A Free-Viewpoint Television System for Horizontal Virtual Navigation

Olgierd Stankiewicz, Marek Domański, Adrian Dziembowski, Adam Grzelka, Dawid Mieloch, Jarosław Samelak

Abstract—**Free-viewpoint television (FTV) and virtual navigation appear to be hot research topics. In this paper, the authors study the practical development of free-viewpoint television systems that provide the functionality of virtual horizontal navigation around real scenes. The considerations are focused on practical systems that use purely optical depth estimation and might be employed in the next few years. The architectures of such systems are discussed in detail, including acquisition, preprocessing, depth estimation, compression and presentation. In particular, the optimization of camera locations is discussed, and it is shown that video acquisition using camera pairs is advantageous for scenes with a substantial amount of occlusions. The theoretical considerations are supported by experimental results obtained for standard test multiview video sequences. Furthermore, the paper describes FTV video acquisition systems that consist of modules with pairs of cameras. The modules are sparsely located in arbitrary positions around a scene. Each camera module is equivalent to a video camera with a depth sensor. The hardware requirements, video processing algorithms and experimental results are reported. In particular, for such systems, a compression technique is discussed that is more efficient than the new 3D-HEVC technology. The paper also describes new test video sequences that are obtained from the camera pairs sparsely distributed around scenes.**

Index Terms—**free-viewpoint television, multiview video, view synthesis, virtual navigation**

I. Introduction

In this paper, we deal with the virtual navigation, i.e. a functionality of future interactive video services that provides a viewer the ability to move freely around a scene and watch it from virtual viewpoints on an arbitrary navigation trajectory. Video communication systems that provide such a functionality are often called free-viewpoint television (FTV) [1], and the respective video content is called free-viewpoint video (FVV) [2]. In this work, we consider such future FTV applications as, e.g. sports broadcasts (like judo, wrestling, sumo, dancing etc.), performances (theater, circus), interactive courses (medicine, cosmetics, dancing etc.), manuals, and school teaching materials. Free-viewpoint television may also be used to produce and deliver augmented-reality content.

In [3], [4], [5], [6], [7], [8], several results on FTV have already been reported. These papers also describe multiview video acquisition systems aimed at the production of test material for research, thus mostly using dense camera arrangements and huge numbers of cameras [3], [4], [5], [7]. They also describe the usage of specialized acquisition hardware [5], specialized processing hardware [4], or sophisticated display devices [3]. For FTV systems, except for football coverages, the general results using sparse distributions of cameras around a scene are still quite limited [7]. Moreover, these results are mostly obtained for very regular (linear or circular) distributions of camera locations, whereas the practical systems have to allow some degree of irregularity due to limitations of real events.

Our goal is to present new results in the design and practical implementations of FTV systems. Here, the aim is to study cost-effective and simple solutions that should lead to practical systems being available in the next very few years. Therefore, we are going to study the systems that are characterized by [16], [17]:

1) The usage of standard moderate-cost cameras;
2) Limited number of cameras – the cameras are sparsely located around a scene;
3) Some irregularity of camera locations due to obstacles in the room (e.g. pillars), people paths, escape ways etc. – the video processing algorithms do not exploit any pre-assumptions on regular patterns of camera locations;
4) Maximum usage of the off-shelf hardware – the specialized hardware is limited to relatively cheap boards produced by the authors – this hardware is used for synchronization signal distribution, system control, and video acquisition for cameras;
5) Limited operational costs – two persons suffice to operate the proposed system.

As regards the issue of efficient camera setups (Section III), the paper deals with the problem of the optimum camera placement. In Section III, we substantially extend the results of [11], [41], and demonstrate that pairing of camera locations results in quality gain for synthesized virtual views.

In particular, we study the system architecture (Section II), efficient sparse camera setups (Section III), a new FTV test video (Section IV), an example of an FTV system (Section V), compression methods (Sections VI and VIII) including

Manuscript received March 7, 2017; revised August 11, 2017. This work was supported by The National Centre for Research and Development, Poland under Project no. TANGO1/266710/NCBR/2015.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This includes two multimedia AVI format movie clips, which show virtual horizontal navigation around real scenes for the proposed test sequences. This material is 3.6 MB in size.

The authors are with the Chair of Multimedia Telecommunications and Microelectronics, Poznań University of Technology, 60965 Poznań, Poland (e-mail: marek.domanski@put.poznan.pl, {adziembowski, dmieloch, ostank}. @multimedia.edu.pl).

Digital Object Identifier

experimental results on appropriate extensions of the 3D-HEVC video compression technology [9] (Section VIII), the practical implementation of the representation server (Section VII), and the rendering server (Section IX).

For the sake of conciseness, we deal with virtual navigation on the horizontal plane only and we leave all audio issues beyond the scope of the paper.

II. FTV System Structure

Throughout this paper, we are going to use the generic architecture of FTV systems [10], [11] (see Fig. 1) that consist of the following functional blocks:

- a video acquisition system,
- a representation server that produces a visual representation of the spatial dynamic scene,
- rendering servers (also called as edge servers) that serve the requests for synthesis of video and audio at particular virtual locations around a scene,
- a user terminal, e.g. tablet, laptop, smartphone, etc.

The video acquisition system produces data necessary to compute the spatial representation of a scene, i.e. video and depth information obtained either from pure multiview video analysis or from depth sensors. The usage of depth sensors is conceptually very attractive (e.g. [12], [13]), but their practical employment still faces severe problems related to limited resolutions of the acquired depth maps, limited distance ranges, synchronization of video and depth cameras, additional infrared illumination of the scene, and sensitivity to environmental factors including solar illumination. In this paper we focus on the multiview recording of real events where additional infrared illumination might be unacceptable. Therefore, we assume that the depth information is obtained by video analysis only, and special depth sensors are not used.

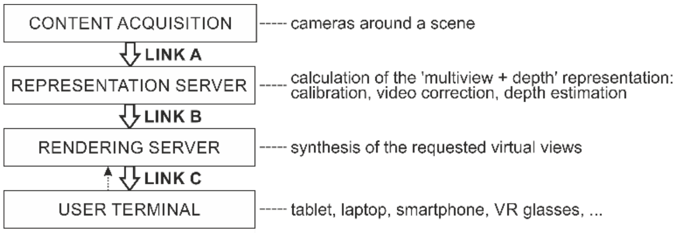
CONTENT ACQUISITION
----- cameras around a scene
LINK A
REPRESENTATION SERVER
----- calculation of the 'multiview + depth' representation: calibration, video correction, depth estimation
LINK B
RENDERING SERVER
----- synthesis of the requested virtual views
LINK C
USER TERMINAL
----- tablet, laptop, smartphone, VR glasses, ...

Fig. 1. A general FTV architecture (modified from [10]).

The video together with the system calibration data are transmitted via Link A. As the video data here is yet neither calibrated nor corrected, standard single-view compression techniques may be used (see Section VI). Link A belongs to the contribution environment, therefore high–fidelity compression is required. A standard approach would be to use intraframe techniques like M-JPEG 2000 [14] or HEVC All Intra [9]. Nevertheless, simple FTV systems will probably rarely use nonlinear editing as the FTV material does not need any choice of the camera or zooming during the production process, as that is done individually by the viewer. If the nonlinear edition is not needed, there is also no need for the random frame access and no need for small error accumulation in multiple encoding–decoding cycles. Therefore, more efficient interframe compression (AVC [15] or HEVC [9]) may be used. This way the requested bitrate may be significantly reduced, but the total bitrate will still be determined by simulcasting multiple video streams. In particular, especially in the initial phase of the FTV development, content hard-disk delivery to the representation server may be acceptable for video-on-demand services [16].

The tasks of the representation server include calibration, correction of the video (correction of lens aberrations, illumination compensation, equalization of the color characteristics of sensors, etc.) and depth estimation (e.g. [16], [17], [18], [19], [52]). The output is a model of the visual scene. The following scene representation types are mostly considered: ray-space [3], [5], object-based [20], [21], point-based [22], and multiview plus depth (MVD) [23], [52], [63], [65]. As the first three types of models are related to quite complex calculations, the MVD representation is used most often and its compression has already been standardized both for AVC [15] and HEVC [9]. Currently, further standardization of MVD compression is also considered [10], [24]. Therefore, the MVD representation is also considered in this paper.

The compressed MVD representation together with the camera parameters and the audio data are transmitted via Link B (Fig. 1). If the representation server and the rendering server are in distant locations a video compression is needed. For the MVD representation, the technology is available and standardized as 3D extensions of the AVC [15] and HEVC [9], [25] standards. Unfortunately, these 3D extensions have been designed and tested for cameras with parallel optical axes, densely located on a line. For cameras sparsely located around a scene, they exhibit compression performance only slightly higher than individual coding (i.e. simulcast coding) of the views and depth maps [17], [26]. For such content, a more efficient MVD compression method is considered in Section VII.

The sink of Link B is in the rendering server as we opt for the centralized model [17], [27] of view synthesis. In this model, the views requested by viewers are synthesized in the servers of the service provider, i.e. in the rendering servers. The number of rendering servers depends on the number of user terminals, as each such server may serve a limited number of user terminals.

Another option would be a distributed model [4], [28], [29], [62] where virtual views are synthesized in each user terminal. Such model requires high transmission bandwidth in order to transmit the MVD representation directly to the user terminals. This model also requires significant processing power in the user terminals. As we are going to avoid problems related to sophisticated video streaming (see e.g. [28], [29]) we opt for the centralized model, following also the conclusions from paper [17]. For more details, please refer to Section VIII.

In the centralized model, the user terminal sends requests for current virtual positions, and the rendering server responds with video frames synthesized for the requested position. The free navigation service will be available as a video-on-demand

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2790162, IEEE Transactions on Multimedia

service on the Internet, as foreseen for the nearest future. The proxy rendering server streams video to the user terminals (Link C in Fig. 1). A user terminal may be as simple as a smartphone or a tablet equipped with any standard video decoder (AVC or HEVC). Requests for a virtual walk around or inside/outside the scene are defined by sliding the touchscreen horizontally or vertically, respectively. In a user terminal, head-mounted devices like VR glasses or VR helmets might also be used allowing a user to control the viewpoint and view direction by head movements. Unfortunately, the practical application of such gadgets is limited due to very rigorous latency restrictions (see Section VIII).

