Point-to-Block Matching in Depth Estimation

Dawid Mieloch Dominika Klóska Magdalena Woźniak
dawid.mieloch@put.poznan.pl

Institute of Multimedia Telecommunications, Poznań University of Technology
Polanka 3, 61-131 Poznań, Poland

ABSTRACT

In this paper, the novel correspondence search method called point-to-block matching was proposed. Recently, in many proposed multimedia systems, depth estimation is performed on compressed input views. To address this problem and increase the quality of such depth maps, in the proposal, a point in a view is not compared simply with a point in another view, but to the most similar point in a small block surrounding it. The introduction of this method in the depth estimation process is beneficial to the quality of depth maps, as it decreases the influence of small shifts in images, caused e.g., by encoding-related errors introduced to input views. The method was implemented in one of the state-of-the-art depth estimation methods and tested in series of experiments. Based on the comparison of synthesized virtual views with the input views, the proposal increases the quality of estimated depth maps in most of the tested configurations.

Keywords

Depth maps, depth estimation, inter-view correspondence, similarity metric, virtual view synthesis.

1. INTRODUCTION

The methods of creating natural three-dimensional content are recently under constant development, as the emergence of new applications of immersive media systems such as free-viewpoint television [Tan12] and virtual reality systems can be easily seen [Dom17] [Laf16].

One of the most common representations of such multimedia content is the use of depth maps that correspond to each view captured by a camera system and show the distance from the acquired 3D point to the camera [Mül11]. Depth maps can be used to synthesize virtual views, so the user can freely move and see an acquired three-dimensional scene from any viewport.

In its essence, the depth estimation methods are based on the inter-view correspondence search. The result and efficiency of this search are dependent on a similarity of a point in one view of the scene to a point that most probably shows the same part of the scene in another view [Tip13]. Unfortunately, this search can be very difficult due to factors that we will summarize below.

Recently, in many proposed multimedia systems, the depth estimation is proposed to be performed using compressed input views, e.g., in the representation server that sends the requested virtual view to the user [Dzi16], or directly, in the client-side decoder of the multi-view image encoder [Gar19]. Therefore, the depth estimation methods should take into account that some encoding-related errors can be present in used videos.

The comparison of uncompressed and compressed input view can be seen in Figure 1, while Figure 2 shows the difference between uncompressed and compressed images, unveiling artifacts introduced by the high compression. The main differences are clearly visible in highly-textured areas and all edges visible in the scene.

Figure 1. Input view of Fencing sequence [Dom16] without compression (A) and with high (B) / low (C) compression.

The presence of high-frequency harmonics is necessary for the correct representation of the edges in the image, unfortunately, due to coding, when the

transform coefficients are quantized, high-frequency components are notched out, which in turn leads to blurring or, in the worst case, an edge shift in the image. Such shift can also occur in highly compressed videos because of the high share of temporally predicted parts of videos [Leo07].

Figure 2. The logarithm of difference between compressed and uncompressed input view of Fencing sequence for high compression. The contrast of the image was increased to increase the visibility of differences.

Depth maps are of course most often estimated before any compression of the representation of a scene. In this case, the input views are usually uncompressed, so the compression-related distortions do not influence the quality of depth maps. On the other hand, this quality can be decreased due to low accuracy of camera parameters (understood as the accuracy of camera distortion models [Tan17] and intrinsic and extrinsic camera parameters [San17] [Che21]), and the use of floating-point representation [God20] and rounding errors.

Other causes of depth estimation errors are related to inter-view correspondence search can be seen also in noise present in input views of natural sequences [Sta14], as such noise influences the original color of points.

In order to address all listed problems, we propose the novel correspondence search method called point-to-block matching. In the proposal, a point in a view is not compared simply with a collocated point in another view, but to the most similar point in a small block surrounding it.

The introduction of such a degree of freedom in the estimation can be very beneficial on the quality of depth maps, as it decreases the influence of small shifts in images. Moreover, the proposal also can decrease the noise-induced errors, as the proposed method lets us find a less noised similar point in the nearest neighborhood.

