Multiple Scanning Order in Fast Depth Estimation Algorithm

Hubert Żabiński
Institute of Multimedia Telecommunications
Poznan University of Technology,
and Mucha Sp. z o.o.
Poznań, Poland
ORCID: 0000-0003-3345-2110

Krzysztof Wegner
Mucha Sp. z o.o.
Poznań, Poland
ORCID: 0000-0002-5671-0541

Olgierd Stankiewicz
Institute of Multimedia Telecommunications
Poznan University of Technology, Poznań, Poland
ORCID: 0000-0001-9691-9094

***Abstract*—Fast algorithms for stereoscopic depth estimation typically employ processing of the input images in a Single Scanning Order (SSO). In this paper we present a novel approach that employs processing in Multiple Scanning Orders (MSO), which are then merged together into a final depth map. We demonstrate the advantages of the proposal on the example of a fast depth estimation technique [1], adaptable both for mobile platforms and FPGA. We show that application of the proposal leads to considerable quality improvement at an acceptable complexity cost.**

Keywords—depth estimation, disparity estimation, multiple scanning order, FPGA.

I. Introduction

Depth estimation is an important tool for modeling 3D scenes from sets of views, important for 3D television (3DTV), immersive 6-DoF video, robot vision, self-driving cars etc. In such applications, efficient real-time depth estimation is still a true challenge, especially when hardware complexity and power consumption is of concern. In order to meet these conditions, the respective depth estimation techniques must be relatively simple but still be able to produce high-fidelity depth.

The most common passive depth estimation methods employ stereo-matching. In the simplest form, stereo matching techniques use image pair analysis and correspondence search between image fragments to determine the best disparity value for each point [2]. Disparity (distance between object positions between distinct views) is then used to determine the depth (e.g. distance to the object in the scene from the camera). The downside of such a simple approach is that each disparity value is estimated independently and therefore the process lacks information in regions without texture and the estimated depth suffers from structural inconsistency.

Therefore, in more advanced depth estimation methods [2, 3, 4], additionally, regularization algorithms are used to enforce structural consistency in the estimated 3D scene. In such approaches, neighboring disparity values are entangled, e.g. optimized together which is commonly expressed in terms of energy minimization. Such energy can be defined globally for the entire image and tackled through means of algorithms like graph cuts [5] or belief propagation [6], which are computationally intensive. Due to this, such works are outside of the scope of this paper.

Less complex solutions typically try to achieve regularization locally, in a greedy approach, in which disparity for newly estimated pixels in inferred from the already estimated pixels. In such an approach pixels are processed (scanned) in some predefined order, e.g. row-by-row from the top to the bottom of the image, and in each row: pixel-by-pixel, from the left to the right, or vice versa. In the methods found in the literature, a Single Scanning Order (SSO) is used. Such SSO-based methods suffer from depth artifacts related to the direction of scanning because the information about the depth is inferred/propagated only in one direction and cannot propagate back. We elaborate on this in Section III of the paper.

In this paper, we present a novel approach that employs processing in Multiple Scanning Orders (MSO). The depth map is estimated multiple times, with the same core algorithm, but each time with the use of different scanning orders, and thus using different inferring directions. Therefore, in each scan, depth information propagates in different directions, each time resulting in different depth artifacts, often in different areas of the image. In the end, the depth maps produced as a result of consecutive scanning orders are merged into a final depth map. For the merging step we propose a few alternative algorithms - the details are presented in Section IV.

The experimental results in Section V show that application of the proposal leads to considerable depth map quality improvement. The proposal can also be efficiently implemented in hardware for which we also show results on an example of an FPGA device (see Section VI).

II. State of the art

In this paper, we consider fast depth estimation methods, applicable for mobile devices and FPGA implementations. Such methods typically [4] perform regularization locally. In particular, we focus on methods that employ scanning of the image in some order, in which depth for already estimated pixels is used for inferring the depth which is currently estimated.

However global optimization methods (graph cuts [5] or belief propagation [6] and its implementations e.g. [7]) are outside of the scope of this paper, there are existing hybrid methods that employ semi-global optimization (e.g. on “inside row” level) and scanning on “between rows” level [8, 9, 10]. For example, Wang et al. use a simpler global reasoning algorithm based on dynamic programming in horizontal scan lines. Such methods can successfully benefit from the proposals of this paper.

