

Investigation and Evaluation of New Filtering Approach for Efficient Video Compression

Ms. Jyoti M. Pingalkar(Lecturer)

Asst. Prof. P.B. Kumbharkar. (Head Of Computer Engineering Department.)

Siddhant College Of Engineering, Pune

Dr.T.R.Sontakke(Principal)

Alard College of Engineering,pune

I. ABSTRACT

As we know that for H.264/AVC standardization the use of in-loop filters was introduced in order to improve the objective and subjective quality of video compression through noise reduction. The in-loop filters such as deblocking filter and adaptive loop filter are currently introduced for the HEVC standardization. However these both filters for HEVC only work under the spatial domain despite the fact of temporal correlation within video sequences. Thus to overcome this problem, in this paper we are presenting new filter which works efficiently with both the domains such as spatial and temporal. In addition to this, the investigation method is based on depth compression technique. This proposed filter introduced with aim of delivering most efficient decoding of video streaming. In this filter, small overhead requires at slice level during the use of temporal information conveyed in the bit stream to reconstruct the individual motion trajectory of every pixel in a frame at both encoder and decoder. Afterward, this information is used for pixel-wise adaptive motion-compensated temporal filtering. Implementation and evaluation study for proposed approach is done using the JAVA frameworks.

Index Terms— H.264/AVC, HEVC, Video Compression, in-loop filter, noise, video frames, decoder, encoder.

II. INTRODUCTION

3D video has gained more attentions as one of the most dominant video formats with a variety of applications such as (3DTV) 3-dimensional TV or (FTV) free view point TV, with the recent development of 3D multimedia/display technologies and the increasing demand for realistic multimedia. FTV can give users the freedom of selections viewpoint, different from conventional TV where the viewpoint is determined by an acquisition camera. 3DTV aims to provide users with 3D depth perception by rendering two and more views on stereoscopic and auto-stereoscopic 3D display [9]. Synthesizing multiple views provided 3D free

view video can also be at the selected viewpoint according to a user preference. Many technical issues should be resolved for successful development of 3D video systems, e.g. capturing and analyzing the stereo or multi view images, transmitting the data and compressing, and rendering multiple images on various 3D displays.

The main challenging issues of FTV and 3DTV are depth estimation, 3D video coding and virtual view synthesis. The virtual view at the receiver side depth maps are used to synthesize, so accurate depth maps should be estimated in an efficient manner for ensuring a seamless view synthesis. Since the performance of 3DTV and FTV heavily depends on the number of multiple views, virtual view synthesis is an important technique in 3D video systems as well. In other words, synthesizing virtual view with the limited number of original views leads to reducing the cost and bandwidth required to capture a number of viewpoints and to transmit the huge amount of data over network [10] [11].

In this paper we are basically presenting the wide study over the depth map used for video compression in section II at first, and then after the TTF (temporal trajectory filter) is investigated in the environment of the HM. In addition, it is shown how the length of individual pixel trajectories can be increased to achieve better filtering performance. The remainder of the paper is structured as follows. Section III presents the literature study over mathematical model for the noise encountered along a pixel trajectory, if the trajectory is completely known. Based on this model, optimal filter coefficients are calculated. In Section IV, three previously described thresholds are revisited that allow the calculation of pixel trajectories directly from the motion vectors conveyed in the bit stream [9].

