Signal, Image and Video Processing (2019) 13:1539–1547
https://doi.org/10.1007/s11760-019-01487-4

ORIGINAL PAPER

Depth estimation using modified cost function for occlusion handling

Krzysztof Wegner[1] · **Olgierd Stankiewicz**[1]

Received: 11 March 2017 / Revised: 5 July 2018 / Accepted: 16 July 2018 / Published online: 29 May 2019
© The Author(s) 2019

Abstract
This paper presents a novel approach to the occlusion handling problem in depth estimation using three views. A solution based on modification of similarity cost function is proposed. During the depth estimation via optimization algorithms, like Graph Cuts, the similarity metric is constantly updated so that only non-occluded fragments in the side views are considered. At each iteration of the algorithm, non-occluded fragments are detected based on side view virtual depth maps synthesized from currently the best estimated depth map of the center view. Then, the similarity metric is updated for correspondence search only in non-occluded regions of the side views. The experimental results, conducted on well-known 3D video test sequences, show that the depth maps estimated with the proposed approach provide about 1.25 dB virtual view quality improvement in comparison with the virtual view synthesized based on depth maps generated with the use of the state-of-the-art MPEG Depth Estimation Reference Software.

Keywords Depth estimation · Disparity estimation · Occlusion handling · MVD · Graph cuts · DERS · Free viewpoint television

1 Introduction

3D video systems have recently gained a lot of attention. Many new 3D video systems have been developed. Among them, super multi-view television and free viewpoint television are examples of such novel 3D systems. In the case of free viewpoint television, a user is able to freely choose a position of a virtual camera. The requested view of a scene is generated from the dynamic 3D representation of the scene.

The most commonly used 3D representation is Multi-view Video plus Depth (MVD) [1], which is composed of multiple videos (e.g., acquired by a set of cameras) accompanied with depth maps for each of the views. Based on transmitted videos and depth data, any view can be easily generated by the means of depth-image-base rendering (DIBR) [2].

Recently, 3D extensions of such standards as AVC [3,4] and HEVC [5] that allow efficient transmission of dynamic 3D scene representation in MVD format have been finalized.

Depth information in such systems can be acquired either directly by depth cameras [6] or indirectly by algorithmic depth estimation from the recorded videos [7]. Commonly, depth information is obtained through conversion from disparity [8]. However, in computer vision, disparity d is often treated as synonymous with depth (distance z), and essentially those terms are the inverse of each other.

$$z \sim \frac{1}{d} \tag{1}$$

Disparity is a displacement between corresponding fragments (pixels, blocks) of two images of the same scene taken from different viewpoints. Those two corresponding fragments represent the same fragment of an observed scene but seen from two different viewpoints.

Stereo correspondence search is an active research topic in computer vision and is one of the basic methods of obtaining disparity information. There are many known stereo disparity estimation methods. A comprehensive study of stereo disparity estimation methods can be found in [9] and on the Middlebury webpage [10] containing an up-to-date benchmark of stereo disparity estimation methods. In the scope of development of multi-view systems, the stereo correspondence search was extended to a multi-view correspondence search [11–13].

For the sake of simplicity and accuracy, many algorithms assume that images are taken by a rectified set of cameras

✉ Krzysztof Wegner
kwegner@multimedia.edu.pl

[1] Chair of Multimedia Telecommunications and Microelectronics, Poznan University of Technology, Poznan, Poland

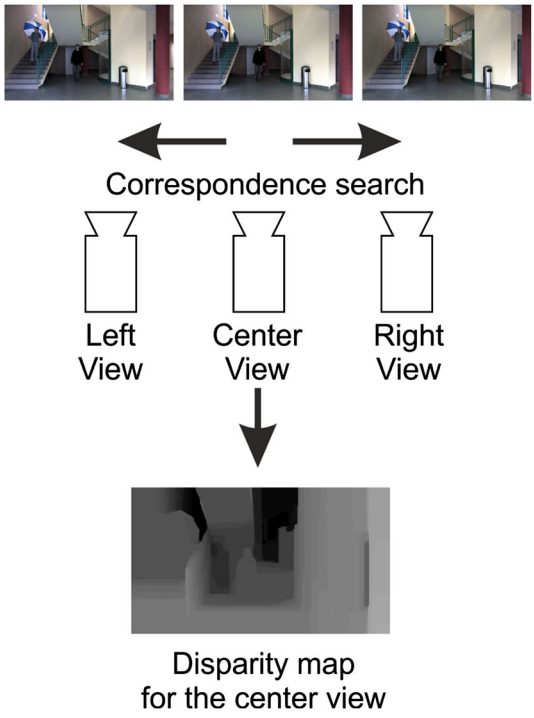


Fig. 1 Three-view disparity estimation

[14,15]. Consecutively, corresponding fragments of a given image can be found on the same horizontal line in the remaining images.

Some algorithms use three views (left, central and right) [16–19] as inputs and produce a disparity/depth map for the central view (Fig. 1). Often, when it is not important which of the left or the right view is referred to, the term "side view" is used instead.

During disparity estimation, for a given fragment of the central view, the algorithm searches for the corresponding fragment in the side views that represents the same fragment of the scene.