In the case of the considered scanning-based approach, all of the methods found in the literature employ just a Single Scanning Order (SSO). In works [11-14] for each pixel in image weights in the block matching cost are calculated by means of bilateral filtering. Unfortunately, bilateral filtering is computationally expensive and thus its various approximations are studied in the literature. Mattoccia et al [11] divide the matching window into small regular blocks in which filter coefficients are kept constant. Wei et al [12] propose two algorithms that employ separable approximation of bilateral filtering and iterative calculation of the matching

cost with an exponential step size. Others, like [13,14] try to use guided filters for fast computation of the matching cost in block-size-independent O(N) time.

In work [1] authors show a hardware-applicable algorithm that employs inferring from neighboring pixels to the newly estimated one. The estimation progresses in one direction and thus this method also can be categorized as SSO. However, the authors mention that the direction of the processing can have an impact on the quality of the depth map, no results or proposals are presented in that matter.

Based on the aforementioned state-of-the-art we have decided to assess the proposal of the paper with the use of an existing scanning-based depth estimation algorithm as a core technique. We have selected the depth estimation algorithm presented in the work [1] because it is fast and applicable to both hardware and mobile applications. It can however be noticed that the approach that we propose can be applied to any scanning-based depth estimation method.

III. Single Scannng Order

In Single Scanning Order (SSO) depth estimation algorithms, pixels are processed (scanned) in some predefined order, e.g. in work [1] row-by-row from the top to the bottom of the image, and in each row: pixel-by-pixel, from the right to the left (Fig. 1).

An explicit order of processing is beneficial because allows usage of already estimated disparity values for inferring currently estimated disparity values. In the work [1] columns on the right are processed first, they can be used to estimate depth for the pixels in columns to the left (Fig. 2a). Thus, a given pixel can use already estimated depth values from its neighboring pixels placed to the right, right-top, and right-bottom. Of course, different inferring schemes, are possible, e.g. using, top, top-right and right (Fig. 2b) limited to the right-top and right neighbor (Fig. 2c), or even limited to the neighbor to the right (Fig. 2d).

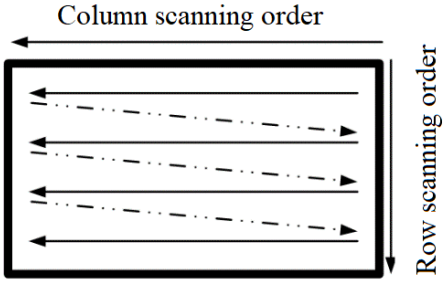

Fig. 1. Exemplary scanning order

Artifacts in estimated depth with SSO methods strongly correlate with the selected direction of scanning/inferring. For example, the pixel at coordinates (x,y) (Fig 2d) uses information about estimated depth from an already estimated pixel at coordinates (x+1, y), but the opposite is impossible. This appears as characteristic “depth leak” artifacts (Fig. 3b), visible mostly and the borders of the objects. In the picture, the depth of the original shape (marked in red) is used for (erroneous) inferring the depth of the left band of the objects.

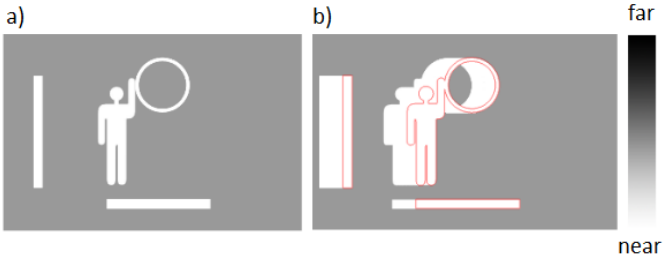

Fig. 3. Exemplary grounth depth map (a) and depth map with“depth leak” artifacts (b) due to information inferring only from one direction – from the right. The original shape of objects in marked on (b) with red line.

