jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2018, VOL. 64, NO. 3, PP. 269–275
Manuscript received May 28, 2018; revised July, 2018. DOI: 10.24425/123518

View and Depth Preprocessing for View Synthesis Enhancement

Adrian Dziembowski and Marek Domański

***Abstract*—In the paper, two preprocessing methods for virtual view synthesis are presented. In the first approach, both horizontal and vertical resolutions of the real views and the corresponding depth maps are doubled in order to perform view synthesis on images with densely arranged points. In the second method, real views are filtered in order to eliminate blurred or improperly shifted edges of the objects. Both methods are performed prior to synthesis, thus they may be applied to different Depth-Image-Based Rendering algorithms. In the paper, for both proposed methods, the achieved quality gains are presented.**

***Keywords*—virtual view synthesis, free-viewpoint television, free navigation, depth-image-based rendering**

I. Introduction

THE Free-Viewpoint Television systems [1], [2] allow a user to freely, virtually navigate around a scene that was captured using a number of real synchronized cameras. In order to provide smooth transition between different points of view, the views from virtual viewpoints should be synthesized [3].

The idea of free navigation is presented in Fig. 1. In the example, 8 real (black) cameras capture a scene. A user can freely change his point of view, watching a scene from a viewpoint of the virtual (orange) camera.

Fig. 1. Idea of a Free-Viewpoint Television system.

In practical low-cost multicamera systems [4], [5], [6], the number of cameras is significantly smaller than in expensive experimental systems, where hundreds of cameras are used [7], [8]. Due to larger distances between the neighboring cameras, in such systems it is much more difficult to synthesize high-quality virtual views. Therefore , we need the techniques that are capable to produce synthetic views with even better quality than the techniques that are in common use.

The presented work has been funded by the Polish Ministry of Science and Higher Education within the status activity (DS) task ”Theory and algorithms of multidimensional signal processing” in 2018.

Adrian Dziembowski (adziembowski@multimedia.edu.pl) and Marek Domański are with Chair of Multimedia Telecommunications and Microelectronics, Poznan University of Technology, Poznań, Poland (e-mail: adziembowski@multimedia.edu.pl, marek.domanski@put.poznan.pl).

In the paper, we propose two simple methods that allow for reducing the artifacts in the virtual views, thus improving their quality. Both proposed methods can be performed before the actual virtual view synthesis algorithms. Therefore, they can be applied for any DIBR (Depth-Image-Based Rendering [9]) algorithm, e.g. state-of-the-art method, MPEG's reference software – VSRS (View Synthesis Reference Software) [10] or the algorithm MVS (MultiView Synthesis) [11] developed by the authors.

The first of the proposed preprocessing methods consists in the upsampling of the real views and the corresponding depth maps. Similar idea was already presented e.g. in [10] or [12] but in the proposed approach a different depth interpolation method was created. Application of the second proposed technique provides improved quality of the synthesized virtual views. The second proposed method allows for filtering blurred edges in the real views in order to improve the quality of the virtual view by eliminating artifacts known as ”ghost” edges [12]. The existing methods allow for removing ghost edges, but they also have some limitations, e.g. all the background points placed at the edges are omitted during view synthesis [13] or they are projected even if their color is wrong [14], [15], [16]. The proposed approach reduces these disadvantages – the ghost edges in the virtual view are eliminated but all the points placed at the edges which color value is proper are projected to the virtual view.

II. View and Depth Map Upsampling

The proposed upsampling operation is performed in the additional, preprocessing step prior to the view synthesis (Fig. 2). We propose to double the initial resolution of the input views and the depth maps, and perform the view synthesis on these interpolated images. For example, in the proposed approach, the FullHD sequences would be processed in the 4K resolution.

Of course, together with the views and the depth maps also the intrinsic parameters of all the cameras (all real cameras and the virtual one) should be changed. The initial matrix of the intrinsic parameters [17] is defined as:

$$\mathbf{K} = \begin{bmatrix} f \cdot s_x & f \cdot s_k & o_x \\ 0 & f \cdot s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where f is the focal length, s_x and s_y are the sampling periods in the horizontal and the vertical direction, s_k – the skew factor of the camera converter, o_x and o_y are the principal point of the camera.