To fully test the proposal, the point to block matching was implemented in one of the publicly available state-of-the-art methods of depth estimation [Mie20a] and tested in 10 different configurations to find the most proper applications in which the proposed matching should be used.

The organization of the paper is as follows: Section 2 shows the description of the most relevant similarity measurement methods, with emphasis on their usefulness in the case of use of compressed views in the depth estimation process; Section 3 describes in details the proposed inter-view matching method; Section 4 includes the results of the comparison of the proposed point-to-block matching with the traditional point-to-point matching.

In the end, Section 5 summarizes the paper and includes conclusions drawn from the performed experiments.

2. RELATED WORKS

Some studies were already performed in order to evaluate the efficiency of the existing matching or similarity measurement methods in various cases, e.g., for noised views. Study [Weg09] has shown that a mixture of census or rank transforms with the simple sum of absolute differences (SAD) or sum of squared differences (SSD) can be very efficient for such multi-view sequences. [Zeg20] proves that the use of structural similarity performs very well when radiometric distortions are present in processed input views.

Authors of [Say15] indicate that similarity measurement should be adapted to the local features of input views. It can be used to change the weight of used mixture similarity metrics and lead to better estimation in problematic areas, such as low-texture regions or occluded parts of views. Another method that tries to adapt to processed views is [Cha18], but in this proposal, the size and shape of the matching window become dependent on the characteristics of the part of the image.

Other methods were proposed to improve the efficiency of inter-view matching by introducing temporal information into the similarity metric [Shi17]. The temporal stability of depth maps is crucial for the quality of the final view synthesis. On the other hand, such stability can be ensured on other stages of the depth estimation process, e.g., by post-processing filtering [Köp13].

An interesting method is proposed by [Suo12]. Its authors propose to perform the matching of views in the transform domain. Such an approach results in exposure-independent estimation, very valuable when multi-view systems are used. However, if the encoding of input views would be performed as simulcast compression (independent in each view, e.g., performed in each camera as post-processing step), the distribution of frequencies could significantly vary in compressed views, negatively influencing the efficiency of transform domain matching.

Matching methods base on minimum spanning tree, such as [Zha19], show high robustness for complex scenes, however, no qualitative results of efficiency for noised or compressed images are provided. This method uses also SURF [Bay08] and KAZE [Alc12] features detectors, which are translation invariant, so they could be also invariant to compression-induced errors, however, these detectors are used only for calculation of disparity range, not depth maps themselves.

To the best knowledge of the authors, contrary to the proposal presented in Section 3, none state-of-the-art method of inter-view correspondence search takes specifically into account the compression-induced artifacts present in the encoded input views.

3. PROPOSED INTER-VIEW MATCHING METHOD

The problem of the depth map estimation can be in most cases presented as a cost (goal) function minimization [Kol02]. In its most commonly used form, the cost function is defined as:

$$E(\bar{d}_p) = \sum_{p \in \mathrm{P}} D_p(\bar{d}_p) + \sum_{p \in \mathrm{P}} \sum_{q \in \mathrm{Q}} V_{p,q}(\bar{d}_p, \bar{d}_q), \quad (1)$$

where P is set of points of an input view, p is point of this view, $\bar{d}_p$ is currently considered depth of a point p, D_p is data term that represents a cost of assigning the depth d_p to the point p, Q is set of points in the neighborhood of p, q is point in this neighborhood, $\bar{d}_q$ is currently considered depth of a point q, $V_{p,q}$ is smoothness term that represents the intra-view discontinuity cost of using the depth values d_p and d_q.

The proposed method changes the data term D_p, which is responsible for the correspondence between points in different views. D_p usually uses some similarity metric (e.g., one of metrics described in Section 2) between a point p and a point p', which is a point in neighboring view that corresponds to the point p for the considered depth of a point ($\bar{d}_p$).

