Enhanced View Synthesis Prediction for Coding of Non-Coplanar 3D Video Sequences

Jens Schneider, Johannes Sauer and Mathias Wien
Institut für Nachrichtentechnik, RWTH Aachen University, Germany

***Abstract*—In many cases, view synthesis prediction (VSP) for 3D video coding suffers from inaccurate depth maps (DMs) caused by non-reliable point correspondences in low textured areas. In addition, occlusion handling is a major issue if prediction is performed from one reference view. In this paper, we address VSP for non-coplanar 3D video sequences modified by a linear DM enhancement model. In this model, points close to edges in the DM are treated as key points, on which depth estimation is assumed to work well. Other points are assumed to have unreliable depth information and the depth is modelled by the solution of the Laplace equation. As another preprocessing stage for DMs, the influence of guided image depthmap filtering is investigated. Moreover, we present an inpainting method for occluded areas, which is based on the solution of the Laplace equation and corresponding depth information. Combining these techniques and applying it to VSP for 3D video sequences, which are captured by an arc-arranged camera array, our approach outperforms state of the art models in coding efficiency. Experimental results show bit rate savings up to 4.3 % compared to HEVC-3D for coding two views and corresponding depth in an All Intra encoder configuration.**

***Index Terms*—Predictive coding, video compression, view synthesis**

I. Introduction

Video coding has evolved from single-view coding to multi-view coding, which was already standardized in AVC [1] and is also part of the latest standard called High Efficiency Video Coding (HEVC) [2]. In addition, the HEVC-3D extension has been developed in order to enable efficient coding for sequences represented by the multiview plus depth (MVD) format. An overview on 3D video coding techniques can be found in [3]. However, standardization so far has focused on coding texture and depth of multiple views on coplanar camera arrangements.

This is expected to change in the future. For example, the MPEG exploration activity on free viewpoint television (FTV) targets at video coding for free view point navigation [4]. This technique shall allow for user controlled modification of the viewing angle under which a scene is displayed on a conventional 2D display or a walk around functionality for super multi-view displays. In consequence, either efficient coding of a large number of views ($\# \rightarrow 100$), or of a small number of views ($\# < 10$), connected with an appropriate view synthesis method, might give a representation of the sequence at acceptable bit rates.

In this paper, we will focus on coding two views and corresponding DMs captured in an arc camera arrangement with large baselines. Note that this is not a limitation because the proposed concept can be easily extended to more views. Generally speaking, view synthesis from texture and depth is an ill-posed problem and one cannot obtain a (true) solution in any case. As a possible implementation, View Synthesis Reference Software (VSRS) has been developed by MPEG as reference software for view synthesis at any point in 3D space from two reference views and corresponding DMs [5].

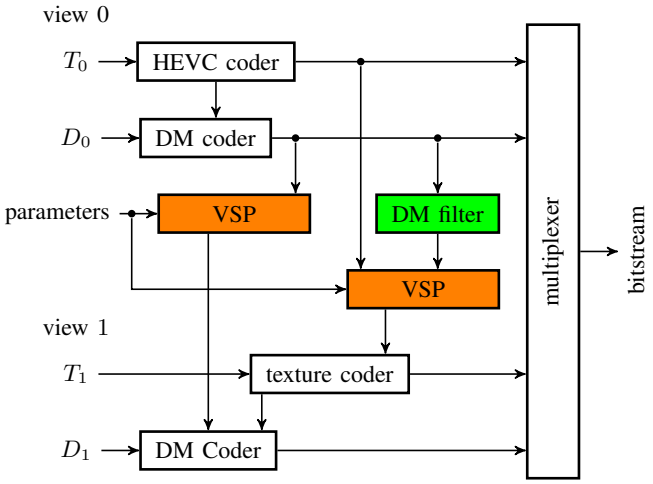

Fig. 1: Overall coding scheme with view synthesis prediction for two views. The input of parameters for the VSP building blocks is related to camera parameters and view synthesis options.