III. RELATED WORK AND MOTIVATION

The MPEG-FTV division, an ad hoc group of MPEG, has made a new standard for FTV. Also, a multi view-plus-depth data format was proposed to make the 3D video systems more flexible. For 3D video coding, (JVT) Joint Video Team from the (MPEG) Moving Pictures Experts Group of ISO/IEC and the (VCEG) Video Coding Experts

Group of ITU-T jointly standardized (MVC) multi view video coding with an extension of H.264/AVC standard. Depth maps are encoded with the corresponding color images together and pre-calculated at the transmitter side. A number of methods have been proposed to efficiently compress the multi view video using view redundancies; the depth video coding has not been studied extensively while at the receiver side, the decoded multi view-plus-depth data are utilized to synthesize the virtual view. Depth video coding aims to reduce a depth bit rate as much as possible while ensuring the quality of the synthesized view. Thus, not the depth map itself but its performance is determined by the quality of the synthesized view. In general, the depth map contains a per-pixel distance between object and camera, usually represented by 8-bit grayscale value. The depth map has unique characteristics are following:

1) Depth value varies smoothly except object boundaries or edges, 2) Edges of the depth map usually coincide with those of the corresponding color image, and 3) Object boundaries should be preserved in order to provide the high-quality synthesized view. Thus, which ultimately affect the synthesized view quality? The straightforward compression of the depth video using the existing video coding standards such as H.264/AVC may cause serious coding artifacts along the depth discontinuities [2].

Morvan et al. proposed a platelet-based method that models depth maps by estimating piecewise-linear functions in the sub-divisions of quad tree with variable sizes under a global rate-distortion constraint. Depth video coding can be classified into two categories according to coding algorithms: post-processing based coding and transform based coding. They show the proposed method out performs JPEG-2000 encoder with a 1-3 dB gain [1].

These methods have better performance than the existing image compression methods, they are difficult to be extended into video domain for exploiting temporal redundancies, and are not compatible with the conventional video coding standards such as H.264/AVC. New intra prediction in H.264/AVC was proposed to encode depth maps by designing an edge-aware intra prediction scheme that can reduce a prediction error in macro blocks. And Maitre and Do proposed a depth compression method based on a shape-adaptive wavelet transform by generating small wavelet coefficients along depth edges [1] [9] [10]. These are different from the platelet or wavelet based coding methods; this scheme can be easily integrated with H.264/AVC. However, the performance of the coding algorithm was evaluated by measuring the depth map itself, but not by the synthesized view.

Depth video coding algorithms have moved interest on reducing compression artifacts that may exist on depth video which is encoded by H.264/AVC in order to meet the compatibility to the advanced H.264/AVC standard. Kim et al. proposed a new distortion metric that considers global video characteristics and camera parameters, and then used

the metric in the rate-distortion optimized mode selection to quantify the effects of depth video compression on the synthesized view quality [12]. Lai et al. showed that a rendering error in the synthesized view is a monotonic function of the coding error, and presented a method to suppress compression artifacts using a sparsity-based de-artifacting filter [11]. The work proposed in exploited an adaptive depth map upsampling algorithm with a corresponding color image in order to obtain coding gain while maintaining the quality of the synthesized view. Oh *et al.* proposed a new coding scheme based on a depth boundary reconstruction filter which considers occurrence frequency, similarity, and closeness of pixel. Liu *et al.* utilized a trilateral filter, which is a variant of bilateral filter, as an in-loop filter in H.264/AVC and a sparse dyadic mode as an intra-mode to reconstruct depth map with sparse representations. Some approaches have been proposed to encode a downsampled depth map and to use a special upsampling filter after decoding to recover the depth edge information.

Author of [1] utilize a (WMF) weighted mode filtering for the post-processing of the compressed depth map, which was proposed to enhance the depth video obtained from depth sensors such as (ToF) Time-of-Flight camera. A joint histogram is generated by first calculating the weight based on spatial and range kernels and then counting each bin on the histogram of the depth map given by the input noisy depth map. Final solution is obtained by seeking a mode with is the maximum value on the histogram. In addition this, author introduce the concept the depth image compression of the weighted mode filtering in generic formulation tailored. Also describe the relation with the bilateral and trilateral filtering methods, which have been used in depth video coding, and show the effectiveness of the proposed method with a variety of experiments. The main contributions of this work as follows.