In the proposal, the data term compares the point p with points in a small block of points R that surround point p' and chooses the point that gives the lowest matching error (highest similarity) to point p. So, if the SAD metric would be used, then:

$$D_p(\bar{d}_p) = \min\big(SAD(p,r)\big), \; r \in R. \quad (2)$$

Therefore, if 3×3 block would be used as R, the $D_p(\bar{d}_p)$ would be equal to the smallest of 9 SAD values, calculated for the central point p' and its 1- and 2- pixel neighborhood, respectively.

Note that any similarity measure can be used with the proposal. Therefore, such a method can enhance almost any depth estimation method with robustness to small shifts in images caused e.g., by image compression and decrease the noise-induced errors.

The drawback of the proposal can be seen in the small increase of the computational complexity, as the similarity is now performed not once for each point, but R^2 times, nevertheless, the calculation of similarity can be easily parallelized.

As the proposal was implemented in the depth estimation software provided in [Mie20a], the Graph Cut method [Kol02] was used to minimize the function (1).

4. EXPERIMENTAL RESULTS

The performed experiments were performed for different configurations of the modified depth estimation software [Mie20a]:

1. Standard parameters:
   - 256 levels of depth,
   - 50,000 superpixels per view,
   - uncompressed input views,
   - all views used in the estimation.
2. Decreased number of depth levels:
   2.1. 16 levels of depth,
   2.2. 128 levels of depth.
3. Changed number of superpixels [Ach12]:
   3.1. 5000 superpixels per view,
   3.2. 100000 superpixels per view.
4. Compression of input views:
   4.1. Low compression,
   4.2. Medium compression,
   4.3. High compression.
5. Decreased number of used cameras:
   5.1. 70% of cameras,
   5.2. 20% of cameras.

All configurations were tested 3 times: for the unmodified depth estimation and the point-to-point matching performed in 3×3 and 5×5 blocks. The use of larger blocks is possible, but as preliminary tests have shown, does not have a large influence on the estimated depth maps.

The quality of estimated depth maps directly affects the final product, which is the most faithful representation of the three-dimensional scene. Nevertheless, in the case of using depth maps for the view synthesis purposes, such faithfulness is understood more as the ability to provide virtual views of high quality, rather than the low absolute error in comparison with ground-truth depth [Fan16]. Moreover, such ground-truth depth maps are not available for natural multi-view sequences, making such comparison not possible.

Therefore, in order to test the proposal, a set of estimated depth maps was used for the synthesis of

virtual views which were placed in the same position as the real cameras that captured a scene. The synthesized views were in the end compared with the original input views. The objective comparison was performed using IV-PSNR [MPEG20], which is the PSNR-based metric used by the ISO/IEC MPEG Video Coding to evaluate immersive video quality. The calculated IV-PSNR was averaged for all views in each test sequence (listed in Table 1).

The experiments were performed for 5 multiview test sequences that vary in the number of input views and their arrangements.

Sequence name	Number of views	Resolution	Source
Carpark	9	1920×1088	[Mie20b]
Fencing	10	1920×1080	[Dom16]
Frog	13	1920×1080	[Sal18]
Painter	16	2048×1088	[Doy18]
Street	9	1920×1088	[Mie20b]

Table 1. Test sequences used in experiments.

The virtual view synthesis software used in the experiments is VVS 2.0 with its default configuration [Boi19], i.e., the synthesis of each virtual view was performed using 4 nearest views (and corresponding depth maps).

The software used for compression of input views is x265 [X265] which is the implementation of the MPEG HEVC encoder. The values of the quality parameter *crf* were equal to 0 (low compression, near-lossless), 26 (medium compression), and 51 (high compression).

The following five subsections include the description of acquired results for each test sequence. Color green in tables denotes that the quality of the proposal (point-to-block matching) was higher than for unmodified depth estimation. The last subsection contains a summary of performed experiments.

Carpark

Table 2. shows the average IV-PSNR of synthesized views for the Carpark sequence.

For 8 out of 10 the point-to-block matching 3×3 provides a gain in the quality, making this option the best for this sequence. On the other hand, the highest gain (more than 3 dB) can be seen for the number of superpixels decreased to 5000 for 5×5 block.