This paper describes a practical and simple FTV system. Other descriptions are either less complete [16], [17] or aim at much more sophisticated systems [3], [5]. Further in this paper, we are going to describe new and original results concerning selected parts of the system.

III. Multiview Video Acquisition for FTV

In a practical FTV system, video is captured by multiple cameras located around a scene. Because of the requirements of low cost and simplicity, the number of cameras should be limited, thus increasing the distances between cameras and influencing the depth estimation. The depth of a point can be estimated if the point is visible by at least two cameras. When the distances between cameras increase, in the individual views, fewer pixels are captured by at least two cameras. For the remaining pixels, called occluded, the depth cannot be undoubtedly estimated but only interpolated or extrapolated. Moreover, even the pixels visible in multiple views are acquired differently by distant cameras due to different lighting conditions and reflections. The occlusions and illumination differences cause difficulties in matching the views, thus significantly deteriorating the estimated depth maps and, in consequence, the synthesized virtual views. In the virtual views, they may cause strange effects like losses of some parts of individual objects, appearance of artificial holes in the objects, flickering of video etc.

Having in mind the two abovementioned negative mechanisms related to sparse camera locations, one may ask if specific placements of cameras may reduce the total influence of these effects on the depth maps and the quality of synthesized virtual views. This problem of efficient camera setups was already considered in the context of computer graphics and object tracking [30], [31], [32], [33], [70]. In particular, nonuniform camera setups have been considered for CAVE and motion capture systems [30], object tracking [31], and representation of simple objects and minimization of occlusions [32], [70]. Unfortunately, the techniques proposed in the abovementioned references for estimation of camera locations need more input information, e.g. about the geometry of objects in a scene, than is available for FTV systems where we are usually unable to predict motions and shapes of many objects that occlude each other. Therefore, we propose to use another approach, being an extension of that from [11] and [41].

In order to reduce the two abovementioned negative effects caused by sparse camera locations, we propose to group the cameras into stereo pairs instead of distributing them uniformly around a scene [11]. In this approach, cameras from the same camera pair acquire a scene from very similar viewpoints. A short base of a camera pair ensures that very few parts of the scene are occluded, i.e. captured by only one camera, or even not visible by any camera. Moreover, the lighting conditions in both views are similar. On the other hand, a short base of a camera pair results in low accuracy of the depth estimated using the two cameras. Very accurate depth may be obtained using long bases created by cameras from different camera pairs. For long bases, many parts of the scene are occluded. For most of the occluded scene parts, at least rough depth estimation is possible using two cameras from the same pair as discussed above.

The two abovementioned contradictory phenomena influence the depth estimation and thus the quality of synthesized virtual views. In order to synthesize virtual views with the highest quality the trade-off between these phenomena has to be found quantitatively. Although it is well-known that *PSNR* of synthesized views is far from perfect as a quality measure [34], we use it because of its simplicity (see e.g. [74], [75] for a similar approach). Therefore, we measure the difference $\Delta_{p-u}PSNR$ between *PSNR* values of the virtual view for paired cameras (denoted with subscript p) and uniformly (e.g. equiangularly) distributed cameras (denoted with subscript u), expressed as a sum of two components related to these two phenomena:

$$\Delta_{p-u}PSNR = \Delta_{p-u}PSNR_b + \Delta_{p-u}PSNR_o \ , \qquad (1)$$

where $\Delta_{p-u}PSNR_b$ is the gain resulting from adjustments of the bases in the system and $\Delta_{p-u}PSNR_o$ is the gain resulting from changes of the amount of occlusions, both expressed as the difference between *PSNR* values for the paired and uniform arrangements of the cameras.

The two components in (1) corresponding to base and occlusion reduction related to camera pairing are considered in the following Sections III-A and III-B.

A. The Influence of the Base on the Virtual View Quality

In this section, we analyze the first factor that influences depth estimation in our proposal. In particular we show that the proposed pairing of the cameras, by changing the base of the cameras, decreases the accuracy of the estimated depth.

First, let us consider the depth estimation (e.g. [3], [17], [35]) for only one camera pair. The focal length of both cameras is f, the base distance is b. The depth of a point object is z and the disparity of the object images is d. Assuming $f \ll z$ we get [36]:

$$z = \frac{f \cdot b}{d} \ . \qquad (2)$$

Let us assume two objects with the depths $z_1$and z_2, respectively. Their positions may be distinguished if the respective disparity difference $|d_1 - d_2|$ exceeds a minimum value Δd:

$$|d_1 - d_2| \geq \Delta d \ . \qquad (3)$$

Δd is the disparity accuracy, i.e. 2 to 3 distances between the centers of the pixels in the sensors. From (2) we get $d_1 = \frac{f \cdot b}{z_1}$, $d_2 = \frac{fb}{z_2}$, and we can denote average depth as $z = \sqrt{z_1 \cdot z_2}$. Therefore, depth values z_1 and z_2 may be distinguished when:

$$|z_1 - z_2| \geq \frac{z^2}{f \cdot b} \Delta d \ . \qquad (4)$$

For example, let us consider a 1/1.2" HD sensor with 1920 pixels per line (like the Sony IMX 174 CMOS sensor from the Basler acA 1920-155uc camera [37]) and disparity accuracy $\Delta d \approx 15$ µm, focal length $f = 16$ mm. For an average depth $z = 25$ m, for base $b = 40$ cm we have the depth resolution of $|z_1 - z_2| \geq 1.5$ m whereas for $b = 4$ m we have $|z_1 - z_2| \geq 0.15$ m. For an average depth $z = 10$ m, these numbers are 0.23 m and 0.023 m, respectively. Please note that abovementioned examples are compliant with the video acquisition project for a sports hall as considered in Section V.

The abovementioned reasoning explains the well-known fact that the depth map can be estimated with a high accuracy for a long base of a camera pair. Therefore, for the sake of the spatial accuracy, the depth estimation should be performed from a camera pair with the longest base. For multiple cameras, the above considerations imply that the depth estimation should be performed with the use of the longest available base, which is between two furthest cameras in the system.

In complex scenes, individual points of a scene are acquired by different sets of cameras. Each camera set exhibits its longest base that corresponds to the two outer cameras of this set. For a uniform (e.g. equiangular) camera arrangement $\bar{b}_u$ denotes the longest base averaged over all points visible in a scene. Similarly, $\bar{b}_p$ is the average for the camera arrangement, where camera pairs are uniformly distributed around the scene. In Appendix I we show how to determine $\bar{b}_u$ and $\bar{b}_p$ for a simplified model of the scene from Section III-C.

As it was mentioned, the shorter the base of the system, the lower is the accuracy of the estimated depth. Depth estimation errors cause horizontal displacements of the objects in a virtual view. For highly textured regions of a scene, these displacements significantly deteriorate the quality of a virtual view, whereas for smooth regions the loss of quality is often negligible. In order to estimate the abovementioned effects quantitatively, we roughly model the average quality of the synthesized views. At first we define a similarity metric $S(n)$ that measures the similarity between an ideal virtual view and that view shifted by n sampling periods from its correct position, i.e. from the position calculated using the ideal depth maps:

$$S(n) = 1 - \frac{1}{W_{img} \cdot H_{img}} \sum_{j=1}^{H_{img}} \sum_{i=1}^{W_{img}-n} \frac{[Y(i,j) - Y(i+\mathrm{n},j)]^2}{255^2} , \qquad (5)$$

where $Y(i,j)$ is the luma of a point, and 255^2 is the maximum possible square of error of the 8-bit sample values. W_{img} and H_{img} are width and height of the image $Y(i,j)$, respectively. The proposed similarity $S(n)$ is defined as a value from 0 to 1, where the unit value means that the view synthesized using the ideal depth maps is the same as that synthesized using the estimated depth maps. Eq. (5) implies that for a synthesized view distorted by a shift by n sampling periods, the luma $PSNR$ can be estimated as:

$$PSNR(n) = -10 \log(1 - S(n)) , \qquad (6)$$

where the reference for $PSNR$ calculation is the view synthesized from the ideal depth maps. With the use of a set of multiview test sequences (see Table I in Section III-C), we have measured $S(n)$ for integer values of shift n. As shown in Fig. 2, the measured $S(n)$ starts at $S(1) = S_1 = 0.995$ and decreases slowly towards 0. In further considerations, we employ an approximate analytical model of $S(n)$ defined for real values of shift $n > 0$:

$$S(n) = {S_1}^n . \qquad (7)$$

This analytical model of $S(n)$ is depicted in Fig. 2 together with $S(n)$ values measured for the set of multiview video test sequences.

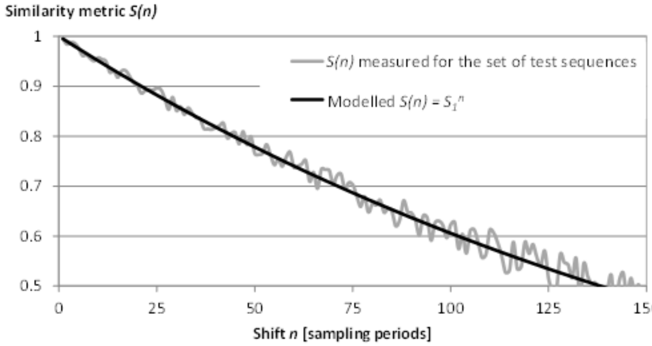

Fig. 2. The model of similarity $S(n)$ between the samples that are horizontally shifted by n sampling periods..