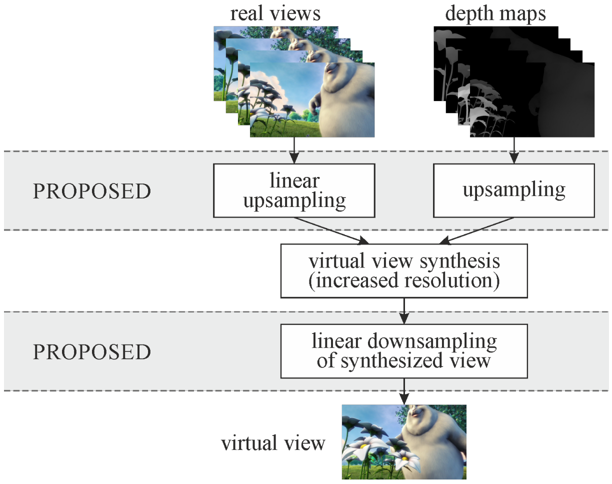

Fig. 2. View synthesis with view and depth map upsampling.

The same matrix for views of doubled resolution is:

$$\mathbf{K} = \begin{bmatrix} 2 \cdot f \cdot s_x & f \cdot s_k & 2 \cdot o_x \\ 0 & 2 \cdot f \cdot s_y & 2 \cdot o_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

The skew factor describes the angle between x and y pixel axes, thus it is not doubled. Also the extrinsic parameters, containing information about rotation and translation of each camera are not modified because the relative camera arrangement does not change.

In the proposed approach, remaining samples in the upsampled real view are interpolated bi-linearly. Obviously, such an approach causes blurring of the edges in the interpolated view but the virtual view's artifacts caused by blurred edges can be removed using algorithm presented in the Section III.

The depth maps differ from the views by the content: they contain sharp edges but nearly no texture. Therefore, in the proposed method, the interpolation of the real depth maps is more sophisticated. In the first step, between each pair of neighboring points of depth map an empty sample is inserted (horizontally and vertically) – Fig. 3.

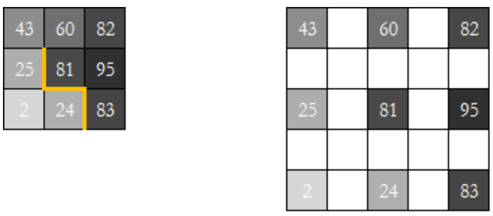

Fig. 3. Depth map upsampling; input (left) and upsampled depth map (right).

Then, empty points of the upsampled depth maps should be filled in. Of course, a simple, bi-linear interpolation of depth samples could be applied (Fig. 4A). However, in such approach the edge of the object (marked with an orange line in Fig. 3A) disappear. Moreover, in the interpolated depth map, there are incorrect depth values at the objects' edges, which may cause appearance of artifacts in the synthesized view.

On the other hand, 0th order interpolation could be used – all the remaining samples could be just copied from the left or top neighboring sample (Fig. 4C). That approach allows for preserving edges in the depth map. However, the method has also a weak point – smooth, continuous regions of the input depth map are stepped in the interpolated depth map.

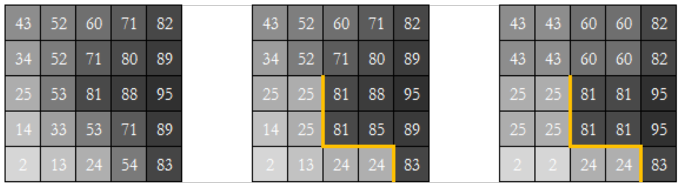

Fig. 4. Upsampled depth map with samples interpolated: linearly (left), using 0th order interpolation (right) and using proposed method (center).

Since two simple interpolation methods have strong disadvantages, the third one was created – mixed interpolation, combining advantages of the two approaches presented above. In this approach remaining depth samples are copied from their neighbors at the edges but they are bi-linearly interpolated in smooth regions.

In order to decide whether an analyzed region of the depth map is smooth or represent an edge, simple thresholding is used. For all the empty samples, the depth values from two neighbors are checked. If their depths differ too much, it is assumed that they represent two different objects in the scene, so there has to be an edge between them. The threshold was empirically set to 1% of depth range, e.g. for 8-bit depth maps it equals 3.

The proposed approach allows for a quality improvement of the synthesized views by 0.1 dB as compared to the two other interpolation approaches [18].