Fencing

For the Fencing sequence, the results, presented in Table 3, show that the proposed matching provides a gain in the quality mainly for the compressed input views. This sequence is the only one with a non-planar camera arrangement, which can suggest that in such a case, the use of point-to-block matching is not beneficial when uncompressed input views are used.

	Average IV-PSNR [dB] of synthesized view		
Depth estimation configuration	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	37.09	37.19	37.11
16 levels of depth	36.09	36.13	36.13
128 levels of depth	37.07	37.20	37.16
5000 superpixels	32.28	32.39	35.55
100000 superpixels	38.28	38.35	38.21
Low compression	37.10	37.23	37.14
Medium compression	37.36	37.28	37.18
High compression	34.08	34.17	34.20
70% of cameras	36.19	36.21	35.95
20% of cameras	35.21	35.17	35.21

Table 2. Results of experiments for the Carpark sequence.

	Average IV-PSNR [dB] of synthesized view		
Depth estimation configuration	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	40.24	40.13	39.90
16 levels of depth	35.40	35.55	35.72
128 levels of depth	40.56	40.23	40.05
5000 superpixels	37.87	37.77	37.71
100000 superpixels	40.89	40.87	40.55
Low compression	40.49	40.10	39.98
Medium compression	38.19	39.03	39.14
High compression	34.09	36.19	36.18
70% of cameras	33.80	39.65	39.06
20% of cameras	38.41	38.40	38.43

Table 3. Results of experiments for the Fencing sequence.

Frog

In the Frog sequence, the best results were obtained for Point-to-block matching in 3×3 block (Table 4).

One of the possible explanations to results observed for Frog can be a very high amount of noise present in this sequence. Table 5 shows the standard deviation of noise estimated for each sequence (from [Dzi20]). As it can be seen, Frog has more than twice the amount of noise than other sequences. Therefore, in configurations without the compression of input views, the point-to-block matching provides a gain in most cases, as it decreases the influence of noise on the inter-view matching.

On the other hand, compression of video highly reduces the noise of sequences, even for low compression. In this sequence, objects have very detailed, rich textures (see Figure 3), so the changes in the position of edges in compressed views do not

occur so often as e.g., in the Fencing sequence, where the foreground and background objects have similar color characteristics. We can see that a high gain in the quality in Frog cannot be seen until high compression is applied.

	Average IV-PSNR [dB] of synthesized view		
Depth estimation configuration	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	38.95	39.00	38.54
16 levels of depth	32.63	32.97	33.05
128 levels of depth	38.99	39.01	38.66
5000 superpixels	37.27	36.97	36.53
100000 superpixels	39.07	39.16	38.62
Low compression	39.04	38.98	38.47
Medium compression	38.99	38.92	38.33
High compression	33.90	34.26	33.88
70% of cameras	38.05	37.89	37.58
20% of cameras	33.28	33.31	33.08

Table 4. Results of experiments for the Frog sequence.

Sequence	σ_Y	σ_U	σ_V
Carpark	10.29	4.23	4.57
Fencing	12.83	3.38	3.83
Frog	26.77	5.59	5.62
Painter	10.17	3.89	3.89
Street	9.85	3.84	4.21

Table 5. Estimated noise standard deviation for luma (σ_Y) and chromas (σ_U and σ_V) [Dzi20]. Evaluated sequences have 10 bits per sample.

Painter

Painter is the only sequence that shows decreased quality in almost all cases of estimation with the point-to-block matching (Table 6). This sequence is the only sequence with lightfield-like arrangement of cameras (matrix of cameras).

	Average IV-PSNR [dB] of synthesized view		
Depth estimation configuration	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	40.55	40.18	39.57
16 levels of depth	39.04	39.12	39.13
128 levels of depth	40.63	40.36	39.73
5000 superpixels	39.51	39.14	38.70
100000 superpixels	40.64	40.24	39.47
Low compression	40.68	40.17	39.86
Medium compression	39.87	39.66	39.04
High compression	36.32	35.85	35.58
70% of cameras	40.22	39.84	39.66
20% of cameras	40.31	40.12	39.70

Table 6. Results of experiments for the Painter sequence.

