

# Content adaptive enhancement of multi-view depth maps for free viewpoint video

Erhan Ekmekcioglu, Vladan Velisavljević, *Member, IEEE*, and Stewart T. Worrall, *Member, IEEE*

## Abstract

Depth map estimation is an important part of the multi-view video coding and virtual view synthesis within the free viewpoint video applications. However, computing an accurate depth map is a computationally complex process, which makes real-time implementation challenging. Alternatively, a simple estimation, though quick and promising for real-time processing, might result in inconsistent multi-view depth map sequences. To exploit this simplicity and to improve the quality of depth map estimation, we propose a novel content adaptive enhancement technique applied to the previously estimated multi-view depth map sequences. The enhancement method is locally adapted to edges, motion and depth-range of the scene to avoid blurring the synthesized views and to reduce the computational complexity. At the same time, and very importantly, the method enforces consistency across the spatial, temporal and inter-view dimensions of the depth maps so that both the coding efficiency and the quality of the synthesized views are improved. We demonstrate these improvements in the experiments, where the enhancement method is applied to several multi-view test sequences and the obtained synthesized views are compared to the views synthesized using other methods in terms of both numerical and perceived visual quality.

## Index Terms

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

## I. INTRODUCTION

In recent years, the development of the 3D video technology brought into focus a new scene representation technique called *multiple view video* (MVV). In this technique, a scene is captured by a set of synchronized cameras located at different spatial locations (*viewpoints*) [1]. The generated signal is represented as a set of video sequences captured at each camera. This technique is often referred to as sampling of *light field* or *plenoptic function* [2].

One of the most popular applications within the MVV framework is *Free Viewpoint Video* (FVV) [3], [4]. In this application, the user freely chooses a viewpoint in the scene, which does not have necessarily to coincide with the camera viewpoints. One of the major challenges in the FVV is how to encode and transmit the large amount of data required to synthesize realistic videos at the chosen viewpoints with an acceptable quality [5].

E. Ekmekcioglu and S. T. Worrall are with the Centre for Comm. Systems Research, University of Surrey, Guildford, UK, e-mail: {E.Ekmekcioglu, S.Worrall}@surrey.ac.uk

V. Velisavljević is with Deutsche Telekom Laboratories, Berlin, Germany, e-mail: Vladan.Velisavljevic@telekom.de.

The current commercial broadcasting systems are not capable of processing and transmitting the signals recorded at all possible viewpoints. Furthermore, the synthesis (rendering) of the views at the receiver side is computationally inefficient. For that reason, several different coding and rendering approaches have recently been proposed to improve this efficiency. In mesh-based FVV coding [6], the underlying 3D geometry of the scene, also called mesh sequences, is encoded and sent to the user along with the corresponding photometric properties. This information is then used to synthesize videos at arbitrary viewpoints. Similarly, in [7], the wavelet transform is applied to the mesh sequences in both spatial and temporal dimensions and the resulting subbands are entropy-coded. These methods significantly reduce the amount of data needed for transmission, but they also require a precise knowledge of the 3D geometry in the scene, which is still challenging to acquire in a general case. Furthermore, the latter condition implies the use of many cameras resulting in a high system cost and complex camera calibration.

By contrast, in image-based rendering (IBR) [8]–[10], the captured MVV signal is encoded as a set of color texture video sequences and the intermediate views are synthesized using only this information. As a result, the synthesized views are more photorealistic. Moreover, even complex objects in the scene can be reconstructed without additional effort or geometrical knowledge. However, the required transmission bandwidth is significantly higher than that of the mesh-based coding. This usually causes difficulties in broadcast system design, raises costs and imposes processing power limitations.

The FVV coding based on depth maps [11], [12] is a balanced solution between the mesh-based coding and IBR. It provides a local approximation of the light-field function at the chosen viewpoints. This approximation allows for a reduced number of camera viewpoints, while the quality of synthesized views is still preserved. A similar approach has been adopted within the standardization group of the Joint Video Team (JVT), where the Multi-View Coding (MVC) tools are used for the compression of both the MVV color texture sequences and associated per-pixel depth map sequences [13], [14]. These two types of sequences are encoded separately using the MVC method based on the hierarchical B-prediction and H.264/AVC, which exploits the intra-frame spatial, inter-frame temporal and inter-view correlation in the multi-view video [5], [15], [16]. 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 sub-optimal coding performance.

In some recent work, this shortcoming has been addressed and the depth maps have been encoded using modified methods based on wedgelets and platelets [17]. The corresponding coding performance has been shown to perform better than the one achieved by H.264/AVC [18], in terms of the view synthesis quality. Another reported depth map coding scheme has been based on shape-adaptive wavelets [19], where the correlation between location of edges in the depth map and the MVV color texture sequences has been exploited. A hybrid view synthesis technique proposed in [20] has reduced the required number of cameras using a simple interpolation step between the captured views. View interpolation has also been analyzed in [21], whereas the MVV coding performance has been improved using the Wyner-Ziv coding in [22]. Depth map information has been coded using a multi-layer representation in [23], whereas an enhancement technique has been applied to the captured MVV sequences in [24]. Finally, in [25], a coding artifact reduction method has been introduced to improve the quality of the synthesized views.

Our goal is to efficiently post-process the estimated multi-view depth maps so that both the resulting coding efficiency and the quality of view synthesis at decoder are improved without significant additional complexity. We propose a novel content adaptive enhancement method based on median filtering of the depth maps to enforce coherence across the spatial, temporal and inter-view dimensions. The median filtering is adapted to the presence of edges and motion in the MVV signal to preserve the sharpness of these features, which is important for the visual quality of the synthesized videos. Furthermore, the resolution of the filtering is adapted to the distance of objects in the scene so that the processing of the closer objects (foreground) is finer than the processing of the farther objects (background). This adaptation reduces the overall computational complexity, while preserving the same visual quality.

The proposed enhancement method allows even a simple and less accurate initial depth estimation to achieve efficient coding and good view rendering at the decoder in terms of both objective and visual quality of the synthesized views. Since our method carries a low computational complexity, it is a promising technique for future real-time implementations. We show that the enhanced depth maps lead to a gain in the coding and rendering performance for several MVV test sequences. We also show an improvement of the visual quality perceived by test subjects for those test videos.

Notice that in our previous work [26], we analyzed the depth map enhancement based on only edge and motion adaptation. In the follow-up [27], we also explained the adaptation to the distance of objects in the scene. However, this work exploited hard-thresholding as the core technique, which involved heuristic methods to determine the values of the thresholds. That motivated us to continue this work and to propose a fully adaptive method without empirically chosen parameters.

The paper is organized as follows. In Section II, we review the state of the art of depth map estimation. Then, in Section III, we explain the principles of the novel content adaptive enhancement method. We present the numerical results of our method and also the assessment of the perceived quality by subjects in Section IV. Finally, we conclude the paper in Section V.

