

EFFICIENT EDGE, MOTION AND DEPTH-RANGE ADAPTIVE PROCESSING FOR ENHANCEMENT OF MULTI-VIEW DEPTH MAP SEQUENCES

Erhan Ekmekcioglu¹, Vladan Velisavljević², Stewart T. Worrall¹

¹Centre for Comm. Systems Research, University of Surrey, Guildford, UK

²Deutsche Telekom Laboratories, Technische Universität Berlin, Germany

ABSTRACT

We present a novel and efficient multi-view depth map enhancement method proposed as a post-processing of initially estimated depth maps. The proposed method is based on edge, motion and scene depth-range adaptive median filtering and allows for an improved quality of virtual view synthesis. To enforce the spatial, temporal and inter-view coherence in the multi-view depth maps, the median filtering is applied to 4-dimensional windows that consist of the spatially neighboring depth map values taken at different viewpoints and time instants. A fast iterative block segmentation approach is adopted to adaptively shrink these windows in the presence of edges and motion for preservation of sharpness and realistic rendering and for improvement of the compression efficiency. We show that our enhancement method leads to a reduction of the coding bit-rate required for representation of the depth maps and also leads to a gain in the quality of synthesized views at arbitrary virtual viewpoints.

Index Terms— Depth estimation, Free viewpoint video, Multi-view coding, Multi-view depth map.

1. INTRODUCTION

Multi-view video signals are typically captured by an array of synchronized cameras, which cover the same scene from different viewpoints. Such a system enables Free-Viewpoint Video (FVV) applications, where the user freely chooses a viewpoint in the scene.

Recent research in the area of FVV brought into focus new technological challenges, like recording, encoding and decoding the FVV signals [1]. The key issue is how to deal with the large amount of data required for representation or realistic approximation of a scene in the FVV. The current commercial broadcasting systems are not capable of processing and transmitting the signals recorded at all possible views. For that reason, several different coding approaches have recently been proposed to improve the coding efficiency.

In mesh-based FVV coding [2], the underlying 3D geometry of the scene is encoded and sent to the user along with the corresponding photometric properties. This method significantly reduces the amount of data needed for transmission, but it also requires a precise knowledge of the 3D geometry in the scene, which is still difficult to acquire in a general case. By contrast, in the depth map-based FVV coding [3, 4], depth map sequences join the color texture sequences in each chosen viewpoint to locally approximate the light-field function. Such a representation does not need a precise geometric knowledge of the scene and, thus, the required number of views (and, consequently, cameras) is reduced. Moreover, arbitrary views can be

generated on demand at the decoder using the image-based rendering (IBR) techniques [5]. A similar approach has been adopted in [6], where the authors propose to use multi-view coding (MVC) tools developed under the standardization group in JVT for the compression of both the multiple viewpoint color texture videos and associated per-pixel depth map sequences. These two types of sequences are encoded separately using the MVC method based on hierarchical B-prediction and H.264/AVC that exploits the intra-frame spatial, inter-frame temporal and inter-view correlation in the multi-view signal [1, 7]. However, the depth map sequences differ from the color texture sequences, having large regions of smoothly changing or even constant gray levels. Thus, applying the same MVC method to both types of sequences might result in a suboptimal coding performance. This shortcoming has already been addressed and the depth maps have been encoded using modified methods based on wedgelets and platelets [8] or on shape-adaptive wavelets [9]. Moreover, the latter method has also exploited a correlation between location of edges in the depth map and color texture sequences.

Our goal is to efficiently post-process the estimated multi-view depth maps so that both the resulting coding efficiency and the quality of IBR at decoder are improved without significant complexity. We propose a novel enhancement method based on an edge, motion and depth range adaptive median filtering of the multi-view depth maps across the spatial, temporal and inter-view dimensions. A simple and quick initial depth estimation process therefore can still result in good rendering at the decoder in terms of both objective and visual quality of the synthesized views. Beside that, owing to an iterative and adaptive block division of the frames, the median filtering is applied jointly to a set of pixels and, thus, our method carries only a low additional complexity.

The paper is organized as follows. In Section 2, we review depth map estimation. Then, we explain the principles of the novel enhancement method in Section 3. We present the results in Section 4 and, finally, we conclude in Section 5.