- Analyze the relation between the WMF and the existing approaches in a localized histogram framework to justify its superior performance in the perspective of robust estimation theoretically.
- Evaluate the effectiveness of the WMF in the depth coding context, where the noise characteristics and the objective measure are different from thoroughly.
- Utilize the WMF in various proposed schemes for depth coding by considering important depth properties for a better synthesized view quality effectively.

In paper [1], using the framework of a conventional video codec author propose a novel scheme that compresses the depth video efficiently. In particular, an efficient post-processing method for the compressed depth map is proposed in a generalized framework, which considers compression artifacts, dynamic range of the depth data and spatial resolution. Proposed post-processing method utilizes

additional guided information from the corresponding color video to reconstruct the depth map while preserving the original depth edge. As an in-loop filter depth video is encoded by a typical transform-based motion compensated video encoder, and compression artifacts are addressed by utilizing the post-processing method. In addition, author design a down/upsampling coding approach for both the spatial resolution and the dynamic range of the depth data. Here basic idea is to reduce the bit rate by encoding the depth data on the reduced spatial resolution and depth dynamic range. Reconstruct the depth video this proposed post-processing filter is then utilized to efficiently.

IV. LITERATURE STUDY OF THEORITICAL BASICS

For any given pixel $(x_0, y_0)^T$ in frame j it is assumed that its locations $(x_i, y_i)^T$, $1 \leq i < N$, in $N-1$ previous frames are also known. If $Y_n(x, y)$ denotes the luminance component of frame n at location $(x, y)^T$ the distorted versions of the original sample $Y_j(x_0, y_0)$ in any of the $N-1$ previous frames $Y_{j-i}(x_i, y_i) = Y_j(x_0, y_0) + n_i$, $1 \leq i < N$ (1)

Even the motion of the pixel is perfectly known, a noise term n_i with variance σ_i is introduced due to the reduced quality of the encoded sequence. As described, it can be assumed that all n_i are uncorrelated. A filtered version of the original luma component can compute by calculating a weighted mean [1]

$$\begin{aligned} Y_j^*(x_0, y_0) &= \frac{1}{N} \sum_{i=0}^{N-1} \beta_i \cdot \hat{Y}_{j-i}(x_i, y_i) \\ &= \frac{1}{N} \sum_{i=0}^{N-1} \beta_i \cdot Y_j(x_0, y_0) + \frac{1}{N} \sum_{i=0}^{N-1} \beta_i \cdot n_i \quad (2) \\ &= \frac{1}{N} Y_j(x_0, y_0) \sum_{i=0}^{N-1} \beta_i + \frac{1}{N} \sum_{i=0}^{N-1} \beta_i \cdot n_i. \end{aligned}$$

Where β_i are the individual weights per frame with $\sum_{i=0}^{N-1} \beta_i = N$ to make the filter unbiased. This leads to the definition of a new noise term \tilde{n}_j for the filtered pixel.

$$Y_j^*(x_0, y_0) = Y_j(x_0, y_0) + \underbrace{\frac{1}{N} \sum_{i=0}^{N-1} \beta_i \cdot n_i}_{\tilde{n}_j} \quad (3)$$

The variance of the filtered noise \tilde{n}_j is subsequently given by

$$\begin{aligned} \sigma_{\tilde{n}}^2 &= E[\tilde{n}_j \tilde{n}_j] \\ &= E \left[\frac{1}{N^2} \sum_{l=0}^{N-1} \beta_l n_l \cdot \sum_{k=0}^{N-1} \beta_k n_k \right] = \frac{1}{N^2} \sum_{m=0}^{N-1} \beta_m^2 \sigma_m^2. \end{aligned} \quad (4)$$