## II. REVIEW OF DEPTH MAP ESTIMATION

Recently, depth map estimation attracted a lot of attention and resulted in a number of methods with different accuracy and complexity. In [28], the depth maps were estimated by a multi-stage segmentation method. A color texture-based segmentation was followed by a coarse disparity assignment obtained by exploiting the inter-view correlation and, finally, the resulting disparity space was smoothed to improve the quality of view synthesis. In [29], the authors estimated depth maps by a stereo matching method and, then, regularized them in a filtering step. In [30], the coarse depth maps obtained by segmentation were refined using belief propagation. Moreover, belief propagation and camera optical flow were used in [31], where depth maps were denoised. Even though these methods provide an accurate representation of the scene geometry, they are computationally complex and make a real-time implementation very difficult.

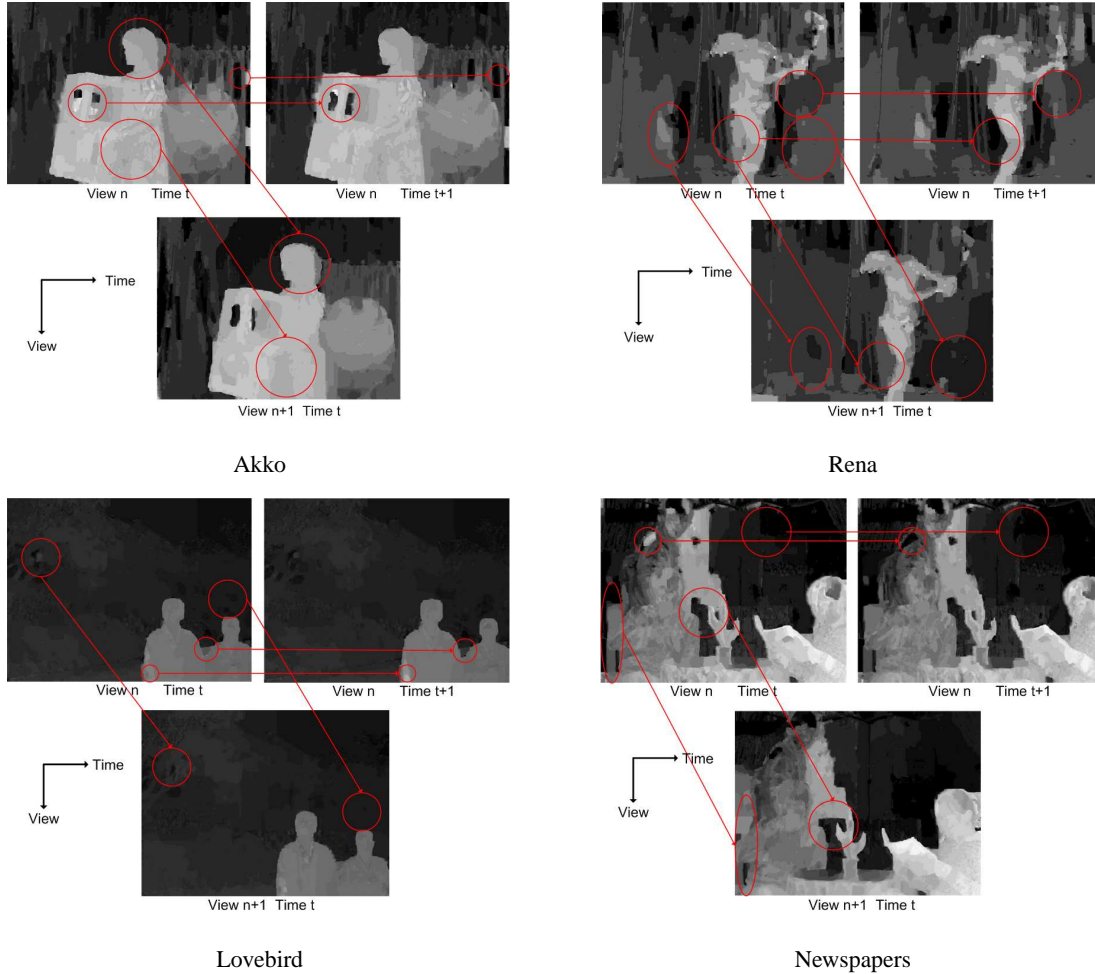


Fig. 1. Inconsistency of the estimated multi-view depth maps using the method from [32]. The method is used because of its computational simplicity. After enhancement, it can achieve a similar estimation results as other more complicated methods. The resulting depth maps are shown for 4 test MVV sequences: Akko, Rena, Lovebird and Newspapers. The most representative inconsistent areas are marked in red.

Another depth estimation method proposed in [32] and also used as a reference approach in the FVV standardization [33] in MPEG is based on stereo matching. The method uses a graph cuts algorithm for energy minimization during a bi-directional disparity search<sup>1</sup> and applies 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 viewpoints and time, since the inter-view and temporal coherence is not fully exploited (see Fig. 1). The lack of inter-view coherence is caused by the estimation using the color texture frames taken only at two neighbor viewpoints. Similarly, the lack of the temporal coherence is caused by the independent generation of each depth map frame across time using only the color texture frames taken at the same time instants.

<sup>1</sup>At the left and right directions across the neighbor viewpoints.

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 the lower prediction efficiency in the encoder, visually annoying distorted object edges and jitter in time. An improvement of the temporal coherence proposed in [34] included a temporal regularization factor in the energy function definition dependent on the depth difference of the temporal neighbor pixels. 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.

Owing to the low computational complexity of this depth estimation algorithm and its use within the MPEG, we exploit it in the enhancement method, despite the explained potential inconsistencies. The proposed enhancement provides a solution to the shortcomings by post-processing the depth maps using an efficient content 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, while the overall computational complexity is retained low.

### III. CONTENT ADAPTIVE DEPTH MAP ENHANCEMENT

Our novel 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 in the sequel. Notice that a part of this method has been also presented in our previous work [26], [27]. However, here, more systematic methods have replaced the empirical thresholding used in those articles.

#### A. View warping

The multi-view per-pixel depth map sequences estimated using [32] are denoted as  $d(k, l, t, n)$ , where  $(k, l)$  is the spatial coordinate,  $t$  time instant and  $n = 1, \dots, N$  camera viewpoints, respectively. 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  $d(k, l, t, n)$  is normalized, that is,  $0 \leq d(k, l, t, n) \leq 255$ , the depth maps are transformed to the real-world depths  $D(k, l, t, n)$  by

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

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

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

where the square matrices  $A(n)$  and  $R(n)$  represent the intrinsic camera parameters and rotation at the  $n$ th viewpoint, respectively, whereas the column-vector  $T(n)$  is the corresponding translation. The three-dimensional coordinates

are further warped to the same  $n_0$ th viewpoint using

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

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

$$\tilde{d}_{n_0}(k, l, t, n) = d(k', l', t, n). \quad (2)$$

Notice that, due to the limitation of the stereo-matching method used in [32], the initially estimated depth maps  $d(k, l, t, n)$  are erroneous. Consequently, this error is propagated to the warped versions  $\tilde{d}_{n_0}(k, l, t, n)$ . However, such errors are incoherent across viewpoints and, thus, they can be efficiently removed by the following adaptive median filtering.