For a uniform camera arrangement, we denote the mean base as $\bar{b}_u$ and the maximum erroneous shift of the virtual point position in a virtual view as Δp_u. The position of a point can only be expressed by integers, therefore assuming that the only source of the errors by estimation of a point position is the rounding, we can assume $\Delta p_u = 0.5$. As follows from (4), for camera pairs, where the mean base $\bar{b}_p$ is shorter, the accuracy of the depth estimation of the object at some distance z decreases. For camera pairs, the accuracy of a point position in a virtual view is $\Delta p_p = \Delta p_u \, \bar{b}_u / \bar{b}_p$. Therefore, the quality gain for a virtual view resulting from a different base of cameras in the paired and the uniform arrangements, with the use of (6) is:

$$\Delta_{p-u} PSNR_b = 10 \log \frac{1 - {S_1}^{\Delta pu}}{1 - {S_1}^{\Delta pp}} \ . \qquad (8)$$

In Section III-C we present results for $\Delta_{p-u} PSNR_b$ attained for a simplified theoretical model.

B. The Influence of Occlusions on the Virtual View Quality

In this section, we consider the second component of the virtual view quality gain $\Delta_{p-u}PSNR$ defined in (1). $\Delta_{p-u}PSNR_O$ is the quality gain resulting from different amount of occlusions in the paired and the uniform arrangements of the cameras. In order to assess this difference, let us first analyze the impact of occlusions on virtual view synthesis process.

A typical view synthesis technique [39] based on DIBR (Depth Image Based Rendering) [40] creates a virtual view in two steps. First, image regions from the input views are rendered to new positions in the virtual view and blended together. At this stage, some regions of the virtual view are unknown, because were occluded in all input views. Such regions are inpainted in the second step. Therefore, the final output virtual image is composed of two kinds of regions: synthesized and inpainted. The quality in the inpainted regions is usually worse, because the inpainting is based on the neighboring synthesized regions, and thus inpainting errors are added to errors of synthesis. As we can see, the ratio between the areas of these regions is related to the amount of occlusions in the scene. Therefore, we can estimate the change in virtual view quality with respect to the amount of occlusions:

$$\Delta_{p-u}PSNR_O = 10\log\frac{OCC_u e_s^2+(1-OCC_u)e_i^2}{OCC_p e_s^2+(1-OCC_p)e_i^2}\ , \qquad (9)$$

where e_s^2 and e_i^2 are mean square errors in the synthesized and inpainted regions, respectively, and OCC_u and OCC_p are the percentages of occluded areas for the uniform camera arrangement and for the camera pairs, respectively. An occluded area is an area of the scene, where depth could not be determined, i.e. fragments of the scene seen by fewer than two cameras.

In Section III-C, we consider a simplified theoretical model, for which we present results attained for $\Delta_{p-u}PSNR_O$. The exact derivation of expressions defined in this section is provided in Appendix I, as it is irrelevant to the general understanding of the paper.

C. Simple Model of the System

Here, we consider a simple theoretical model of a multicamera system, which is used to derive the quality gain $\Delta_{p-u}PSNR$ due to grouping the cameras into pairs (1).

As we consider the uniform arrangement of either single cameras or camera pairs, it is enough to consider only 4 neighboring cameras (Fig. 3). Therefore, we locate them on a line for the sake of simplicity. The cameras are placed at the locations x_0 to x_3 in x direction and at $z = 0$. All cameras have the same FOV (field of view). The scene is modeled with a single foreground object that occludes the background. The foreground object has width w_O and its center is at (x_O, z_O). The background has infinite size in the x dimension and is placed at distance z_B from the cameras.We use this model to estimate the $PSNR$ gain from the camera pairing $\Delta_{p-u}PSNR$.

For the calculations of the gain $\Delta_{p-u}PSNR$ from (1), (8) and (9), we assume normalized values $x_0 = 0$, $x_3 = 3$.

For the paired camera arrangement we use $x_1 = 0.4$, $x_3 = 2.6$. The cameras have FOV = 70 degrees and a FullHD sensor and $\Delta p_u = 0.5$. The background is at $z_B = 6$. In order to model various occlusion levels, w_O varies from 0.2 to 2.8, and z_O from 0.2 to 5.8. These parameters reasonably model the real scenes used in the experiments and test video shooting, for the unit of about 2 to 3 m.

For the abovementioned parameters, Fig. 4 presents the gains $\Delta_{p-u}PSNR$, $\Delta_{p-u}PSNR_b$, and $\Delta_{p-u}PSNR_O$ for various occlusion levels.

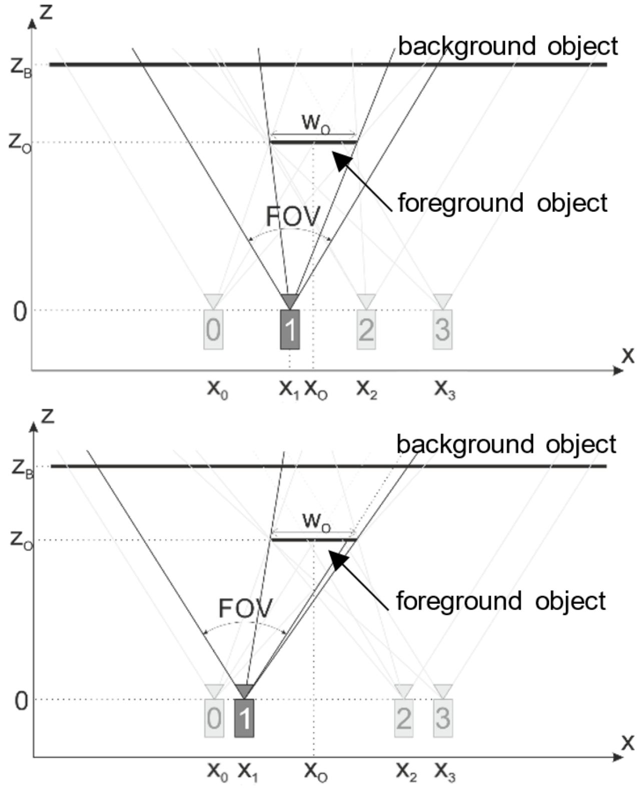

Fig. 3. A simple model of a multicamera system for uniform arrangement of cameras (top) and for camera pairs (bottom).

The results show that for assumed model of a scene usage of camera pairs instead of uniformly arranged cameras is beneficial when percentage of occluded area is relatively high (greater than 25%).

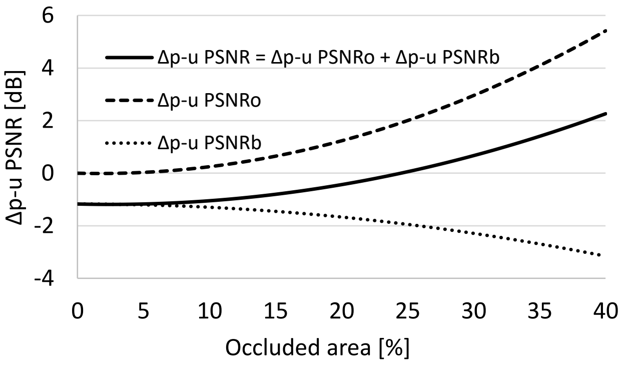

Fig. 4. Theoretical curve for camera pairing gain $\Delta_{p-u}PSNR$ as a function of the occluded area.

D. Experimental Results

The goal of the experiment (initially proposed in [41]) is to verify the proposed theoretical dependency between the virtual view quality and the camera arrangement. We assume the abovementioned arrangement of 4 cameras with fixed positions of the outer ones and variable positions of the 2 inner cameras. Therefore, the normalized base of each camera pair varies from 1 (uniform arrangement of all cameras) to 0 (collocated cameras in each pair). We performed the experiment on a set of 11 multiview MPEG test sequences obtained from at least 10 cameras located either on a line or on an arc.

In the experiment, the virtual views are synthesized using depth maps estimated with the use of different camera arrangements. The virtual video quality was estimated as luma $PSNR$ between the virtual and collocated reference views, i.e. the real view is was used as the ground-truth for view synthesis. For each sequence, the average percentage of occluded areas OCC_u for the uniform camera arrangement was calculated (Table I). The sequences are classified as those with insignificant occlusions (OCC_u < 25%) and those with significant occlusions (OCC_u > 25%).

For the sequences with insignificant occlusions, the quality gain for all non-uniform arrangements is negative (Fig. 5), and the best camera distribution is the uniform one. For the sequences with significant occlusions, camera pairing allows to obtain better quality of the depth maps, and therefore better quality of the virtual views.

For camera pairs, the highest gain was achieved for the base within a camera pair equal to 0.4 (Fig. 5). In the arrangements characterized by shorter base, the depth accuracy was too low, whereas the views captured using the systems with longer bases have too many occlusions.

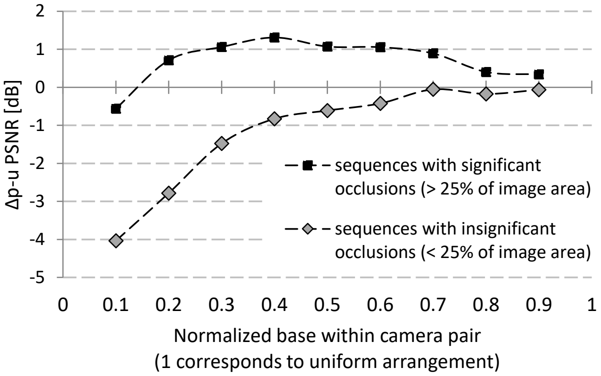

Fig. 5. Average quality gain over uniform camera arrangement for variable bases of the camera pairs.

In Fig. 6, for individual multiview tests sequences, the experimentally estimated pairing gains $\Delta_{p-u}PSNR$ are shown together with the theoretical curve, thus confirming the presented theoretical considerations.

