, we propose the use of a grayscale-based similarity criterion.

The rest of the article is organized as follows: **section** 2 presents the proposed algorithm. In **section** 3, the FPGA architecture for the proposed algorithm is described. Experimental results for different synthetic stereo pairs and performance comparisons regarding to other algorithms in the literature are detailed in **section** 4. Finally, **section** 5 concludes this article.

2 The proposed algorithm

Fig. 1.(a) shows many objects at different depths. When any correlation coefficient is computed using all the pixels of the correlation window the averaging effect yields errors on the estimated disparity as shown in **Fig.** 1.(b). On the other hand, **Fig.** 1.(c) shows a neighborhood in which only the pixels that are the projections of the same object are used. In this case retained pixels are similar to the central pixel and they have the same disparity as shown in **Fig.** 1.(d). Although separating objects and background can improve the accuracy of the stereo match algorithms, in previous work it has been addressed via segmentation algorithms or super pixels which have high mathematical complexity and whose real-time implementation is complex, [3,7,22]. However, in this research, we separate objects and background using a similarity criterion based on the grayscale levels of the correlation window. The proposed similarity criterion allows implementing in dedicated hardware for real-time processing with low hardware resource consumption and parallel-pipelined design.



a) Conventional window b) Disparities

c) Proposed window d) Disparities

Fig. 1: Tsukuba scene

In the proposed algorithm, a fixed size window is centering on each pixel of the reference image, but only the pixels selected by the similarity criterion are used to compute the correlation coefficient. Any grayscale-based correlation coefficient such as Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD), Normalized Cross Correlation (NCC) and so on can be modified using this technique. However, to reach real-time processing, the proposed algorithm is inspired by the Sum of Hamming Distances (SHD) [8] since it consists in binary register operations and it can be implemented in FPGA devices with low hardware resource consumption and parallel-pipelined design. In this case, we propose to use **Eq** 1-2,

$$C(x,y,z) = \sum_{i=-w,j=-w}^{i=w,j=w} \mathcal{H}\{Q\}, \tag{1}$$

$$\begin{aligned} Q = \text{XOR}(&I_l(x+i,y+j)\cdot\lambda(x,y,i,j),\\ &I_r(x+z+i,y+j)\cdot\lambda(x,y,i,j)), \end{aligned} \tag{2}$$

where the parameter $\lambda(x,y,i,j)$, is equal to one for all pixels in the correlation window which correspond to same object that the center pixel, zero otherwise. $(2\cdot w+1)^2$ is the correlation window size. z is range from 0 to $z_{\max}$ (maximum expected disparity). $I_l(a,b)$, $I_r(a,b)$ are binary registers for the pixels from the left and right images, respectively and the $\mathcal{H}$ operator is defined as shown in **Eq** 3,

$$\mathcal{H}\{Q\} = \sum_{k=1}^{k=\text{bpp}} q(k), \tag{3}$$

where Q is a binary register and *bpp* is the size of the register Q, i.e., the bits per pixel for the input images.

2.1 Technique to define the similarity criterion

We can assume that two pixels have different disparity levels when there is a significant difference between their greyscale values. Hence, we define $\lambda(x,y,i,j)$ as one only when the grey level $I_l(x+i,y+j)$ is close to the grey level of the pixel $I_l(x,y)$, as shown in **Eq** 2. Where $\varphi(x,y)$ is the maximum acceptable difference between the greyscale values defined as shown in **Eq** 4,

$$\lambda(x,y,i,j) = \begin{cases} 1, & |I_l(x+i,y+j) - I_l(x,y)| \le \varphi(x,y) \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

the value of the φ parameter must be related to the uniformity of the greyscale values in the correlation window. To compute the φ value it is proposed to use **Eq** 5.

$$\varphi(x,y) = \frac{\sum_{i=-w}^{i=w}\sum_{j=-w}^{j=w}|I_l(x,y) - I_l(x+i,y+j)|}{(2\cdot w+1)^2} \tag{5}$$

2.2 Disparity computation

In standard stereo algorithms, the disparity $d_l(x, y)$ is defined as the shift z which gives the maximum (or minimum) of the correlation values in **Eq.** 1. To detect occlusions, the left-right consistency is used [17]. For each pixel, if the disparity $d_l(x, y)$ computed using the left image as a reference is equal to the disparity $d_r(x + z_{\max}, y)$ computed using the right image as the reference, use **Eq.** 6 instead of **Eq.** 2,

$$Q = \mathrm{XOR}(I_l(x+i, y+j) \cdot \lambda(x, y, i, j), \\ I_r(x - z + i, y + j) \cdot \lambda(x, y, i, j)), \tag{6}$$

the solution is considered as correct. Otherwise the pixels are marked as occluded, however, the disparity can be assigned as the minimum value between $d_l(x, y)$ and $d_r(x + z_{\max}, y)$.

