A Practical Approach to Acquisition and Processing of Free Viewpoint Video

M. Domański, A. Dziembowski, D. Mieloch, A. Łuczak, O. Stankiewicz, K. Wegner
Chair of Multimedia Telecommunications and Microelectronics
Poznań University of Technology, Poznań, Poland

***Abstract*—We deal with the processing of multiview video acquired by the use of practical thus relatively simple acquisition systems that have a limited number of cameras located around a scene on independent tripods. The real-camera locations are nearly arbitrary as it would be required in the real-world Free-Viewpoint Television systems. The appropriate test video sequences are also reported. We describe a family of original extensions and adaptations of the multiview video processing algorithms adapted to arbitrary camera positions around a scene. The techniques constitute the video processing chain for Free-Viewpoint Television as they are aimed at estimating the parameters of such a multi-camera system, video correction, depth estimation and virtual view synthesis. Moreover, we demonstrate the need for new compression technology capable of efficient compression of sparse convergent views. The experimental results for processing the proposed test sequences are reported.**

I. Introduction

Free-Viewpoint Television (FTV) is an interactive video service that provides the ability for a viewer to navigate freely around a scene [1],[2]. A viewer watches the scene from virtual viewpoints on an arbitrary navigation trajectory. At each virtual viewpoint, the corresponding view has to be synthesized and made available at the receiver. Possibly many viewers share the same FTV service, and each viewer navigates independently. View synthesis may use either the distributed model where views are synthesized independently in each receiver, or the centralized model where views requested by all viewers are synthesized in the servers of the service provider [3],[4]. The distributed model requires high transmission bandwidth in server-to-viewer downlinks and significant processing power of viewer terminals [5],[6]. On the other hand, the centralized model suffers from delays in the bidirectional server-to-terminal communications [7], similarly to networked gaming. Therefore, both models are considered for future applications.

An FTV system requires efficient techniques for multicamera system calibration and video correction [8], depth estimation and view synthesis [9]. Many techniques have been already proposed for the abovementioned tasks, but mostly for the linear camera setup and small distances between cameras. Fewer works dealt with circular camera setup (Fig. 1) and arbitrary locations of a limited number of cameras [1],[10]. On the other hand, in a practical FTV system the number of cameras should be limited, and therefore the distances between cameras are large. The cameras are located around a scene, in a roughly circular camera setup. Therefore, this paper focuses on the chain of multiview video processing in an FTV system with circular camera setup and limited number of cameras. The considerations will be appropriate for both distributed and centralized systems. Therefore, the communications between the servers and the viewer terminals as well as audio processing will be left beyond the scope of this short paper.

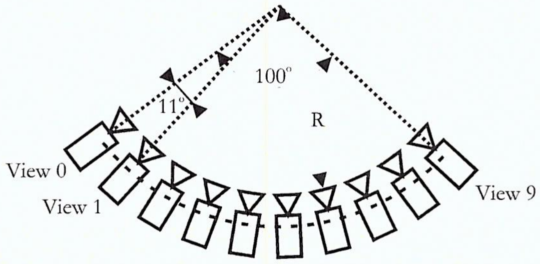

Figure 1. The circular camera setup used in the experiments reported in this paper.

II. Video acquisition

An important requirement for an FTV multi-camera system is its practicality, including moderate cost, portability, easy operation etc. These features may be obtained by the application of standard portable cameras mounted on individual tripods of approximately the same height that stand roughly on an arc around a scene (Fig. 2). For experiments and FTV-system prototyping, we have designed and developed a system with 10 HDTV global-shutter cameras (Canon XH G1) as in Fig. 2.

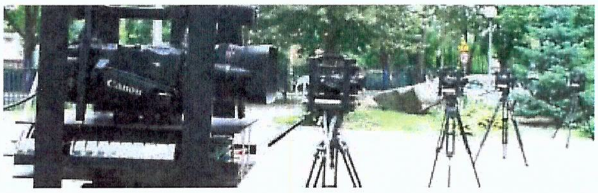

Figure 2. Tripods with wireless camera modules designed and produced at Poznań University of Technology.