The correspondence search is done on the basis of the similarity metric, which expresses how probable it is that a certain fragment of one image is the corresponding fragment of the second image. Although the metric used is often called similarity, it actually expresses dissimilarity between fragments. There are many similarity metrics known from the literature: Sum of Absolute Difference (SAD), Sum of Squared Difference (SSD), Rank Transform, Census Transform, cross-correlation and others [20,21].

The correspondence search is often defined as an optimization problem in which for every fragment of the central image the best (the most similar) fragment in the side view is selected. This optimization problem may be also expressed in terms of energy function, using Markov Random Field (MRF), and optimized via an optimization algorithm such as belief propagation [22,23], dynamic programming [24] or Graph Cuts [25].

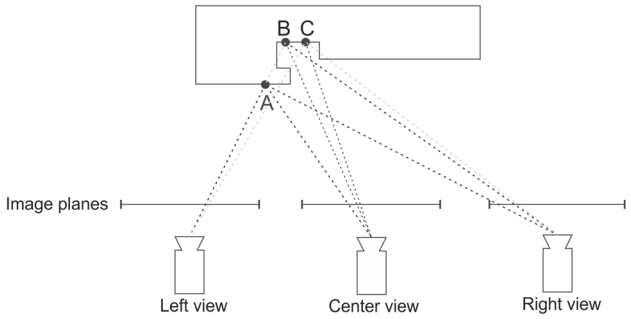


Fig. 2 Occlusion in a three-view disparity estimation problem

Since input videos are captured by multiple cameras with different positions, some parts of the observed scene can be occluded and thus not visible within some of the views. Disparity estimation for those fragments of a scene is challenging and requires special care. If the possible occlusions within the scene are not properly taken into account in the algorithm, the estimated disparity can be wrong, e.g., the estimated disparity may indicate fragments which are not truly corresponding to each other.

In this paper, a novel approach to occlusion handling is proposed, designed to work in three-view disparity estimation algorithms.

2 Occlusion problem in disparity estimation

Given three images: center I_C, left I_L and right I_R, all of same size, we search for such a displacement t for every pixel P of the center view (at coordinates (x, y)) that minimizes the cost function expressing similarity between pixel P (or small fragment around the pixel P, like block) and a corresponding pixel P' (or small fragment around pixel P') displaced by t in the side views (at positions $(x+t, y)$ in the left and $(x-t, y)$ in the right view). Such displacement is then outputted as disparity of a given pixel P of a center view.

$$d_{\text{Center}}(x, y) = \min_t \text{Cost}(x, y, t); \tag{2}$$

In disparity estimation based on three views (see Fig. 2), a given point of the scene, visible from the center view, can be visible from both of the side views (point A), only from one of the side views (left or right, point B) or from neither of them (point C).

If the given fragment of the scene visible from center view is not visible from one or both of the side views, we say that a given fragment of the scene is occluded in the side view (is not visible from that particular side view).

The simplest method for detecting occluded fragments is cross-checking [26]. Cross-checking tests the consistency of the estimated disparity value for pixels from the center view with those estimated for pixels in the left and in the right views. If the disparity value estimated in each view is different for a correspondent triple of pixels from the center, the left and the right views, a given pixel is assumed to be occluded. Next, the disparity value for occluded pixels is extrapolated (inpainted) from the nearest pixels that are not occluded. In order to perform cross-checking, disparity maps for all of the three views are required. Unfortunately, estimation of three disparity maps is not always possible or desired. Even if the estimation of three disparity maps instead of one is possible, it is resource and time-consuming.

Occlusion handling is performed by adding/putting additional constraints, such as ordering constraint or uniqueness constraint to the objective function of optimization procedures, like graph cuts (GC), dynamic programming (DP) or belief propagation (BP) used to estimate the disparity map.

The ordering constraint [27] imposes the same ordering of corresponding pixels in all views. If in the center view, a pixel A is on the left side of the pixel B, then also in the side view pixel A' (that is a corresponding pixel of pixel A) must be on the left side of pixel B' (a corresponding pixel of pixel B). In real scenes, the ordering constraint is often violated in the case of big perspective changes or in the case of thin objects. In such cases, ordering constraint can introduce errors in the estimated disparity maps.

The uniqueness constraint [28,29] imposes one-to-one correspondence between the pixel in the center and in the side views. If a given pixel A in the central view is assigned to a corresponding pixel B in the side view, then no other pixel in the central view can be assigned to correspondence with pixel B in the side view. This way, a unique pixel to pixel correspondence is forced across all of the views.

There are many disparity estimation algorithms known that handle occlusion in an efficient way [29–31]. The main drawback of all of those algorithms are additional constraints (terms) imposed in optimization procedures which increase complexity and thus execution time of the disparity estimation.

Another approach to occlusion handling is to modify the cost term (Eq. 2) composed of a similarity metric in optimization algorithms. As we search for a corresponding fragment of a central view in both side views simultaneously, there are many ways of defining a $\mathrm{Cost}(x, y, t)$ function.