As the filter is to minimize $\sigma_{\tilde{n}}^2$ constraint to $\sum_{i=0}^{N-1} \beta_i = N$, the minimum may be found by Lagrangian minimization:

$$\frac{\partial}{\partial \beta_i} \left[\sum_{m=0}^{N-1} \beta_m^2 \sigma_m^2 - \lambda \left(N - \sum_{k=0}^{N-1} \beta_k \right) \right] = 0 \quad (5)$$

$$\begin{aligned} 2\beta_i \sigma_i^2 + \lambda &= 0 \Rightarrow \beta_i = -\frac{\lambda}{2\sigma_i^2} \\ \sum_{k=0}^{N-1} \beta_k &= N \Leftrightarrow \sum_{k=0}^{N-1} -\frac{\lambda}{2\sigma_k^2} = N \\ \lambda &= \frac{N}{\sum_{k=0}^{N-1} -\frac{1}{2\sigma_k^2}} \\ \beta_i &= -\frac{\lambda}{2\sigma_i^2} = \frac{N/\sigma_i^2}{\sum_{k=0}^{N-1} \frac{1}{\sigma_k^2}}. \end{aligned} \quad (6)$$

The reconstruction error variance σ_k^2 for each pixel along with the trajectory would be required to calculate the optimal filter weight β_i . According to Wiegand and Girod the distortion variance in a reconstructed frame is given [1]:

$$D_{\text{REC}} = \frac{Q_{\text{step}}^2}{3} \text{ with zero mean.} \quad (7)$$

Where Q_{step} is the quantizer step size selected by the quantization parameter QP . Both in H.264/AVC and in HEVC, Q_{step} is roughly:

$$Q_{\text{step}} = 0.625 \cdot 2^{\frac{QP}{6}}. \quad (8)$$

Subsequently, the optimal filter weight for frame i according to its QP may be calculated

$$\sigma_i^2 = D_{REC,i} = \frac{1}{3} \left(0.625 \cdot 2^{\frac{QP_i}{6}} \right)^2$$

$$\beta_i(QP_i) = 3 \cdot \left(0.625 \cdot 2^{\frac{QP_i}{6}} \right)^{-2} \quad (9)$$

V. INVESTIGATED FILTER DESIGN

FILTER DESIGN

Low-delay high efficiency setting of IBBB coding structure and HEVC with an every is B-predicted block. There are two motion vectors pointing to one of the last four encoded pictures. Assumed that the motion vector for a given block and also describes the individual motion of each pixel [4].

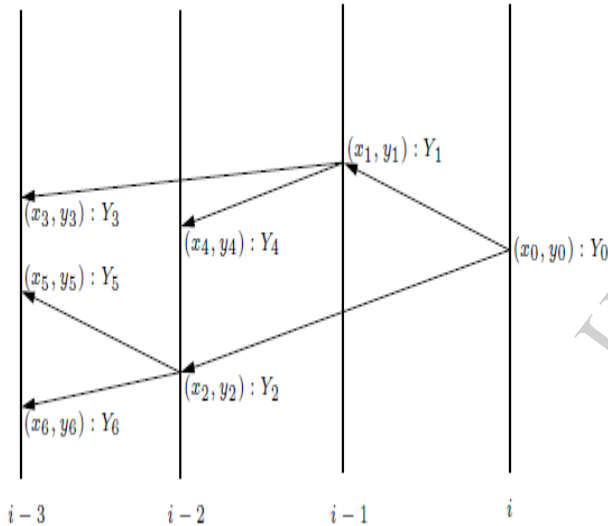


Figure 1

In Fig. 1. Starting at a pixel (x_0, y_0) with luminance Y_0 in an arbitrary B-frame i , possible trajectory locations derived through the concatenation of motion vectors pointing to previously encoded B-frames.

Block. The components of the two resulting motion vector fields for frame i shall be denoted by $(dx_{i,0}, dy_{i,0})^T$ and $(dx_{i,1}, dy_{i,1})^T$.