III. Gradient-Based View Filtering

The second proposed preprocessing method is the filtering of the real views or – more precisely – edges of the objects in these views. It is performed separately for all the real views. This step should be performed in order to eliminate two main problems causing artifacts in the virtual view:

1. edge dislocation between a real view and the corresponding depth map (Fig. 5A and Fig. 5B),
2. blurred edges in a real view (Fig. 5C and Fig. 5D).

The first problem is caused by erroneous estimation of the depth, the second one – by imperfect spatial sampling in a camera sensor (nonzero pixel size and limited spatial sampling frequency).

The abovementioned problems affect the quality of the synthesized virtual view. When they are not eliminated, the ghost edges appear in the virtual view (Fig. 6A and Fig. 6B). In Fig. 6C and Fig. 6D, the effects of the proposed real view filtering method are presented.

The virtual view synthesis with the proposed algorithm is illustrated in Fig. 7. As it is shown, the depth information is used to guide the filtering of the real views. Once the preprocessing step is finished, the filtered real views and unmodified

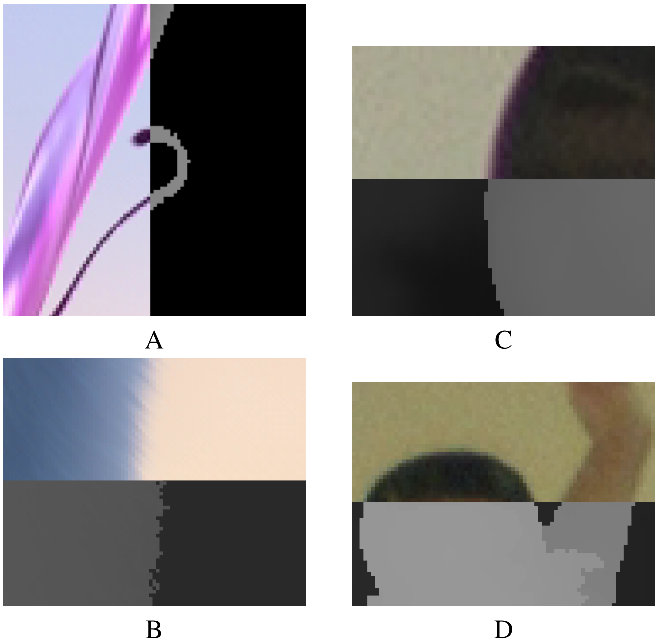

Fig. 5. Problems caused by imperfect objects' edges: edge dislocation between the view and the depth map (A, B), blurred edge in the real view (C, D). Fragments of the real views are presented at the left side of A and at the bottom of B, C and D. Fragments of the depth maps are presented at the right side of A and at the top of B, C and D.

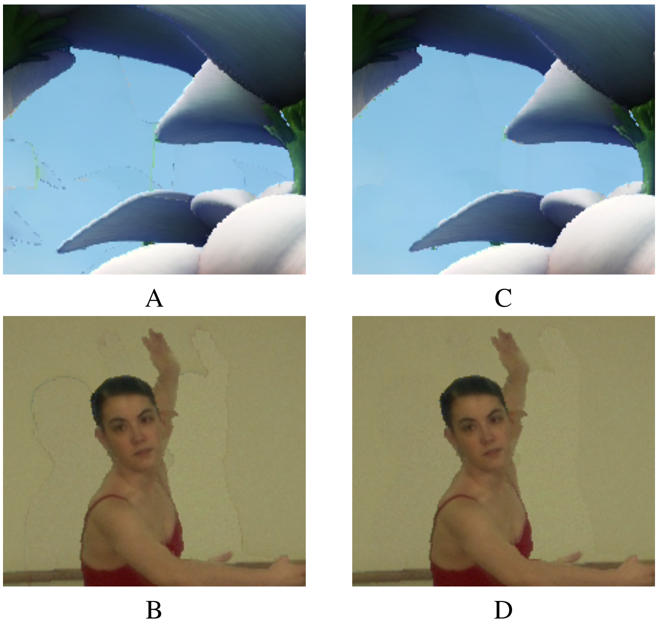

Fig. 6. Virtual view synthesis for blurred objects in the real views: without view filtering (A, B) and with gradient-based view filtering (C, D).

input depth maps are used for virtual view synthesis. Filtered points of the real view have to satisfy two conditions:

1. are placed at the edges (horizontal or vertical) in the depth map,

2. represent farther object (point at the other side of the edge has lower depth value), because the artifacts in the virtual view appear in the background.