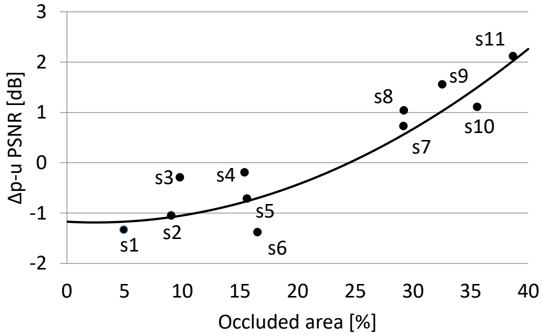

Fig. 6. The relation between camera pairing gain $\Delta_{p-u}PSNR$ and percentage of occluded area. The dots s_i correspond to the video test sequences mentioned in Table I.

In Fig. 6, the theoretical curve is estimated for a simple scene with its parameters roughly corresponding to the available test sequences. These sequences are representative to the applications of free-viewpoint television that are considerable for the near future. Therefore, we conclude that the curve depicted in Fig. 6 is a very rough estimate of the gains due to camera pairing in simple free-viewpoint television systems.

TABLE I
PERCENTAGE OF OCCLUDED AREAS AND CAMERA PAIRING GAIN IN USED TEST SEQUENCES

ID	Sequence name	OCC_u [%]	$\Delta_{p-u}PSNR$ [dB]
s1	BBB Rabbit Arc [43]	4.93	-1.33
s2	BBB Butterfly Arc [43]	9.05	-1.05
s3	Dog [66]	9.81	-0.29
s4	BBB Rabbit Linear [43]	15.41	-0.19
s5	Pantomime [66]	15.61	-0.71
s6	BBB Butterfly Linear [43]	16.53	-1.38
s7	BBB Flowers Linear [43]	29.18	0.73
s8	San Miguel [67]	29.21	1.04
s9	Champagne [66]	32.55	1.56
s10	Bee [64]	35.57	1.11
s11	BBB Flowers Arc [43]	38.68	2.12

IV. NEW MULTIVIEW-VIDEO TEST SEQUENCES FROM CAMERA PAIRS

Hitherto, most of the multiview video test material is available for uniformly spaced cameras, and only few sequences are available for cameras located on an arc [42], [43], [44]. Therefore, we have produced new multiview test sequences (Figs. 7-9) acquired using camera pairs located on an arc. In each pair, the cameras were aligned in parallel with base of 22 cm. There were 5 camera pairs placed over 60° arc (thus the angle between the optical axes of neighboring pairs is 15°). The radius of the arc was 3 m for Poznan Blocks2 and 3.5 m for the other sequences. The video data (10 views in total for each sequence) were captured in raw YUV format (4:2:0 chroma subsampling) with the resolution of 1920 × 1080, 25 frames per second. The length of each sequence is 20 seconds. To the best of our knowledge, these are the first such sequences offered to the research community under the condition of citing this paper (for access please contact the authors).

Fig. 7. Two views from the sequence Poznan Fencing2.

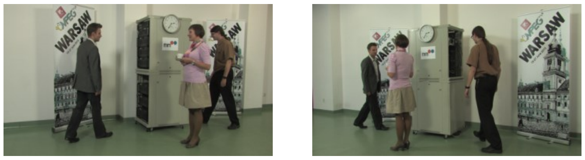

Fig. 8. Two views from the sequence Poznan Service2.

Fig. 9. Two views from the sequence Poznan Blocks2.

V. Design of an FTV Acquisition System

One of the most expected applications of free-viewpoint television systems are sports events coverages. Thus, let us consider a designing process for a system in a sports hall.

In dynamic sports (e.g. in basketball) many players can often be placed in a relatively small area, thus in the region of viewer's interest many occluded areas can be expected. According to the considerations described in Section III, for such scenes cameras should be arranged in pairs.

Simultaneously, for all points of the scene, the estimated depth should have the highest possible spatial resolution, ensured by large bases of cameras. Therefore, we assume that all points of a scene should be visible by at least three cameras in order to estimate the depth using information from the cameras within a pair (to reduce the influence of occlusions) and at least one camera from another pair (to increase depth accuracy). Fig. 10 presents such a camera arrangement – any point of the court seen by cameras from pair i is also seen by at least one camera from pair $i-1$ or $i+1$. It is true if point P of intersection of the right camera field of view from pair $i-1$ and left camera from pair $i+1$, is placed on the sideline or nearer.

The maximum distance between neighboring camera pairs depends on three factors: distance between the sideline and cameras, FOV of the cameras and base within a camera pair. For further considerations we assume size 52 × 34 m and arrangement of the real academic sports hall in Poznań. In the middle, there is a typical basketball court (28 × 15 m), so the distance between the court's sideline and the walls is 9.5 meter. The audience stands are placed by the sidewalls. In order to avoid the spectators occluding the court, cameras on the wall should be placed at 4.5 m above the floor, which gives 10.5 m between cameras and the sideline.

FOV of the cameras was chosen as 44° (angular degrees). Assuming a 16:9 sensor, vertical FOV is 25°, which covers the whole court and the players.

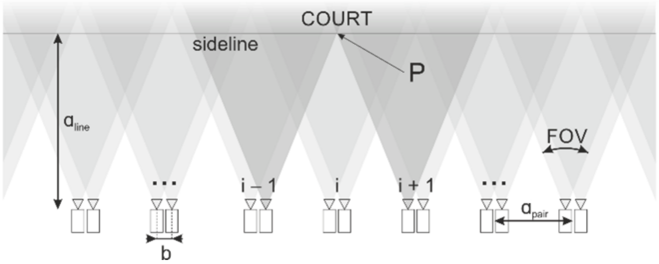

Fig. 10. The camera arrangement proposed for a sports hall.

The maximum distance between neighboring pairs can be estimated as:

$$a_{pair} \leq \frac{a_{line}}{\cot\left(\frac{FOV}{2}\right)} + \frac{b}{2} \ , \tag{10}$$

where a_{line} is the distance between cameras and the sideline, FOV is the horizontal field of view of each camera and b is the base within each camera pair. Assuming a uniform arrangement of camera pairs around the whole hall, the number of camera pairs can now be roughly estimated:

$$N_{pair} = \frac{2 \cdot (W_{hall} + L_{hall})}{a_{pair}} \ , \tag{11}$$

where W_{hall} and L_{hall} are width and length of the hall, respectively.

In order to estimate the base of each camera pair, the assumed depth accuracy has to be set. In general, the depth of the objects placed farther from the cameras may be estimated with lower accuracy. We assume that in the background the accuracy of the depth should be 0.5 m (two objects should be distinguishable if one of them is 0.5 m closer to the cameras than the second one). It implies, that the disparity between two views within one camera pair should differ by one sampling period Ts for objects placed at W_{hall} and $W_{hall} - 0.5$ m, so after the normalization of disparity d by the sampling period:

$$\frac{d}{d+Ts} \geq \frac{W_{hall} - 0.5}{W_{hall}} \ . \tag{12}$$

From (12), the minimum disparity on the camera sensor is $d \geq 67$ sampling periods. The minimum base of each camera pair:

$$b = d \cdot \frac{W_{hall}}{\cot\left(\frac{FOV}{2}\right) \cdot W_{cam}} \ , \tag{13}$$

where W_{cam} is the width of each camera sensor. Assuming HD cameras with W_{cam} equal to 1920 sampling periods, b is 94 cm. After adding some margin, the requested base of each camera pair is equal to 1 m.

Using the estimated base, we can calculate that the distance between two pairs a_{pair} is 4.75 m, thus the approximate number of camera pairs required for the entire hall is 36.

Another issue that should be considered is the synchronization of cameras. The synchronization error between cameras should be negligible as compared to the shutter time. Moreover, the cameras should be well synchronized with audio sampling, which is very prone to errors in the acquisition of spatial audio signals.

VI. Compression for Link A of FTV Systems

In the free-viewpoint television system described in Section V, there are 72 Full-HD cameras in a sports hall. Assuming that each camera captures video in 25 fps and the chroma subsampling is 4:2:0, more than 42 Gbps of throughput would be needed to transmit all uncompressed video. It is possible to build an infrastructure gathering such amount of data, but the cost of such a system would be relatively high.

The compression in link A significantly reduces the bitrate of video streams, although it influences the quality of depth map estimation, and thus the quality of virtual view synthesis. The experiment that verifies if the multiview sequence could be initially compressed with no significant loss of the quality of estimated depth is proposed [45].

For the purpose of experiments, 7 multiview sequences with more than 30 available views were used (see Table III in Appendix II). Four chosen views were encoded and decoded independently (using AVC and HEVC encoders) and further used for the depth estimation. Estimated depth maps were used to synthesize virtual views in the positions of the remaining 27 real views. The quality of virtual views was measured as luma *PSNR* between them and the corresponding real view. For reference, the process was repeated for uncompressed views.

The publicly available optimized encoders were used in the experiment: for AVC the x264 [46] and for HEVC the x265 [47]. Both encoders have been configured in the "fast" operation mode in order to simulate real-world low-power embedded encoders. For the "random access mode," the GOP size was 13 and frame arrangement was: I BB P BB P BB P BB P.

Fig. 11 depicts the results averaged over all sequences. In order to reduce the latency of compression, the all-intra mode can be used. The results for all-intra compression are shown in Fig. 12. The detailed results for Figs. 11 and 12 are provided in Tables IV and V in Appendix II.

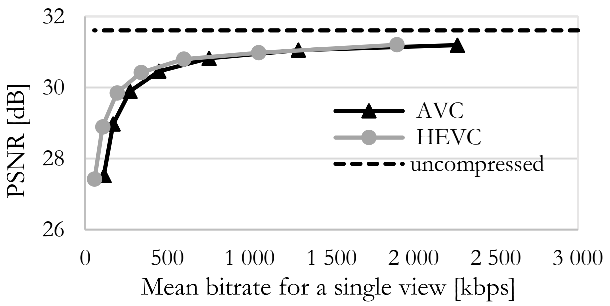

Fig. 11. Mean luma *PSNR* for virtual views synthesized from the compressed real views.