Street

Results for the Street sequence (Table 7) clearly benefit from using the proposed point-to-block matching for all tested configurations of depth estimation.

The gain of quality is very high in most cases (up to 3 dB). As it can be seen in Figure 4, most of the differences between synthesis that used depth maps with point-to-point matching and point-to-block matching are still visible mainly on the edges of objects.

	Average IV-PSNR [dB] of synthesized view		
Depth estimation configuration	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	37.34	40.23	40.09
16 levels of depth	36.14	39.09	39.03
128 levels of depth	37.23	40.23	39.98
5000 superpixels	31.99	32.75	32.83
100000 superpixels	38.34	40.43	39.95
Low compression	37.37	40.35	40.06
Medium compression	40.24	40.08	40.53
High compression	32.52	34.59	34.49
70% of cameras	35.34	37.99	38.04
20% of cameras	38.97	38.32	40.10

Table 7. Results of experiments for the Street sequence.

Summary of experiments

In order to summarize the performed experiments, the results were averaged for all sequences and presented in Table 8.

	Average IV-PSNR [dB] of synthesized view		
Depth estimation configuration	Point-to-point matching	Point-to-block matching (3×3)	Point-to-block matching (5×5)
Standard parameters	38.83	39.35	39.04
16 levels of depth	35.86	36.57	36.61
128 levels of depth	38.90	39.41	39.12
5000 superpixels	35.78	35.80	36.26
100000 superpixels	39.44	39.81	39.36
Low compression	38.94	39.37	39.10
Medium compression	38.93	38.99	38.84
High compression	34.18	35.01	34.87
70% of cameras	36.72	38.32	38.06
20% of cameras	37.24	37.06	37.30

Table 8. Results of experiments averaged for all sequences.

Although the results vary for different sequences, as was presented in previous subsections, these results show that for the used set of test sequences the point-to-block matching in 3×3 block was on average better than standard point-to-point matching in all but one

Figure 3. Comparison of virtual views synthesized using depth maps estimated using point-to-point matching (left column), point-to-block matching with 3×3 block (center), and with 5×5 block (right). Test sequences (top to bottom): Painter, Frog, Fencing, Carpark, Street.

tested configurations. Most significant gains can be observed for the reduced number of cameras, high compression, and levels of depth reduced from 256 to 16.

5. CONCLUSIONS

This paper presents the novel method of the inter-view correspondence search called point-to-block matching. In the proposal, a point in a view is not compared simply with a point in another view, but with the most similar point in a small block surrounding it.

Series of performed experiments based on the comparison of synthesized virtual views with the input views showed that on average the proposal increases the quality of estimated depth maps in almost all configurations. The results for individual sequences seem to confirm that the method provides the best results for highly compressed input views or when the