Commonly [17,18] the $\mathrm{Cost}(x, y, t)$ function is a sum of similarity metrics between a fragment in the center view and corresponding fragments in the left and in the right view.

$$\begin{aligned}\mathrm{Cost}(x, y, t) = {} & \mathrm{Similarity}(I_{\mathrm{C}}(x, y), I_{\mathrm{L}}(x + t, y)) \\ & + \mathrm{Similarity}(I_{\mathrm{C}}(x, y), I_{\mathrm{R}}(x - t, y))\end{aligned} \tag{3}$$

Because of the occlusions, Tanimoto [16] proposed to pick just the most similar fragment from either the left or from the right view. The intuition is that the occluded fragment will be less similar, thus the minimum of similarity metrics from the left and the right view is used.

$$\begin{aligned}\mathrm{Cost}(x, y, t) = {} & \min(\mathrm{Similarity}(I_{\mathrm{C}}(x, y), I_{\mathrm{L}}(x + t, y)), \\ & \mathrm{Similarity}(I_{\mathrm{C}}(x, y), I_{\mathrm{R}}(x - t, y)))\end{aligned} \tag{4}$$

In this paper, we propose yet another way to define the cost function which takes into account an occlusion possible within the scene.

3 Proposed occlusion handling

As it was said before, a given fragment of a scene visible from the center view can be occluded in one or both side views (left or/and right) (Fig. 2). In such a case, searching for a correspondence in this particular side view (left or right) is pointless, as the given fragment of the scene is not visible from that particular side view. Considering the correspondence for an occluded fragment of an image can cause errors in estimated disparity.

$$\mathrm{Cost}(x, y, t) = \frac{\mathrm{NotOcc}_{\mathrm{L}}(x, y, t) \cdot \mathrm{Sim}(I_{\mathrm{C}}(x, y), I_{\mathrm{L}}(x + t, y) + \mathrm{NotOcc}_{\mathrm{R}}(x, y, t) \cdot \mathrm{Sim}(I_{\mathrm{C}}(x, y), I_{\mathrm{R}}(x - t, y)))}{\mathrm{NotOcc}_{\mathrm{L}}(x, y, t) + \mathrm{NotOcc}_{\mathrm{R}}(x, y, t)} \tag{5}$$

Therefore the correspondence search should be performed only in those side views in which a considered fragment of a center view is not occluded. The cost function should be constructed in such a way that it considers only similarity metrics from non-occluded views. If a given fragment is visible in both views, then the cost function should be an average of both similarity metrics, in order to reduce the influence of noise which is present in all views. We propose to define the cost function in a way that it considers only similarity metrics of fragments from non-occluded views (either left or right) (Eq. 5) where $\mathrm{NotOcc}_{\mathrm{L}}(x, y, t)$, $\mathrm{NotOcc}_{\mathrm{R}}(x, y, t)$ expresses whether a given pixel of a center view is not occluded in the left and in the right view, respectively. Depending on the existence of occlusion in the views, the sum $\mathrm{NotOcc}_{\mathrm{L}}(x, y, t) + \mathrm{NotOcc}_{\mathrm{R}}(x, y, t)$ in the denominator of Eq. 5 can be 2 if a pixel is not occluded in

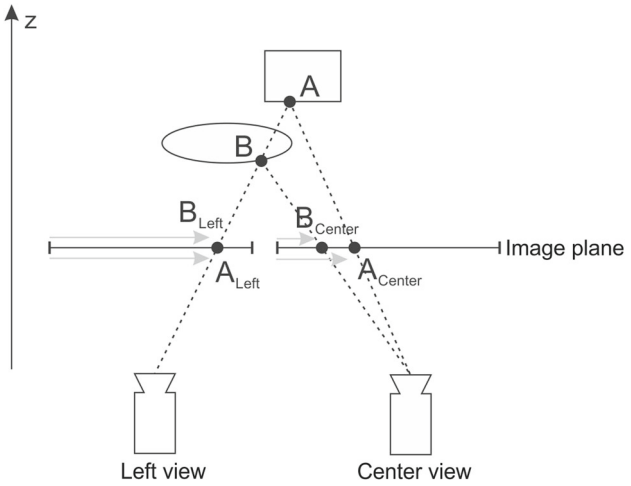


Fig. 3 Occlusion problem in correspondence search

both views, 1 if it is occluded in one of the side views (either left or right), and 0 if it is occluded in both side views. If a given pixel is occluded in both side views, Eq. 5 looses its meaning, thus in such a case constant penalty value is used as a cost value.

$$\mathrm{Cost}(x, y, t) = \mathrm{const} \tag{6}$$

Consider the example in Fig. 3 where two points A and B are observed by two cameras (left and center). Point B is closer to the cameras and point A is farther. Point B is visible in both views (left and center) at pixel position B_{Left} and B_{Center}, respectively. Due to occlusion, point A is visible only in the center view at pixel position A_{Center}. If there would be no point B, point A would be visible in the left view at pixel position A_{left}. The disparity of point B in the left view is the difference of the pixel position B_{Left} and B_{Center}, and the disparity of point A in the left view would be (if the point was/would be visible) the difference of pixel position A_{Left} and A_{Center}.