The multiple cameras would need common control and exact synchronization in addition to signal and power supply cables. In order to avoid problems with huge number of cables, a wireless camera module was developed by the authors (see left side of Fig. 2). Each module has wireless control and synchronization links, local power supply and hard disk for captured video. The whole system has no cables and can be easily operated by a single person. It is used for the acquisition of compressed or uncompressed multicamera video, both indoor and outdoor. For the hardware details please refer to [10].

978-1-4799-7783-3/15/$31.00 ©2015 IEEE

Further in this paper, video processing will be considered in the context of the acquisition with the use of the abovementioned system.

The experimental system was built having in mind such future FTV applications, e.g. sport broadcasts (like judo, wrestling, sumo, dance etc.), performances (theater, circus), interactive courses (medical, cosmetics, dance etc.), manuals and school teaching materials.

III. Test sequences

Very few multiview sequences, recorded using non-linear camera arrangements, are available. Moreover, none of them is available in Full-HD resolution. Therefore, for experimental purposes, we have produced a set of 10-view test sequences recorded using the abovementioned multicamera system with nearly-circular camera arrangement. The created sequences (Fig. 3) have registered both indoor ("Poznan Blocks" [11]) and outdoor ("Poznan Team") scenes, and are offered to the FTV research community. Basic specification of sequences is: resolution 1920×1080, length 1000 frames, 25 frames per second, the camera setup radius $R = 3$ or 15 meters (Fig. 1), for the indoor and outdoor video, respectively.

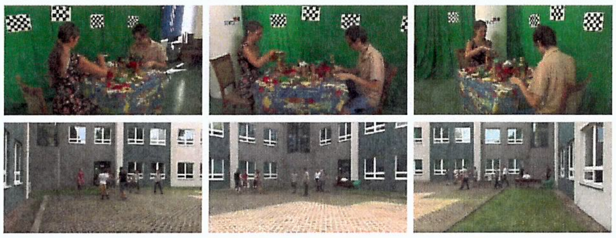

Figure 3. Views 0, 5 and 9 of indoor and outdoor test sequences: "Poznan Blocks" (top) and "Poznan Team" (bottom).

IV. System calibration

A. Estimation of intrinsic parameters and lens distortions

This step of calibration is performed independently for each camera, so existing algorithms can be used. We used the method described in [8]. For each camera, a number of different views of a checkerboard has to be acquired (3 – 6 views are sufficient to provide repeatability of estimated parameters). The method was shown to provide high calibration accuracy [12].

B. Extrinsic parameters estimation

We propose to use a modification of the technique from [13] that requires a set of corresponding characteristic points for each camera. Those points can be obtained by the use of calibration objects, feature extractors and matching algorithms, or they can be provided manually. For a circular camera setup, a halogen lamp is a good choice for a calibration object [14], as it is simultaneously visible by all cameras.

Every point of the image obtained from one of the cameras, in another view should lie on an epipolar line, which directly depends on extrinsic parameters [15]. For each camera we estimate 6 parameters: the 3-dimensional vector of translation $\vec{t}$ (between camera's optical center and global reference system) and 3 angles of camera's rotation in 3D-space: ϕ, θ, ψ. These parameters can be obtained through minimization of the sum of the distances between every characteristic point and the corresponding epipolar line. We propose calculation of the error summed over all combinations of two cameras in every group of C neighboring cameras, where the final parameters are calculated with respect to the global coordinate system. We have chosen $C = 4$, so the groups contain cameras 1-2-3-4, 2-3-4-5, etc. This size of camera groups yields a reduction of error propagation among all cameras, simultaneously ensuring global consistency of the extrinsic parameters. The error function is defined as:

$$E(\vec{t},\phi,\theta,\psi) = \sum_{n=0}^{N-C} \sum_{c_{ref}=n}^{C+n} \sum_{\substack{c_t=n \\ c_t \neq c_{ref}}}^{C+n} \sum_{p=0}^{P} d(n, c_{ref}, c_t, p)\,, \quad (1)$$