Starting again with pixel $(x_0, y_0)^T$ in frame i , two possible locations of the pixel in the referenced frames are therefore given by

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} dx_{i,0}([x_0], [y_0]) \\ dy_{i,0}([x_0], [y_0]) \end{pmatrix}, \quad (10)$$

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} dx_{i,1}([x_0], [y_0]) \\ dy_{i,1}([x_0], [y_0]) \end{pmatrix}.$$

In Figure 1 show how the concatenation of motion vectors is used to derive possible pixel locations over a GOP of four frames. Not all of these describe the true motion of the pixel. It is necessary, to discard those motion vectors that have purely been chosen due to rate distortion Optimization and thus May not relate to the true motion of pixels [5]. To this end three thresholds are used. In each of the following equations the motion vectors are scaled according to the temporal distance that they span.

A. Absolute Error along the Trajectory

For every pixel the absolute difference of two consecutive luminance samples $\Delta Y_i = Y_{i+1} - Y_i$ together with the respective chrominance differences ΔU_i and ΔV_i are calculated. A sudden change in one of these differences is assumed to indicate that a motion vector no longer describes the true motion of a pixel. The trajectory is only continued, if

$$\Delta Y_i < T, \Delta U_i < T, \Delta V_i < T, T = \begin{cases} 2T_Y, QP < 30 \\ 4T_Y, QP \geq 30 \end{cases} \quad (11)$$

for a given threshold $T_Y, 0 \leq T_Y \leq 7$.

B. Temporal Motion Consistency

In addition, Consecutive motion vectors are tested in the similarity. A trajectory is expected to be correct as long as its motion does not change significantly over time. When examining a new motion vector for list 0 for any given pixel of the trajectory, its Euclidean distance to the vector pointing to

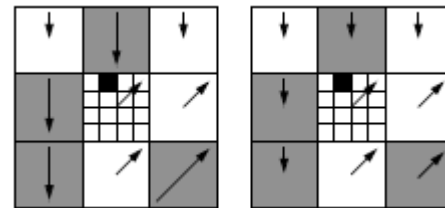


Figure 2

Fig. 2. On the left the original motion vectors for all 4x4 blocks surrounding the trajectory's current location are shown. Vectors that span a temporal distance greater than 1 are marked in gray. After the scaling (right) the BV-metric (i.e. the number of vectors differing from the current one) for the current block is 5

current location is calculated. The trajectory is continued, as long as.

$$\sqrt{(dx_{r0} - dx_i)^2 + (dy_{r0} - dy_i)^2} \leq T_{TC} \quad (12)$$

With $0 \leq TTC \leq 7$ in quarter-pel, where dx_{r0} and dy_{r0} are the components of the motion vector for list 0 pointing from the current frame to reference frame r0. The temporal motion consistency is also checked for the vectors of reference list 1.

C. Spatial Motion Consistency

At each frame the motion vector for a pixel of the trajectory is compared with its eight neighbors on 4×4 block level. In this context the block-vote metric BV denotes the number of neighboring motion vectors that differ significantly from the current one. A side from a temporal comparison of motion vectors, spatial similarity is also examined, which is a measure for the reliability of a motion vector [2]. The allowed maximum difference is 30% of the original motion vector's length or at least 0.3 quarter-pel. Both values were chosen empirically and proved to be well suited for all sequences. In the scaling of motion vectors and the block-vote metric are illustrated by Figure 2. The filtering along the trajectory is continued only, if the block-vote metric for the current pixel satisfies

$$BV_{i,0}(x, y) \leq T_{SC}, 0 < T_{SC} \leq 8. \quad (13)$$

D. Long Trajectories

Previous implementation described in, a trajectory was interrupted as soon as the luminance difference became bigger than the threshold T_Y . This makes the coder biased towards shorter trajectories. In theory, the quality of the filtering process is increased with the number of samples used. The new design, therefore, does not stop the trajectory formation all together, but simply omits the luma sample in question from the filtering process and continues the trajectory. For the threshold T_Y the last filtered luminance samples and the respective chrominance samples are now used for the calculation of $\Delta Y_i, \Delta U_i,$ and ΔV_i .