$$d_{\mathrm{Left}}(B_{\mathrm{Left}}) = B_{\mathrm{Left}} - B_{\mathrm{Center}} \tag{7}$$

$$d_{\mathrm{Left}}(A_{\mathrm{Left}}) = A_{\mathrm{Left}} - A_{\mathrm{Center}} \tag{8}$$

The distance to the camera z is reciprocal to disparity. Therefore, a fragment of an image representing a closer object (point B) has bigger disparity than the fragment representing the farther object (point A).

$$z_{\mathrm{Left}}(A_{\mathrm{Left}}) > z_{\mathrm{Left}}(B_{\mathrm{Left}}) <=> d_{\mathrm{Left}}(A_{\mathrm{Left}}) < d_{\mathrm{Left}}(B_{\mathrm{Left}}) \tag{9}$$

For a given pixel A_{Center} of the center view at coordinates (x, y) and considered displacement t, the corresponding pixel A_{Left} in the left view should be at coordinates $(x+t, y)$. Thus, if we want to check whether a fragment A of a scene is occluded in the left view we have to check the disparity (distance) assigned to the considered corresponding pixel A_{Left} in the left view. If a disparity $d_{\mathrm{Left}}(x+t, y)$, assigned already to the considered corresponding pixel A_{Left}, is bigger than the considered displacement t, then pixel A_{left} probably is not a fragment of the same object A. Rather, it is a fragment of some other, closer object B that occludes object A in the left view.

Based on such a consideration, we can create a function assessing whether for a pixel at coordinates (x, y) and displacement t the corresponding pixel is/can be/will be occluded or not in the left and in the right view.

$$\mathrm{NotOcc_L}(x, y, t) = \begin{cases} 1 & \text{for } t \geq d_{\mathrm{Left}}(x+t, y) \\ 0 & \text{for } t < d_{\mathrm{Left}}(x+t, y) \end{cases} \tag{10}$$

$$\mathrm{NotOcc_R}(x, y, t) = \begin{cases} 1 & \text{for } t \geq d_{\mathrm{Right}}(x-t, y) \\ 0 & \text{for } t < d_{\mathrm{Right}}(x-t, y) \end{cases} \tag{11}$$

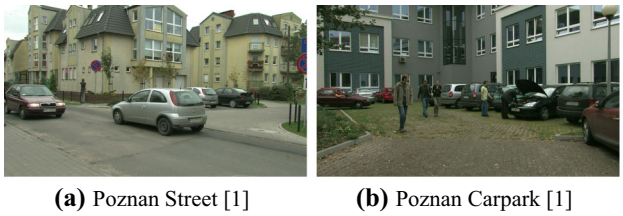
$\mathrm{NotOcc}(x, y, t)$ equal to 1 means that the corresponding pixel in the side view at a given displacement is probably not occluded.

4 Application of the proposed idea

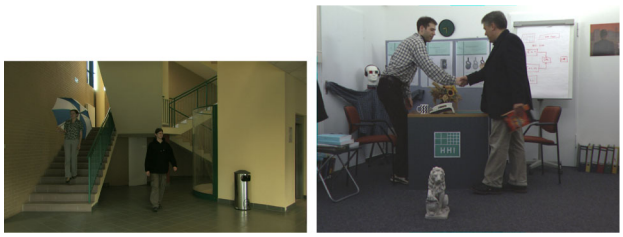
The proposed idea is general as it does not impose any particular source of disparity maps d_{Left} for the left and d_{Right} for the right view. Obviously, in general, disparity maps for the left and the right views may be unknown before estimating the disparity for the central view.

Commonly, disparity maps are estimated iteratively with the use of algorithms like belief propagation or graph cuts. In such algorithms, at each iteration of the estimation, the algorithm maintains up-to-date/the best already estimated disparity map for the center view. This disparity map is further refined in the further iterations of the algorithm.

For our occlusion detection, we propose to use disparity maps of the side views created based on the disparity map of the center view through Depth-Image-Based Rendering (DIBR). After each iteration of a disparity estimation algorithm, we create disparity maps of the side views (d_{Left} and d_{Right}) from the best already estimated disparity map of the center view. This way, if the estimation algorithm used, already assigned some disparity $d_{\mathrm{Center}}(B)$ to some pixel B_{Center}, then pixel A_{Center} cannot have such a disparity that the corresponding pixel A_{Left} (Fig. 3), is at the same position as corresponding pixel B_{Left} of pixel B_{Center}. In other words, fragment B of a scene represented by pixel B_{Center} in the center view should occlude a fragment A of a scene (represented by pixel A_{Center} in the center view) seen from the left view.