In many cases the view synthesis suffers from inaccurate DMs which were estimated from the epipolar geometry of the scene. In this paper, we propose utilizing enhanced DMs for inter-view prediction. The proposed enhancement model overcomes inaccuracies within homogeneous regions in DMs which results in larger consistent image regions after warping than obtainable with the original DMs. As will be shown, VSP with enhanced DMs outperforms prediction with original DMs. Moreover, we introduce improvements to VSRS in order to make it feasible for generating a reasonable reference picture for prediction.

The structure of the paper is as follows. Section II presents the experimental VSP framework and some adjustments made to the view synthesis algorithm implemented in VSRS. In section III our enhancement model and a suitable preprocessing filter for DMs are detailed. Coding results and conclusions are finally presented in Sections IV and V respectively.

II. View Synthesis Prediction

View synthesis prediction utilizes a reference picture for prediction which is generated by 3D warping. The overall coding scheme for two views using this type of prediction is

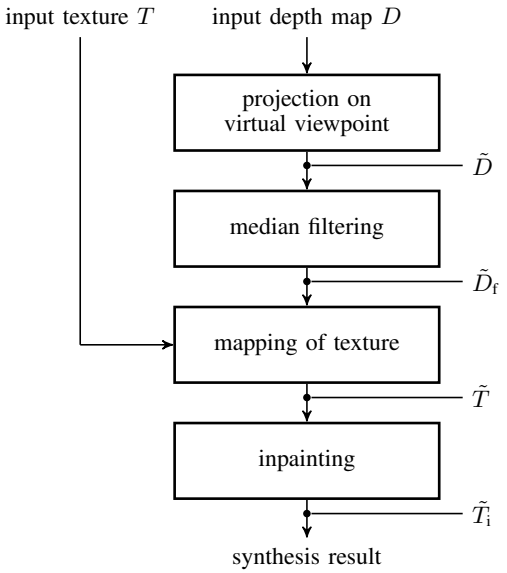

Fig. 2: Reverse warping algorithm

depicted in Fig. 1. In general, the system can be extended to more views. The 3D warping technique referred to the VSP blocks in Fig. 1 is not based on forward per pixel warping of the texture but an algorithm called reverse warping, which results in better synthesis quality than forward warping [5]. The algorithm is shown in Fig. 2 and it's key stages are median filtering of the warped DM $\tilde{D}$ and inpainting of the preliminary synthesis result $\tilde{T}$. The building blocks "projection on virtual viewpoint" and "mapping of texture" comprise simple depth-based warping algorithms, which are described for example in [6]. In the following subsections we will focus on median filtering and the inpainting for this view synthesis scheme.

A. Width of the median filter

The median filter is used to fill occlusions caused by rounding pixels to integer positions after 3D forward warping of the DM. The default median filter size used in the VSRS software is $w_{\mathrm{M}} = 3$. This is intended for synthesizing intermediate views from two reference views and corresponding DMs, but proved unsuitable for predicting the reference views from another as can be seen in Fig. 3. The synthesized view shows many tiny holes when using the lower median filter width. With this synthesized view as a reference picture for inter-view prediction, the encoder decides to not include blocks with holes for prediction. Increasing the size of the median filter fills the holes in the synthesized DM but also introduces some distortion to it, as e.g. slender structures are displayed even more slender. In consequence, the texture view synthesized using this DM does not show holes, except for occlusions caused by the changed perspective but some distortion, which will not be investigated further here. Thus, larger hole-less blocks are available for inter-view prediction. We chose the parameter of the median filter width based on experimental data and set it to $w_{\mathrm{M}} = 9$, as small holes are filled and distortion does not matter for $w_{\mathrm{M}} = 9$ in the case of using the synthesis result for inter-view prediction.