### B. Adaptive median filtering

To apply the adaptive median filtering, the depth maps  $d(k, l, t, n)$  are first warped to the central viewpoint  $n_0 = \lceil N/2 \rceil$  using (2). Then, the warped depth map values  $\tilde{d}_{n_0}(k, l, t, n)$  are transformed to the real-world depths  $\tilde{D}_{n_0}(k, l, t, n)$  at the viewpoint  $n_0$ , as in (1).

Median filtering is applied to  $\tilde{D}_{n_0}(k, l, t, n)$  locally to enforce the four-dimensional spatial, temporal and inter-view coherence. The filtering is adaptive to edges, motion and depth-range in the MVV sequences to prevent visually annoying distortions in synthesized views.

This adaptation is based on three parameters that locally measure the depth-range, the presence of edges and the motion activity. The parameters are denoted as (1) local mean  $m_d$  of the depth values, (2) local variance  $v_d$  of the depth values and (3) local mean  $m_c$  of the absolute difference between the luminance components in the two consecutive color texture frames. The first two parameters,  $m_d$  and  $v_d$ , are computed from the depth map sequences and they scale the size of the local spatial window denoted as  $\mathcal{S}_t$ , whereas the third parameter  $m_c$  is calculated from the color texture sequences and it determines the weight factor for the depth map pixels at the previous time instant  $(t - 1)$ . All the parameters are computed in a square  $2^m \times 2^m$  spatial neighborhood  $\mathcal{N}_{t, n_0, m}(i, j)$  that consists of the integer pixel coordinates  $(k, l)$ , such that  $i - \frac{1}{2}2^m \leq k < i + \frac{1}{2}2^m$  and  $j - \frac{1}{2}2^m \leq l < j + \frac{1}{2}2^m$ , taken at the time instant  $t$  and viewpoint  $n_0$ . The scaling integer  $m$  takes values from the interval  $1 \leq m \leq M$ , where the choice of  $M$  does not affect significantly the enhancement performance (in the experiments in Section IV,  $M = 6$ ). We next explain the computation of the three parameters and their influence on the median filtering process in detail. Notice that, for a convenient notation, we drop the index  $(i, j)$  whenever possible.

1) *Parameter  $m_d$* : The parameter  $m_d$  represents the local average depth-range in the scene and it is obtained as

$$m_d(t, m) = \text{mean} \left[ \tilde{D}_{n_0}(\mathcal{N}_{t, n_0, m}) \right]. \quad (3)$$

This parameter is used to separate the background and foreground areas in the scene and to scale appropriately the size of the spatial window  $\mathcal{S}_t$ . In the background areas, adaptation to edges and motion is less important for

the visual quality of the rendered views than the same in the foreground areas because of the larger distance from the camera, that is, the smaller disparity. For that reason, the spatial partition imposed by the spatial window  $\mathcal{S}_t$  is coarser in the background areas and the window remains larger. Then, the median filtering is applied jointly to larger sets of pixels and the computational complexity is reduced without affecting the visual quality.

To determine the background area, the parameter  $m_d$  is hard-thresholded with the threshold  $T_d$  and a pixel is considered as background if  $m_d > T_d$ . The threshold  $T_d$  is computed as the minimal depth such that the disparity shift between two neighbor pixels after warping is quantized to zero. That is, assume that  $D_1 = D(k, l, t, n)$  and  $D_2 = D(k + 1, l, t, n)$  are the real-world distances of two neighbor pixels at the viewpoint  $n$  and at the coordinates  $(k, l)$  and  $(k + 1, l)$ , respectively. The disparity shift between these two pixels after warping to the viewpoint  $n_0$  is given by

$$p = a \cdot \frac{|\tilde{D}_{n_0}(k, l, t, n) - \tilde{D}_{n_0}(k + 1, l, t, n)|}{\tilde{D}_{n_0}(k, l, t, n) \cdot \tilde{D}_{n_0}(k + 1, l, t, n)},$$

where  $a$  is derived from the rotational, translational and affine parameters of the source viewpoint  $n$  and target viewpoint  $n_0$  as

$$[a, a_1, a_2]^T = A(n_0) \cdot R^{-1}(n_0) \cdot (T(n_0) - T(n)).$$

Assume  $|D_1 - D_2|$  is equal to the minimally quantized difference (depth resolution). Then, the threshold  $T_d$  is the distance  $\min(D_1, D_2)$  such that  $\lfloor p \rfloor = 0$ . Hence, all the neighbor points with smoothly changing real-world distances larger than  $T_d$  retain their relative positions after warping to the viewpoint  $n_0$ .

2) *Parameter  $v_d$* : The second parameter  $v_d$  is computed from the transformed depth values  $\tilde{D}_{n_0}(k, l, t, n_0)$  as

$$v_d(t, m) = \text{var} \left[ \tilde{D}_{n_0}(\mathcal{N}_{t, n_0, m}) \right], \quad (4)$$

which measures the local changes in the depth map frames. To detect the presence of edges in the sequences, the parameter  $v_d$  is compared to the threshold  $T_v$ .<sup>2</sup> If no edge is detected, that is, if  $v_d \leq T_v$ , then the spatial coherence is fully exploited and the spatial window  $\mathcal{S}_t = \cup_{n=1}^N \mathcal{N}_{t, n, m}$  includes the entire  $2^m \times 2^m$  neighborhood across all viewpoints. Otherwise, for  $v_d > T_v$ , a finer segmentation is applied to avoid smoothing the edge and reducing the visual quality of the reconstructed view. The finer segmentation is obtained by iterative spatial partition of the neighborhood into 4 equal  $\mathcal{N}_{t, n_0, m-1}$  neighborhoods of the size  $2^{m-1} \times 2^{m-1}$ . Then,  $v_d$  is recomputed in each of the new neighborhoods using (4).

Following the adaptation to the depth-range explained in Section III-B1 and to preserve bigger neighborhoods in the background areas, the threshold  $T_v$  should have smaller values in the background and larger values in the foreground. Thus,  $T_v$  is constrained to be an increasing function of  $m_d$ . For the reasons of conceptual simplicity,

<sup>2</sup>Recall that edges can be efficiently detected using a number of edge detectors, but, due to the conceptual and computational simplicity of our method, we exploit and process the already computed and warped depth map pixels instead of using another edge detection.