(a) Poznan Street [1] **(b)** Poznan Carpark [1]



(c) Poznan Hall2 [1] **(d)** Book Arrival [2]

Fig. 4 Exemplary frames from multi-view test sequences used in experiments

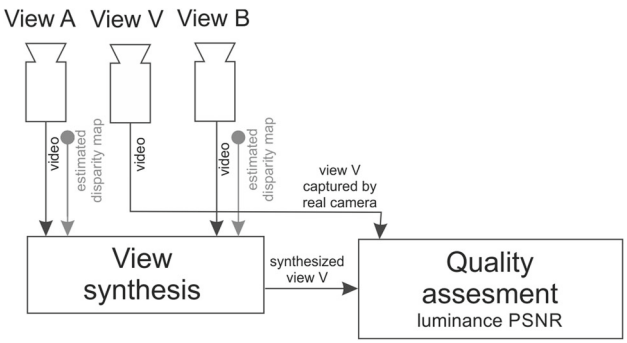


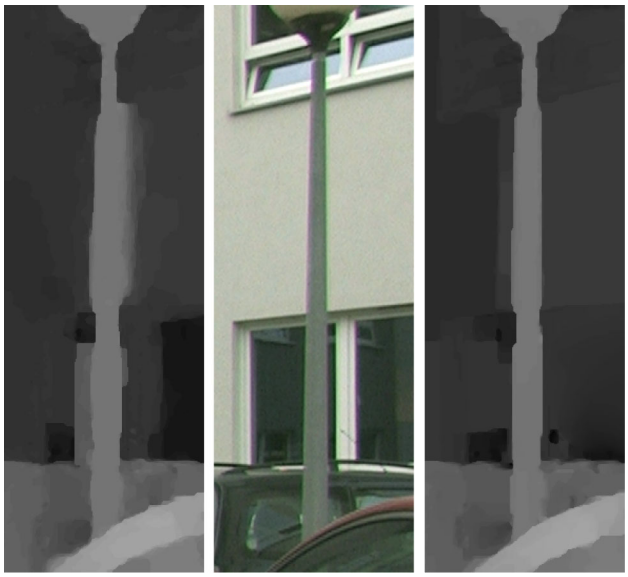
Fig. 5 Disparity map quality evaluation methodology

Table 1 Positions of views used for evaluation of quality of estimated disparity maps

Sequence name	View *A*	View *B*	View *V*
Poznan Street	3	5	4
Poznan Hall 2	5	7	6
Poznan Carpark	3	5	4
Book Arrival	7	9	8

5 Experiments

Our idea can be applied to any depth estimation algorithm, as it modifies only similarity metrics. For the sake of experimentation, we have implemented our idea in Depth Estimation Reference Software (DERS) [32] version 5.0 developed by Moving Picture Experts Group (MPEG) of International Standardization Organization (ISO) during works on 3D video compression standardization. DERS is a state-of-the-art disparity estimation technique, designed with 3D video application in mind. It uses graph cuts as an optimization algorithm along with many other techniques that improve



(a) DERS **(b)** Texture **(c)** Proposed

Fig. 6 Comparison of depth maps fragment estimated for Poznan Carpark [33]

or/and speed up disparity estimation. It uses three input videos and produces a single output disparity map.

The proposed approach was tested on four 3D video test sequences recommended by the MPEG committee (Fig. 4): Poznan Street, Poznan Carpark, Poznan Hall 2 [33], and Book Arrival [34].

In many applications which use depth maps, such as free viewpoint television, depth maps are never presented directly to the viewer, but they are mainly used for the purpose of creating an additional view of the scene by means of view synthesis [7,35]. Therefore, we have evaluated our proposed method indirectly, by assessing the quality of synthesized views. Such methodology is widely recognized and accepted in the literature for assessing depth maps quality [1,6,30,35].

In order to compare the influence of the proposed idea, we have estimated disparity maps for two views: A and B (Fig. 5), with the use of the proposed method and the original, unmodified DERS software. Based on the views A and B and the estimated disparity maps for views A and B, a view V that is positioned in between views A and B was synthesized. Exact view numbers for each of the test sequences used during experiments are provided in Table 1.

The quality of the estimated disparity maps for views A and B is measured as a quality of synthesized view V. The quality of synthesized view V is expressed with the PSNR of luminance in comparison with the view V captured by a real camera positioned at the same spatial position (see Fig. 5).

In the course of evaluation, disparity maps were estimated for every frame of the sequences (mostly 250 frames per view). This allowed evaluation of our algorithm on a wide