where N is the number of cameras in the system ($N = 10$ in our case), C – the number of cameras in each group, n – the group index, c_{ref} – the reference camera index, c_t – the target camera index, P – the quantity of the visible characteristic points in a camera group, p – the index of a characteristic point and d – the distance between point p and the corresponding epipolar line. Obviously, the above formula (1) defines the cost function, but its practical implementation should avoid multiple calculations of individual distances. Moreover, when the rotation of camera is represented with Euler angles, a situation called "gimbal lock" can occur [16]: for some angles the rotation loses one degree of freedom, because two rotation axes are on the same plane. In order to prevent this situation, we use quaternion representation of the rotation angles ϕ, θ, ψ, as in [17].

The extrinsic parameters are estimated by minimization of E using a non-gradient optimization algorithm. The starting values of the parameters are roughly calculated from the camera system geometry, i.e. the angles between the cameras and the distance from the center of the scene.

The accuracy of the proposed procedure was estimated using the average value of d (see (1)) for both automatically and manually selected characteristic points. For 50 automatically detected positions of a halogen lamp, the mean d was 0.11 sampling period (with standard deviation 0.17). For 20 manually selected points, the error was 0.46 sampling period (with standard deviation 0.62). For automatic acquisition of characteristic points, very low errors indicate both the high accuracy of proposed algorithms and proper operation of the camera synchronization modules. Moreover, with manually selected points, the algorithm still provides sufficiently low error, that enables using these parameters for depth estimation.

C. Color correction

The algorithm of color correction is based on histogram adjustment and it is performed independently for each RGB component. In our adaptation of the technique [18], in the first step, we estimate the differences in histograms between all neighboring cameras. Then, in added second step, for each RGB component, we adjust a histogram of every view to the histogram of center view.

The proposed method [19] was previously used to correct sequences recorded by a system with linear camera setup. In a nearly-circular system, the described approach provides a slight improvement of synthesis quality, increasing the PSNR value by almost 0.5 dB.

V. Depth Estimation

The core of the depth estimation algorithm implemented by the authors uses the workflow of the enhanced version of the Depth Estimation Reference Software [20]. However, in this paper, some new original improvements dedicated to a system with arbitrary camera locations are proposed: the global depth map levels, various matching block sizes, inter-view consistency in the Graph Cut optimization and independent depth estimation for stationary background or moving objects.

A. *Matching error estimation*

Depth information about the scene is obtained automatically from real views. The matching error is calculated by matching potentially corresponding pixels in each view. As opposed to linear camera setup, where corresponding points were located at the same horizontal lines in each view, now they lie on the epipolar lines.

The search for corresponding points within the epipolar line is performed by projecting points between views using the projection matrices, combining both intrinsic and extrinsic parameters of each camera. Any point visible in one camera can be projected to another one using the equation:

$$[x_2\ y_2\ z_2\ 1]^T = P_2 \cdot P_1^{-1} \cdot [x_1\ y_1\ z_1\ 1]^T, \tag{2}$$

where P_2 and P_1 are the projection matrices of two cameras, (x_1, y_1) are the coordinates of a particular point in one view, z_1 represents its depth. By changing the value of z_1 it is possible to reach every point on the specified epipolar line.

Similarity measurement between potentially corresponding points is performed in blocks of different sizes. An increase of block sizes enables avoiding the influence of noise, but also it causes the removal of a significant amount of image details, we can preserve with the use of small blocks (e.g. 1×1). The proposed method, that merges the advantages of both methods, uses blocks of various sizes. The improvement of depth quality, compared to constant blocks sizes, is shown below (Fig. 4).

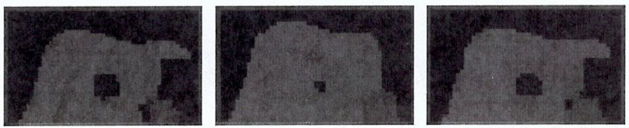

Figure 4. A part of a depth map estimated using pixel matching (left), 7×7 blocks (center), and various-sized blocks (right).