All the remaining points of the real views are not modified.

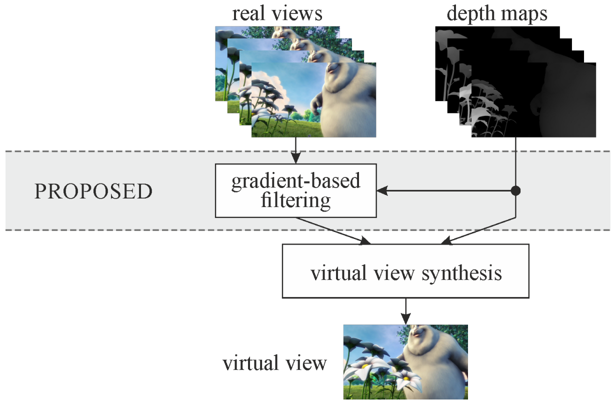

Fig. 7. View synthesis process with gradient-based view filtering.

In the proposed approach, for every point of the real view 6 gradients are estimated – for each color component YCbCr the vertical and horizontal gradient is estimated.

For any point satisfying two abovementioned conditions (placed at the edges and representing farther object) it is checked, whether its absolute value of each gradient is smaller than the averaged absolute value for three neighboring points. If this condition is not satisfied for at least 1 of 6 gradients, that particular point is filtered out thus will not be projected to the virtual view.

In Fig. 8, an example for analyzing one of the gradients (horizontal luma gradient) is presented.

50	50	49	48	47	**30**	3	2	1
51	50	49	47	46	**6**	4	1	2
50	49	45	43	**23**	3	0	2	0
48	45	41	42	**20**	6	0	1	1
45	48	44	45	**6**	4	1	3	1
47	48	47	**15**	3	2	1	2	1
50	48	44	**5**	3	0	1	1	2
48	50	**30**	3	0	0	2	0	3

A

0	1	1	1	17	**27**	1	1	1
1	1	2	1	40	**2**	3	1	2
1	4	2	20	**20**	3	2	2	0
3	4	1	22	**14**	6	1	0	1
3	4	1	39	**2**	3	2	2	1
1	1	32	**12**	1	1	1	1	1
2	4	39	**2**	3	1	0	1	2
2	20	**27**	3	0	2	2	3	3

C

5	5	5	5	5	**10**	10	10	10
5	5	5	5	5	**10**	10	10	10
5	5	5	5	**10**	10	10	10	10
5	5	5	5	**10**	10	10	10	10
5	5	5	5	**10**	10	10	10	10
5	5	5	**10**	10	10	10	10	10
5	5	5	**10**	10	10	10	10	10
5	5	**10**	10	10	10	10	10	10

B

50	50	49	48	47		3	2	1
51	50	49	47	46	**6**	4	1	2
50	49	45	43		3	0	2	0
48	45	41	42		6	0	1	1
45	48	44	45	**6**	4	1	3	1
47	48	47		3	2	1	2	1
50	48	44	**5**	3	0	1	1	2
48	50		3	0	0	2	0	3

D

Fig. 8. Fragment of the real view (A), corresponding depth map (B), absolute values of horizontal gradients in the real view (C) and real view with eliminated blurred edges (D). Analyzed points are highlighted by bolded font and thick frame.

In Fig. 8B, a fragment of the real depth map is presented (higher value means, that the particular point represents farther object). Analyzed points (highlighted by thick frame and bolded font) are the points with higher depth value, neighboring to points with lower depth value.

Fig. 8A contains corresponding fragment of the real view. Highlighted (bolded font) points are analyzed and, possibly, filtered.

In Fig. 8C, the absolute values of horizontal gradients in fragment from Fig. 9A are shown.

For each analyzed point (bolded font) it is checked whether the absolute value of its gradient is higher than the averaged absolute gradient of three closest points at the right side. If it is true, the analyzed point is eliminated and not used in virtual view synthesis (white points in Fig. 8D).

IV. Combined preprocessing

Both presented methods provide solutions to different problems of virtual view synthesis. Therefore, they can be combined in order to improve the quality of a synthesized virtual view even more.

The scheme of the proposed approach is presented in Fig. 9. In the first step, the real views are filtered in order to remove corrupted edges of the objects. Then, the previously filtered real views and input depth maps are upsampled.