3 The FPGA design

In **Fig.** 2, an overview of the FPGA design is shown. This design has five inputs, `clk_pixel` as the pixel clock for the input stereo pairs, `left_image [7:0]` and `right_image [7:0]` as grayscale values of the pixels from the images of left and right, respectively. `x_resolution [10:0]` as the horizontal resolution of the input images. `n [4:0]` as the number of lines in the correlation window. On the other hand, the design has one output, `final_disparity [7:0]`, corresponding to disparity values for the output image. Its general behavior can be described as following: first, the **stereopair_buffer** module stores the grayscale values for all pixels in the correlation windows from 0 up to $z_{\max}$. Then, the proposed similarity criterion is computed. Then, left-disparity and right-disparity modules compute the disparity value using the left and right image as reference. Finally, a multiplexer (**mux**) sets the final disparity value as the minimum value between the left and right disparity values.

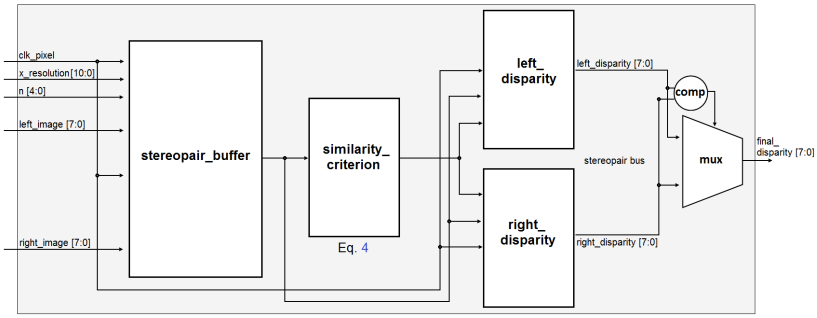


Fig. 2: General diagram for the FPGA design

3.1 The stereopair_buffer module

The **stereopair_buffer** module manages an array of $n + 1 \cdot 2$ BRAM cores which store n lines of two images as shown in **Fig.** 3. The **RAM_controller** module assigns to each BRAM core their corresponding address and write-read values. The **RAM_controller** module has two inputs, `x_resolution [10:0]` as the horizontal resolution of the input images and `n [4:0]` as the number of lines in the correlation window. There are two outputs, where `w/r [`$n + 1$`:0]` consist on a logic vector with $n + 1$ bits of size, the write-read value of each BRAM are determined by each bit of the vector. `address [10:0]` consists of a logic vector, which corresponds to a read/write address for all the BRAM cores. Each BRAM core only provides the grayscale value of one pixel from one horizontal line in the correlation window. In order to store the others horizontal values necessary for the disparity computation, we used the **line_vector** module. Using the **line_vector** module, the value of the first horizontal pixels of the $n + 1$ BRAMs are read in parallel form and, the pixels of the n BRAMs in read mode are placed in the bits 7-0 of n storage vectors, respectively. Then, the first horizontal pixels are placed in the bits 15-8 and the second horizontal pixels are placed in the bits 7-0. This process is repeated until all horizontal pixels necessary for computing all disparity levels are stored.