As an option, depth maps may be estimated separately for stationary background and for moving objects after segmentation of pictures. We have implemented this option and used it in the experiments described further in this paper. That improvement may enhance the quality of depth map, especially for sequences with relatively small moving objects, like e.g. “Poznan Team” (Fig. 3).

B. *Optimization*

To perform the optimization of calculated matching error, Graph Cut algorithm is used [21]. Graph Cut is essentially an optimization algorithm for binary-valued variables. The approach from [21] assumes the following goal function:

$$T(f) = \sum_{p} D_p(f_p) + \sum_{(p,q)} V_{p,q}(f_p, f_q), \tag{3}$$

where p and q are pixels in the image, f is the depth of a point, D_p is a cost function of assigning the depth f_p to the pixel p and $V_{p,q}$ is consequently a cost function of assigning depth f_p to p and f_q to q. D_p is equal to the previously computed matching error, $V_{p,q}$ (called smoothness term) is an absolute difference of f_p and f_q in the simplest yet practical case.

For each level of depth, a graph is created. It bonds all processed pixels (represented as nodes) with edges weighted by the goal function T. All nodes can be assigned to one of two sets – one represents the previously calculated depth level of a point, the other – the currently processed depth level (expansion move algorithm [22]) To find a cut of graph, which minimizes the global energy, the maximum flow is computed.

In order to achieve spatial (inter-view) consistency of depth maps computed for neighboring cameras, a different goal function is proposed:

$$T(f) = \sum_{(c,d)} \sum_{(p,p')} V_{p,p'}(f_p, f_{p'}) + \sum_{(p,q)} V_{p,q}(f_p, f_q), \tag{4}$$

where p' is a pixel in the view c that corresponds to the pixel p from view d. Therefore, the depth of each point is estimated using information from all views simultaneously.

When using the abovementioned goal function (4), it is necessary to change the interpretation of the depth map. Typically, for each point the depth is expressed as the distance between a camera and a plane containing the 3D point, perpendicular to camera’s optical axis. When cameras optical axes are parallel, the points corresponding to the same point P in 3D space in different cameras have the same depth (Fig. 5a). This statement is not true for a circular arrangement of cameras (Fig. 5b). For that setup, the depth map has to be calculated as a distance from the center camera of the system, so in spite of cameras placed on an arc, the depth levels are arranged like parallel planes as shown in Fig. 5c. We define the depth levels that are common for all cameras as global depth map levels.

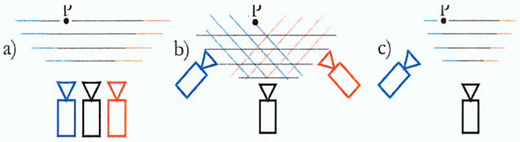

Figure 5. Depth map levels for: a) the parallel optical axes, b) the circular arrangement of cameras, c) the circular arrangement with global depth map levels.

VI. Virtual View Synthesis

In that step, views representing images captured by virtual cameras are being created. We have developed an algorithm of virtual view synthesis which uses MPEG’s reference software (VSRS [23]) core, adding several improvements for the adaptation of the algorithm for circular camera arrangement: global depth map levels, median filtering of occluded regions and multi-camera synthesis.

The main idea of the synthesis algorithm is depicted in Fig. 6. It consists of two major phases: virtual depth map calculation (blue arrows) and backward texture projection (orange).

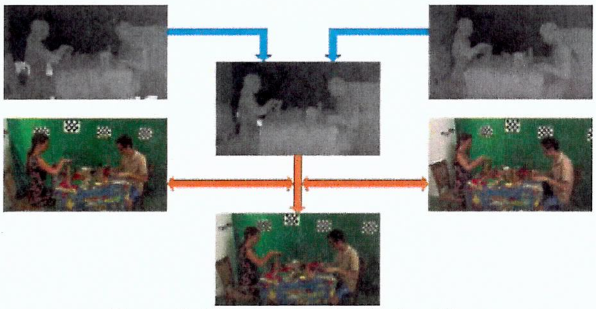

Figure 6. The view synthesis algorithm.