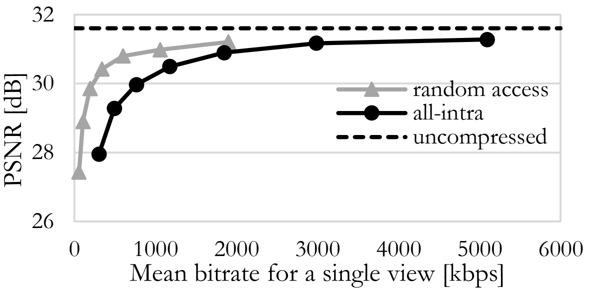

Fig. 12. Mean luma *PSNR* for virtual views synthesized from the HEVC-compressed real views estimated for two standard configurations: all-intra and random access [55].

VII. Representation Server

The representation server processes data captured by all cameras and estimates the MVD representation of a scene, i.e. real views with corresponding depth maps.

In the representation server, three main operations are performed: calibration of the system, correction of input views and estimation of depth maps. All these operations (especially depth estimation) are very time-consuming, so in a simple, low-cost FTV system the representation server operates off-line – the viewer cannot watch livestreams but only previously recorded events.

The calibration of the system comprises the estimation of intrinsic and extrinsic camera parameters. The input data for calibration are collected before or/and after actual video acquisition, or additionally even during pauses of the covered event. The experience of the authors proves that the calibration device may very simple – e.g. just one light spot (e.g. one LED) that is in motion through the scene. Such a calibration device allows cameras to be located in any positions around the scene.

The intrinsic parameters are estimated for each camera independently, thus well-known methods of calibration are used [71]. In order to estimate the extrinsic camera parameters, a technique adapted to cope with arbitrary camera locations was developed by the authors [17].

The depth estimation process is crucial for the high quality of experience of the user of FTV system. In a simple, practical system with arbitrarily located cameras, typical depth estimation algorithms mostly cannot be used, because of their limitations (required number of cameras [72], specific camera arrangements [73], etc.). Therefore, the authors proposed a new technique that can be used for any number of arbitrarily positioned cameras [69].

VIII. 3D HEVC Extension for Link B of FTV Systems

As it was mentioned in Section II, usually, the rendering server is distant from the representation server. Therefore, the MVD representation used in in Link B (Fig. 1) should be compressed. Unfortunately, standard 3D extensions of AVC and HEVC are optimized for a linear, dense arrangement of cameras and are not efficient for cameras sparsely distributed around a scene. A more efficient extension has already been proposed in [26]. In this paper we use the approach from [26] to develop a more efficient MVD codec suitable for arbitrarily located cameras. In order to attain this, several modifications have been introduced to low-level coding tools. The developed codec exploits the derivation of disparity vectors with nonzero vertical components. This also implies modifications of the following tools: Disparity Compensated Prediction, Neighboring Block Disparity Vector (NBDV), Depth-oriented NBDV, View Synthesis Prediction, Inter-view Motion Prediction, Illumination Compensation. These modifications are similar to those in [26], but they are embedded into another implementation.

Unfortunately, for compression efficiency quite few results are available for higher numbers of cameras sparsely located on an arc [48]. We examine three available techniques (MV-HEVC, 3D-HEVC, [9], [25] and our implementation based on [26]) in the conditions that we expect in Link B.

For the experiments, we have used the HTM 13.0 software [50] for 3D-HEVC and MV-HEVC, which are references for our technique, and our implementation built on top of HTM 13.0. The coding experiments have been performed for sequences obtained from two camera arrangements: with and without camera pairing. The experiments without camera pairing have been carried out for 7 views with corresponding depth maps for the following test sequences: *Poznan Blocks* (all views except the utmost left and right) [51], *Big Buck Bunny Flowers* (views 6, 19, 32, 45, 58, 71, 84) [43], *Ballet* and *Breakdancers* (all views) [53]. The experiments with cameras arranged in pairs have been performed for 5 pairs of views with corresponding depth maps for the following test sequences: *Poznan Fencing2, Poznan Blocks2 and Poznan Service2* [54]. The configuration for all codecs is similar as in [55], i.e. Main Profile, GOP size = 8, intra period = 24, hierarchical GOPs on, 4 reference frames, Neighboring Block Disparity Vector turned on, Depth oriented NBDV turned on, View Synthesis Prediction turned on, Inter-view Motion Prediction turned on, Illumination Compensation on but View Synthesis Optimization for Depth Coding switched off. The comparison of compression performance is made using *PSNR* for luma (Table II). The detailed raw results are provided in Table III in the Appendix II.

TABLE II
AVERAGE LUMA BITRATE REDUCTIONS CALCULATED ACCORDING TO THE BJØNTEGAARD FORMULA [49].

	Ours vs. 3D-HEVC	Ours vs. MV-HEVC	3D-HEVC vs. MV-HEVC
Poznan Blocks	-6.44%	-4.20%	2.37%
BBB_Flowers	-3.03%	-2.80%	0.21%
Ballet	-8.64%	-12.55%	-4.32%
Breakdancers	-9.79%	-13.71%	-4.39%
Avg. without pairs	**-6.97%**	**-8.32%**	**-1.53%**
Poznan Fencing2	-4.30%	-1.29%	3.11%
Poznan Blocks2	-5.90%	-4.62%	1.35%
Poznan Service2	-5.29%	-4.31%	1.02%
Avg. with pairs	**-5.16%**	**-3.41%**	**1.83%**

As mentioned in Section VI, the amount of data produced by the proposed video acquisition system would exceed 42 Gbps. After depth estimation, the bitrate increases by a factor 1.5. Using state-of-the-art MVD HEVC-based compression techniques, the bitrate can be reduced to roughly 300 Mbps, retaining high quality of video. The results obtained by the authors demonstrate that for the sequences obtained from camera pairs, 3D-HEVC may be less efficient than the simpler MV-HEVC. This is an astonishing result as usually the compression efficiency for 3D-HEVC is slightly higher (by less than 2%) than that for MV-HEVC. This issue needs more extensive study when more test sequences produced by camera pairs are available.

For the sequences with circular camera arrangements, the technique proposed by the authors results in average bitrate reduction of 6% (similar like in [26]) versus the state-of-the-art 3D-HEVC. This average bitrate reduction is similar for the test video sequences obtained with and without camera pairs. Therefore, this result encourages further research on the MVD compression for the FTV.

IX. RENDERING SERVER

The task of the rendering server is to respond to the requests from a user and to stream video for the requested viewpoint. Therefore, the video frames need to be synthesized according to the current viewpoint defined by a user. Unlike some other works [27], currently we aim at internet delivery only, because the terrestrial and satellite broadcasting are too expensive for a small number of initial users.

For MVD content obtained from cameras located on a straight line, real-time implementations of view-synthesis are known for graphical processing units (GPUs) [56], [57]. For camera located around a scene, the synthesis is significantly more complex [58], [59], [68] but still feasible on a GPU in real time. Thus, we designed the video processing architecture as a set of GPUs, each serving some users at a time. The remaining parts of the required functionality: connection request service, position calculations and connection and processing control are implemented in the software. All of them form a virtual processing block (Fig. 13) that is lent to a user for the time of a viewing session. One rendering server with full MVD representation can provide service for many users independently. The number of user terminals which can be supported depends on rendering algorithm complexity and computational power of the server.

The indicative latency budget is set to 350 ms including 150 ms given to position calculation, view synthesis, video coding and buffering, 100 ms for video decoding and buffering, and 100 ms for the round-trip packet travel time including operational system response times. These latency limits are demanding but realistic for the contemporary video technology [60]. In order to test the efficiency of the proposed rendering server, the authors prepared its straight-forward CPU implementation. For Intel Core i7-4770 and Full-HD multiview sequences the latency introduced by the rendering server was 100 to 130 ms and varied because of different complexity of scenes (e.g. the area which has to be inpainted). The overall latency (i.e. time between user request for a new view and decoding of this view on user terminal) is highly dependent on the characteristics of the network used. For a local wired network the overall latency was less than 150 ms.

The results of Section VI demonstrate that even for compression typical for broadcasting, compression errors have a very limited impact on the final quality of the virtual video.

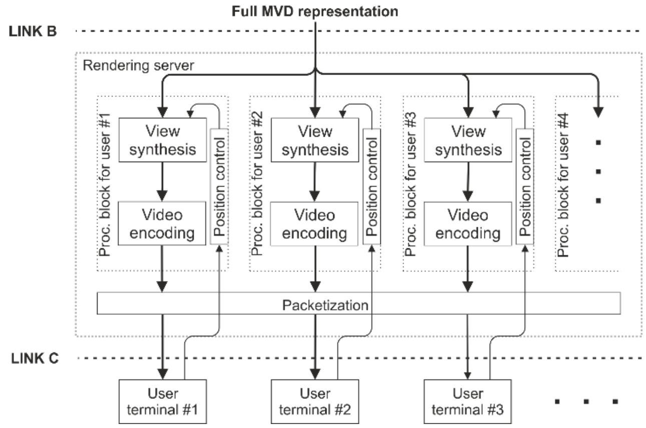

Fig. 13. The rendering server. User entitlement control and the user connection control blocks are not shown.

For the centralized model considered in this paper (see Section I), a realistic delay between viewpoint request and video delivery is much higher than the one allowed for head-mounted devices controlled by head or gaze movements. In such a case the latency should be below 3 ms [61]. This limit is far below the values usually achievable in the centralized model, and is extremely challenging even for a distributed model where the view synthesis is performed in the terminal.

Two examples of virtual walks around scenes are available as video clips attached to this paper as Supplementary Material. The clips were obtained using the publicly available DERS software [58], although virtual video quality may be further improved using newer techniques, e.g. the one developed by the authors and described in [68].

X. Conclusions

In the paper, we have considered an original architecture of a simple, practical low-cost FTV system in contrast to the sophisticated systems usually considered in the references. Straightforward schemes for the rendering server and for video streaming are also proposed. The novelty of the paper is also related to the proposal to build the acquisition system using two-camera modules and to analyze such systems in the context of virtual video quality. The advantages of such a system are demonstrated both theoretically, on the basis of some assumptions regarding occlusions and the accuracy of the depth estimation, and experimentally, using new video test sequences with cameras arranged in pairs and uniformly over an arc.