After virtual view synthesis the resolution of the synthesized view is reduced in order to produce output image of the same size as input images.

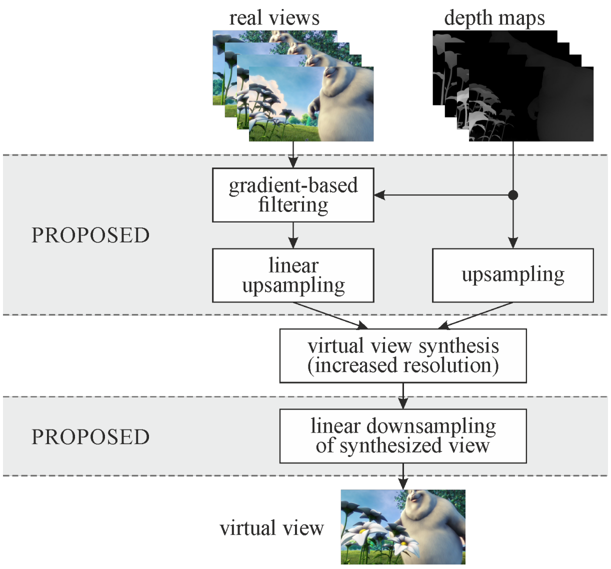

Fig. 9. View synthesis process with preprocessing.

V. Experiments

For the experiments, the MVS technique [11] was chosen as a view synthesis algorithm, because it leads to significantly better quality of the synthesized views than the state-of-the-art method VSRS [10].

In order to estimate quality gain provided by the proposed preprocessing methods, all the virtual views were synthesized in four ways:

1. without preprocessing,
2. with view and depth map upsampling,
3. with gradient-based view filtering,
4. with both preprocessing steps.

Quality measured for setups 2 – 4 was compared to the quality of the views synthesized without any preprocessing step.

For quality comparison, two widely used quality metrics were chosen: PSNR and SSIM [19]. In order to estimate values of these metrics, virtual views were synthesized in the viewpoints of some real cameras. For example, views from cameras 0, 1, 3, 4 were used to synthesize virtual view placed in position of camera 2.

The proposed techniques were tested on the set of 12 miscellaneous test sequences presented in Table I.

7 of 12 test sequences (from *Poznan Blocks* to *Soccer Arc* in Table I) were acquired using sparse multicamera systems with cameras arranged on an arc. Both *Big Buck Bunny* sequences are synthetic, with 79 real views. For the tests only 7 evenly distributed views were used, providing sparse camera arrangement. Last three sequences were captured by linear camera systems.

TABLE I
Test Sequences

Sequence	Source
Big Buck Bunny Butterfly	Holografika [20]
Big Buck Bunny Flowers	Holografika [20]
Poznan Blocks	Poznan Univ. of Technology [21]
Poznan Blocks2	Poznan Univ. of Technology [22]
Poznan Fencing2	Poznan Univ. of Technology [22]
Poznan Service2	Poznan Univ. of Technology [22]
Ballet	Microsoft Research [23]
Breakdancers	Microsoft Research [23]
Soccer Arc	Hasselt University [24]
Soccer Linear	Hasselt Univeristy [24]
Poznan Carpark	Poznan Univ. of Technology [25]
Poznan Street	Poznan Univ. of Technology [25]

For all the sequences, at least 5 different virtual views were synthesized and compared. Duration of all test sequences was limited to 10 consecutive frames.

VI. Experimental results

A. View and depth map upsampling

Results presented in Tables II and III show that upsampling of real views and depth maps provides higher quality of synthesized virtual views. In average, additional upsampling increases PSNR by 0.9 dB. For sequences captured by sparse multicamera systems with cameras arranged on an arc, the gain is higher than for linear systems: 1.01 dB vs. 0.47 dB.

Similar results were obtained for the SSIM measure – overall gain is 0.017 and for non-linear camera setups: 0.023. For linear multicamera systems, the gain is negligible.