E. Parameter Calculation

The parameter combination the minimum mean square error is selected. In which all possible parameter combinations can be tested simultaneously at the encoder. Each of the thresholds is transmitted to the decoder requiring 9 additional bits per frame. A tenth bit can be used to disable the filter for the current frame all together, in which case the other thresholds are simply omitted.

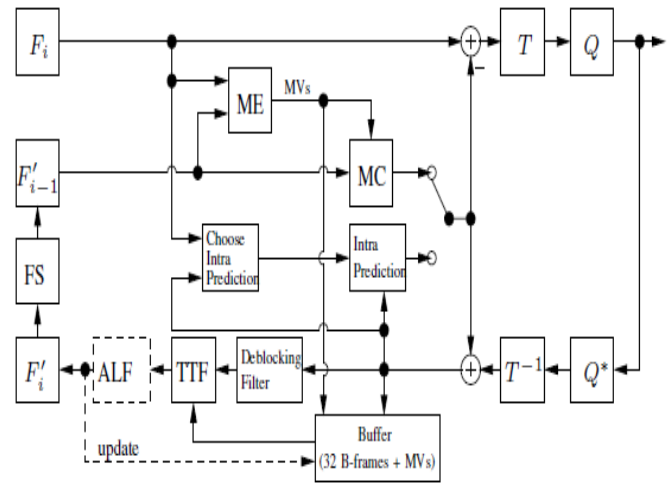


Figure 3

In Fig. 3, The TTF is included in the local decoder loop of the encoder after the deblocking filter. Both ALF and TTF are used together, for the first test the ALF was disabled, the respective frame in the TTF's buffer is updated after the ALF has been applied.

VI. WORK DONE

In [2], the evaluation of this proposed TTF method has been carried out. During the practical experiment of this method using JAVA, the TTF is integrated into the HEVC test model HM 3.0. Up to 32 previously decoded unfiltered frames are kept in a buffer to be used for the trajectory formation. Figure 3, above showing the practical design of this approach. In this setting the ALF (dotted connections) was disabled. Tests have been conducted for a variety of sequences listed in Table I below. The exact configuration for the low-delay high efficiency setting may be found in [3]. From [2], results showing that TTF outperforms the existing method. The average BD-rate for the simplified filter is only -0.9%, which provides evidence for the effectiveness of the weighted filtering.

TABLE I
BD-RATES AND AVERAGE PSNR-GAIN FOR THE SEQUENCES USED IN THE EXPERIMENTS

Sequence	Resolution, framerate in Hz	TTF vs HEVC		ALF vs HEVC		TTF+ALF vs HEVC		TTF+ALF vs HEVC+ALF		TTF+ALF simple average	
		Δ PSNR in dB	BD-rate in %	Δ PSNR in dB	BD-rate in %	Δ PSNR in dB	BD-rate in %	Δ PSNR in dB	BD-rate in %	Δ PSNR in dB	BD-rate in %
BlowingBubbles	416x240, 30	0.03	-0.70	0.05	-1.29	0.08	-1.92	0.03	-0.64	0.02	-0.42
BQSquare	416x240, 60	0.18	-4.77	0.21	-5.71	0.44	-11.48	0.23	-6.06	0.08	-2.21
RaceHorses	416x240, 30	0.00	-0.05	0.05	-0.99	0.00	-0.05	0.01	-0.13	0.01	-0.12
PartyScene	832x480, 50	0.02	-0.53	0.13	-3.13	0.15	-3.53	0.02	-0.42	0.01	-0.17
Vidyo1	1280x720, 60	0.02	-0.60	0.17	-4.95	0.17	-4.79	0.00	0.18	0.01	-0.35
Vidyo3	1280x720, 60	0.00	0.07	0.37	-10.38	0.38	-11.09	0.01	-0.27	0.00	-0.15
Vidyo4	1280x720, 60	0.01	-0.30	0.17	-5.42	0.18	-5.80	0.01	-0.38	0.01	-0.33
BQTerrace	1920x1080, 60	0.02	-1.46	0.17	-9.73	0.19	-10.41	0.01	-0.74	0.00	-0.21
ParkScene	1920x1080, 24	0.01	-0.38	0.07	-2.07	0.07	-2.29	0.01	-0.21	0.01	-0.11
Allstars	704x576, 25	0.01	-0.25	0.25	-7.46	0.25	-7.66	0.01	-0.21	0.01	-0.26
BBCpan-13	720x576, 25	0.00	-0.11	0.21	-6.53	0.20	-6.41	0.00	0.11	0.00	0.10
Waterfall	704x480, 25	0.26	-8.95	0.29	-9.40	0.51	-16.50	0.23	-7.61	0.20	-6.83
Average		0.05	-1.52	0.18	-5.60	0.22	-6.83	0.05	-1.37	0.03	-0.92