B. Inpainting of occlusions

There are occluded regions in the synthesized view which have a homogeneous texture in the original view. These regions can be filled in order to improve the inter-view prediction. The occluded domain and the contained pixels will be denoted as Ω and p^* ,respectively, in the following. ∇p^*_{n} represents the gradient normal to the boundary. Our approach fills Ω based on the solution of the Laplace equation. So we obtain the boundary value problem for Dirichlet boundaries $\partial\Omega_{\mathrm{D}}$ and Neumann boundaries $\partial\Omega_{\mathrm{N}}$:

$$\begin{aligned} \nabla \cdot (\nabla p^*) &= 0\,, \\ p^*|_{\partial\Omega_{\mathrm{D}}} &= p|_{\partial\Omega_{\mathrm{D}}} \text{ and} \\ \nabla p^*_{\mathrm{n}}|_{\partial\Omega_{\mathrm{N}}} &= 0\,. \end{aligned} \tag{1}$$

Solving (1) results in smooth intensity values based on the data at the boundary $\partial\Omega_{\mathrm{D}}$. In order to find the boundaries of Ω, edges in the inpainting mask are detected by convolution with a Laplace operator. The boundaries are classified into foreground and background based on a simple threshold, which is chosen such that 75% of the farthest pixels are classified as background. The threshold is chosen based on the assumption that most likely the occluded area belongs to the background of the image. Note, that this thresholding value is appropriate for the tested sequences but may not be useful in general. Consequently, background boundaries of the inpainting mask are decided to be treated as Dirichlet boundaries and all other boundaries are treated as Neumann boundaries. An example of the inpainting of occlusions is shown in Fig. 3c. As can be seen from this, the method works well for homogeneous regions, for example the occluded area next to the flower which shows sky in the original.

III. Depth Map Enhancement for View Synthesis Prediction

Depth maps typically exhibit sharp edges and regions which vary smoothly. However, estimated DMs can violate this model especially in homogeneous regions [7]. This leads to a lower consistency of the synthesis result and, in consequence, blocks which are synthesized from low textured areas are not chosen for prediction by the encoder. In the following, we introduce two different methods for depth map enhancement in the context of VSP.

A. A linear Model for Depth Maps

In order to overcome the fact of inaccurate DMs, we set up a linear model for DMs based on the assumption of reliable depth information close to edges in the original DM. This assumption holds due to the fact that edges in the original DM can be seen as feature points for depth estimation and so the estimation error should be small close to edges. We propose to extract the reliable data out of the DM and regenerate the depth map by applying a model which smoothly connects the extracted data. When the resulting DM is used to warp texture from a view to another, larger consistent regions are created than would be obtainable with the original DM. This preprocessing enables the usage of inter-view prediction for

(a) $w_{\mathrm{M}} = 3$ (b) $w_{\mathrm{M}} = 9$ (c) $w_{\mathrm{M}} = 3$ with inpainting

Fig. 3: An example for view synthesis with different median filter width and inpainting. The images show the first frame of BigBuckBunny Flowers sequence (view 32).

(a) original DM (b) linear modelled DM (c) guided image filtered DM ($w = 25, \epsilon = 1000$)

Fig. 4: Illustration of different preprocessing filters for DMs. The first frame of Poznan Blocks DMs are shown (view 5). Note, that the background of filtered DMs appears smoother than in original DM.

these regions, leading to an improved coding performance. The integration of the DM preprocessing operation into the VSP based coding scheme is illustrated in Fig. 1 as the DM filter block. Note that the filter does not affect the actual coded DM but only the DM used for VSP.