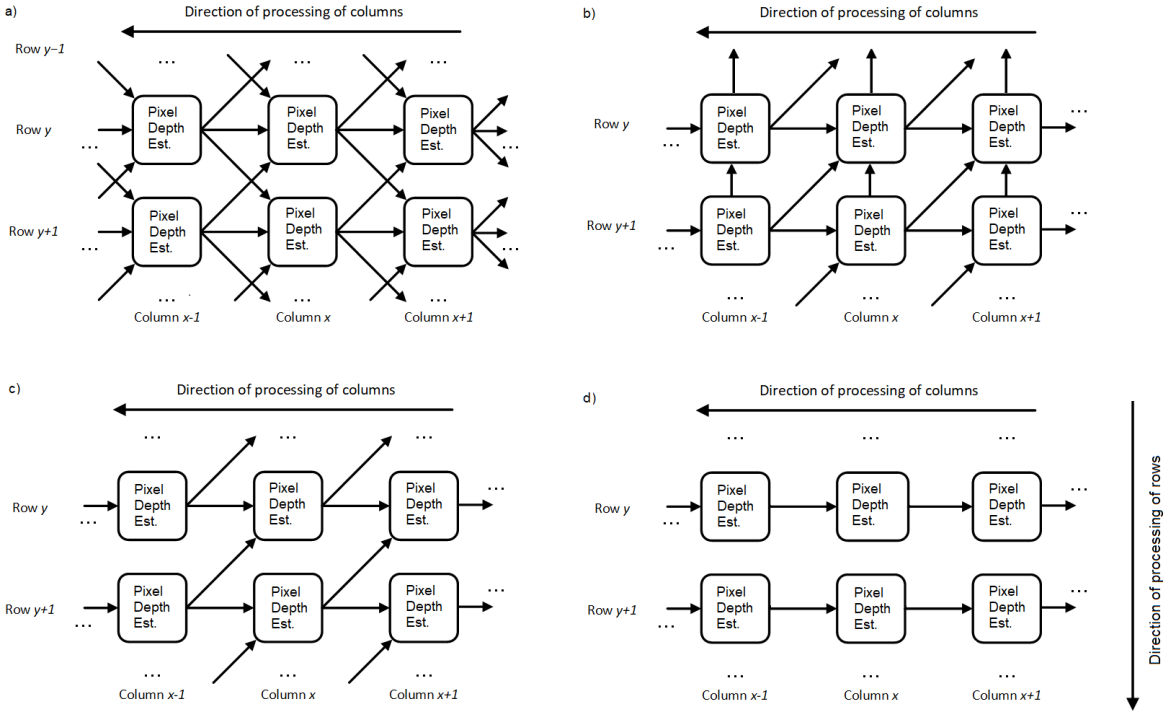

Fig. 2. Possible directions of information inferring in SSO depth estimation. Estimation scans considered in the paper (A,B,C,D) employ inferring presented in (b).

IV. Multiple Scanning Order

The idea of this paper is to replace SSO approach with Multiple Scanning Orders (MSO). Instead of estimating depth once with only one selected scanning order, which is vulnerable to direction-characteristic depth leaks, we propose to perform multiple passes of depth estimation scanning. In each scan, the same core algorithm is used, but with different scanning order, and thus also with different inferring directions. In the end, the depth maps produced as a result of consecutive scanning orders are merged in to a final depth map.

A. Scanning orders and inferring dirrections

For the sake of experimentation, we have decided to employ 4 scans: A, B, C, D, each of which uses three-neighboring pixels for inferring, as shown in Fig. 2b, but rotated accordingly (Fig.4). It can be noted that 90° rotations of inferring direction altogether with scanning order can be implemented through the same computational kernel, but working on transformed (flipped) data (Table I and Fig. 4). In the case of three-view depth estimation, where the depth is estimated for the center view, additionally there is a need to switches left with right image and vice-versa. This operation is controlled identically to horizontal flipping.

For example, variant A means inferring data from previously processed pixels located to the left, to the left-top, and to the top, relatively to the currently processed pixels, top-to-bottom row scanning, left-to-tight column scanning. In such a case:

- Variant B is implemented as variant A but with horizontal flipping of the input image.
- Variant C is implemented as variant A but with vertical flipping of the input image.
- Variant D is implemented as variant A but with both horizontal and vertical flipping of the input image.