A. *Virtual depth map calculation*

In the first phase, only depth maps for real views are used. A set of depth maps may comprise two or more depth maps. By increasing the number of depth maps many occlusions can be eliminated, which increases the quality of the virtual depth maps (the depth maps for the virtual views).

A virtual depth map is created using the depth information and the camera parameters, precisely: the projection matrices of the real cameras: P_R, and the virtual ones: P_V. For each real camera the workflow is identical: every point from its depth map is projected to virtual depth map:

$$[x_V\ y_V\ z_V\ 1]^T = P_V \cdot P_R^{-1} \cdot [x_R\ y_R\ z_R\ 1]^T. \quad (5)$$

As a result, we obtain a vector containing the position and the depth of the same point in the virtual camera.

At this point, there are several partial depth maps for one virtual view – each of them projected from only one input depth map. They subsequently have to be merged into one proper virtual depth map. The final depth value has to be chosen independently for every point in the virtual depth map. We performed the merging because of occlusions in the real views and their different lighting characteristics.

After merging, there can be some regions with no information about depth in the virtual depth map – non-synthesized regions. Before texture projection, these regions have to be filled up. The proper way to achieve that is to filter the depth map by a median filter. However, the entire image filtering would cause the displacement of the edges. Therefore, the depth map is filtered only in non-synthesized regions, leaving remaining areas unchanged. The influence of such filtering, in comparison with disabling that feature is presented on Fig. 7.

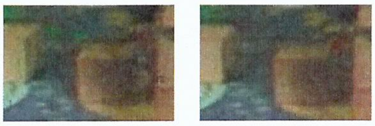

Figure 7. A part of a virtual view: without depth map filtering (left), with median filtering in non-synthesized regions (right).

B. *Backward texture projection*

In order to obtain the final virtual view, it is necessary to estimate the proper color of each image pixel. The texture is projected from the real views to the virtual ones according to the previously estimated values of the virtual depth maps.

Just like in the virtual depth map calculation, every pixel is projected from one view (virtual) into another one (real). Then, the color values from the projected position in a real view are copied into a virtual view. The color of every pixel can be copied from any view. Different lighting conditions in the real views may cause color artifacts in the synthesized view. To avoid it, the colors of the corresponding pixels should be averaged, for example using weighted average with two separate weights: the distance to virtual camera and the value of the projected depth from each real camera.

VII. Accuracy Estimation

For individual multiview video processing tasks, the quality assessment methods were already presented in several papers (e.g. [6],[24]). These methods are based on a common idea, that the quality can be evaluated by the comparison of the virtual view and the real view from the same viewpoint. For example, in our test sequences, the real View 1 (Fig. 1) may be compared to a virtual view at this location. This virtual view is synthesized using Views 0 and 2, where the angle between these views is 22°. We are also able to synthesize virtual views for real views having an angle of 44°, 66° and 88° in between. Therefore, we are able to estimate the error expressed as PSNR for all mentioned angles.

Nevertheless, in an FTV system, the virtual views are synthesized using real views with angle 11° in between. The PSNR value for this angle may be obtained by extrapolation of PSNR values for 22°, 44°, 66° and 88° (see Fig. 8). Therefore, the estimated luma PSNR for 11° is also a good estimate for the overall accuracy of the system. This estimated PSNR (about 31 dB) is an indicator for the acceptable quality of the whole video processing chain.

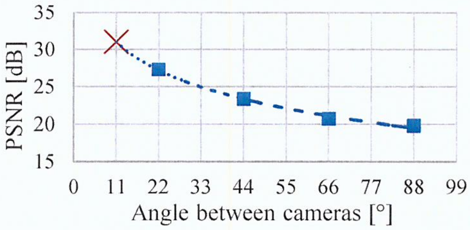

Figure 8. Average luma PSNR for the synthesized frames with respect to the corresponding reference from a real camera; data are extrapolated for angles less than 22° between cameras (for “Poznan Blocks” sequence).