Our approach of selecting reliable data in the original DM is based on extracting its edges through a variance filter and a simple threshold operation. This filter acts on a 9×9 window, as it is important to obtain reliable data from both regions, which are separated by the edge. The edges are required to be positioned between pixels. Thus an edge could be described with at least one pixel from each region, the edge separates. If one of the regions is foreground and the other background, this achieves a clear separation of the both. In order to determine whether a pixel belongs to an edge, the window is centered at it. The mean variance in the window is calculated and used as a threshold for the classification of the pixel. If the variance of the center pixel is higher than the threshold, it belongs to an edge. This is expressed in the following equations:

$$p_{\mathrm{f}} = \mathrm{VAR}_{9\times 9}(p) = \mathrm{MEAN}_{9\times 9}(p^2) - \left(\mathrm{MEAN}_{9\times 9}(p)\right)^2 , \quad (2)$$

$$e = \varepsilon\left(p_{\mathrm{f}} - \mathrm{MEAN}_{9\times 9}(p_{\mathrm{f}})\right) , \quad (3)$$

where $\mathrm{VAR}_{9\times 9}(\cdot)$ is the variance in a 9×9 neighborhood centered at pixel p, $\mathrm{MEAN}_{9\times 9}(\cdot)$ is the mean value in the same area and p_{f} the value of a filtered pixel. The function $\varepsilon(\cdot)$ denotes the thresholding operation in form of a per pixel unit step function and e represents an edge flag, which indicates whether a pixel belongs to an edge or not.

From the detected edges, the complete linearly modelled DM is generated by solving the following boundary value problem:

$$\begin{aligned} \nabla \cdot (\nabla Z^*) &= 0 , \\ Z^*(i,j)|_{e(i,j)=1} &= Z(i,j)|_{e(i,j)=1} \text{ and} \\ \nabla Z^*(i,j)|_{p(i,j)\in\partial D} &= 0 . \end{aligned} \quad (4)$$

The variable Z in (4) denotes the distances of the pixels from the camera and relates to the DM D by

$$Z(i,\, j) = \frac{1}{\frac{D(i,j)}{A_{\mathrm{max}}}\left(\frac{1}{Z_{\mathrm{n}}} - \frac{1}{Z_{\mathrm{f}}}\right) + \frac{1}{Z_{\mathrm{f}}}} \quad (5)$$

with Z_{n} and Z_{f} representing near and far clipping planes of the DM, $A_{\mathrm{max}} = 255$ for 8 bit video and (i, j) referring to an integer pixel position. This transformation cannot be treated as optional, because the linear model can only be assumed for distances but not for depth values provided in the DM. Note that this is due to the fact that the mapping between distances and DM values in (5) is non-linear. A possible way to solve (4) is to approximate the divergence operator by a suitable discrete realization such as finite differences. The input to the equation set are the detected DM edges $Z(i,j)|_{e(i,j)=1}$ and the positions of the image borders ∂D. The output is the regenerated DM Z^*. This approach is generic and allows for DM enhancement resulting in perfectly smoothed DMs with preserved edges. As will be seen in the results, enhancing the DM by the model described above leads to a higher coding efficiency.

B. Guided Image Depth Map Filtering

As the previous method, guided image depthmap filtering (GIDMF) aims at obtaining a smoother DM in order to

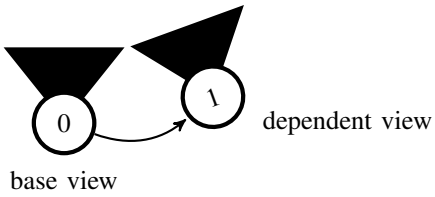

Fig. 5: Inter-view prediction for two non-coplanar views

increase the number of regions available for inter-view prediction. The linear model in (4) is memory and time consuming since it does not act locally on the image. In order to overcome those limitations of the linear model, GIDMF could e.g. be used for DM preprocessing. The method is based on the filter suggested in [8], which is applied on the DM. This filter is similar to the bilateral filter but can be implemented more efficiently and achieves a smoothing of the DM while preserving sharp edges. The filter is modelled by following equation set:

$$\begin{aligned} p_\mathrm{f} &= \mathrm{MEAN}_{w\times w}(a)\cdot g + \mathrm{MEAN}_{w\times w}(b)\,, \\ a &= \frac{\mathrm{COV}_{w\times w}(g,\,p)}{\mathrm{VAR}_{w\times w}(g)+\epsilon} \text{ and} \\ b &= \mathrm{MEAN}_{w\times w}(p) - a\cdot\mathrm{MEAN}_{w\times w}(g)\,. \end{aligned} \tag{6}$$