TABLE I. Information inferring directions and implied scanning order and practical implementation by image flipping (Horizontal/Vertical)

	Inferring direction (relative to currently processed pixel)	Scanning		Flipping	
		Row	Col.	Horz. (FH)	Vert. (FV)
A	Left (←), left-top (↖), top (↑)	→	↓	-	-
B	Right (→), right-top (↗), top (↑)	←	↓	+	-
C	Left (←), left-bottom (↙), bottom (↓)	→	↑	-	+
D	Right (→), right-bottom (↘), bottom (↓)	←	↑	+	+

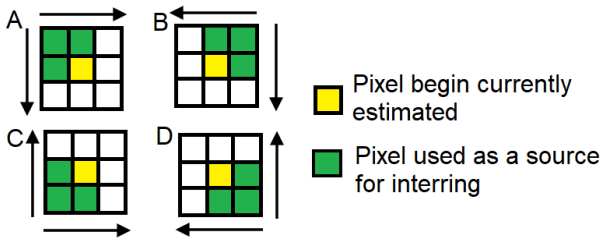

Fig. 4. Information inferring directions and implied scanning orders.

B. Depth map merging

Usage of Multiple Scanning Orders (MSO) results in four estimated depth maps. Therefore, for each image point there are available 4 disparity values - one from each scan: A, B, C and D. Basing on these four values, merging is performed in order to produce the final depth map (Fig. 4). As a first step, the process involves sorting of set {A,B,C,D}, which results in set {E,F,G,H} such that $E \le F \le G \le H$.

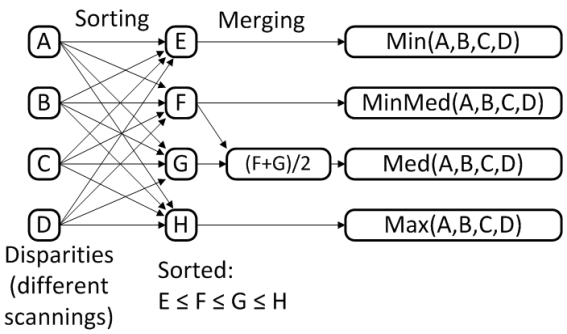

Fig. 4. Proposd merging methods of MSO results.

We propose four alternative methods for merging of disparity values and producing the final disparity value R:

- Min – minimal disparity value is selected, thus the algorithm has a preference for distant (far) objects (1):

$$R \leftarrow \min(A, B, C, D) \quad (1).$$

- Max – maximal disparity value is selected, thus the algorithm has a preference for near objects (2):

$$R \leftarrow \max(A,B,C,D) = H \quad (2).$$

- Med – may result in a disparity value not present in the scene (not found by the core depth estimation algorithm) as because 4 scans are used, median operation (3) involves division by 2 (integer shift right by 1 bit):

$$R \leftarrow (F+G)/2 \quad (3).$$

- MinMed – "median" with a preference for smaller disparity value (preference for slightly further objects) (4):

$$R \leftarrow F \quad (4).$$

As we show in the results section, the selection of the voting algorithm has a significant impact on the depth estimation quality.

C. MSO Realization

Realization of the proposed MSO can be done with the use of the resources (e.g. hardware) suitable for SSO, only at the cost of 4 passes of execution of the algorithm and a merging step which has negligible complexity. In particular, additional needed hardware can be summarized as:

- Controllable vertical/horizontal coordinate flipping. This can be implemented by means of an adder and a multiplexer. For example, controllable horizontal coordinate flipping, controlled with horizontal flipping flag FH (Table I), can be expressed as (5):

$$x' = FH \ ? \ (WIDTH-1) - x \ : \ x \quad (5),$$

  where ? : is a conditional expression, WIDTH is the width of the image, x is original horizontal coordinate and x' is (optionally) flipped horizontal coordinate.

- In the case of three-view depth estimation, where the depth is estimated for the center view - image data multiplexer which is switches left with right image and vice-versa. This operation is controlled identically to horizontal flipping, e.g. with the use of horizontal flipping flag FH (Table I).