B. Gradient-based view filtering

The proposed gradient-based filtering leads to (in average) 0.1 dB higher PSNR of the synthesized views (0.16 dB for

TABLE II
QUALITY OF VIEWS SYNTHESIZED WITH AND WITHOUT VIEW AND DEPTH MAP UPSAMPLING (LUMA PSNR [DB])

Sequence	Without upsampling	With upsampling
BBB Butterfly	31.638	32.846
BBB Flowers	24.498	25.031
Poznan Blocks	24.667	26.141
Poznan Blocks2	29.159	29.425
Poznan Fencing2	29.282	30.107
Poznan Service2	24.161	26.917
Ballet	28.534	29.675
Breakdancers	30.566	31.441
Soccer Arc	20.964	21.303
Soccer Linear	33.191	33.191
Poznan Carpark	32.371	33.066
Poznan Street	33.942	34.650
Average	28.581	29.483

TABLE III
QUALITY OF VIEWS SYNTHESIZED WITH AND WITHOUT VIEW AND DEPTH MAP UPSAMPLING (LUMA SSIM)

Sequence	Without upsampling	With upsampling
BBB Butterfly	0.953	0.959
BBB Flowers	0.845	0.854
Poznan Blocks	0.769	0.794
Poznan Blocks2	0.844	0.854
Poznan Fencing2	0.870	0.887
Poznan Service2	0.796	0.871
Ballet	0.817	0.833
Breakdancers	0.815	0.840
Soccer Arc	0.749	0.770
Soccer Linear	0.885	0.885
Poznan Carpark	0.922	0.922
Poznan Street	0.926	0.927
Average	0.849	0.866

non-linear camera setups and -0.09 dB for linear arrangements). For SSIM, overall gain is 0.004, while for arc setups it is 0.005. Similar to upsampling method, SSIM decrease for linear setups in negligible.

Quality improvement caused by the proposed filtering method is significantly lower than for view and depth upsampling, but – what is important for free navigation systems – it significantly reduces amount of artifacts in the virtual views, thus improving subjective quality of virtual navigation.

C. Combined preprocessing

The results presented in Tables VI and VII show that combination of the two presented preprocessing methods leads to a further increase of the quality of the synthesized views.

For PSNR, overall quality gain is 0.93 dB, while for non-linear camera setups it is 1.1 dB compared to views synthesized without preprocessing. Also SSIM values are higher when two proposed methods are used: 0.019 overall and 0.025 for cameras located on an arc.

TABLE IV
QUALITY OF VIEWS SYNTHESIZED WITH AND WITHOUT PROPOSED FILTERING METHOD (LUMA PSNR [DB])

Sequence	Without filtering	With filtering
BBB Butterfly	31.638	32.245
BBB Flowers	24.498	24.632
Poznan Blocks	24.667	24.676
Poznan Blocks2	29.159	29.335
Poznan Fencing2	29.282	29.437
Poznan Service2	24.161	24.204
Ballet	28.534	28.689
Breakdancers	30.566	30.685
Soccer Arc	20.964	21.045
Soccer Linear	33.191	33.116
Poznan Carpark	32.371	32.234
Poznan Street	33.942	33.878
Average	28.581	28.681

TABLE V
QUALITY OF VIEWS SYNTHESIZED WITH AND WITHOUT PROPOSED FILTERING METHOD (LUMA SSIM)

Sequence	Without filtering	With filtering
BBB Butterfly	0.953	0.959
BBB Flowers	0.845	0.854
Poznan Blocks	0.769	0.779
Poznan Blocks2	0.844	0.847
Poznan Fencing2	0.870	0.872
Poznan Service2	0.796	0.801
Ballet	0.817	0.824
Breakdancers	0.815	0.816
Soccer Arc	0.749	0.750
Soccer Linear	0.885	0.886
Poznan Carpark	0.922	0.921
Poznan Street	0.926	0.925
Average	0.849	0.853

TABLE VI
QUALITY OF VIEWS SYNTHESIZED WITH AND WITHOUT THE WHOLE PROPOSED PREPROCESSING (LUMA PSNR [DB])

Sequence	Without preprocessing	With preprocessing
BBB Butterfly	31.638	32.947
BBB Flowers	24.498	25.036
Poznan Blocks	24.667	26.145
Poznan Blocks2	29.159	29.563
Poznan Fencing2	29.282	30.184
Poznan Service2	24.161	26.946
Ballet	28.534	29.687
Breakdancers	30.566	31.512
Soccer Arc	20.964	21.330
Soccer Linear	33.191	33.177
Poznan Carpark	32.371	32.942
Poznan Street	33.942	34.635
Average	28.581	29.509