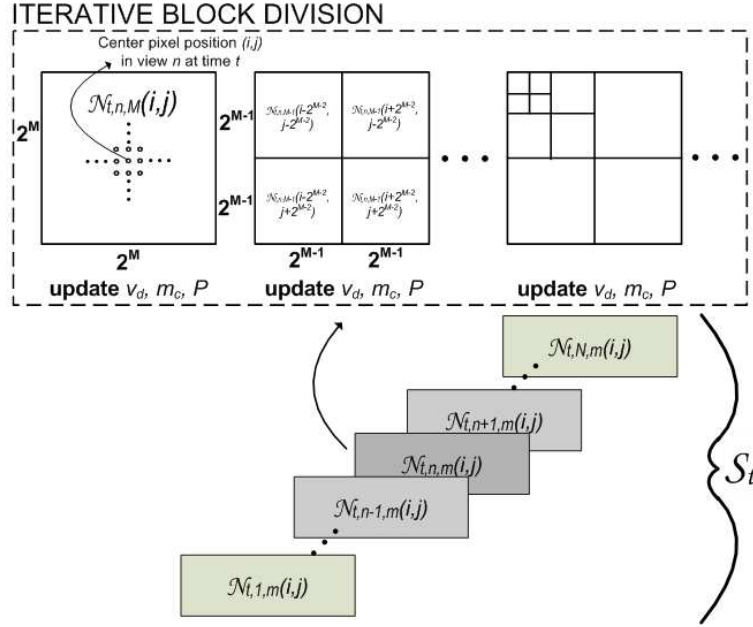


Fig. 2. Median filtering is applied to all the viewpoints  $n = 1, \dots, N$  within the spatial neighborhood  $\mathcal{S}_t$  computed after an iterative block partition. The parameters  $v_d$  and  $m_d$  influence the shape and size of the neighborhood  $\mathcal{S}_t$ , whereas the parameter  $m_c$  determines the impact of the neighborhood  $\mathcal{S}_{t-1}$  taken at the previous time instant ( $t-1$ ) on the set of coefficients  $\mathcal{P}$ . These parameters and the set  $\mathcal{P}$  are recomputed in each iteration. In this way, the spatial, temporal and inter-view coherence is exploited without distortions in the synthesized views around edges or motion.

it is defined as a piecewise linear function, that is,

$$T_v(m_d) = \begin{cases} T_{min}, & m_d \leq T_d - \frac{1}{2}T_\delta \\ m_d - T_d + (T_{max} + T_{min})/2, & |m_d - T_d| \leq \frac{1}{2}T_\delta \\ T_{max}, & m_d \geq T_d + \frac{1}{2}T_\delta \end{cases},$$

where  $T_d$  is computed as in Section III-B1 and  $T_\delta = T_{max} - T_{min}$ . The choice of the values for  $T_{max}$  and  $T_{min}$  does not affect significantly the performance of the method as long as they are close to the maximal and minimal possible values of  $v_d$ , respectively.

3) *Parameter  $m_c$* : 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 as

$$m_c(t, m) = \text{mean} \left[ \left| c(\mathcal{N}_{t,n_0,m}) - c(\mathcal{N}_{t-1,n_0,m}) \right| \right]. \quad (5)$$

This parameter captures the motion across the consecutive frames in a local neighborhood by a simple difference operator. If the motion is detected, then the correlation of the depth maps across the time dimension is reduced and the impact of the pixels from the previous time instant ( $t-1$ ) on the median filtering is attenuated. This impact is modeled by a linear combination of the pixels at the subsequent time instants  $t$  and ( $t-1$ ) as

$$\mathcal{P} = \alpha_c \cdot \tilde{D}_{n_0}(\mathcal{S}_t) + (1 - \alpha_c) \cdot \tilde{D}_{n_0}(\mathcal{S}_{t-1}), \quad (6)$$

where  $0 \leq \alpha_c \leq 1$  is a normalized version of the parameter  $m_c$ . Note that the length of the filtering in (6) is limited only to two consecutive time instants. A longer filtering with motion compensation might result in a better enhancement of the depth maps, but with a cost of a significant additional computational and conceptual complexity and processing delay. Since our goal is to retain the low overall complexity, we use only the simple temporal filtering, as shown in (6). The computations in (5) and (6) are also updated in each iteration of refining the neighborhood  $\mathcal{N}_{t,n_0,m}$  in Section III-B2.

The resulting depth values are obtained by applying the median filtering to the values in  $\mathcal{P}$ , that is,

$$\tilde{D}_{n_0}(\mathcal{N}_{t,n,m}) = \text{median}[\mathcal{P}], \quad (7)$$

for all  $n = 1, \dots, N$ . The described process is illustrated in Fig. 2.

Notice that the resulting real-world depth values are used for each viewpoint  $n = 1, \dots, N$  to exploit the inter-view coherence. Notice also that the iterative processing within square neighborhoods might lead to a blocking effect. However, in the experiments, this effect is not observable due to the non-linear median filtering (as also shown in Section IV).

### C. Inverse view warping

The obtained enhanced values of  $\tilde{D}_{n_0}(k, l, t, n)$  are warped back to the original viewpoints. First, the resulting depth map sequence is transformed to  $\tilde{d}_{n_0}(k, l, t, n)$  using the inversion of (1) and, then, these values are inverse-warped to the original viewpoints. This inversion is implemented following the opposite order of steps from Section III-A.

### D. Occluded pixels

Notice that the corresponding depth values for the pixels visible in one of the viewpoints, but occluded in the central viewpoint  $\lceil N/2 \rceil$ , cannot be computed during the median filtering. To improve also the consistency of these values, the three previously explained stages are iterated for these spatial pixel coordinates 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. 3 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.

## IV. RESULTS

A common FVV scenario comprises a depth map estimation step from the input signal and two MVC steps applied to both the multi-view depth map and color texture sequences, as shown in Fig. 4. The novel proposed enhancement method is integrated in this scenario as a post-processing of the estimated multi-view depth maps.

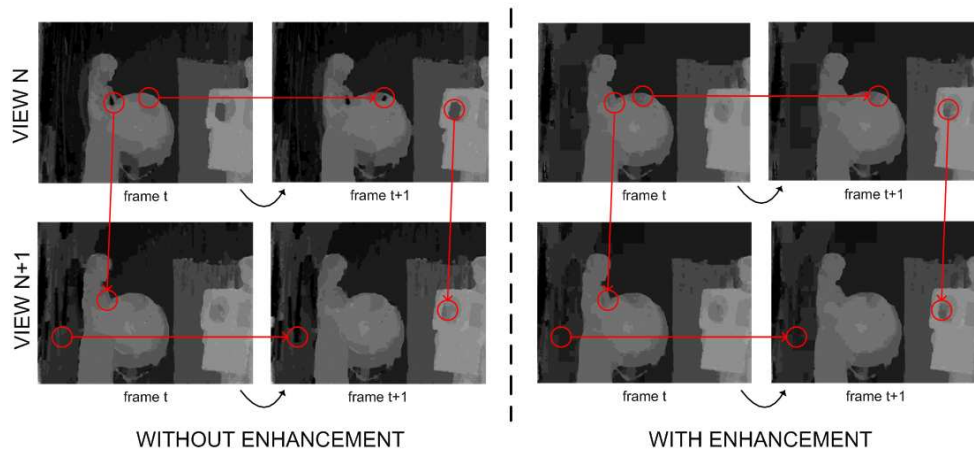


Fig. 3. An example of the effect of median filtering on the depth maps. Four frames of Akko are shown for two successive viewpoints and time instants. Notice the improvement of the spatial, temporal and inter-view coherence in the processed depth maps.