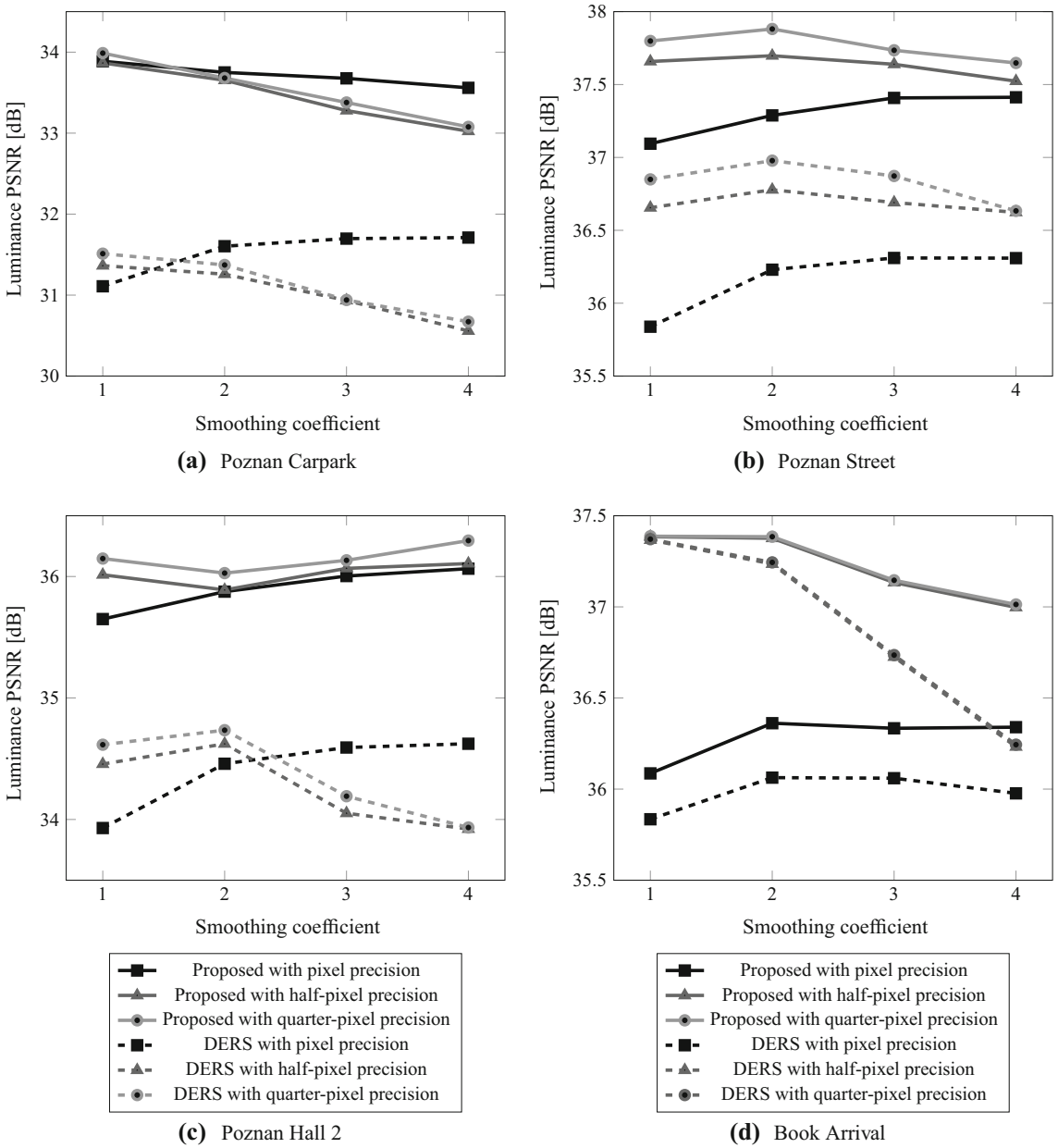
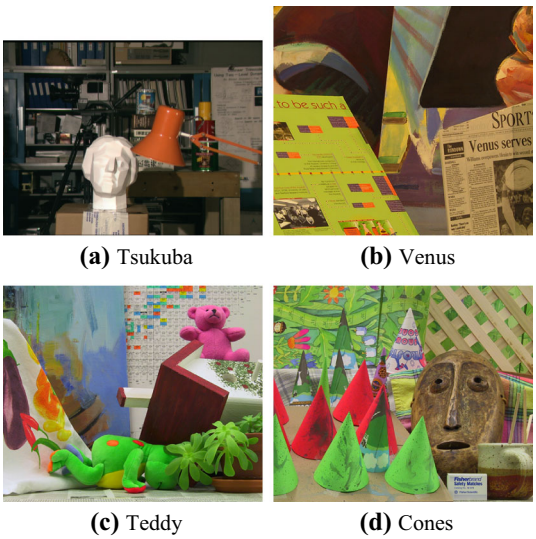


Fig. 7 Performance comparison of depth estimation with the use of the proposed method and DERS

Table 2 Quality comparison by PSNR of a synthesized view for the best depth maps with respect to the smoothing coefficient

Sequence Name	Pixel precision			Half-pixel precision			Quarter-pixel precision		
	DERS (dB)	Proposed (dB)	Gain	DERS (dB)	Proposed (dB)	Gain	DERS (dB)	Proposed (dB)	Gain
Poznan Street	36.31	37.41	1.10	36.78	37.70	0.92	36.96	37.88	0.90
Poznan Hall2	34.62	36.06	1.44	34.62	36.11	1.48	34.74	36.39	1.56
Poznan Carpark	31.71	33.89	2.18	31.36	33.87	2.50	31.51	33.99	2.48
Book Arrival	36.06	36.36	0.30	37.37	37.38	0.02	37.37	37.39	0.02
Average	–		1.26	–		1.23	–		1.24



(a) Tsukuba **(b)** Venus

(c) Teddy **(d)** Cones

Fig. 8 Standard Middlebury dataset [36] used for evaluation of the proposed algorithm

Table 3 Specification of three views used for disparity estimation for each Middlebury dataset

Dataset name	Additional view	Standard stereo pair	
	Left view	Center view	Right view
Tsukuba	2	3	4
Venus	0	2	6
Teddy	0	2	6
Cones	0	2	6

range of different images. The disparity estimation was done with various precisions: per-pixel, half-pixel and quarter-pixel precision. Also, a wide range of regularization terms used in Graph Cut algorithm has been evaluated. In DERS, the regularization is controlled by a "smoothing coefficient". In the experiments, a range of 1–4 for the smoothing coefficient was explored.