Here, p denotes a pixel of the input image, while g represents a pixel of the guidance image. The parameter ϵ refers to the degree of smoothing. The mean values are calculated over a square window of the size $w \times w$ centered at pixel p. Due to this model, the output image will only have an edge when the guide image has an edge [8]. For the purpose of DM filtering, the DM is chosen to be both the input and the guidance image. The goal is to smoothen the DM while its edges are preserved. Smoothing large image areas could be achieved by choosing a large filter size w, or applying the filter several times. For this reason, an additional parameter n_It controls the number of applied filter cycles. Preprocessing the DM by this filter results in a higher consistency of the synthesis result and consequently in a higher chance for utilizing inter-view prediction. Fig. 4 illustrates the results of modeling the DM by our approach and guided image filtering with one filter cycle.

IV. Results

The influence of the proposed methods on the coding performance was evaluated on the free navigation video sequences Poznan Blocks (PB), BigBuckBunny Flowers (BBB) and Soccer-Arc 1 (SA1) [4]. Table I shows coded views and encoder settings for the sequences. The used camera arrangement is illustrated in Fig. 5, which is non-coplanar. The base view and one dependent view were encoded. For the encoding of the dependent view, VSP and the previously described methods were applied. The proposed methods for depth map enhancement were applied separately in order to compare the performance. The encoding was using an All Intra configuration. Consequently, no temporal prediction candidates were available. For reference the same views were encoded using the HEVC-3D codec with All Intra configuration. For comparison of coding performance Bjøntegaard Delta

TABLE I: Coded views and encoder settings

	BigBuckBunny	Poznan Blocks	Soccer-Arc 1
base view	45	5	3
dependent view	32	4	4
coded frames	120	250	250
QPs (T/D)	37/43	30/39	30/39
	40/45	35/42	37/43
	44/47	40/45	44/47
	47/50	45/48	47/50

TABLE II: BD measurements on dependent texture and dependent depth for VSP with linear modelled DMs

	BigBuckBunny		Poznan Blocks		Soccer-Arc 1	
	rate	PSNR	rate	PSNR	rate	PSNR
texture	−5.3 %	0.23 dB	−8.9 %	0.36 dB	−8.8 %	0.32 dB
depth	−3.3 %	0.19 dB	−0.32 %	0.01 dB	−4.0 %	0.33 dB

(BD) was measured according to [9], which gives a measure in terms of average bit rate savings and average Peak Signal to Noise Ratio (PSNR) improvements.

A. VSP using varying median filter sizes and inpainting

Measuring the influence of increased median filter size and inpainting on the performance of VSP, we get the results shown in Fig. 6. The results show increasing bit rate savings for increasing sizes of the median filter and the use of inpainting. By these modifications the bit rate savings in case of e.g. BBB more than double compared to applying VSP without modifying parameters and algorithms in VSRS. From Fig. 6 it also becomes clear that a median filter size of $w_\mathrm{M} = 9$ is a reasonable choice. Note that this choice only holds for tested sequences and may be investigated further in the future.