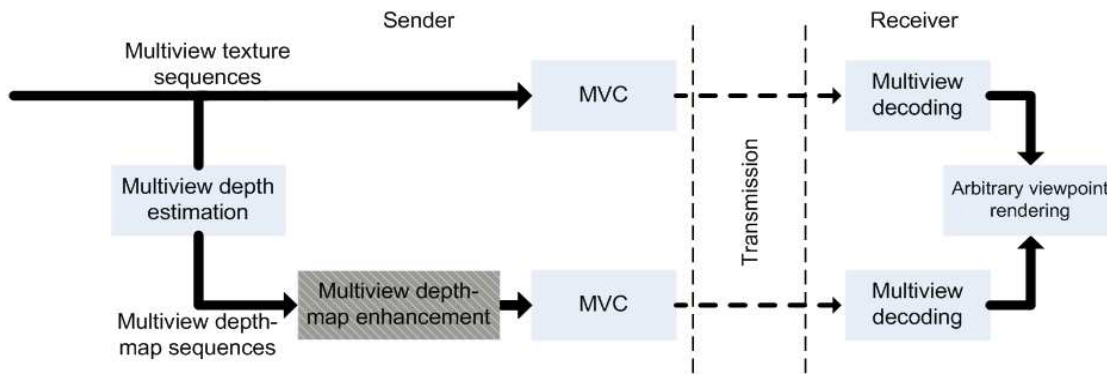


Fig. 4. A common FVV scenario: the input MVV color texture sequences are used for depth map estimation and, then, both signals are encoded using MVC and sent to the receiver. Our enhancement method is implemented as a post-processing of the estimated depth maps.

Both sequences are encoded using the Joint multi-view Video Model (JMVM), version 6.0, developed within the JVT, which exploits the hierarchical B-prediction across time and viewpoint. The temporal Group of Pictures (GOP) size is set to 8. The MVC blocks are tuned in such a way that the depth map overhead consumes between 20% and 40% of the total bit-rate sent to the receiver.

The experiments are made with four multi-view test videos, where the first two have been also used in [32]: 1) Akko with 5 consecutive viewpoints (#26 – #30), 2) Rena with 7 consecutive viewpoints (#41 – #47), 3) Lovebird with 11 consecutive viewpoints (#0 – #10) and 4) Newspapers with 9 consecutive viewpoints (#0 – #8). The cameras are arranged on a line and the sequences are rectified, which removes the effect of any possible offset in the image positions. The encoding is performed at 4 different quantization points for both the multi-view color texture and multi-view depth map sequences.

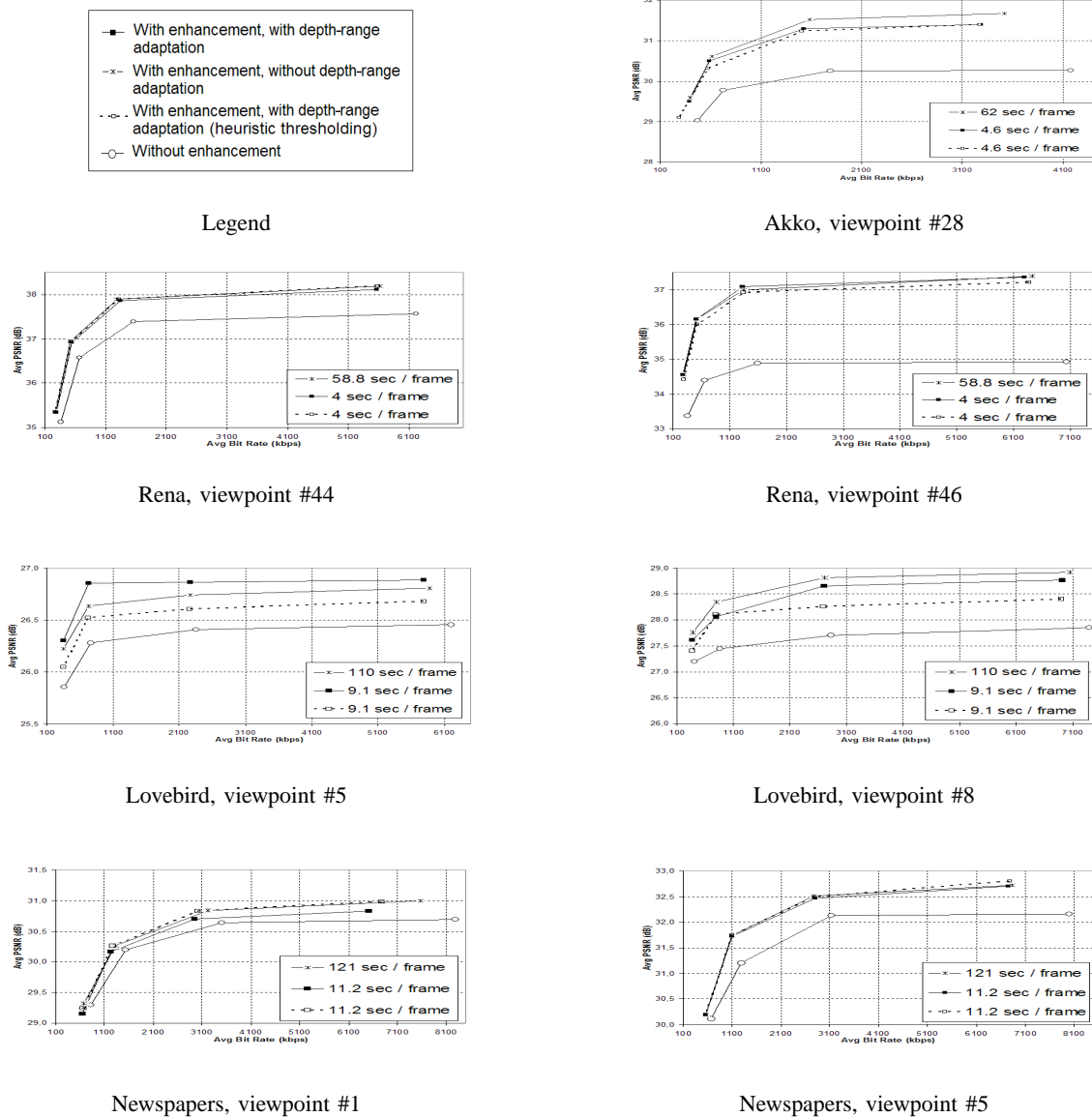


Fig. 5. The view synthesis is performed at 7 viewpoints: #28 of Akko, #44 and #46 of Rena, #5 and #8 of Lovebird and #1 and #5 of Newspapers. The synthesized views are compared in terms of PSNR to the videos captured at the same viewpoints. Four encoding and rendering algorithms are used in the comparison: 1) with original depth maps (no enhancement), 2) with enhanced depth maps adapted only to edges and motion (no depth-range adaptation) [26], 3) with fully adaptive enhancement obtained using hard-thresholding [27] and 4) the novel method with fully adaptive enhancement that avoids heuristic thresholding. The enhanced depth maps improve significantly the quality of the view synthesis. We emphasize several important results: a) all enhancement methods strongly outperform the method without depth map enhancement; b) depth-range adaptation reduces the computational complexity and elapsed time, while retaining a similar quality of the synthesized views; c) the novel method results in a comparable quality, while avoiding the heuristic thresholding used in our previous work in [27].