In Fig. 6, the exemplary frame of depth map estimated for Poznan Carpark [33] with DERS and with the proposed algorithm has been presented. Obviously, depth maps near the edge of a foreground object (the lamp) have been improved.

The comparison of quality of the estimated disparity maps for the proposed method versus the original DERS can be found in Fig. 7. As it can be noticed, the smoothing coefficient can have a significant impact on the quality of disparity maps estimated with DERS. It can be expected that in a real-world-use scenario, this parameter will be automatically controlled to provide the best results. Therefore, in the summarized Table 2, we have presented only the best performing cases. Depending on the case, the proposed occlusion handling brings a gain of 0.02–2.50 dB of luminance PSNR of the synthesized view, related to the original unmodified DERS. On average, the proposal provides an improvement of 1.26 dB for pixel-precise disparity estimation, 1.23 dB for half-precise disparity estimation, and 1.18 dB for quarter-pixel-precise disparity estimation.

We have also evaluated our algorithm using a different methodology developed in [36] and used in widely recognized Middlebury test bench. In the Middlebury methodology, the quality of depth maps is evaluated directly by counting the number of pixels where the estimated disparity differs from ground truth disparity obtained by means of structured lighting.

We have used the default Middlebury dataset [36]: Tsukuba, Venus, Teddy and Cones (Fig. 8). For the evaluation, we have modified the DERS algorithm to directly output raw disparity maps in the format required by the Middlebury evaluation webpage [10]. Because both evaluated methods—the proposed one and the DERS algorithm—are designed to work with three input images, we have extended the recommended/standard stereo pair with the third image as specified in Table 3.

The application of the proposed occlusion handing algorithm on Middlebury images results in an improvement of maximally 0.15 percentage points of non-occluded bad pixel numbers (Table 4). Please keep in mind that Middlebury datasets have very little occlusions and thus the attained gains cannot be significant.

Table 4 Comparison of the proposed method with DERS on Middlebury datasets

Algorithm	Tsukuba			Venus			Teddy			Cones		
	Nonocc	All	Disc	Nonocc	All	Disc	Nonocc	All	Disc	Nonocc	All	Disc
GC + occ	1.19	2.01	6.24	1.64	2.19	6.75	11.2	17.4	19.8	5.36	12.4	13.0
2OP + occ [13]	2.91	3.56	7.33	0.24	0.49	2.76	10.9	15.4	20.6	5.42	10.8	12.5
Putv3	1.77	3.86	9.42	0.42	0.95	5.72	7.02	14.2	18.3	2.40	9.11	6.56
CostAggr + occ [19]	1.38	1.96	7.14	0.44	1.13	4.87	6.80	11.9	17.3	3.60	8.57	9.36
DERS	2.70	3.30	12.10	0.67	1.25	8.53	10.2	11.5	23.3	5.17	7.33	9.50
Proposed	2.65	3.01	11.20	0.63	1.02	8.34	9.96	10.97	–	5.02	7.12	–

6 Conclusion

We have presented a novel approach to occlusion handling in disparity estimation, based on a modification of the similarity cost function. The proposed approach has been tested in a three-view disparity estimation scenario. For occlusion detection, synthesized disparity maps of the left and the right view have been used.

For well-known multi-view video test sequences, the experimental results show that the proposed approach provides improvement of virtual view quality of about 1.25 dB of luminance PSNR over the state-of-the-art technique implemented in MPEG Depth Estimation Reference Software (DERS). Moreover, direct quality evaluation of estimated disparity, based on the Middlebury dataset, reveals that the proposed approach reduces the number of bad pixels by 0.15 p.p.