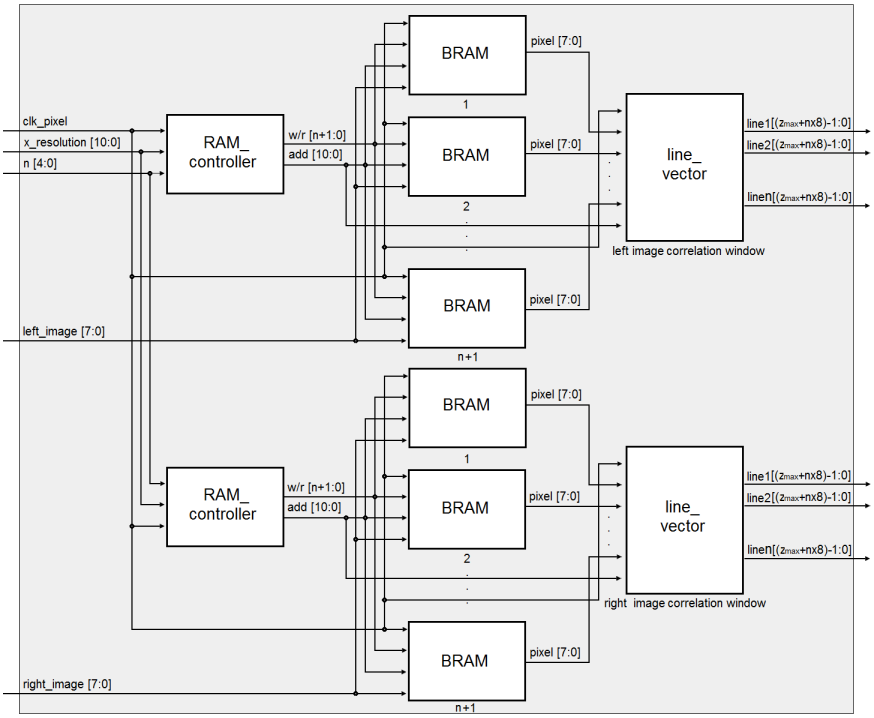


Fig. 3: FPGA design for the **stereopair_buffer** module

3.2 The disparity module

For the computation of the disparity map via the proposed algorithm, a pixel-parallel and window parallel architecture was designed. The architecture of the **disparity** module is presented in **Fig.** 4, its general behavior is as follows: first, the **XOR** modules compute the XOR operation between the pixels from left and right images of the correlation window. This process is executed in each of the $z_{\max} + 1$ **XOR** modules, implemented in parallel, which are configured for expected disparity levels from 0 until $z_{\max}$, where each module process only one disparity level and computes the XOR opperation only for pixels selected by the proposed similarity criterion (**Eq.** 2). Then, the output of each of the **XOR** modules are sent to its corresponding **binary_opperator** module. The **binary_opperator** module corresponds to the $\mathcal{H}$ operator (**Eq.** 3). Then, the **adder** module computes the sum of the values for all pixels retained in the correlation window (**Eq.** 1). Finally, the **mux_tree** module which consist in a multiplexer tree assigns the corresponding index for all correlation values, then, determines the minimum correlation value and set the disparity value as the index of the minimum correlation value. In the developed FPGA design, two **disparity** modules were implemented in parallel form where the first module uses the left image as reference while the second module uses the right image as reference.