2. REVIEW OF DEPTH MAP ESTIMATION

Depth map estimation attracted a lot of attention and resulted in a number of methods with different accuracy and complexity.

In [10], the depth maps are estimated by a multi-stage segmentation method. A color texture-based segmentation is followed by a coarse disparity assignment obtained by exploiting the inter-view correlation and, finally, the resulting disparity space is smoothed to improve the quality of view synthesis. In [11], the authors estimated depth maps by a stereo matching method and, then, regularized them in a filtering step. In [12], the coarse depth maps obtained by segmentation are refined using belief propagation. Even though these methods are efficient in representing accurately the scene geometry,

This work was in part developed within VISNET II, a European Network of Excellence (<http://www.visnetnoe.org>), funded under the European Commission IST FP6 programme.

they are computationally complex and require powerful processors for a real-time implementation.

Another depth estimation method proposed in [13] (and also used as a reference approach in the FVV standardization [14]) is based on stereo matching. The method uses a graph cuts algorithm for energy minimization during a bi-directional disparity search¹ and applies a regularization to a spatial neighborhood of pixels. The resulting computational complexity is low, as compared to the previous depth map estimation methods. However, the generated depth maps can be inconsistent across views and time, since the inter-view and temporal coherence is not fully exploited. A lack of inter-view coherence is caused by the estimation using only the color texture frames taken at two neighbor viewpoints. Similarly, a lack of the temporal coherence is caused by an independent generation of each depth frame across time using only the color texture frames taken at the same time instants. As a consequence, the depth maps can be locally erroneous with spot noise in some regions. This reduces both the coding performance and quality of view synthesis because of a lower prediction efficiency in the encoder, visually annoying distorted object edges and jitter in time. An improvement of the temporal coherence proposed in [15] included a depth range dependant temporal regularization factor in the energy function definition. However, this approach is not adaptive to the local motion activity in the scene and, thus, the temporal consistency can not be improved in case of pixels that are occluded at some time instants.

Our depth map enhancement method exploits the algorithm in [13] and provides a solution to the shortcomings by post-processing the depth maps using an efficient edge, motion and depth range adaptive median filtering. The filtering is applied within the multi-dimensional windows that are iteratively resized to achieve the best adaptation to the content of the signal. The novel method results in an improved coding performance and the quality of view synthesis with a limited additional complexity, as explained in the sequel.

3. DEPTH MAP ENHANCEMENT

The proposed enhancement method of the initially estimated multi-view depth maps consists of three stages: 1) warping all viewpoints to the central viewpoint, 2) application of the adaptive progressive block segmentation median filtering to the warped viewpoints to produce consistent depth maps and 3) inverse view warping of the filtered depth map to all initial viewpoints. We explain each stage next.

3.1. View warping

The multi-view per-pixel depth map sequences $d(x, y, t, n)$ are estimated at each viewpoint $n = 1, \dots, N$, time instant t and spatial location (x, y) using the algorithm from [13]. Since the goal of our method is to enforce the spatial, temporal and inter-view coherence in the depth map sequences, they have to be warped to the same viewpoint to ensure spatial alignment.

First, assuming $0 \leq d(x, y, t, n) \leq 255$, the depth maps are transformed to the real-world depths $D(x, y, t, n)$ by

$$D(x, y, t, n) = \left(\frac{d(x, y, t, n)}{255} \cdot (z_{near}^{-1} - z_{far}^{-1}) + z_{far}^{-1} \right)^{-1}, \quad (1)$$

where z_{near} and z_{far} are the smallest and the largest depths in the scene, respectively.