TABLE VII
QUALITY OF VIEWS SYNTHESIZED WITH AND WITHOUT THE WHOLE PROPOSED PREPROCESSING (LUMA SSIM)

Sequence	Without preprocessing	With preprocessing
BBB Butterfly	0.953	0.961
BBB Flowers	0.845	0.856
Poznan Blocks	0.769	0.802
Poznan Blocks2	0.844	0.855
Poznan Fencing2	0.870	0.887
Poznan Service2	0.796	0.875
Ballet	0.817	0.836
Breakdancers	0.815	0.840
Soccer Arc	0.749	0.770
Soccer Linear	0.885	0.885
Poznan Carpark	0.922	0.921
Poznan Street	0.926	0.926
Average	0.849	0.868

For linear camera arrangements the best choice is to simply increase resolution of input views and depth maps – without filtering.

VII. CONCLUSIONS

In this paper, preprocessing methods for the real views and the corresponding depth maps are studied in the context of improvements of fidelity of the virtual views produced by Depth-Image-Based rendering (DIBR).

Two basic preprocessing techniques are considered in detail:

- The interpolation of the real views acquired from cameras, and interpolation of the depth maps that correspond to these views,
- The edge filtering in real views the filtering is guided by depth.

The abovementioned preprocessing provide improvements of the fidelity of the virtual views obtained by DIBR. In order to obtain the results that are relevant to the state-of-the-art techniques, the experiments were designed in the context of the technique MVS previously proposed by the authors. The experimental results reported in the references state that the MVS technique clearly outperforms [11] the well-known and widely-used VSRS of MPEG.

Therefore, the fidelity of the synthesis of virtual views is assessed by the comparisons between the real views and the collocated virtual views synthesized by the use of the preprocessing described in the paper and the MVS technique. The influence of the preprocessing is measured by repeating the abovementioned experiments with the preprocessing (or its part) switched on and off. As the respective quality differences are small, only objective measures of the quality differences are used whereas the subjective quality comparisons are omitted due to the problems in subjective assessment of differences between views with similar quality.

For the synthesis of the virtual views, the extensive experiments demonstrate that the proposed preprocessing provides significant quality improvement in average of about 1 dB in PSNR for luma. What is promising, the preprocessing techniques seem to be particularly beneficial for the multiview video obtained from the cameras located relatively sparsely on an arc around a scene. The experiments demonstrate that for such sequences the increase in PSNR is higher than the average.

Therefore, the preprocessing techniques may be useful in particular in applications to virtual navigation, free-viewpoint television and other virtual reality systems.

REFERENCES

[1] M. Tanimoto, M. P. Tehrani, T. Fujii, T. Yendo ”FTV for 3-D spatial communication”, *Proc. IEEE*, vol. 100, no. 4, pp. 905-917, 2012.

[2] G. Lafruit, M. Domanski, K. Wegner, T. Grajek, T. Senoh, J. Jung, P. Kovacs, P. Goorts, L. Jorissen, A. Munteanu, B. Ceulemans, P. Carballeira, S. Garcia, M. Tanimoto, ”New visual coding exploration in MPEG: Super-MultiView and Free Navigation in Free viewpoint TV”, *IST Electronic Imaging, Stereoscopic Displays and Applications XXVII*, San Francisco 2016.

[3] M. Domanski, M. Gotfryd, K. Wegner, ”View synthesis for multiview video transmission”, *The Int. Conf. Image Proc., Computer Vision, and Pattern Recognition*, 2009.

[4] P. Goorts, M. Dumont, S. Rogmans, P. Bekaert, ”An end-to-end system for free viewpoint video for smooth camera transitions”, *International Conference on 3D Imaging, IC3D 2012*, Liege, Belgium, 3-5.12.2012.

[5] M. Domanski, A. Dziembowski, A. Kuehn, M. Kurc, A. Luczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner, ”Experiments on acquisition and processing of video for free-viewpoint television”, *3DTV Conference 2014*, Budapest, Hungary, 2-4.07.2014.

[6] O. Stankiewicz, M. Domanski, A. Dziembowski, A. Grzelka, D. Mieloch, J. Samelak, ”A free-viewpoint television system for horizontal virtual navigation”, *IEEE Transactions on Multimedia*, early access, 2018.