The depth maps are initially estimated using the algorithm proposed in [32] and then post-processed using the novel enhancement algorithm. The parameter  $M$  is set to 6, that is, the coarsest spatial window has the size  $64 \times 64$ . To compare the quality of the view synthesis using the original and enhanced depth maps for the four test MVV sequences, we synthesize the color texture sequences at the viewpoints #28 for Akko, #44 and #46 for Rena, #5 and #8 for Lovebird and #1 and #5 for Newspapers using the neighbor color texture and depth map sequences. The view synthesis is implemented using the reference View Synthesis Software developed under the MPEG FTV research group [35], which retains the depth values and applies a visual improvement to the synthesized frames to remove the artifacts produced by the occluded pixels. The synthesized views are compared in terms of PSNR to the original views captured by the cameras at the same viewpoints. The PSNR values are plotted in Fig. 5 for the specified seven viewpoints and for a wide interval of total coding bit-rates. The resulting graphs are obtained for four cases of encoding and rendering: 1) with original depth maps (no enhancement), 2) with enhanced depth maps adapted only to edges and motion (no depth-range adaptation; as also presented in [26]), 3) with fully adaptive enhancement achieved by an empirical hard-thresholding (as presented in [27]) and 4) the novel adaptive enhancement presented in this paper.

We emphasize few achievements noticeable in Fig. 5. First, all the enhancement schemes result in a significant improvement of the quality of the synthesized views as compared to the views synthesized using the non-enhanced depth maps. Furthermore, the schemes with the depth-range adaptation reduce the required computational time, while they retain a comparable quality as the method without depth-range adaptation [26].<sup>3</sup> Finally, the novel method provides similar results as the method in [27], but with an advantage of not using hard-thresholding with empirical thresholds.

In addition, the visual quality of the views synthesized using the enhanced depth maps is also noticeably better than the same obtained using the original non-enhanced depth maps. Fig. 6-8 illustrate three examples of the view synthesis for the sequences Akko, Rena and Newspapers, respectively. The synthesized sequences show an improved consistency across space and time. Moreover, the visually annoying artifacts that are especially visible around edges and occluded pixels are suppressed. This improvement is emphasized and shown separately and magnified in those three figures.

To obtain a more relevant comparison of the visual quality, the resulting synthesized videos are tested using an adjectival categorical judgement method for assessment of the subjective quality of television pictures, as suggested in the ITU-R recommendation BT.500-11 [36]. Three versions of the synthesized views for the sequences Akko, Lovebird and Newspapers are generated using the methods 1) without depth map enhancement, 2) with enhanced depth maps, but without depth-range adaptation and 3) the novel method with a full adaptation. For comparison, these versions are showed pairwise and simultaneously (side-by-side) to 20 non-expert subjects on a 32-inches screen. A snapshot of the experiment setup is shown in Fig. 9. In each test cycle, which takes around 6-10 seconds,

<sup>3</sup>Note that even though the current computational time does not indeed allow for a real-time implementation of the entire method, there is a large potential for parallel processing (e.g. forward/inverse warping of the views and the median filtering applied in separated spatial windows).

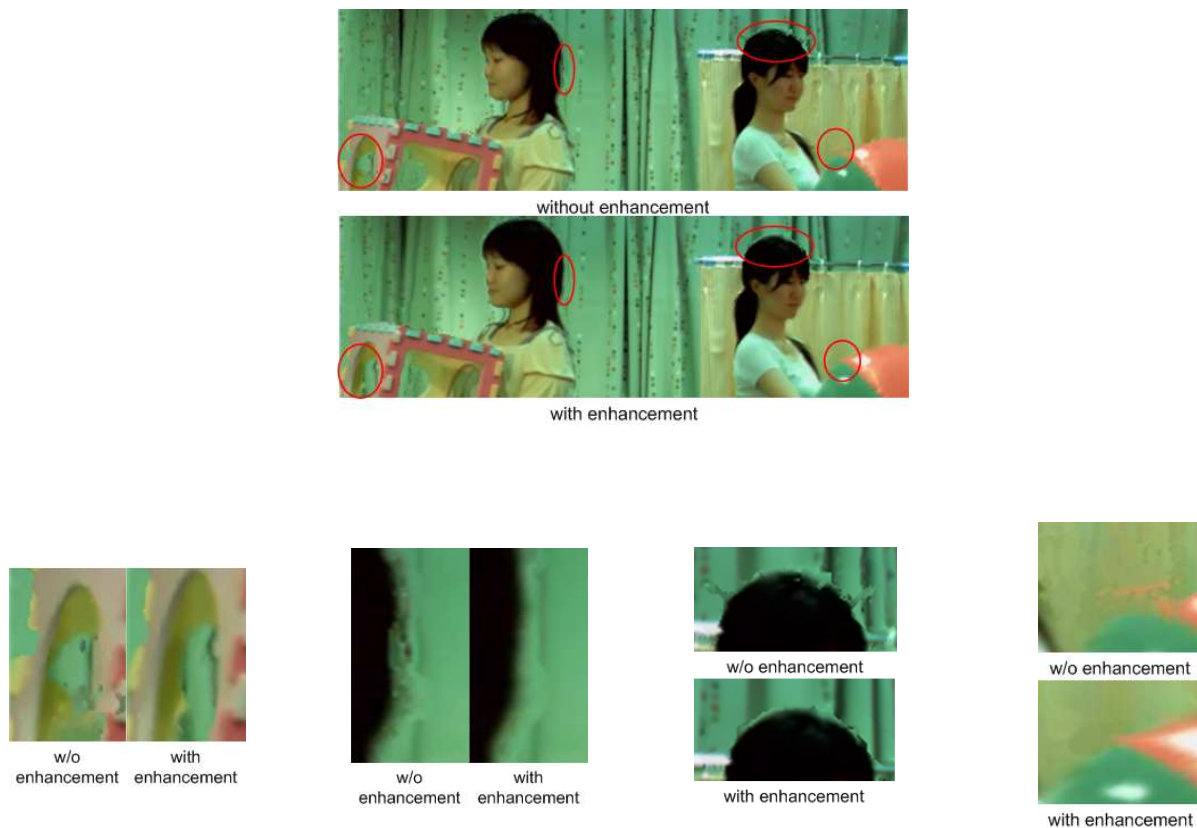


Fig. 6. An example of the synthesized frames of Akko using the original and enhanced depth maps with the novel adaptive scheme. The artifacts that appear in the synthesized frames without the enhancement are suppressed. This improvement is emphasized in the zoomed regions.

the subjects are asked to compare the shown video at the left-hand side to the counterpart reference video at the right-hand side using a qualitative comparison scale.