Acknowledgements This research was supported by National Science Centre, Poland, according to decision DEC-2012/05/B/ST7/01279.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Muller, K., Merkle, P., Wiegand, T.: 3-D video representation using depth maps. Proc. IEEE **99**(4), 643–656 (2011)
2. Zhang, L., Tam, W.J.: Stereoscopic image generation based on depth images for 3DTV. IEEE Trans. Broadcast. **51**(2), 191–199 (2005)
3. Annex, I.: Multiview and depth video coding of ISO/IEC 14496-10. International Standard Generic coding of audio-visual objects—Part 10: Advanced Video Coding, 8th ed., 2013, also: ITU-T Rec. H.264, Edition 8.0 (2013)
4. 3D-AVC Draft Text 9, JCT-3V of ITU T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JCT3V-G1003, San Jose, USA (2014)
5. Tech, G., Wegner, K., Chen, Y., Yea, S.: 3D-HEVC draft text 6 joint collaborative team on 3D video coding extension development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 Doc. JCT3V-J1001, 10th Meeting: Strasbourg, FR, 1824 (2014)
6. Kim, S.Y., Cho, J.H., Koschan, A.: 3D video generation and service based on a TOF depth sensor in MPEG-4 multimedia framework. IEEE Trans. Consum. Electron. **56**(3), 1730–1738 (2010)
7. Domański, M., Dziembowski, A., Kuehn, A., Kurc, M., Łuczak, A., Mieloch, D., Siast, J., Stankiewicz, O., Wegner, K.: Experiments on acquisition and processing of video for free-viewpoint television. In: 3DTV Conference 2014, Budapest, Hungary (2014)
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn, pp. 262–278. Cambridge University Press, Cambridge (2003)
9. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vis. **47**(1/2/3), 7–42 (2002)
10. Middlebury Stereo Evaluation—Version 2. Webpage visited 2015-01-24. http://vision.middlebury.edu/stereo/eval
11. Okutomi, M., Kanade, T.: A multiple baseline stereo. IEEE Trans. PAMI **15**(4), 353–363 (1993)
12. Collins, R.T.: A space-sweep approach to true multi-image matching. In: CVPR96, San Francisco, pp. 358–363 (1996)
13. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 06), pp. 519–526 (2006)
14. Stankowski, J., Klimaszewski, K., Stankiewicz, O., Wegner, K., Domański, M.: Preprocessing methods used for Poznan 3D/FTV test sequences. ISO/IEC JTC1/SC29/WG11 MPEG 2010/M17174, Doc. m17174, Kyoto, Japan (2010)
15. Stankowski, J., Klimaszewski, K.: Application of epipolar rectification algorithm in 3D Television. In: Image Processing and Communications Challenges 2. Advances in Intelligent and Soft Computing, vol. 84, pp. 345-352. Springer, Berlin (2010). ISBN: 978-3-642-16294-7
16. Tanimoto, M., Fujii, T., Suzuki, K., Fukushima, N., Mori, Y.: Reference softwares for depth estimation and view synthesis. ISO/IEC JTC1/SC29/WG11, Doc. M15377, Archamps, France (2008)
17. Wildeboer, M., Fukushima, N., Yendo, T., Panahpour, M.T., Fujii, T., Tanimoto, M.: A semi-automatic multi-view depth estimation method. In: Proceedings of the SPIE, vol. 7744 (2010)
18. Lee, S.-B., Ho, Y.-S.: Multi-view depth map estimation enhancing temporal consistency. In: 23rd International Technical Conference on Circuits/Systems, Computers and Communications
19. Stankiewicz, O.: Stereoscopic depth map estimation and coding techniques for multiview video systems. Ph.D. Dissertation at Poznan University of Technology, Faculty of Electronics and Telecommunications (2014)
20. Wegner, K., Stankiewicz, O.: Similiarity measures for depth estimation. In: 3DTV-Conference 2009, Potsdam, Germany (2009)
21. Birchfield, S., Tomasi, C.: A pixel dissimilarity measure that is insensitive to image sampling. IEEE Trans. Pattern Anal. Mach. Intell. **20**(4), 401–406 (1998)
22. Sun, J., Zheng, N.N., Shum, H.Y.: Stereo matching using belief propagation. IEEE Trans. Pattern Anal. Mach. Intell. **25**(7), 787–800 (2003)
23. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 261–268 (2004)
24. Veksler, O.: Stereo correspondence by dynamic programming on a tree. In: CVPR (2005)
25. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
26. Egnal, G., Wildes, R.: Detecting binocular halfocclusions: empirical comparisons of five approaches. IEEE Trans. Pattern Anal. Mach. Intell. **24**(8), 1127–1133 (2002)
27. Bobick, A., Intille, S.: Large occlusion stereo. Int. J. Comput. Vis. **33**(3), 181–200 (1999)
28. Marr, D., Poggio, T.A.: Cooperative computation of stereo disparity. Science **194**(4262), 283–287 (1976)
29. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 508–515 (2001)
30. Jang, W.-S., Ho, Y.-S.: Efficient depth map generation with occlusion handling for various camera arrays. Signal Image Video Process. **8**(2), 287–297 (2014)
31. Ben-Ari, R., Sochen, N.: Stereo matching with Mumford–Shah regularization and occlusion handling. IEEE Trans. Pattern Anal. Mach. Intell. **32**(11), 2071–2084 (2010)

32. Wildeboer, M., Stankiewicz, O., Wegner, K.: A soft-segmentation matching in depth estimation reference software (DERS) 5.0. ISO/IEC JTC1/SC29/WG11 Doc. M17049, Xian, China (2009)
33. Domański, M., Stankiewicz, O., Wegner, K., et al.: Pozna multiview video test sequences and camera parameters. ISO/IEC JTC1/SC29/WG11 Doc. M17050, Xian, China, October (2009)
34. Feldmann, I., Smolic, A., et al.: HHI test material for 3D video. ISO/IEC JTC1/SC29/WG11, Doc. M15413, Archamps, France (2008)
35. Jinmi, K., Kidong, C.: High-performance depth map coding for 3D-AVC Signal. Image Video Process. **10**(6), 1017–1024 (2016)
36. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), vol. 1, pp. 195–202, Madison, WI (2003)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.