[7] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, Y. Suenaga, ”Multipoint measuring system for video and sound - 100-camera and microphone system”, *IEEE International Conference on Multimedia and Expo, ICME 2006*, Toronto, Canada, 9-12.07.2006.

[8] M. Tanimoto, ”Overview of FTV (free-viewpoint television)”, *IEEE International Conference on Multimedia and Expo, ICME 2009*, New York, United States, 28.06-3.07.2009.

[9] Z. Sun; Ch. Jung ”Real-Time Depth-Image-Based Rendering on GPU,” *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Xi’an, 2015, pp: 324 – 328.

[10] T. Senoh, K. Yamamoto, N. Tetsutani, H. Yasuda, K. Wegner, ”View Synthesis Reference Software (VSRS) 4.2 with improved inpainting and hole filling”, *ISO/IEC JTC1/SC29/WG11 MPEG2013, M40657*, Hobart, Australia, 3-7.04.2017.

[11] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, K. Wegner, M. Domanski, ”Multiview Synthesis – improved view synthesis for virtual navigation”, *Picture Coding Symposium*, 2016.

[12] S.Zinger, Q. Do, P. de With, ”Free-viewpoint depth image based rendering”, *Journal of Visual Communication and Image Representation*, vol. 21, no. 5-6, pp. 533-541, 2010.

[13] I. Ahn, C. Kim, ”A novel depth-based virtual view synthesis method for free viewpoint video”, *IEEE Transactions on Broadcasting*, vol. 59, pp. 614-626, 2013.

[14] B. Ceulemans, S.P. Lu, G. Lafruit, A. Munteanu, ”Robust multiview synthesis for wide-baseline camera arrays”, *IEEE Transactions on Multimedia*, vol. PP, no. 99, pp. 1-1, 2018.

[15] C. Lee, Y.S. Ho, ”Boundary filtering on synthesized views of 3D video”, *2nd International Conference on Future Generation Communication and Networking, FGCN 2008*, Sanya, China, 13-15.12.2008.

[16] K. Mueller, A. Smolic, K. Dix, P. Merkle, P. Kauff, T. Wiegand, ”View synthesis for advanced 3D video systems”, *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1-11, 2008.

[17] A. Heyden, M. Pollefeys, ”Multiple view geometry”, in: G. Medioni, S.B. Kang (eds.) ”Emerging Topics in Computer Vision”, *Prentice Hall*, s. 63-75, 2005.

[18] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, M. Domanski, ”Enhancing view synthesis with image and depth map upsampling”, *International Conference on Systems, Signals and Image Processing, IWSSIP 2017*, Pozna, Poland, 22-24.05.2017.

[19] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, ”Image quality assessment: from error visibility to structural similarity”, *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 04.2004.

[20] P.T. Kovacs, A. Fekete, K. Lackner, V.K. Adhikarla, A. Zare, T. Balogh, ”[FTV AHG] Big Buck Bunny light-field test sequences”, *ISO/IEC JTC1/SC29/WG11 MPEG2013, M36500*, Warsaw, Poland, 22-26.06.2015.

[21] M. Domanski, A. Dziembowski, A. Kuehn, M. Kurc, A. Luczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner, ”Poznan Blocks - a multiview video test sequence and camera parameters for free viewpoint television”, *ISO/IEC JTC1/SC29/WG11 MPEG2014, M32243*, San Jose, United States, 13-17.01.2014.

[22] M. Domanski, A. Dziembowski, A. Grzelka, A. Luczak, D. Mieloch, O. Stankiewicz, K. Wegner, ”Multiview test video sequences for free navigation exploration obtained using pairs of cameras”, *ISO/IEC JTC1/SC29/WG11 MPEG2016, M38247*, Geneva, Switzerland, 30.05-3.06.2016.

[23] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, ”High-quality video view interpolation using a layered representation”, *ACM Transactions on Graphics*, vol. 3, no. 23, pp. 600-608, 08.2004.

[24] P. Goorts, ”Real-time adaptive plane sweeping for free viewpoint navigation in soccer scenes”, *doctoral dissertation*, Hasselt University, Belgium, pp. 175-186, 16.06.2014.

[25] M. Domanski, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, K. Wegner, ”Poznan multiview video test sequences and camera parameters”, *ISO/IEC JTC1/SC29/WG11 MPEG2009, M17050*, Xi’an, China, 26-30.10.2009.