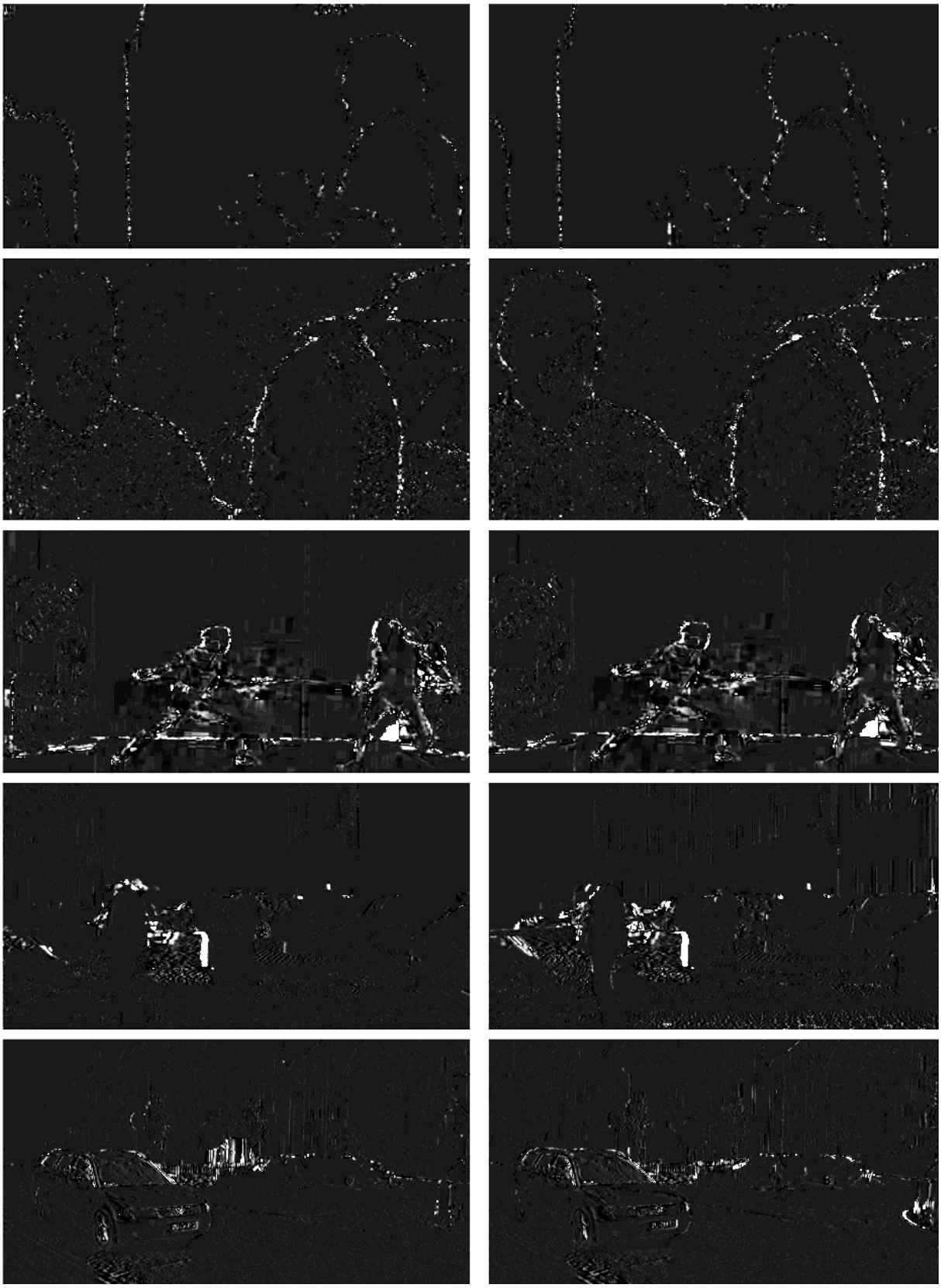

Figure 4. Logarithm of differences between virtual views synthesized using depth maps estimated using point-to-point matching and point-to-block matching with 3×3 block (left) / 5×5 block (right). Test sequences (top to bottom): Painter, Frog, Fencing, Carpark, Street. The contrast of the image was increased to increase the visibility of differences.

amount of noise present in a multi-view sequence is significant.

A gain in quality was also observed for all sequences in cases of decreased number of depth levels tested during the estimation process. Such estimation is much less computationally expensive, which indicates that the method could also be very effective in visual systems with limited resources (e.g., a typical PC with two internet cameras). Such systems almost always operate using compressed input video streams, therefore, the abovementioned efficiency of the proposal in such cases further increases its usability in such real-life configuration.

The need for further research on depth estimation methods adapted to be used with compressed input views can be seen also in the current state of the development of the new MPEG Immersive Video codec [Boy21] for virtual and augmented reality applications. This soon-to-be video encoding standard includes Geometry Absent profile [MPEG21] which can be used to send a subset of input views to the decoder, where the depth estimation process is performed to synthesize any viewport of the three-dimensional scene.

6. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education and Science of Republic of Poland.

7. REFERENCES

[Ach12] Achanta R., Shaji A., Smith K., Lucchi A., Fua P., and Süsstrunk S. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. 2012 IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, 2012.

[Alc12] Alcantarilla P. F., Bartoli A., and Davison A. J. KAZE features. Proceedings of European Conference on Computer Vision, pp. 214–227, 2012.