B. VSP using linear modelling of depth maps

Table II shows the BD statistics for all tested sequences and the use of the proposed linear model as a DM preprocessing step in the VSP scheme. For all simulations the median filter

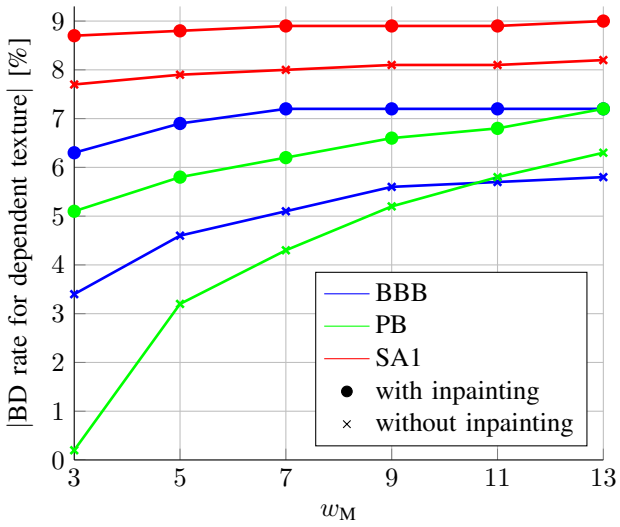

Fig. 6: BD rate (absolute value) of the dependent texture for VSP without inpainting and with inpainting

TABLE III: BD measurements on dependent texture and dependent depth for VSP with GIDMF

	BigBuckBunny		Poznan Blocks		Soccer-Arc 1	
	rate	PSNR	rate	PSNR	rate	PSNR
texture	$-8.3\,\%$	$0.37\,\mathrm{dB}$	$-10.3\,\%$	$0.41\,\mathrm{dB}$	$-6.0\,\%$	$0.23\,\mathrm{dB}$
depth	$-3.6\,\%$	$0.21\,\mathrm{dB}$	$-2.1\,\%$	$0.09\,\mathrm{dB}$	$-3.7\,\%$	$0.31\,\mathrm{dB}$

TABLE IV: BD for total rate, I: inpainting, LM: linear DM model, GIDMF with $w = 25$, $\epsilon = 5$, $n_{\mathrm{It}} = 25$

tools	BigBuckBunny		Poznan Blocks		Soccer-Arc 1	
	rate	PSNR	rate	PSNR	rate	PSNR
	$-2.1\,\%$	$0.10\,\mathrm{dB}$	$-1.5\,\%$	$0.07\,\mathrm{dB}$	$-3.9\,\%$	$0.15\,\mathrm{dB}$
I	$-2.7\,\%$	$0.13\,\mathrm{dB}$	$-2.0\,\%$	$0.09\,\mathrm{dB}$	$-4.3\,\%$	$0.17\,\mathrm{dB}$
I and LM	$-2.0\,\%$	$0.10\,\mathrm{dB}$	$-2.8\,\%$	$0.12\,\mathrm{dB}$	$-4.2\,\%$	$0.16\,\mathrm{dB}$
I and GIDMF	$-3.1\,\%$	$0.15\,\mathrm{dB}$	$-3.1\,\%$	$0.14\,\mathrm{dB}$	$-2.9\,\%$	$0.11\,\mathrm{dB}$

size was set to $w_{\mathrm{M}} = 9$ and occluded areas after view synthesis were inpainted. Although bit rate savings decrease for BBB compared to results shown in Fig. 6, VSP with linear modelled depth maps outperforms the HEVC-3D anchor. For SA1 our approach of DM preprocessing does not lead to worse coding performance and in the case of PB our approach clearly outperforms VSP with the original DM. So in conclusion, the proposed DM enhancement overcomes inaccuracies within DMs and the assumption of linear behaviour of DMs in homogeneous regions is feasible, as for every sequence bit rate savings at the same quality are observable.