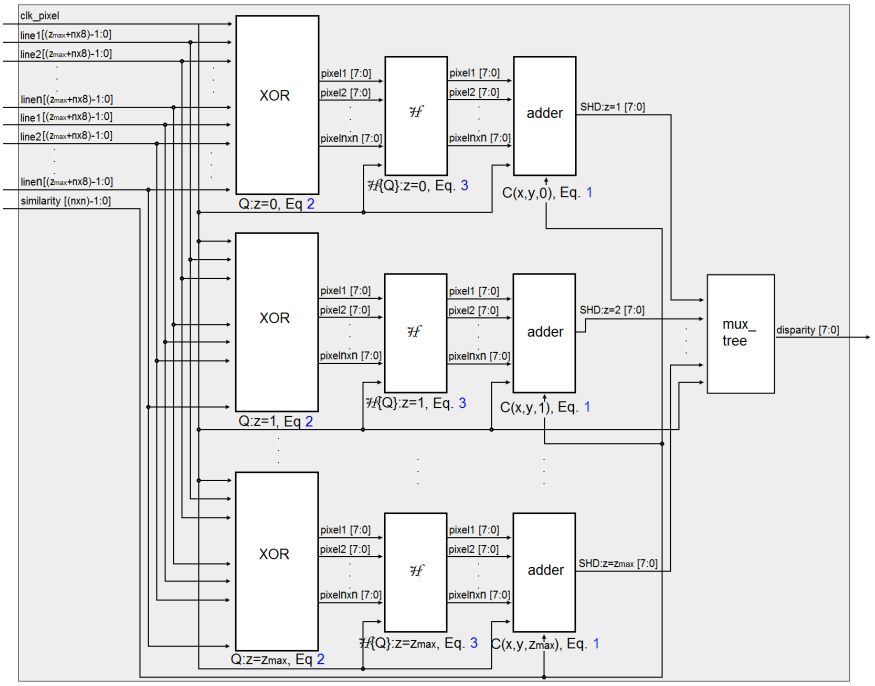


Fig. 4: FPGA design for the **disparity** module

4 Results and discussions

The developed FPGA architecture was implemented in an FPGA Cyclone IV EP4CGX150CF23C8 of Altera. All modules were designed via the CAPH design tool, which allows high-level programing and friendly design for video stream processing [20]. Furthermore, all modules were validated via post-synthesis simulations performed in ModelSim Altera. The selected configuration for the implemented algorithm was set as: correlation window size equal to $19 \cdot 19$, i.e., $2 \cdot w + 1 = 19$ and maximum disparity level equal to 31, i.e., $z_{\max} = 31$. This setup allows to read 19x19 pixels from 32 correlation windows in parallel form, i.e., it computes the correlation index value for all disparity levels in the reference pixel in the same clock cycle. The setup requires 40 BRAM cores, 64 **XOR**, $\mathcal{H}$ and **adder** modules, and two **mux_tree** module implemented in parallel-pipelined form. The hardware resource consumption of the developed FPGA architecture is shown in **Table** 1.

Table 1: Hardware resource consumption for the FPGA implementation

Resource (FPGA:EP4CGX150CF23C8)	**Consumption**
Total logic elements	62,689/149,760 (41.85%)
Total pins	25/287 (11.48%)
Total Memory Bits	242,064/6,635,920 (27.41%)
Embedded multiplier elements	0/720 (0%)
Total PLLs	0/6 (0%)

In **Table** 2, quantitative results of the number of erroneous pixels obtained by the proposed algorithm compared with other FPGA-based stereo matching algorithms in the literature are presented. **Table** 2 demonstrates that the proposed algorithm improves most algorithms in the literature. The algorithms presented in [2,23,11] allow estimating dense disparity maps. However, the averaging effect due to large window sizes used generates errors at depth discontinuities. To avoid the averaging effect algorithms with small window sizes have been developed [6,4,5]. However, noise sensibility is increased and erroneous values in untextured regions are generated. In both cases erroneous pixels near to 12% are reached. Other approach such as [1], which uses a geometric feature, or [15], which is inspired in the Census transform and introduces and adaptive coefficient reach erroneous pixels near to 10%. Finally, although the algorithm presented in [14] uses a grayscale-based similarity criterion to improve the performance of the SAD algorithm, it is possible to affirm that using the similarity criterion proposed in **Eq.** 1-6 of this article, allows to increase the accuracy, near to 10% with respect to [14]. Furthermore, due to the proposed algorithm allows decreasing the correlation window size with respect to [14], the FPGA resource usage for the proposed algorithm is more efficient than [14].

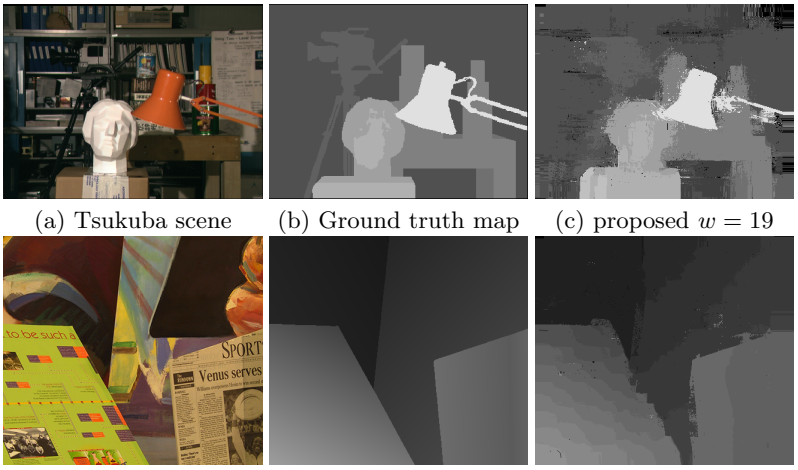
In **Fig.** 5 the disparity maps for the Tsukuba and Venus scenes generated by the proposed method are shown. Although the obtained results retain some noise, previous algorithms in the literature [6,4,5,1,14,15] have been improved quantitatively and qualitatively, disparity maps for the Tsukuba and Venus scenes of most algorithms compared in **Table** 2 can be consulted in the Middlebury Stereo Vision Page [18]. On the other hand, in **Table** 3 comparison of processing speed regarding to other real-time stereo matching algorithms reported in the literature is presented. It is observed high increase with respect to several FPGA-based stereo matching algorithms in the literature [15,2,23,11].