The results of these comparisons are shown in Fig. 10. The comparison is performed at high and low bitrate transmission, which correspond to 2.6 Mbps and 400 Kbps for Akko, 6.0 Mbps and 750 Kbps for Lovebird and 5.5 Mbps and 700 Kbps for Newspapers, respectively. For each example, encoding of the depth maps is tuned to spend between 25% and 40% of the total bit-rate, whereas the sequences are recorded at the rate of 25 frames per second. The results shown in the graphs in Fig. 10 represent the normalized differential mean opinion scores between the two compared videos with 95%-confidence intervals. The normalization scales the scores to the interval  $(-1, 1)$ . The smaller the absolute value of the score is, the smaller the perceived differences between the two videos are.

The presented scores coincide well with the numerical comparisons in Fig. 5. The comparison indicates that the proposed enhancement methods provide a better quality than the approach without any enhancement. The visual difference is even more noticeable at high bit-rates. Furthermore, the enhancement with depth-range adaption achieves a comparable performance as the other method, while it reduces the computational time.



Fig. 7. The synthesized frames of Rena using the two methods are compared. The improved details are zoomed in the right-hand side.

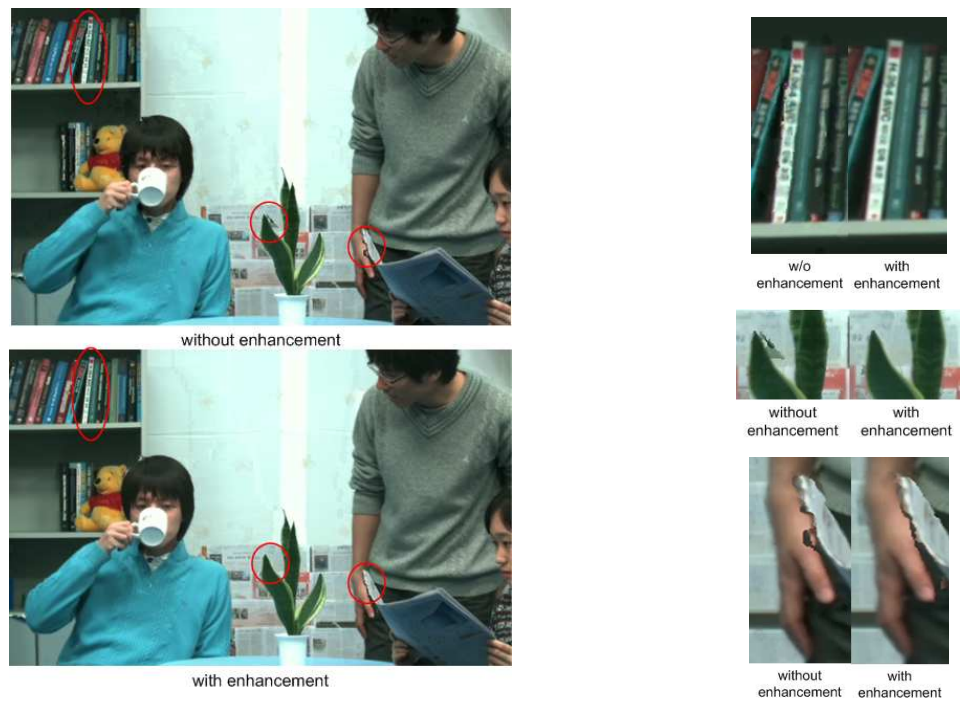


Fig. 8. The synthesized frames of Newspapers are also compared and the zoomed details are shown in the right-hand side.

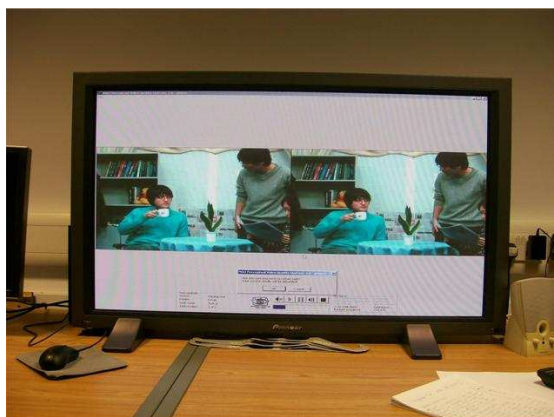


Fig. 9. A snapshot of the 32-inches screen used to compare the perceived subjective quality of the three versions of view synthesis.