VII. CONCLUSION AND FUTURE WORK

As in this research study we investigated the best approach for improved video compression, however in literature we studied the one more approach which is resulting into the better improvement for video compression, this approach is called as depth video coding. So along with the use of TTF, in future we can also use the depth coding technique for more improved results for video compression. From the work done section over TTF, it shows that this filter produces an average BD-rate of -1.4% when included in the HEVC test model. Additional improvements may be achieved by further investigating both long trajectories and weighted averaging separately. Thus our further work will be focusing over possible interactions between in-loop filtering approaches, TTF encode complexity reduction and use of depth map coding technique.

VIII. REFERENCE

- [1] "Efficient Techniques for Depth Video Compression Using Weighted Mode Filtering", Viet-Anh Nguyen, Dongbo Min, *Member, IEEE*, and Minh N. Do, *Senior Member, IEEE*, 2011.
- [2] "Weighted Temporal Long Trajectory Filtering for Video Compression", Marko Esche, Alexander Glantz, Andreas Krutz, Michael Tok, and Thomas Sikora, 2012.
- [3] T. Chujoh, N. Wada, T. Watanabe, G. Yasuda, and T. Yamakage, "Specification and experimental results of quadtree-based adaptive loop filter," ITU-T SG16/Q.6 VCEG document VCEG-AK22, Apr 2009.
- [4] A. Golwelkar and J. Woods, "Motion-compensated temporal filtering and motion vector coding using biorthogonal filters," *IEEE TCSVT*, vol. 17, no. 4, pp. 417–428, April 2007.
- [5] A. Glantz, A. Krutz, M. Haller, and T. Sikora, "Video coding using global motion temporal filtering," *Proceedings of the 16th International Conference on Image Processing (ICIP)*, pp. 1053–1056, Nov 2009.
- [6] M. Esche, A. Krutz, A. Glantz, and T. Sikora, "A novel in-loop filter for video-compression based on temporal pixel trajectories," *Proceedings of the 26th PCS*, pp. 514–517, Dec 2010.
- [7] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," *Proceedings of the International Conference on Image Processing (ICIP)*, vol. 3, pp. 542–545, 2001.

[8] G. Bjøntegaard, "Calculation of average PSNR differences between RDcurves," ITU-T SG16/Q.6 VCEG document VCEG-M33, Mar 2001.

[9] MPEG document, N9760, Text of ISO/IEC 14496-10:2008/FDAM "Multiview Video Coding," Oct. 2008, Busan, Korea.

[10] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, and A. Koz, "Coding Algorithms for 3DTV-A Survey," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1606-1621, 2007.

[11] MPEG document, N9992, "Results of 3D Video Expert Viewing," Jul. 2008, Hannover, Germany.

[12] MPEG document, w11061, "Applications and requirements on 3D video coding," MPEG, Xi'an, China, Oct. 2009.