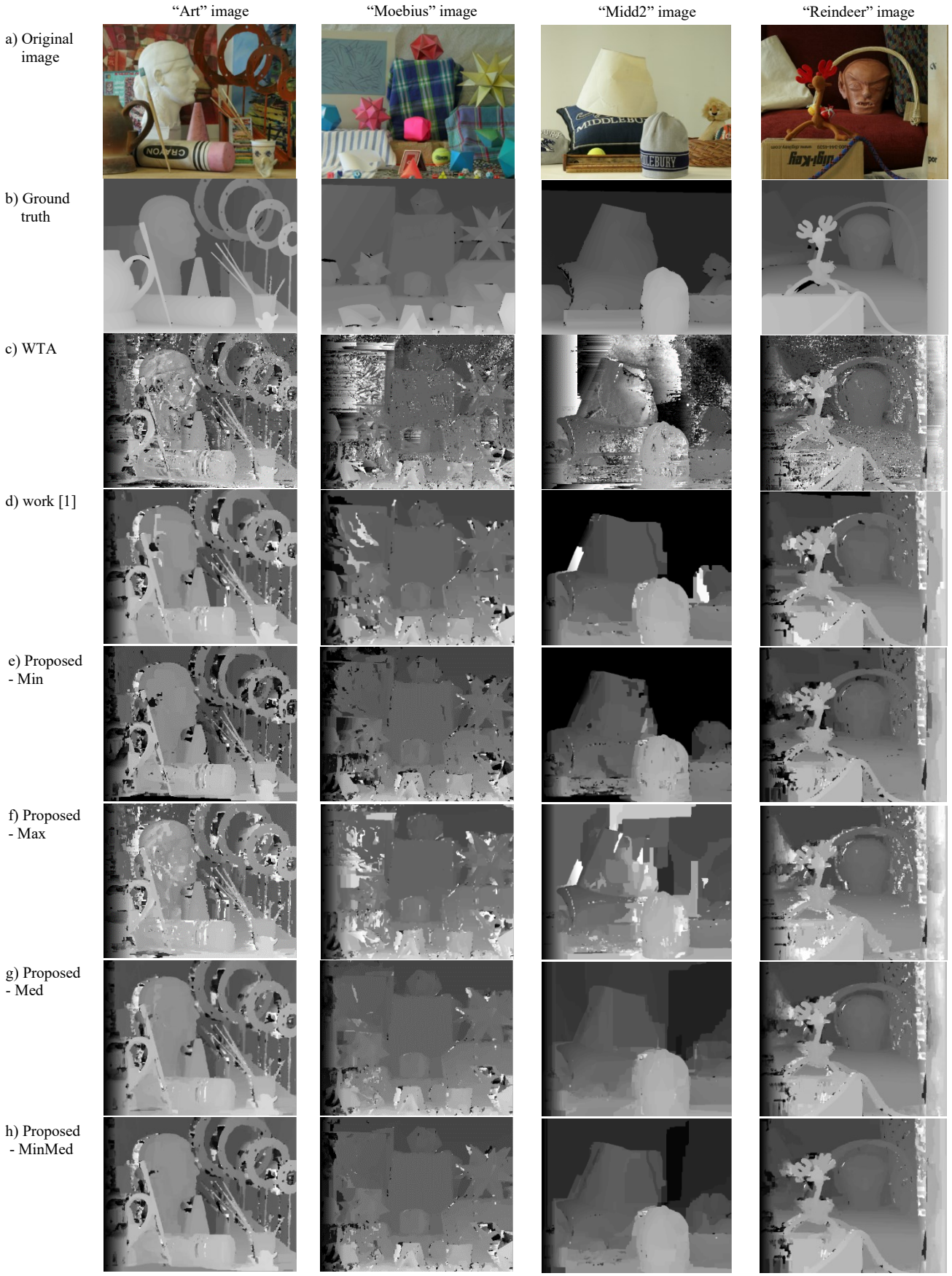

Fig. 5. Experimental results – "Art" and "Moebius" imags from Middlebury dataset [15]:
a) the original image, b) ground-truth depth map, c) Winner Takes All (WTA) technique , d) technique from work [1],
e) proposed – Min, f) proposed – Max, g) proposed – Med, h) proposed – MinMed.