Then, these depths are used to obtain the corresponding three-dimensional coordinates as

$$(u, v, w) = R(n) \cdot A^{-1}(n) \cdot (x, y, 1) \cdot D(x, y, t, n) + T(n),$$

where the functions $A(n)$, $R(n)$ and $T(n)$ represent the intrinsic camera parameters, rotation and translation, respectively, at the n th viewpoint. The three-dimensional coordinates are further warped to the same n_0 th viewpoint using

$$(u', v', w') = A(n_0) \cdot R^{-1}(n_0) \cdot \{(u, v, w) - T(n_0)\}$$

to ensure spatial alignment of the depth maps estimated at different viewpoints. Finally, the warped coordinates (x', y') are expressed in a homogenous two-dimensional form as $x' = u'/w'$ and $y' = v'/w'$. The corresponding warped depth map is denoted as

$$d'_{n_0}(x, y, t, n) = d(x', y', t, n). \quad (2)$$

3.2. Adaptive median filtering

The depth maps $d(x, y, t, n)$ are first warped using (2) to the central viewpoint $n_0 = \lceil N/2 \rceil$. Then, the warped depth map values $d'_{n_0}(x, y, t, n)$ are transformed to the real-world depths $D'_{n_0}(x, y, t, n)$ at the viewpoint n_0 based on the transform in (1).

To exploit spatial, temporal and inter-view coherence, median filtering is applied to $D'_{n_0}(x, y, t, n)$ within a 4-dimensional window \mathcal{S} . The shape and size of the window \mathcal{S} are adaptive to edges, motion and depth range in the multi-view sequences to prevent a visually annoying distortion in synthesized views. This adaptation is based on three parameters that measure locally a presence of edges, depth range and motion activity. The parameters are denoted as 1) local variance v of the depth values, 2) local mean m_d of the depth values and 3) local mean m_c of the absolute difference between the luminance components of the two consecutive color texture frames, respectively. All parameters are computed in a $2^m \times 2^m$ spatial neighborhood $\mathcal{N}_{t, n_0, m}(i, j)$ at the time instant t and viewpoint n_0 that consists of the pixels (x, y) , such that $i \leq x < i + 2^m$, $j \leq y < j + 2^m$ and the integer $1 \leq m \leq M$.² Hence, the first parameter is computed from the transformed depth values $D'_{n_0}(x, y, t, n_0)$ as

$$v(t, m) = \text{var}[D'_{n_0}(\mathcal{N}_{t, n_0, m})].$$

The parameter m_d is obtained as the local average of the depth value

$$m_d(t, m) = \text{mean}[D'_{n_0}(\mathcal{N}_{t, n_0, m})].$$

Finally, the third parameter is computed from the luminance components c of the color texture sequences at two consecutive time instants $(t-1)$ and t taken at the n_0 th viewpoint, that is,

$$m_c(t, m) = \text{mean}[|c(\mathcal{N}_{t, n_0, m}) - c(\mathcal{N}_{t-1, n_0, m})|].$$

The first parameter v is compared to the threshold T_v to detect a presence of edges. If no edge is detected ($v \leq T_v$), then the spatial coherence is exploited and the window \mathcal{S} consists of the $2^m \times 2^m$ neighborhoods across all views, that is, $\mathcal{S}_v = \cup_{n=1}^N \mathcal{N}_{t, n, m}$. Otherwise, a finer segmentation is obtained by iterative partition of the spatial neighborhood into 4 equal $\mathcal{N}_{t, n_0, m-1}$ neighborhoods of the size $2^{m-1} \times 2^{m-1}$. The sensitivity to edges is scaled with depth, that is, a higher threshold T_v is chosen in the area far from camera (for $m_d > T_d$) and vice versa. In this way, a coarser partition is applied to the background area using larger windows \mathcal{S} because of a

¹At the left and right directions across the neighbor viewpoints.

²For a convenient notation, we drop the index (i, j) whenever possible.

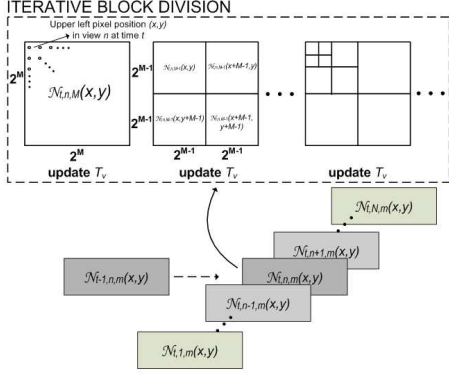


Fig. 1. Median filtering is applied within the window \mathcal{S} computed after an iterative block partition, where the corresponding thresholds T_v and T_m are updated in each iteration. The window includes the depth map values from the current t and previous $(t-1)$ time instants at the views $n = 1, \dots, N$. In this way, the spatial, temporal and inter-view coherence is exploited without distortions in the synthesized views around edges or motion.