C. VSP using GIDMF

Simulations using GIDMF were done for all test sequences using ($w = 25, \epsilon = 5, n_{\mathrm{It}} = 25$) as parametrization of the filter. The median filter width used for view synthesis was kept fixed at a value of $w_{\mathrm{M}} = 9$. Occlusions in the virtual views were filled with inpainting. Table III shows the results of the BD against the reference coded with the HEVC-3D codec. Note that in these results, sequence dependent (and most likely non-optimal) parameter settings for the filtering process were used. However, using GIDMF gave up to $10.3\,\%$ of rate savings for coding of the dependent texture but further investigations into choosing the parameters optimally and automatically are required. In conclusion, this investigation shows that using GIDMF as a preprocessing step for DMs is appropriate for VSP if parameters are chosen with care. A detailed performance analysis which compares the linear model and GIDMF is not part of this paper, but it can be stated that the complexity of the DM preprocessing could be reduced by the use of GIDMF instead of the linear model as the filter is naturally of type $\mathcal{O}(N)$ [8], while solving (4) e.g. with Gaussian elimination costs $\mathcal{O}(N^3)$ operations.

D. Overall coding results

The overall coding performance of the proposed VSP scheme in terms of total bit rate savings and average PSNR gain of the two views is depicted in Table IV. Different tools or combinations of them were used as indicated. Note that the methods of linear DM modelling and GIDMF were not used in combination of each other, as they rely on different concepts. The width of the median filter was fixed to $w_{\mathrm{M}} = 9$. The rate refers to the total rate needed for the encoding of both textures and DMs. PSNR was measured and averaged on the textures only. The best results were achieved for SA1 sequence, where up to $4.3\,\%$ of rate could be saved without diminishing the quality of the encoded textures.

V. Conclusion

In this paper we present a coding scheme utilizing VSP for non-coplanar 3D video sequences. For this purpose, we propose improvements to state of the art view synthesis algorithms. Moreover, we introduce a linear model, which enables a preprocessing for DMs. This leads to additional coding gain in the case of inaccurate DMs. Investigations on the guided image filter as another method of DM preprocessing are detailed.

The experimental results of the proposed coding scheme show improvements relative to the 3D extension of the HEVC standard. The proposed linear model for DM enhancement leads to better coding performance using VSP and consequently outperforms VSP with original DMs. Even better results were obtained using GIDMF as preprocessing stage for DMs leading to a less complex implementation, where however parametrization of the filter still is an issue. Finding parameters which allow for a high coding gain and acceptable run time will be of major interest in further research activities. However, the presented results show clearly that GIDMF can outperform the method of linear modelling and is superior in terms of flexibility and complexity. Future research directions should concentrate on low complex implementations of the proposed concepts. The results we presented show clearly that DM enhancement techniques are essential for VSP in the general case of estimated DMs, as they introduce a more efficient coding.

References

[1] *Advanced video coding for generic audiovisual services*, Recommendation ITU-T H.264, ITU-T Std. Recommendation ITU-T H.264, Feb. 2014.

[2] *High efficiency video coding*, Recommendation ITU-T H.265, ITU-T Std. Recommendation ITU-T H.265, Apr. 2015.

[3] K. Müller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D high-efficiency video coding for multi-view video and depth data," *Image Processing, IEEE Transactions on*, vol. 22, no. 9, pp. 3366–3378, Sep. 2013.

[4] G. Lafruit, K. Wegner, and M. Tanimoto, "Call for evidence on free-viewpoint television: Super-multiview and free navigation," ISO/IEC JTC1/SC29/WG11, 112th Meeting, Warsaw, Tech. Rep. n15348, Jun. 2015.

[5] Y. Chen, G. Tech, K. Wegner, and S. Yea, "Test model 10 of 3D-HEVC and MV-HEVC," JCT-3V, 110th Meeting, Strasbourg, Tech. Rep. JCT3V-J1003, Oct. 2014.

[6] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[7] F. Garcia, D. Aouada, H. K. Abdella, T. Solignac, B. Mirbach, and B. Ottersten, "Depth enhancement by fusion for passive and active sensing," in *Computer Vision–ECCV 2012. Workshops and Demonstrations*. Springer, 2012, pp. 506–515.

[8] K. He, J. Sun, and X. Tang, "Guided image filtering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[9] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," ITU-T SG16/Q6 VCEG, Austin, USA, Tech. Rep. Doc. VCEG-M33, 2001.