Table 2: Quantitative results of FPGA-based stereo matching algorithms

Algorithm	**Tsukuba**$_{(all)}$	**Venus**$_{(all)}$	**Correlation window size**
Aguilar-González et al., [1]	10.9%	6.93%	$3 \cdot 3$
Georgoulas et al., [6]	12.0%	8.0%	$7 \cdot 7$
Alba et al., [2]	13.92%	12.6%	$19 \cdot 19$
Georgoulas and Andreadis, [4]	11%	8%	$7 \cdot 7$
Georgoulas and Andreadis, [5]	12%	9%	$7 \cdot 7$
Perri et al., [15]	11.8%	7.2%	$13 \cdot 13$
Ttofis et al., [23]	10.4%	12.1%	$11 \cdot 11$
Pérez-Patricio and Aguilar-González, [14]	7.6%	3.2%	$29 \cdot 29$
Jin et al., [11]	11.57%	5.27%	$15 \cdot 15$
Proposed*	6.8%	2.8%	$19 \cdot 19$

Table 3: Processing speed of FPGA-based stereo matching algorithms

Algorithm	**Resolution**	**Frames/s**	**Pixeles/s**
Georgoulas et al., [6]	1280·1024	65	85,196,800
Perri et al., [15]	640·480	68	20,889,600
Alba et al., [2]	256·256	100	6,553,600
Ttofis et al., [23]	1280·1024	50	65,536,000
Aguilar-González et al., [1]	1280·1024	75	98,304,000
Georgoulas and Andreadis, [5]	1024·1024	102	106,954,752
Georgoulas and Andreadis, [4]	384·288	833	92,123,136
Pérez-Patricio and Aguilar-González, [14]	450·375	592	99,900,000
Jin et al., [11]	640·480	630	70,656,000
Proposed	1280·720	117	107,827,200



(a) Tsukuba scene (b) Ground truth map (c) proposed $w = 19$

(d) Venus scene (e) Ground truth map (f) proposed $w = 19$

Fig. 5: Disparity maps generated for different test synthetic stereo pairs

5 Conclusions

In this article an area-based algorithm for stereo matching using a similarity criterion, which is used as pixel selector in the correlation window was presented. It was demonstrated that using the modified Sum of Hamming Distance algorithm proposed in this article, it is possible to increase the accuracy of most real-time FPGA-based stereo matching algorithms in the literature and reach parallel-pipelined design that increases the processing speed. The best performance of the proposed algorithm was obtained with a large window, appropriated for untextured areas. However, due to only pixels of the same object are used in the correlation window, blurring effects are avoided, and therefore, erroneous values at depth discontinuities are reduced.

References

1. Aguilar-González A., Pérez-Patricio M., Arias-Estrada M., Camas-Anzueto J.L., Hernández-De león H.R., Sánchez-Alegría A.: An FPGA correlation-edge distance approach for disparity map. In: Proceedings of IEEE International Conference on Electronics, Communications and Computers (CONIELECOMP15), pp. 21–28, IEEE Press, Cholula (2015)
2. Alba A., Arce-Santana E., Aguilar-Ponce R.M., Campos-Delgado D.U.: Phase-correlation guided area matching for realtime vision and video encoding. J Real-Time Image Proc 9, 621–63(2012)
3. Bleyer, M., Rother, C., Kohli, P., Scharstein, D., Sinha, S.: Object stereo Joint stereo matching and object segmentation. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3081–3088, IEEE Press, Providence, RI (2011)