smaller disparity. Furthermore, such an adaptation reduces the overall computational complexity, because of joint processing performed on many pixels, and it improves the visual quality of the synthesized views. The threshold T_d is chosen so that the disparity between two neighbor pixels after the view warping is equal to zero for an average physical distance in the scene. Assuming $\Delta D'$ is this distance, the disparity p is given by

$$p = \left\lfloor \frac{a \cdot \Delta D'}{D'_{n_0}(x, y, t, n) \cdot D'_{n_0}(x+1, y, t, n)} \right\rfloor, \quad (3)$$

where a is a constant derived from the rotational, translational and affine parameters of the source and target viewpoints. The value of T_v is updated throughout the iteration.

If no motion is detected in the window (that is, $m_c(t, m) \leq T_m$), then the temporal coherence is enforced by including the locations at the previous time instant $(t-1)$ in the window and, thus, $\mathcal{S} = \mathcal{S}_v \cup \mathcal{N}_{t-1, n_0, m}$. Otherwise, $\mathcal{S} = \mathcal{S}_v$.

Finally, the resulting depth values are obtained by applying the median filter to the adaptive window \mathcal{S} as

$$D'_{n_0}(\mathcal{N}_{t, n_0, m}) = \text{median}[D'_{n_0}(\mathcal{S})], \quad (4)$$

for all $n = 1, \dots, N$. Notice that the inter-view coherence is exploited by using the same resulting real-world depth value $D'_{n_0}(x, y, t, n)$ for all views $n = 1, \dots, N$. The described process is illustrated in Fig. 1.

3.3. Inverse view warping

The resulting depth map sequence $D'_{n_0}(x, y, t, n)$ is first transformed back to $d'_{n_0}(x, y, t, n)$ using the inversion of (1) and, then, inverse warped to the original viewpoints. This inversion is implemented following the opposite order of steps from Section 3.1.

3.4. Occluded pixels

Notice that the depth values for the pixels visible in one of the viewpoints, but occluded in the central viewpoint $\lceil N/2 \rceil$, are not computed during the median filtering. To improve also the consistency of these values, the three stages explained in Sections 3.1-3.3 are

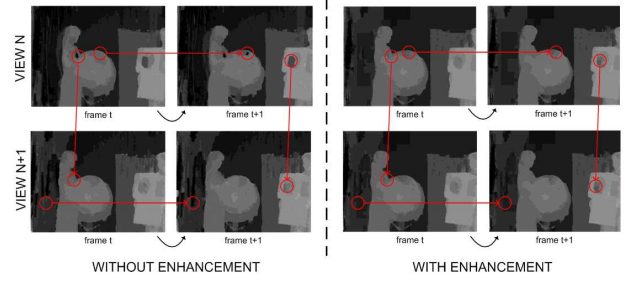


Fig. 2. An example of the effect of median filtering on the depth maps. Four frames are shown for two successive viewpoints and time instants. Notice the improvement of the spatial, temporal and inter-view coherence in the processed depth maps.

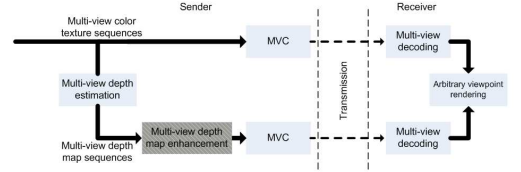


Fig. 3. A common FVV scenario: the input multi-view color texture sequences are used for depth map estimation and, then, both signals are encoded using MVC and sent to the receiver. Our enhancement method is implemented as a post-processing of the estimated depth-maps.

iterated for these pixel positions with the target warping viewpoint $n_0 = \lceil N/2 \rceil \pm 1, \lceil N/2 \rceil \pm 2, \dots$ instead of $\lceil N/2 \rceil$. This iteration continues until either all depth values are processed or all possible target viewpoints n_0 are exhausted. Notice also that this additional procedure does not carry a significant overhead complexity because the occluded regions have typically small size.