[Bay08] Bay H., Ess A., Tuytelaars T., and Van Gool T. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, vol. 110, Issue 3, pp. 346-359, 2008.

[Boi19] Boissonade P., and Jung J. [MPEG-I Visual] Improvement of VVS1.0.1. ISO/IEC JTC1/SC29/WG11/MPEG2019/m46263, Marrakesh, Jan. 2019.

[Boy21] Boyce J., Doré R., Dziembowski A., Fleureau J., Jung J., Kroon B., Salahieh B., Vadakital V., and Yu L. MPEG Immersive Video Coding Standard. Proceedings of the IEEE, early access. 2021.

[Cha18] Chai Y. and Cao X. Stereo Matching Algorithm Based on Joint Matching Cost and Adaptive Window. 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, pp. 442-446, 2018.

[Che21] Chen R., Chen F., Xu G., et al. Precision analysis model and experimentation of vision reconstruction with two cameras and 3D orientation reference. Scientific Reports, No. 11, 3875, 2021.

[Dom16] Domański M., Dziembowski A., Grzelka A., Mieloch D., Stankiewicz O., and Wegner K. Multiview test video sequences for free navigation exploration obtained using pairs of cameras. Document ISO/IEC JTC1/SC29/WG11, MPEG M38247, 2016.

[Dom17] Domański M., Stankiewicz O., Wegner K. and Grajek T. Immersive visual media – MPEG-I: 360 video, virtual navigation and beyond. 2017 IEEE International Conference on Systems, Signals and Image Processing (IWSSIP), Poznań, pp. 1–9, 2017.

[Doy18] Doyen D., Boisson G., and Gendrot R. [MPEG-I Visual] New Version of the Pseudo-Rectified Technicolorpainter Content. Document ISO/IEC JTC1/SC29/WG11 MPEG/M43366, Ljublana, 2018.

[Dzi16] Dziembowski A., Domański M., Grzelka A., Mieloch D., Stankowski J., and Wegner K. The influence of a lossy compression on the quality of estimated depth maps. 2016 International Conference on Systems, Signals and Image Processing (IWSSIP), Bratislava, pp. 1-4, 2016.

[Dzi20] Dziembowski A., Mieloch D., Stankiewicz O., Nikonowicz J., Domański M. Noise modeling in TMIV. Document ISO/IEC JTC1/SC29/WG11 MPEG/M54895, Online, June 2020.

[Fan16] Fang L., Xiang Y., Cheung N. M., and Wu F. Estimation of virtual view synthesis distortion toward virtual view position. 2016 IEEE Transactions on Image Processing, vol. 25, no. 5, pp. 1961–1976, May 2016.

[Gar19] Garus P., Jung J., Maugey T., and Guillemot C. Bypassing Depth Maps Transmission for Immersive Video Coding. 2019 Picture Coding Symposium (PCS), Ningbo, pp. 1-5, 2019.

[God20] Godbolt M. Optimizations in C++ compilers. Communications of the ACM, vol. 63, no. 2 pp. 41-49, 2020.

[Kol02] Kolmogorov V., and Zabih R. Multi-camera Scene Reconstruction via Graph Cuts. Proceedings of the 7th European Conference on Computer

Vision-Part III (ECCV '02), London, pp. 82-96, 2002.

[Köp13] Köppel M., Makhlouf M. B., Müller M., and Ndjiki-Nya P. Temporally consistent adaptive depth map preprocessing for view synthesis. 2013 Visual Communications and Image Processing (VCIP), Kuching, Malaysia, pp. 1-6, 2013.

[Laf16] Lafruit G., Domański M., Wegner K., Grajek T., Senoh T., Jung J., Kovács P., Goorts P., Jorissen L., Munteanu A., Ceulemans B., Carballeira P., García S., and Tanimoto M. New visual coding exploration in MPEG: Super-MultiView and Free Navigation in Free viewpoint TV. 2016 Proceedings of the Electronic Imaging Conference: Stereoscopic Displays and Applications, San Francisco, pp. 1–9, 2016.