VIII. Video Compression

The experiments were designed to assess the available compression efficiency for the sequences with a circular camera arrangement. We compare the HEVC-based state-of-the-art codecs: simulcast HEVC [25], MV-HEVC [26] and 3D-HEVC [27]. The configuration parameters for all encoders were the same: intra-period = 24, GOP = 8, 1 slice per picture, SAO and VSO switched on.

The results of the experiments (Fig. 9) show, that the specialized MV-HEVC and 3D-HEVC codecs provide only a small improvement over HEVC simulcast, i.e. bitrate is reduced only by up to 13% (for "Poznan Blocks" by 10.9 % for MV-HEVC, and 13.1% for 3D-HEVC, and for "Poznan Team" by 3.1% and 3.6%, respectively).

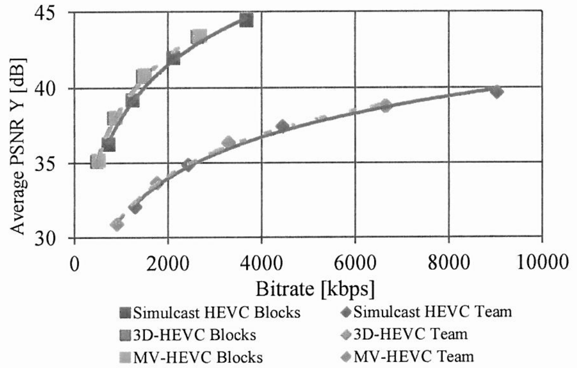

Figure 9. Compression of "Poznan Blocks" and "Poznan Team" sequences (the first 50 frames) for 3 views.

IX. CONCLUSIONS

We propose the entire processing chain for multiview video acquired by a system with a nearly-circular camera arrangement. In order to achieve high quality of the synthesized virtual views, the usage of existing methods is not satisfactory. Therefore, we have adapted widely-known algorithms by adding some improvements and modifications, e.g. optimization-based extrinsic parameters estimation with modified goal function, spatial-consistent optimization of depth maps, or global depth map levels. Moreover, the paper presents the experimental results, that show the need for new compression techniques that will provide higher compression efficiency for nearly-circular arrangements of cameras. For research purposes a set of multiview test sequences was created. Two 10-view sequences with respective camera parameters are now available to the FTV research community and can be obtained from the authors: {ostank, kwegner}@multimedia.edu.pl.

ACKNOWLEDGEMENT

Research project was supported by National Science Centre, Poland, according to the decision DEC-2012/05/B/ST7/01279.

The authors express their sincere thanks to the reviewers for their valuable comments.

REFERENCES

[1] M. Tanimoto, M. Tehrani, T. Fujii, T. Yendo, "FTV for 3-D spatial communication", Proc. IEEE, Vol. 100, pp. 905-917, April 2012.

[2] 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 JTC 1/SC 29/WG 11, Doc. M30229/M30232, Vienna, Austria, July/Aug. 2013.

[3] M. Domański, "Practicing free-viewpoint television: multiview video capture and processing," in: M. Tanimoto, T. Senoh, "FTV seminar report," ISO/IEC JTC 1/SC 29/WG 11, Doc. MPEG M34564, July 2014.

[4] J. Kim, J. Jang, D. Ho Kim, "Design of platform and packet structure for the free-viewpoint television", 18th IEEE International Symposium on Consumer Electronics, Jeju Island, 2014.

[5] E. Bondarev, R. Miquel, M. Imbert, S. Zinger, P. de With, "On the technology roadmap of free-viewpoint 3DTV receivers", IEEE International Conference on Consumer Electronics, Las Vegas, pp. 687 – 688, 2011.

[6] 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", 3DTV-CON, Budapest 2014.

[7] J. Osada, N. Fukushima, Y. Ishibashi, "Influence of network delay on viewpoint change in free-viewpoint video transmission", 18th Asia-Pacific Conference on Communications, Jeju Island, pp. 110 –115, 2012.