Fig. 2 shows an example of depth maps processed by median filtering. The four frames are sampled at two successive viewpoints and time instants and processed by the proposed algorithm. The spatial, temporal and inter-view coherence is apparently improved after filtering.

4. RESULTS

A common FVV scenario comprises a depth map estimation from the input signal and two MVCs applied to both multi-view depth maps and color texture sequences, as shown in Fig. 3. The proposed efficient enhancement method is integrated in this scenario as a post-processing of the estimated multi-view depth maps. Both sequences are encoded using the Joint Multi-view Video Model (JMVM), version 6.0, developed within the JVT. A hierarchical B-prediction across time and view is used and the temporal Group of Pictures (GOP) size is set to 8. The MVC blocks are tuned in such a way that the encoding of depth maps spends between 20% and 40% of the total bit-rate sent to the receiver.

The experiments are made with two test multi-view videos (also used in [13]): 1) Akko with 5 consecutive viewpoints (#26 - #30) and 2) Rena with 7 consecutive viewpoints (#41 - #47). The cameras are arranged on a line and the sequences are rectified. The encoding is performed at 4 different quantization points for both the multi-view color texture and multi-view depth map sequences.

To compare the quality of the view synthesis using the original and enhanced depth maps, we estimate the color texture sequence at the view #28 in case of Akko and at the views #42 and #46 in case of Rena using the neighbor color texture and depth map sequences. The threshold T_v is chosen as $T_v = 20$, for $m_d \leq T_d$, and

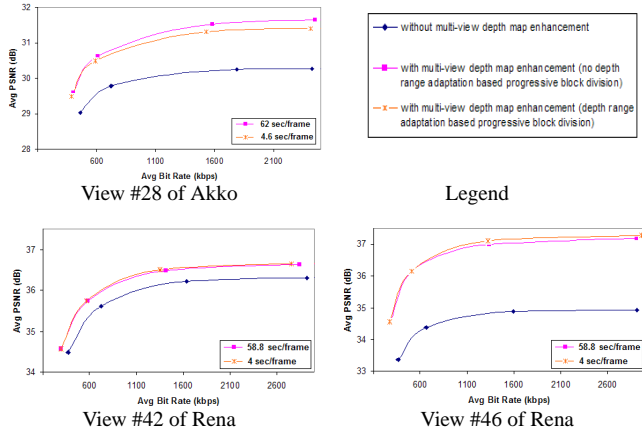


Fig. 4. The color texture sequences at the viewpoint #28 of Akko and the viewpoints #42 and #46 of Rena are synthesized using the neighbor color texture and depth map sequences. The synthesis is performed using 3 types of depth maps: a) original (non-enhanced), b) enhanced using only edge and motion adaptation with small windows and c) enhanced using the novel method with edge, motion and depth range adaptation. The novel adaptive method outperforms the scheme without enhancement and it reduces the computational time as compared to the second scheme.

$T_v = 100$, otherwise, where $T_d = 70$. The threshold $T_m = 50$, whereas the maximal $M = 6$. The quality of the synthesized sequences is measured in terms of PSNR with the original sequences at the same viewpoints as reference signals. The PSNR values are plotted in Fig. 4 for these three views and for a wide interval of total coding bit-rates. Notice that the novel method strongly outperforms the method with the original depth maps. The results are also compared to the performance of the method using only edge and motion adaptation (without depth range adaptation) with a small window size. While the PSNR is comparable, the computational time is significantly reduced in the novel method, as also shown in Fig. 4. The obtained visual quality of the synthesized views is noticeably better than in the case of the original depth maps, as depicted in Fig. 5. The annoying artifacts that are especially visible around edges and occluded pixels are suppressed and the synthesized sequences show a higher coherence across space and time.

5. CONCLUSIONS

In this work, we propose a novel and computationally efficient enhancement method for the multi-view depth map sequences based on a locally edge, motion and depth range adaptive median filtering. To avoid vast median filter computation at every pixel location, an iterative block partition median filtering is used. The method allows for an improvement of the coding performance in a common FVV scenario and provides a better quality of synthesized color texture sequences at arbitrary virtual viewpoints. Furthermore, the additional computational complexity added by the method is kept low. Hence, it is possible to incorporate the proposed enhancement scheme in fast depth map estimation techniques to strengthen the performance of compression and virtual view rendering without a substantial increase in computational complexity.