[Leo07] Leontaris A., Cosman P. C., and Reibman A. R. Quality Evaluation of Motion-Compensated Edge Artifacts in Compressed Video. 2007 IEEE Transactions on Image Processing, vol. 16, no. 4, pp. 943-956, Apr. 2007.

[Mie20a] Mieloch D., Stankiewicz O., and Domański M. Depth Map Estimation for Free-Viewpoint Television and Virtual Navigation. 2020 IEEE Access, vol. 8, pp. 5760-5776, 2020.

[Mie20b] Mieloch D., Dziembowski A., and Domański M. [MPEG-I Visual] Natural Outdoor Test Sequences. Document ISO/IEC JTC1/SC29/WG11 MPEG/M51598, Brussels, Jan. 2020.

[MPEG20] Software manual of IV-PSNR for Immersive Video. Document ISO/IEC JTC1/SC29/WG04 MPEG VC, N0013, Online, Oct. 2020.

[MPEG21] Test Model 8 for MPEG Immersive Video. Document ISO/IEC JTC1/SC29/WG04 MPEG VC, N0050, Online, Jan. 2021.

[Mül11] Müller K., Merkle P., and Wiegand T. 3-D Video Representation Using Depth Maps. 2011 Proceedings of the IEEE, vol. 99, no. 4, pp. 643-656, 2011.

[Sal18] Salahieh B., Marvar B., Nentedem M., Kumar A., Popvic V., Seshadrinathan K., Nestares O. and Boyce J. Kermit test sequence for Windowed 6DoF Activities. Document ISO/IEC JTC1/SC29/WG11 MPEG/M43748, Ljublana, Slovenia, Jul. 2018.

[San17] Sankowski W., Włodarczyk M., Kacperski D., and Grabowski K. Estimation of measurement uncertainty in stereo vision system, Image and Vision Computing, vol 61, pp. 70-81, 2017.

[Say15] Saygili G., Van der Maaten L., Hendriks E. A. Adaptive stereo similarity fusion using confidence measures. Computer Vision and Image Understanding, vol. 135, pp. 95-108, 2015.

[Shi17] Shin Y., and Yoon K. Adaptive spatiotemporal similarity measure for a consistent depth maps. 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Kuta, pp. 1-4, 2017.

[Sta14] Stankiewicz O., Domański M., and Wegner K. Analysis of noise in multi-camera systems. 3DTV Conference 2014, Budapest, 2014.

[Suo12] Suominen O., Gotchev A., and Hannuksela M. Transform domain similarity measures in stereo matching. 2012 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), Zurich, pp. 1-4, 2012.

[Tan12] Tanimoto M., Tehrani M. P., Fujii T., and Yendo T. FTV for 3-D Spatial Communication. 2012 Proceedings of the IEEE, vol. 100, no. 4, pp. 905-917, 2012.

[Tan17] Tang Z., Grompone von Gioi R., Monasse P., and Morel J. A Precision Analysis of Camera Distortion Models. IEEE Transactions on Image Processing, vol. 26, no. 6, pp. 2694-2704, 2017.

[Tip13] Tippetts B., Jye Lee D., Lillywhite K., and Archibald J. Review of stereo vision algorithms and their suitability for resource-limited systems. Journal of Real-Time Image Processing, vol. 11, no. 1, pp. 5-25, 2013.

[Weg09] Wegner K., and Stankiewicz O. Similarity measures for depth estimation. 2009 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, Potsdam, pp. 1-4, 2009.

[X265] MulticoreWare Inc. x265 HEVC Encoder/ H.265 Video Codec. Available on http://x265.org

[Zeg20] Zeglazi O, Rziza M, Amine A, Demonceaux C. Structural Similarity Measurement Based Cost Function for Stereo Matching of Automotive Applications. Journal of Imaging, vol. 6, no. 8, 2020.

[Zha19] Zhang C., He C., Chen Z., Liu W., Li M. and Wu J. Edge-Preserving Stereo Matching Using Minimum Spanning Tree. IEEE Access, vol. 7, pp. 177909-177921, 2019.