The mentioned new test sequences, acquired with the use of camera pairs, constitute another novelty of the paper. To the best of our knowledge, these are the first such sequences to the research community. The high quality of the video obtained during a virtual walk around scenes is demonstrated in the video clips provided as supplementary materials that may be downloaded from: http://ieeexplore.ieee.org.

The paper includes experimental results concerning the influence of compression errors on virtual video quality. These results prove that efficient video compression may be used in FTV systems. The paper also provides original results on the improvements of the state-of-the-art 3D-HEVC compression technology. All these results, together with the other results cited in the paper, encourage us to believe that the development of usable FTV systems will be possible within the very next few years.

Appendix I

For the simple model from Fig. 3, the set of points visible by an individual camera is shown in Fig. 14 (as denoted by a dotted line).

For the i-th camera, there are four specific points per object: $F_L(i,z_P)$, $F_R(i,z_P)$ which are the intersections of the boundaries of the i-th camera FOV with any plane P placed at the distance z_P, and $O_L(i,z_P)$, $O_R(i,z_P)$ which denote intersections of lines connecting camera i with leftmost and rightmost point of the foreground object with plane at distance z_P. The entire set of points visible by the i-th camera can be defined as:

$$\mathbb{C}_i = \mathbb{B}_i \cup \mathbb{O}_i \ , \tag{14}$$

where $\mathbb{B}_i$ is set of points of the background seen by camera i, $\mathbb{O}_i$ is set of points of the foreground object visible in camera i:

$$\mathbb{B}_i = [F_L(i,z_B), F_R(i,z_B)] - (O_L(i,z_B), O_R(i,z_B)), \tag{15}$$
$$\mathbb{O}_i = [F_L(i,z_O), F_R(i,z_O)] \cap [O_L(i,z_O), O_R(i,z_O)], \tag{16}$$

where $[q,r]$ denotes a set of points of the scene between horizontal coordinates q and r. For the simplified 2-dimensional scene presented in Fig. 14, the operator indicates a section between points q and r.

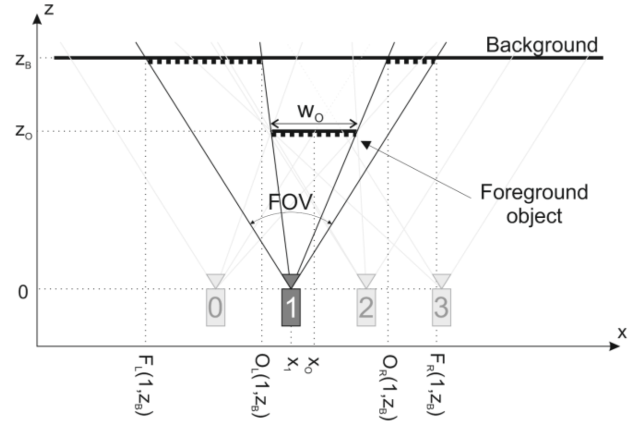

Fig. 14. A simplified model of a multi-camera acquisition system.

A. Derivation of $\bar{b}_u$ and $\bar{b}_p$

Assuming the set of points seen by each camera as in (14), fragments of the scene captured by cameras with particular base can be defined. We distinguish 4 sets of the scene points:

$$\mathbb{K} = \mathbb{C}_0 \cap \mathbb{C}_3 \ , \tag{17}$$

$$\mathbb{L} = \big((\mathbb{C}_0 \cap \mathbb{C}_2) \cup (\mathbb{C}_1 \cap \mathbb{C}_3)\big) - \mathbb{K} \ , \tag{18}$$

$$\mathbb{M} = (\mathbb{C}_1 \cap \mathbb{C}_2) - (\mathbb{K} \cup \mathbb{L}) \ , \tag{19}$$

$$\mathbb{N} = \big((\mathbb{C}_0 \cap \mathbb{C}_1) \cup (\mathbb{C}_2 \cap \mathbb{C}_3)\big) - (\mathbb{K} \cup \mathbb{L} \cup \mathbb{M}) \ , \tag{20}$$

where $\mathbb{K}$ is a set of points with the maximal base, seen by outer cameras, $\mathbb{L}$ is a set of points visible only for one of the outer cameras and the inner camera from another pair, $\mathbb{M}$ is a set of points seen by both inner cameras and $\mathbb{N}$ is a set of points visible only by cameras within one of two camera pairs.

Depth information cannot be obtained for points visible by fewer than two cameras. Therefore, a set of points with determinable depth ($\mathbb{D}$) can be defined as a union of all possible intersections of two sets of points seen by individual cameras.

The mean base distance for uniform arrangement of cameras $\bar{b}_u$ can be calculated as a weighted average of bases for the entire scene:

$$\bar{b}_{\text{u}} = \frac{b_{max} \cdot |\mathbb{K}| + b_1 \cdot |\mathbb{L}| + b_2 \cdot |\mathbb{M}| + b_{min} \cdot |\mathbb{N}|}{|\mathbb{D}|} \ , \tag{21}$$

where $b_{max} = x_3 - x_0$ is the distance between two furthest cameras, b_{min} is the distance between the cameras within one camera pair, $b_1 = b_{max} - b_{min}$, $b_2 = b_{max} - 2 \cdot b_{min}$. $|\mathbb{X}|$ operator depicts the aggregative length (area for 3-D case) of $\mathbb{X}$ (of all continuous subsets of points in set $\mathbb{X}$). The mean base for cameras as pairs $\bar{b}_p$ can be evaluated in the same way.

B. Derivation of OCC_u, OCC_p and $\Delta_{p-u}PSNR_o$

The set of all points of the scene $\mathbb{S}$ can be defined as a union of sets of points seen by all the cameras. Fragments of the background occluded in all cameras are not considered, because they are not visible from any virtual viewpoint in between real cameras. In order to add the fragments of the object that were not seen by any cameras we define set $\mathbb{S}'$:

$$\mathbb{S}' = \mathbb{S} \cup \left[x_O - \frac{w_O}{2}, x_O + \frac{w_O}{2}\right] \ , \tag{22}$$

which contains all points of the scene participating in any virtual view.

The ratio between a set of points with indeterminable depth and a set of points of the entire scene for the uniform camera arrangement describes the number of occlusions in the scene and can be calculated as follows:

$$OCC_u = \frac{|\mathbb{S}' - \mathbb{D}|}{|\mathbb{S}'|} \ . \tag{23}$$

OCC_p can be evaluated in the same way. Assuming that the mean square error of synthesized region (e_s^2) is k times smaller than the mean square error of inpainted region ($e_i^2 = k e_s^2$) (7) can be presented as:

$$\Delta_{p-u}PSNR_o = 10 \log \frac{1 + (1/k^2 - 1) \cdot OCC_u}{1 + (1/k^2 - 1) \cdot OCC_p} \ . \tag{24}$$

Assuming the scene arrangement from Fig. 14 there are two occluded areas (on the left and the right side of the scene), so the largest disoccluded area is of size $OCC_u/2$ for uniform camera arrangement and $OCC_p/2$ for paired cameras..

In the simplest case the inpainting is based on a neighborhood of a disoccluded area, therefore, the error of inpainting is related to similarity $s(n) = {s_1}^n$. The larger the disoccluded area, the higher is the mean error of inpainting. We propose to evaluate k as a mean similarity of disoccluded points to the nearest synthesized point. The similarity between the following points of disocclusion forms a geometric series, therefore (for uniform camera arrangement):

$$k = \frac{\frac{s_1}{1 - s_1} \cdot \left(1 - {s_1}^{W_{cam} \cdot OCC_u/2}\right)}{W_{cam} \cdot OCC_u/2} \ , \tag{25}$$

where W_{cam} is the width of cameras sensor. The evaluation of k for paired cameras is analogous.

APPENDIX II

TABLE III
COMPRESSION OF 7 VIEWS WITH THE DEPTH MAPS

Sequence	QP	MV-HEVC Bitrate [Mbps]	MV-HEVC PSNR [dB]	3D-HEVC Bitrate [Mbps]	3D-HEVC PSNR [dB]	Our Bitrate [Mbps]	Our PSNR [dB]
Poznan Blocks [51]	25	6.85	43.0	6.81	42.9	6.61	43.0
	30	3.76	40.4	3.75	40.2	3.59	40.3
	35	2.14	37.6	2.11	37.4	2.01	37.5
	40	1.22	34.7	1.21	34.5	1.13	34.6
BBB_Flowers [43]	25	6.19	40.5	6.10	40.4	6.01	40.4
	30	3.25	37.7	3.20	37.6	3.13	37.6
	35	1.80	35.0	1.76	34.9	1.71	34.9
	40	1.01	32.2	0.99	32.1	0.95	32.1
Ballet [53]	25	2.06	41.4	1.89	41.4	1.82	41.4
	30	1.05	39.9	0.95	39.7	0.91	39.8
	35	0.59	37.9	0.52	37.6	0.50	37.8
	40	0.33	35.6	0.30	35.4	0.28	35.5
Breakdancers [53]	25	4.66	39.0	4.31	39.0	4.15	39.0
	30	1.96	37.6	1.76	37.4	1.67	37.5
	35	1.02	35.8	0.92	35.7	0.85	35.8
	40	0.54	33.8	0.49	33.7	0.45	33.8
Poznan Fencing2 [54]	25	5.76	40.37	5.76	40.3	5.70	40.35
	30	2.94	38.65	2.92	38.6	2.88	38.59
	35	1.59	36.45	1.59	36.30	1.54	36.39
	40	0.84	33.95	0.84	33.78	0.80	33.89
Poznan Blocks2 [54]	25	4.59	40.08	4.59	40.07	4.49	40.10
	30	2.04	38.48	2.03	38.44	1.96	38.49
	35	1.06	36.62	1.06	36.55	1.01	36.63
	40	0.57	34.50	0.57	34.41	0.54	34.51
Poznan Service2 [54]	25	5.38	40.33	5.35	40.30	5.27	40.31
	30	2.74	38.55	2.73	38.48	2.63	38.52
	35	1.50	36.27	1.48	36.18	1.41	36.23
	40	0.81	33.64	0.80	33.57	0.76	33.64