6. REFERENCES

[1] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.

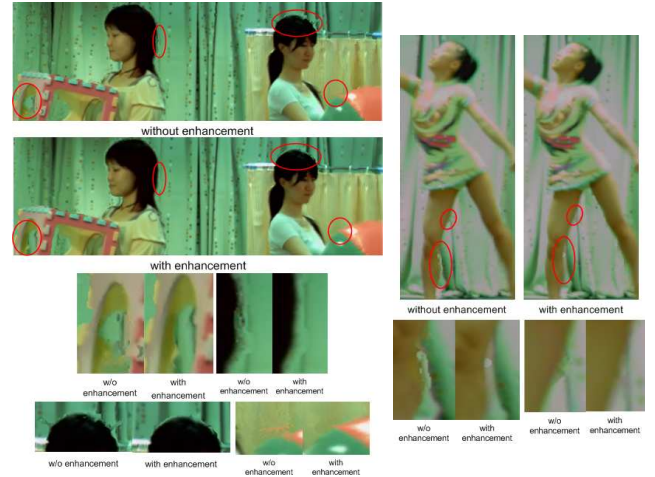


Fig. 5. An example of synthesized frames of Akko and Rena with the original depth maps and with enhanced depth maps using the novel adaptive scheme. The difference between the performance of the two schemes is shown in the magnified images. The artifacts that appear in the synthesized frames without the enhancement are suppressed using the adaptive filtering.

[2] K. Mueller, A. Smolic, M. Kautzner, P. Eisert, and T. Wiegand, "Rate-distortion-optimized predictive compression of dynamic 3-D mesh sequences," *Signal Proc.: Image Comm.*, vol. 21, no. 9, pp. 812–828, 2007.

[3] E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun, "Extensions of H.264/AVC for multiview video compression," in *IEEE Int. Conf. on Image Proc.*, Atlanta, GA, 2006.

[4] E. Ekmekcioglu, S. Worrall, and A. Kondoz, "Bit-rate adaptive downsampling for the coding of multi-view video with depth information," in *Proc. IEEE 3DTV Conference 2008*, Istanbul, Turkey, May 2008.

[5] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*, Springer-Verlag, 2007.

[6] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *IEEE Int. Conf. on Image Proc.*, San Antonio, TX, 2007.

[7] K. Mueller, P. Merkle, H. Schwarz, T. Hinz, A. Smolic, and T. Wiegand, "Multi-view video coding based on H.264/AVC using hierarchical B-frames," in *Picture Coding Symposium*, Beijing, China, Apr. 2006.

[8] Y. Morvan, D. Farin, and P. H. N. de With, "Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images," in *IEEE Int. Conf. on Image Proc.*, San Antonio, TX, 2007.

[9] M. Maitre and M. N. Do, "Joint encoding of the depth image based representation using shape-adaptive wavelets," in *IEEE Int. Conf. on Image Proc.*, San Diego, CA, 2008.

[10] L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. ACM SIGGRAPH*, New York, NY, 2004.

[11] C. Cigla, X. Zabulis, and A. Alatan, "Region-based dense depth extraction from multi-view video," in *IEEE Int. Conf. on Image Proc.*, San Antonio, TX, 2007.

[12] S. Lee, K. Oh, and Y. Ho, "Segment-based multi-view depth map estimation using belief propagation from dense multi-view video," in *IEEE 3D-TV Conf.*, Istanbul, Turkey, 2008.

[13] M. Tanimoto, T. Fujii, and K. Suzuki, "Multi-view depth map of Rena and Akko & Kayo," *MPEG Doc. M14888*, Oct. 2007.

[14] ISO/IEC JTC1/SC29/WG11, "Description of exploration experiments in 3D video coding," *MPEG Doc. N9991*, July 2008.

[15] S. Lee and Y. Ho, "Enhancement of temporal consistency for multi-view depth map estimation," *MPEG Doc. M15594*, July 2008.