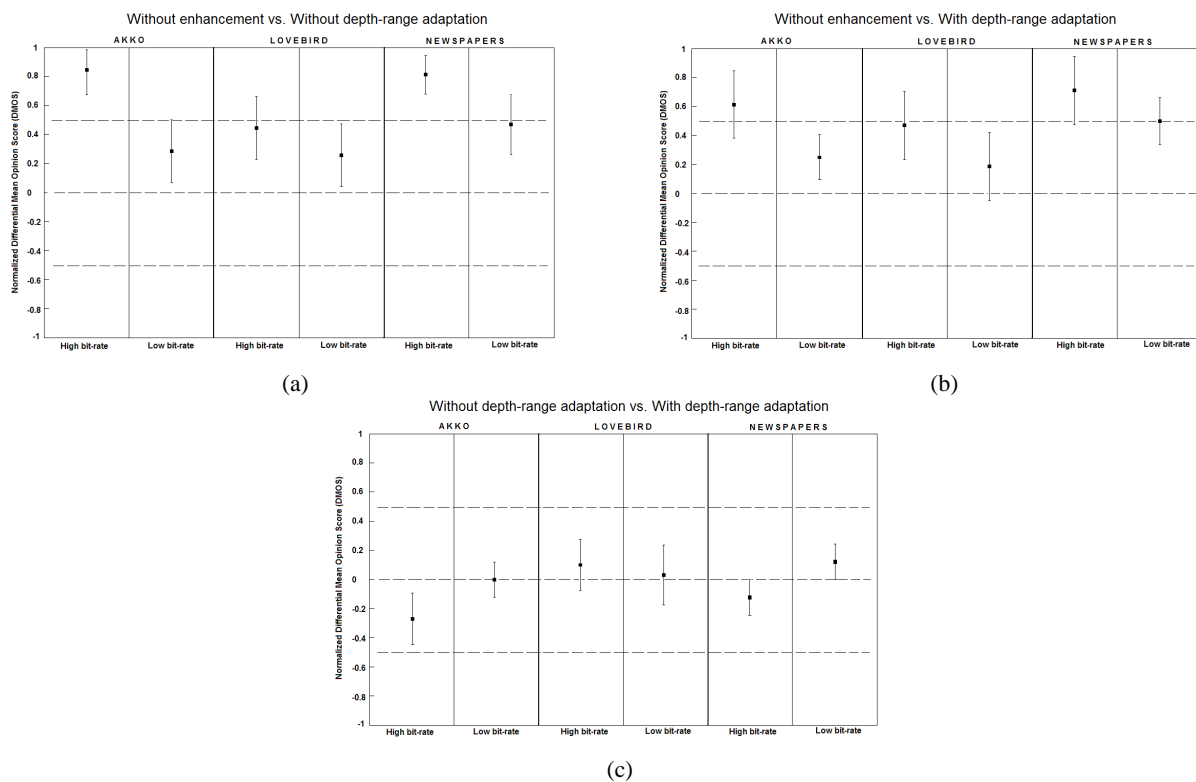


Fig. 10. The results of an adjectival categorical judgement method for assessment of the subjective quality of television pictures (according to the ITU-R recommendation BT.500-11 [36]). The three methods are compared pairwise by showing the two videos side-by-side to 20 subjects. The graphs show the normalized differential mean opinion scores with 95%-confidence intervals. The results demonstrate a large perceived difference between the videos synthesized using non-enhanced and enhanced depth maps.

## V. CONCLUSION

We propose a novel and computationally efficient enhancement method for the multi-view depth map sequences based on locally content adaptive median filtering. The filtering is adapted to edges, motion and depth-range to preserve the sharpness of the reconstructed (synthesized) views and to reduce the overall computational complexity. The enhancement scheme increases the consistency across the spatial, temporal and inter-view dimensions in the depth maps, which leads to an improved coding and rendering performance. The method allows for a quick and less accurate depth map estimation used in a common FVV scenario to achieve a comparable quality of the synthesized views at virtual viewpoints as more complex estimation algorithms. We present the improvements of this quality in terms of both the numerical PSNR factor and perceived visual quality as compared to the same obtained without the enhancement. Furthermore, we compare the elapsed time-per-frame in case of different enhancement methods and we show that the depth-range adaptation is capable of significantly reducing the computational complexity because of the joint processing of large background blocks in the depth maps, while preserving a similar quality.

## ACKNOWLEDGMENT

This work was in part developed within VISNET II, a European Network of Excellence, funded under the European Commission IST FP6 programme.

## REFERENCES

- [1] B. Wilburn, M. Smulski, K. Lee, and M. A. Horowitz, "The light field video camera," in *Proc. Media Processors SPIE Electron. Imag.*, San Jose, CA, Jan. 2002, pp. 29–36.
- [2] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computation Models of Visual Processing*, M. Landy and J. A. Movshon, Eds. Cambridge, MA: MIT Press, 1991, pp. 3–20.
- [3] M. Tanimoto and T. Fujii, "FTV: Achievements and challenges," *ISO/IEC JTCl/SC29/WG11 M11259*, Oct. 2004.
- [4] M. Tanimoto and M. Wildeboer, "Frameworks for FTV coding," in *Picture Coding Symposium (PCS2009)*, Chicago, IL, May 2009.
- [5] 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.
- [6] 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.
- [7] J.-W. Cho, M.-S. Kim, S. Valette, H.-Y. Jung, and R. Prost, "A 3-D mesh sequence coding using the combination of spatial and temporal wavelet analysis," in *Lecture Notes in Computer Science*. Berlin/Heidelberg, Germany: Springer, 2007, vol. 4418, pp. 389–399.
- [8] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. ACM SIGGRAPH*, New Orleans, LA, 1996, pp. 31–42.
- [9] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. ACM SIGGRAPH*, New Orleans, LA, 1996, pp. 43–54.
- [10] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*. Springer-Verlag, 2007.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] A. Vetro, P. Pandit, H. Kimata, and A. Smolic, "Joint draft 9.0 on multiview video coding," *Joint Video Team Document JVT-AB204*, 2008.

- [15] 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.
- [16] *Advanced video coding for generic audiovisual services*, ITU-T Recommendation ISO/IEC 14496-10, 2009.
- [17] 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.
- [18] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, P. H. N. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Processing: Image Communication*, vol. 24, pp. 73–88, 2009.
- [19] M. Maitre, Y. Shinagawa, and M. N. Do, "Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering," *IEEE Trans. Image Processing*, vol. 17, no. 6, pp. 946–957, June 2008.
- [20] W. Li, J. Zhou, B. Li, and M. I. Sezan, "Virtual view specification and synthesis for free viewpoint television," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 4, pp. 533–546, Apr. 2009.
- [21] X. Xiu and J. Liang, "Projective rectification-based view interpolation for multiview video coding and free viewpoint generation," in *Picture Coding Symposium (PCS2009)*, Chicago, IL, May 2009.
- [22] X. Guo, Y. Lu, F. Wu, D. Zhao, and W. Gao, "Wyner-Ziv-based multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 713–724, June 2008.
- [23] S. Yea and A. Vetro, "Multi-layered coding of depth for virtual view synthesis," in *Picture Coding Symposium (PCS2009)*, Chicago, IL, May 2009.
- [24] U. Fecker, M. Barkowsky, and A. Kaup, "Histogram-based prefiltering for luminance and chrominance compensation of multiview video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 9, pp. 1258–1267, Sept. 2008.
- [25] G. H. Park, M. W. Park, S.-C. Lim, W. S. Shim, and Y.-L. Lee, "Deblocking filtering for illumination compensation in multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 10, pp. 1457–1461, Oct. 2008.
- [26] E. Ekmekcioglu, V. Velisavljević, and S. T. Worrall, "Edge and motion-adaptive median filtering for multi-view depth map enhancement," in *Picture Coding Symposium (PCS2009)*, Chicago, IL, May 2009.
- [27] —, "Efficient edge, motion and depth-range adaptive processing for enhancement of multi-view depth map sequences," in *IEEE Int. Conf. on Image Proc.*, Cairo, Egypt, Nov. 2009.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] E. S. Larsen, P. Mordohai, M. Pollefeys, and H. Fuchs, "Temporally consistent reconstruction from multiple video streams using enhanced belief propagation," in *IEEE Int. Conf. on Computer Vision (ICCV)-2007*, Rio de Janeiro, Brasil, Oct. 2007.
- [32] M. Tanimoto, T. Fujii, and K. Suzuki, "Multi-view depth map of Rena and Akko & Kayo," *MPEG Doc. M14888*, Oct. 2007.
- [33] I. JTC1/SC29/WG11, "Description of exploration experiments in 3D video coding," *MPEG Doc. N9991*, July 2008.
- [34] S. Lee and Y. Ho, "Enhancement of temporal consistency for multi-view depth map estimation," *MPEG Doc. M15594*, July 2008.
- [35] M. Tanimoto, T. Fujii, and K. Suzuki, "View synthesis algorithm in view synthesis reference software 2.0 (VSRS2.0)," *MPEG Doc. M16090*, Feb. 2008.
- [36] *Methodology for the subjective assessment of the quality of television pictures*, ITU-R Recommendation BT.500-11, 2002.