TABLE IV
MEAN PSNR VALUES FOR EACH TEST SEQUENCE
(AVC ENCODING)

Sequence name	PSNR for original	PSNR [dB] for different QP						
		20	25	30	35	40	45	50
BBB Butterfly Arc	36.9	36.3	36.1	35.7	35.1	34.2	33.0	30.6
BBB Butterfly Linear	35.7	35.4	35.2	34.9	34.4	33.7	32.5	30.4
Dog	30.0	29.5	29.5	29.4	29.2	28.8	28.1	26.9
BBB Flowers Linear	27.5	26.8	26.6	26.5	26.2	25.8	25.3	24.6
Pantomime	30.3	30.0	30.0	29.9	29.9	29.5	28.7	27.6
BBB Rabbit Arc	31.2	30.8	30.6	30.2	29.7	29.0	27.8	26.4
BBB Rabbit Linear	29.8	29.6	29.4	29.2	28.8	28.2	27.3	26.2

TABLE V
MEAN PSNR VALUES FOR EACH TEST SEQUENCE
(HEVC ENCODING)

Sequence name	PSNR for original	PSNR [dB] for different QP						
		20	25	30	35	40	45	50
BBB Butterfly Arc	36.9	35.8	35.4	35.1	34.7	33.9	33.1	30.3
BBB Butterfly Linear	35.7	35.4	35.1	34.8	33.9	33.2	31.4	30.1
Dog	30.0	29.5	29.5	29.4	29.2	28.8	28.1	26.9
BBB Flowers Linear	27.5	26.7	26.2	26.2	25.9	25.5	25.1	24.4
Pantomime	30.3	30.0	29.9	29.9	29.7	29.3	28.9	28.1
BBB Rabbit Arc	31.2	31.2	31.1	30.8	30.4	29.5	28.1	26.2
BBB Rabbit Linear	29.8	29.8	29.7	29.5	29.2	28.8	27.5	25.9

REFERENCES

[1] M. Tanimoto and T. Fujii, “FTV－ Free Viewpoint Television,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M8595, Klagenfurt, July 2002.

[2] S. Wurmlin, E. Lamboray, O. G. Staadt and M. H. Gross, “3D video recorder,” in *10th Pacific Conference on Computer Graphics and Applications*, 2002. Proceedings., 2002, pp. 325-334.

[3] M. Tanimoto, M. Panahpour, T. Fujii, T. Yendo, “FTV for 3-D spatial communication,” *Proceedings of the IEEE*, vol. 100, Issue 4, pp. 905-917, Feb. 2012

[4] E. Bondarev, R. Miquel, M. Imbert, S. Zinger and P. H. N. de With, “On the technology roadmap of Free-Viewpoint 3DTV receivers,” in *2011 IEEE International Conference on Consumer Electronics (ICCE),* Las Vegas, NV, USA, 2011, pp. 687-688.

[5] M. Tanimoto, “Overview of FTV (free-viewpoint television),” in *2009 IEEE International Conference on Multimedia and Expo*, New York, NY, USA, 2009, pp. 1552-1553.

[6] G. Lafruit, K.Wegner, M. Tanimoto, “FTV software framework,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG N15349, Warsaw, Poland, June 2015.

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

[8] C.-C. Lee, A. Tabatabai, K. Tashiro, “Free viewpoint video (FVV) survey and future research direction,” APSIPA Transactions on Signal and Information Processing, vol. 4, Oct. 2015.

[9] “High efficiency coding and media delivery in heterogeneous environments - Part 2: High Efficiency Video Coding,” ISO/IEC IS 23008-2, ITU-T Rec. H.265, 2015.

[10] M. Domański, A. Dziembowski, K. Klimaszewski, A. Łuczak, D. Mieloch, O. Stankiewicz, K. Wegner, “Comments on further standardization for free-viewpoint television,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M35842, Geneva, Switzerland, Feb. 2015.

[11] M. Domański, M. Bartkowiak, A. Dziembowski, T. Grajek, A. Grzelka, A. Łuczak, D. Mieloch, J. Samelak, O. Stankiewicz, J. Stankowski, Krzysztof Wegner, “New results in free-viewpoint television systems for horizontal virtual navigation,” in *2016 IEEE International Conference on Multimedia and Expo (ICME),* Seattle, WA, 2016, pp. 1-6.

[12] I. Stamos and P. K. Allen, “Integration of range and image sensing for photo-realistic 3D modeling,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, San Francisco, CA, 2000, pp. 1435-1440 vol.2.

[13] D. Sandberg, P. E. Forssen and J. Ogniewski, “Model-based video coding using colour and depth cameras,” in *2011 International Conference on Digital Image Computing: Techniques and Applications*, Noosa, QLD, 2011, pp. 158-163.

[14] “JPEG 2000 image coding system, Part 3: Motion JPEG2000,” ISO/IEC IS 15444-3, ITU-T Rec. T.802, 2007.

[15] “Coding of audio-visual objects, Part 10: Advanced Video Coding,” ISO/IEC IS 14496-10, 2014.

[16] M. Domański, A. Dziembowski, A. Kuehn, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner, “Experiments on acquisition and processing of video for free-viewpoint television,” in *2014 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, Budapest, 2014, pp. 1-4.

[17] M. Domański, A. Dziembowski, D. Mieloch, A. Łuczak, O. Stankiewicz and K. Wegner, “A practical approach to acquisition and processing of free viewpoint video,” in *2015 Picture Coding Symposium (PCS)*, Cairns, QLD, 2015, pp. 10-14.

[18] T. Maugey, I. Daribo, G. Cheung and P. Frossard, “Navigation domain representation for interactive multiview imaging,” *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3459-3472, Sept. 2013.

[19] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, 1999, pp. 666-673 vol.1.

[20] G. Miller, J. Starck and A. Hilton, “Projective surface refinement for free-viewpoint video,” in *The 3rd European Conference on Visual Media Production (CVMP 2006),* London, 2006, pp. 153-162.

[21] A. Smolic, K. Müller, P. Merkle, M. Kautzner and T. Wiegand, “3D video objects for interactive applications,” in *2005 13th European Signal Processing Conference*, Antalya, 2005, pp. 1-4.

[22] K. C. Wei, Y. L. Huang and S. Y. Chien, “Point-based model construction for free-viewpoint TV,” in *2013 IEEE Third International Conference on Consumer Electronics Berlin (ICCE-Berlin),* Berlin, 2013, pp. 220-221.

[23] K. Muller, P. Merkle and T. Wiegand, “3-D video representation using depth maps,” *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643-656, April 2011.

[24] M. Tanimoto, T. Senoh, S. Naito, S. Shimizu, H. Horimai, M. Domański, A. Vetro, M. Preda, K. Mueller, “Proposal on a new activity for the third phase of FTV,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M30232, Vienna, 2013.

[25] G. J. Sullivan, J. M. Boyce, Y. Chen, J. R. Ohm, C. A. Segall and A. Vetro, “Standardized extensions of High Efficiency Video Coding (HEVC),” in *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1001-1016, Dec. 2013.

[26] J. Stankowski, Ł. Kowalski, J. Samelak, M. Domański, T. Grajek and K. Wegner, “3D-HEVC extension for circular camera arrangements,” in *2015 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON),* Lisbon, 2015, pp. 1-4.

[27] J. Kim, J. Jang and D. H. Kim, “Design of platform and packet structure for the free-viewpoint television,” in *The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014),* JeJu Island, 2014, pp. 1-2.

[28] L. Toni, G. Cheung and P. Frossard, “In-network view re-sampling for interactive free viewpoint video streaming,” in *2015 IEEE International Conference on Image Processing (ICIP),* Quebec City, QC, 2015, pp. 4486-4490.

[29] T. Fujihashi, Z. Pan and T. Watanabe, “UMSM: A traffic reduction method on multi-view video streaming for multiple users,” *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 228-241, Jan. 2014.

[30] P. Rahimian and J. K. Kearney, “Optimal camera placement for motion capture systems,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 3, pp. 1209-1221, March 1 2017.

[31] X. Chen and J. Davis, “An occlusion metric for selecting robust camera configurations,” Mach. Vis. Appl., vol. 19, pp. 217–222, 2008.

[32] G. Olague and R. Mohr, “Optimal camera placement for accurate reconstruction,” *Pattern Recognit.*, vol. 35, pp. 927–944, 2002.

[33] N. Qian, C.-Y. Lo, "Optimizing camera positions for multi-view 3D reconstruction," in *2015 International Conference on 3D Imaging (IC3D)*, Liege, 2015, pp. 1-8.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, April 2004.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2790162, IEEE Transactions on Multimedia

[35] O. Stankiewicz, K. Wegner, M. Tanimoto, M. Domański, “Enhanced Depth Estimation Reference Software (DERS) for Free-viewpoint Television,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M31518, Geneva, 2013.

[36] R. Hartley, A. Zisserman, “Multiple view geometry in computer vision” (2nd ed.), Cambridge Univ. Press, 2004.

[37] Basler Ace camera documentation [Online]. Available: http://s.baslerweb.com/media/documents/BAS1701_ace_Brochure_SAP0025_EN_web.pdf

[38] E. P. Simoncelli and B. A. Olshausen, “Natural image statistics and neural representation,” *Annual Review of Neuroscience*, vol. 24, pp. 1193–216, May 2001.

[39] K. Wegner, O. Stankiewicz, M. Domański, “Depth based view blending in View Synthesis Reference Software (VSRS),” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M37232, Geneva, Oct. 2015.

[40] Z. Sun; Ch. Jung “Real-time depth-image-based rendering on GPU,” in *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Xi'an, 2015, pp: 324 – 328.