V. Experimental results

For the quality evaluation of our method, we have used Middlebury stereoscopic images [15]. As an objective quality index bad-pixel ratio [16] has been used, which is common in literature. Bad-pixel ratio presents the percentage of pixels for which disparity is estimated wrongly, in comparison to ground-truth disparity maps, with a margin of 1 disparity level. The percentage of bad pixels was calculated only for non-occluded regions of the images. In Table II we present results for Winner Takes All algorithm (the most straight-forward reference), the results of the unaltered [1] technique (used as the core for our experiments) with SSO, and the results for 4 alternative merging algorithms proposed in this paper: Min, Max, Med, and MinMed. The evaluation has been done for various window sizes for the stereo-matching step: 3×3, 5×5, 7×7 and with different setups of color components: Luminance only (Y) and RGB. We also show some exemplary images (Fig. 5) to show the visual improvement attained with our proposal. It can be seen that the usage of depth merging from competitive orthogonal scanning orders allows canceling of some of the "depth leak" artifacts.

Table II. Bad-pixel ratio [%] results on Middlebury [15] dataset for various block sizes and component settings

Algorithm	Y – Luminance only			RGB		
	3x3	5x5	7x7	3x3	5x5	7x7
WTA	53.91	45.67	42.12	48.78	42.12	39.18
SSO [1]	37.32	35.8	35.28	34.99	33.57	33.42
Proposed MSO - Min	39.11	36.54	35.66	36.42	33.74	33.05
Proposed MSO - Max	44.56	41.81	40.27	41.59	39.07	37.84
Proposed MSO - Med	33.34	32.04	32.11	30.51	30.10	30.43
Proposed MSO - MinMed	33.01	31.39	31.81	30.77	29.35	29.69
Gain of MSO – MinMed versus SSO [1]	4.31	4.41	3.47	4.22	4.22	3.73

It can be noticed that the best results are attained with MinMed merging method. It allows for an improvement of about 4 percent points, as compared to the SSO method [1]. Notably this gain is much higher than one attainable by the usage of larger windows. For example, usage of 7×7 window improves the results by about 1 percent point, as compared to the usage of 3×3 window size.

VI. Implementation

The hardware implementation has been prepared in Verilog language. It has been synthesized, verified and tested on a proprietary Mucha Development Board equipped with Lattice ECP5 FPGA (Fig. 6). The source of video data is a 3-camera rig equipped with OmniVision ov4689 sensor. Also, for demonstration application purposes, an end-user-friendly Arduino-based microcontroller (Espressif ESP32) was used.

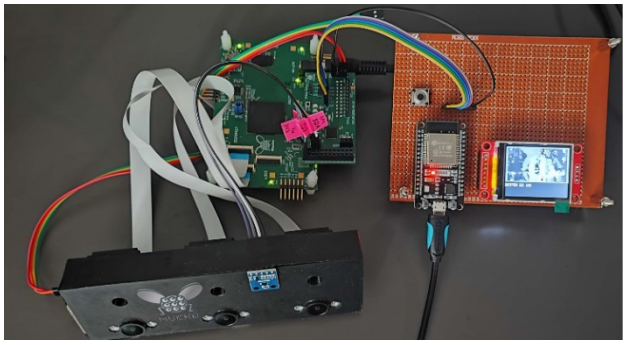

Fig. 6. Hardware implemention of the algorith, using proprietry Mucha Development Board.

Table III. Synthesis results for FPGA hardware implementation

	Window Size (RGB mode)		
Synthesis parameter	**3×3**	**5×5**	**7×7**
Clock frequency – P&R [MHz]	112.7	102.4	74.8
Clock frequency – MAP [MHz]	158.8	176.1	159.7
CLBs	12 180	16 536	22 853
EBRs	174	178	182

In Table III we present synthesis results for various window sizes. In all cases RGB color space has been used in similarity metric calculation and in all cases MinMed variant of the proposed algorithm has been used. The synthesized IP is can run at 100 MHz and consumes 83 640 Cell Logic Blocks (CLBs) and 208 blocks of memory (EBR), 18 kbit each.

Conclussions

A novel approach to stereoscopic depth estimation has been proposed that employs Multiple Scanning Orders (MSO) as opposed to commonly used Single Scanning Order (SSO). MSO in our proposal is appended with a merging step which may involve one of four proposed voting methods: Min, Max, Med, and MinMed.

The experimental results are presented on the example of MSO applied on top of a state-of-the-art algorithm known from the literature [1] with four competitive scanning orders, which infer depth information from four orthogonal directions. As it has been presented - both visually and objectively by means of bad-pixel ratio – the usage of the proposal allows for significant improvement of the quality of the estimated depth map. The highest gains are observed for the MinMed method (around 5 percentage points reduction of bad-pixel ratio), but very comparable results can be achieved with the Med method.