[8] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations", IEEE International Conference on Computer Vision, vol. 1, pp. 666-673, 1999.

[9] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, Y. Mori, "Reference Softwares for Depth Estimation and View Synthesis", ISO/IEC JTC1/SC29/WG11 MPEG2008/M15377, Archamps, France, 2008.

[10] 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", 3DTV-CON, Budapest 2014.

[11] M. Domański, A. Dziembowski, A. Kuehn, M. Kurc, A. Łuczak, D. Mieloch, J. Siast, O. Stankiewicz, K. Wegner, "Poznan Blocks - a multiview video test sequence and camera parameters for Free Viewpoint Television", ISO/IEC JTC1/SC29/WG11, Doc. M32243, 2014.

[12] W. Sun and J. R. Cooperstock, "Requirements for camera calibration: Must accuracy come with a high price?" in Proceedings of the Seventh IEEE Workshops on App. of Comp. Vision, vol. 1, pp. 356-361, 2005.

[13] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review",Int. Journal .Computer Vision, 27(2), pp. 161–195, 1998.

[14] T. Svoboda, D. Martinec, T. Pajdla, "A convenient multi-camera self-calibration for virtual environments", Presence, 14, pp. 407-422, 2005.

[15] A. Heyden, M. Pollefeys, "Multiple view geometry", in G. Medioni, S. B. Kang (Eds.), "Emerging topics in computer vision", pp. 63-75, Prentice Hall, 2004.

[16] D. S. Brezov, C. D. Mladenova, I. M. Mladenov, "New perspective on the gimbal lock problem, AIP Conference Proc. 1570, Sozopol, 2013.

[17] J. Yan-Bin, "Quaternions and Rotations", Com S 477/577 Notes, Iowa State Univ., 2014.

[18] U. Fecker, M. Barkowsky, A. Kaup. "Histogram-based prefiltering for luminance and chrominance compensation of multiview video coding". IEEE Trans. Circ. Syst. Video Tech., vol. 18, pp. 1258-1267, 2008.

[19] J. Stankowski, K. Klimaszewski, O. Stankiewicz, K. Wegner, M. Domański, "Preprocessing methods used for Poznan 3D/FTV test sequences", MPEG 2010 / M17174, Kyoto, Japan, 2010.

[20] O. Stankiewicz, K. Wegner, M. Tanimoto, M. Domański, "Enhanced depth estimation reference software (DERS) for Free-viewpoint Television", ISO/IEC JTC 1/SC 29/WG 11, Doc. MPEG M31518, 2013.

[21] V. Kolmogorow, R. Zabih, S.J. Gortler, "Generalized multi-camera scene reconstruction using graph cuts", IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 501-516, 2003.

[22] V. Kolmogorov, R. Zabih. "What energy functions can be minimized via graph cuts?", 7th Eur. Conf. on Comp. Vision, vol. 3, pp. 65–81, 2002.

[23] O. Stankiewicz, K. Wegner, M. Tanimoto, M. Domański, "Enhanced view synthesis reference software (VSRS) for Free-viewpoint Television", ISO/IEC JTC 1/SC 29/WG 11, Doc. M31520, Oct. 2013.

[24] L. Yu, S. Xiang, H. Deng, P. Zhou, "Depth Based View Synthesis with Artifacts Removal for FTV", 6th Int. Conf. on Image and Graphics, 2011.

[25] "High Efficency Video Coding", ISO/IEC Int. Standard 23008-2, ITU-T Rec. H.265, 2013.

[26] G. Tech, K. Wegner, Y. Chen, M. Hannuksela, J. Boyce, "MV-HEVC Draft Text 8", JCT-3V of ITU-T, ISO/IEC Doc. JTC3V-H1004, 2014.

[27] G. Tech, K. Wegner, Y. Chen, S. Yea, "3D-HEVC Draft Text 4", JCT-3V of ITU-T and ISO/IEC Doc. JTC3VH1001, 2014.