[41] M. Domański, A. Dziembowski, A. Grzelka and D. Mieloch, “Optimization of camera positions for free-navigation applications,” in *2016 International Conference on Signals and Electronic Systems* (ICSES), Krakow, 2016, pp. 118-123.

[42] T. Senoh, K. Wegner, G. Lafruit, “Status of test sequences for free-viewpoint television (FTV),” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M35804, Geneva, Feb. 2015.

[43] P. Kovacs, “[FTV AHG] Big Buck Bunny light-field test sequences,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M35721, Geneva, 2015.

[44] P. Goorts, S. Maesen, M. Dumont, S. Rogmans and P. Bekaert, “Free viewpoint video for soccer using histogram-based validity maps in plane sweeping,” in *2014 International Conference on Computer Vision Theory and Applications (VISAPP),* Lisbon, Portugal, 2014, pp. 378-386.

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

[46] x264 encoder [Online]. Available: www.videolan.org/developers/x264.html

[47] x265 encoder [Online]. Available: www.x265.org

[48] G. Lafruit, K. Wegner, M. Tanimoto, “Call for Evidence on Free-Viewpoint Television: Super-Multiview and Free Navigation,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG N15348, Warsaw, June 2015.

[49] G. Bjøntegaard, “Calculation of average PSNR differences between RD-curves,” Video Coding Experts Group, Doc. VCEG-M33, Austin, Apr. 2001. M15378

[50] 3D HEVC reference codec [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-13.0

[51] K. Wegner, O. Stankiewicz, K. Klimaszewski, M. Domański, “Poznan Blocks – a multiview video test sequence and camera parameters for FTV,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M32243, San Jose, Jan. 2014.

[52] P. Wu, Y. Liu, M. Ye, J. Li and S. Du, "Fast and adaptive 3d reconstruction with extensively high completeness," in IEEE Transactions on Multimedia, vol. 19, no. 2, pp. 266-278, Feb. 2017.

[53] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, “High-quality video view interpolation using a layered representation,” in *ACM Trans. Graphics*, vol. 23, pp. 600-608, Aug. 2004.

[54] A. Grzelka, D. Mieloch, O. Stankiewicz, K. Wegner, “Multiview test video sequences for free navigation exploration obtained using pairs of cameras,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M38247, Geneva, 2016.

[55] K. Müller, A. Vetro, “Common Test Conditions of 3DV Core Experiments,” ITU-T SG 16 WP 3, Doc. JCT3V G1100, San José, Jan. 2014.

[56] L. Do, G. Bravo, S. Zinger and P. H. N. de With, “Real-time free-viewpoint DIBR on GPUs for large base-line multi-view 3DTV videos,” in *2011 Visual Communications and Image Processing (VCIP),* Tainan, 2011, pp. 1-4.

[57] A. Akin, R. Capoccia, J. Narinx, J. Masur, A. Schmid and Y. Leblebici, “Real-time free viewpoint synthesis using three-camera disparity estimation hardware,” in *2015 IEEE International Symposium on Circuits and Systems (ISCAS),* Lisbon, 2015, pp. 2525-2528.

[58] O. Stankiewicz, K. Wegner, M. Tanimoto, M. Domański, “Enhanced view synthesis reference software (VSRS) for Free-viewpoint Television,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M31520, Geneva, 2013.

[59] L. Jorissen, P. Goorts, B. Bex, N. Michiels, S. Rogmans, P. Bekaert, G. Lafruit, “A qualitative comparison of MPEG view synthesis and light field rendering,” in *2014 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON),* Budapest, 2014, pp. 1-4.

[60] T. Kämäräinen, M. Siekkinen, A. Ylä-Jääski, W. Zhang, P. Hui, “Dissecting the end-to-end latency of interactive mobile video applications”, in *HotMobile '17 Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, Sonoma, USA, Feb 2017, pp 61-66.

[61] J. Jerald and M. Whitton, “Relating scene-motion thresholds to latency thresholds for head-mounted displays,” in *2009 IEEE Virtual Reality Conference*, Lafayette, LA, 2009, pp. 211-218.

[62] X. Xiu, G. Cheung and J. Liang, “Delay-cognizant interactive streaming of multiview video with free viewpoint synthesis,” *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1109-1126, Aug. 2012.

[63] C. Yao, J. Xiao, T. Tillo, Y. Zhao, C. Lin and H. Bai, “Depth map down-sampling and coding based on synthesized view distortion,” *IEEE Transactions on Multimedia*, vol. 18, no. 10, pp. 2015-2022, Oct. 2016.

[64] T. Senoh, A. Ishikawa, M. Okui, K. Yamamoto, N. Inoue, “FTV AHG: EE1 and EE2 results with Bee by NICT,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M32995, Valencia, Spain, Apr. 2014

[65] F. Shao, G. Jiang, M. Yu, K. Chen and Y. S. Ho, “Asymmetric coding of multi-view video plus depth based 3-d video for view rendering,” *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 157-167, Feb. 2012.

[66] M. Tanimoto, T. Fujii, N. Fukushima, “1D Parallel test sequences for MPEG-FTV,” ISO/IEC JTC1/SC29/WG11, Doc. MPEG M15378, Archamps, Apr. 2008.

[67] P. Goorts, M. Javadi, S. Rogmans, G. Lafruit, “San miguel test images with depth ground truth”, ISO/IEC JTC1/SC29/WG11, Doc. MPEG M33163, Valencia, Mar. 2014.

[68] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, K. Wegner, M. Domański, “Multiview synthesis – improved view synthesis for virtual navigation”, in *32nd Picture Coding Symposium PCS 2016*, Nuremberg, Germany, Dec. 2016.

[69] D. Mieloch, A. Dziembowski, A. Grzelka, O. Stankiewicz, M. Domański, “Graph-based multiview depth estimation using segmentation,” in *2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong*, 2017.

[70] P. Gargallo, E. Prados, P. Sturm, “Minimizing the reprojection error in surface reconstruction from images,” in *IEEE 11th International Conference on Computer Vision*, 2007.

[71] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations”, in *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999.

[72] F. Zilly, C. Riechert., M. Muller., P. Eisert, T. Sikora, P. Kauff, “Real-time generation of multi-view video plus depth content using mixed narrow and wide baseline”, *Journal of Visual Communication and Image Representation*, vol. 25, no. 4, pp. 632–648, 2014.

[73] L. Jorissen, P. Goorts, S. Rogmans, G. Lafruit, P. Bekaert, “Multi-camera epipolar plane image feature detection for robust view synthesis”, in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2015.

[74] F. Shao, W. Lin, G. Jiang, M. Yu, Q. Dai, "Depth map coding for view synthesis based on distortion analyses," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 1, pp. 106-117, March 2014.

[75] R. Wang et al., "Accelerating image-domain-warping virtual view synthesis on GPGPU," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1392-1400, June 2017.

Olgierd Stankiewicz received his M.Sc. and Ph.D. from the Faculty of Electronics and Telecommunications, Poznań University of Technology in 2014. Currently, he is an assistant professor at the Chair of Multimedia Telecommunications and Microelectronics. In 2005 he won the second place in IEEE Computer Society International Design Competition (CSIDC), held in Washington D.C. He is actively involved in ISO standardization activities where he contributes to the development of the 3D video coding standards. In years 2011-2014 he was a coordinator of development of MPEG reference software for 3D-video coding standards based on AVC. Now he contributes to MPEG Free viewpoint TV and JPEG-PLENO standardization activities. He has published over ninety MPEG/JPEG standardization documents as well as about thirty papers on free view television, depth estimation, view synthesis and hardware implementation in FPGA. His professional interests include signal processing, video compression algorithms, computer graphics and hardware solutions.

Marek Domański received M.Sc., Ph.D. and Habilitation degrees from Poznań University of Technology, Poland in 1978, 1983 and 1990, respectively. Since 1993, he is a professor at Poznań Univ. of Technology, where he leads Chair of Multimedia Telecommunications and Microelectronics. He coauthored one of the very first AVC decoders for TV set-top boxes (2004) as well as highly ranked technology proposals to MPEG for scalable video compression (2004) and 3D video coding (2011). He authored 3 books and over 300 papers in journals and conference proceedings. The contributions were mostly on image, video and audio compression, virtual navigation, free-viewpoint television, image processing, multimedia systems, 3D video and color image technology, digital filters and multidimensional signal processing. He was General Chairman/Co-Chairman and host of several international conferences: Picture Coding Symposium, PCS 2012; IEEE Int. Conf. Advanced & Signal-based Surveillance, AVSS 2013, European Signal Processing Conference, EUSIPCO 2007; 73rd and 112nd Meetings of MPEG; Int. Workshop on Signals, Systems and Image Processing, IWSSIP 1997 and 2004; Int. Conf. Signals and Electronic Systems, ICSES 2004 and others. He served as a member of various steering, program and editorial committees of international journals and international conferences.

Adrian Dziembowski received his M.Sc. in 2014. Currently, he is a research assistant and a Ph.D. student at the Chair of Multimedia Telecommunications and Microelectronics, where he has been working on the projects in the field of free-viewpoint television systems since receiving his Engineer's degree in 2012. His professional interests include FTV, virtual view synthesis and computer graphics.

Adam Grzelka received M.Sc. degree from Poznan University of Technology in 2014. He is a Ph.D. student at the Chair of Multimedia Telecommunications and Microelectronics. The main area of his professional activities are image processing, FTV (free-viewpoint television) and hardware programing.

Dawid Mieloch received his M.Sc. from Poznań University of Technology in 2014. Currently, he is a research assistant and a Ph.D. student at the Chair of Multimedia Telecommunications and Microelectronics. He has been involved in several projects focused on multiview and 3-D video processing. His professional interests include free-viewpoint television, depth estimation and camera calibration.

Jarosław Samelak received M.Sc. degree from Poznan University of Technology in 2015. He is a Ph.D. student at the Chair of Multimedia Telecommunications and Microelectronics. His current research interests include video compression, optimized video processing algorithms, software optimization techniques.