The considered algorithm has been implemented and tested in hardware FPGA devices (Lattice ECP5). It allowed us to practically show that the implementation of the proposed MSO approach can be done with the use of the same hardware which is suitable for SSO, only at the cost of 4 passes of execution of the algorithm (and a negligible merging step).

References

[1] Marek Domański, Jacek Konieczny, Maciej Kurc, Adam Łuczak, Jakub Siast, Olgierd Stankiewicz, Krzysztof Wegner, "Fast Depth Estimation on Mobile Platforms and FPGA Devices", 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 3DTV- Con 2015, Lisbon, Portugal, 8-10 July 2015.

[2] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms," 2006 IEEE Computer Society Conf. on Comp. Vision and Pattern Recogn. (CVPR'06), New York, NY, USA, 2006, pp. 519-528.

[3] O. Johannsen et al., "A Taxonomy and Evaluation of Dense Light Field Depth Estimation Algorithms," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 2017, pp. 1795-1812 226.

[4] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, et al., "High-resolution stereo datasets with subpixel-accurate ground truth", German Conf. Pattern Recognition (GCPR 2014), Münster, September 2014.

[5] M. Bleyer, M. Gelautz, "Graph-based surface reconstruction from stereo pairs using image segmentation", SPIE Electronic Imaging Conf., San Jose, pp. 288–299, January 2005.

[6] T. Montserrat, J. Civit, O.D. Escoda, J.-L. Landabaso, "Depth estimation based on multiview matching with depth/color

segmentation and memory efficient Belief Propagation", IEEE Int. Conf. .Image Proc. (ICIP), pp. 2353-2356, Cairo, Nov. 2009.

[7] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, D. Nister, "Realtime global stereo matching using hierarchical belief propagation", British Machine Vision Conference, pp. 989–998, Edinburgh, 2006.

[8] C. Stentoumis, E. Karkalou and G. Karras, "A review and evaluation of penalty functions for Semi-Global Matching," 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 2015, pp. 167-172.

[9] L. Wang, M. Liao, M. Gong, R. Yang, D. Nister, "High-quality realtime stereo using adaptive cost aggregation and dynamic programming", Third Int. Symp. 3D Data Processing, Visualization, and Transmission (3DPVT'06), Washington, pp. 798–805, 2006.

[10] C. Banz, S. Hesselbarth, H. Flatt, H. Blume, P. Pirsch, "Real-time stereo vision system using semi-global matching disparity estimation: Architecture and FPGA-implementation", Int. Conf. on Embedded Computer Systems (SAMOS), pp.93,101, July 2010.

[11] S. Mattoccia, M. Viti, and F. Ries, "Near real-time fast bilateral stereo on the GPU", IEEE Comp. Society Conf. Computer Vision and Pattern Recog. Workshops (CVPRW 2011), pp. 136 –143, Colorado Springs, June 2011.

[12] W. Yu, T. Chen, F. Franchetti, J. C. Hoe, "High performance stereo vision designed for massively data parallel platforms," IEEE Trans. Circuits Syst. Video Techn., vol. 20, pp. 1509–1519, Nov. 2010.

[13] A. Hosni, C. Rhemann et al, "Temporally consistent disparity and optical flow via efficient spatio-temporal filtering", Advances in Image and Video Technology (Y.-S. Ho, ed.), vol. 7087, Lecture Notes in Comp. Science, pp. 165–177, Springer, 2012.

[14] C. Richardt, D. Orr, I. Davies, A. Criminisi, N.A.Dodgson, "Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid", European Conf. Computer Vision (ECCV), Lecture Notes in Computer Science, pp. 510–523, September 2010.

[15] D. Scharstein, R. Szeliski. Middlebury stereo evaluation - version 2, 2010, http://vision.middlebury.edu/stereo/eval/.

[16] D. Scharstein, R. Szeliski, "High-accuracy stereo depth maps using structured light", IEEE Comp. Society Conf. Computer Vision and Pattern Recogn. (CVPR 2003), vol. 1, pp. 195-202, Madison, June 2003.