4. Georgoulas C., Andreadis I.: A real-time fuzzy hardware structure for disparity map computation. J Real-Time Image Proc 6, 257–273 (2011)
5. Georgoulas C., Andreadis I.: FPGA based disparity map computation with vergence control. Microprocessors and Microsystems 34, 259–273 (2010)
6. Georgoulas C., Kotoulas L., Sirakoulis G.C., Andreadis I., Gasteratos A.: Real-time disparity map computation module. Microprocessors and Microsystems 32, 159–170 (2008)
7. Gerrits, M., Bekaert, P.,: Local Stereo Matching with Segmentation-based Outlier Rejection. IEEE 2006. The 3rd Canadian Conference on Computer and Robot Vision, pp. 66–72, IEEE Press (2006)
8. Hamming, R. W.: Error detecting and error correcting codes. Bell System Technical Journal 29(2), 147-160 (1950)
9. Hu T., Qi B., Wu T., Xu X., He H.: Stereo matching using weighted dynamic programming on a single-direction four-connected tree. Computer Vision and Image Understanding 116, 908–921 (2012)
10. Kalomiros J.A., Lygouras J.: Design and hardware implementation of a stereo-matching system based on dynamic programming. Microprocessors and Microsystems 35, 496–509 (2011)
11. Jin S., Cho J., Pham X.D., Lee K.M., Park S.K., Kim M., Jeon J.W.: FPGA design and implementation of a real-time stereo vision system. IEEE Transactions on Circuits and Systems for Video Technology 20, 15–26 (2009)
12. Parrilla E., Torregrosa J.R., Riera J., Hueso J.L.: Fuzzy control for obstacle detection in stereo video sequences. Mathematical and Computer Modelling 54, 1813–1817 (2011)
13. Pérez-Patricio M., Aguilar-González A., Arias-Estrada M., Camas-Anzueto J.L.: A Fuzzy Logic Approach for Stereo Matching Suited for Real-Time. International Journal of Computer Applications 113, 1–8 (2015)
14. Pérez-Patricio M., Aguilar-González A.: FPGA implementation of an efficient similarity-based adaptive window algorithm for real-time stereo matching. J Real-Time Image Proc. (2015). DOI 10.1007/s11554-015-0530-6
15. Perri, S., Corsonello, P., Cocorullo G.: Adaptive census transform: A novel hardware-oriented stereo vision algorithm. Computer Vision and Image Understanding 117, 29–41 (2013)
16. Rong X., Huanyu J., Yibin Y.: Recognition of clustered tomatoes based on binocular stereo vision. Computers and Electronics in Agriculture 106, 75–90 (2014)
17. Scharstein D., Szeliski R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision 47(1), 7–42 (2002)
18. Scharstein, D., Szeliski R., Hirschmller H.: Middlebury Stereo Vision Page. http://vision.middlebury.edu/stereo/
19. Schauwecker K., Klette R., Zell A.: A New Feature Detector and Stereo Matching Method for Accurate High-Performance Sparse Stereo Matching. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5171–5176 IEEE Press, Vilamoura (2012)
20. Serot J., Berry F., Ahmed S.: FPGA: A Language for Implementing Stream-Processing Applications on FPGAs, Embedded Systems Design with FPGAs. Springer, chap FPGA: A La, pp 201–224. In: Athanas, P., Pnevmatikatos, D., Sklavos, N. (eds.) Embedded Systems Design with FPGAs. pp. 1148–1158. Springer New York (2012)
21. Stefano L.D., Marchionni M., Mattoccia S.: A fast area-based stereo matching algorithm. Image and Vision Computing 22, 983–1005 (2004)

22. Trinh, H,: Efficient Stereo Algorithm using Multiscale Belief Propagation on Segmented Images. Proceedings of the British Machine Vision Conference, pp. 33.1–33.10, BMVA Press (2008)
23. Ttofis C., Hadjitheophanous S., Georghiades A.S., Theocharides T.: Edge-directed hardware architecture for real-time disparity map computation. IEEE TRANSACTIONS ON COMPUTERS 62, 690–704 (2013)
24. Wang L., Liu Z., Zhang Z.: Feature Based Stereo Matching Using Two-Step Expansion. Mathematical Problems in Engineering 2014, 1–14 (2014)