INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2011/ M22697
November 2011, Geneva, Switzerland

Source	**Poznań University of Technology,** **Chair of Multimedia Telecommunications and Microelectronics, Poznań, Poland**
Status	**Contribution to the 96th MPEG Meeting**
Title	**Technical Desciption of Poznan University of Technology proposal for Call on 3D Video Coding Technology**
Author	**Marek Domański** (domanski@et.put.poznan.pl), Tomasz Grajek (tgrajek@multimedia.edu.pl), Damian Karwowski (dkarwow@multimedia.edu.pl), Krzysztof Klimaszewski (kklima@et.put.poznan.pl), Jacek Konieczny (jkonieczny@multimedia.edu.pl), Maciej Kurc (mkurc@multimedia.edu.pl), Adam Łuczak (aluczak@multimedia.edu.pl), Robert Ratajczak (rratajczak@multimedia.edu.pl), Jakub Siast (jstast@multimedia.edu.pl), **Olgierd Stankiewicz** (ostank@multimedia.edu.pl), Jakub Stankowski (jstankowski@multimedia.edu.pl), **Krzysztof Wegner** (kwegner@multimedia.edu.pl)

Table of contents

1 Introduction

This documents presents a technical description of compression technology prepared at Poznan University of Technology in response to Call for Proposals on 3D Video Coding Technology [1]. The proposed technology is HEVC-based and the bitstream of the base view is HEVC-compatible. In the codec implementation for view synthesis standard VSRS has been used.

2 Overview

The proposed technology is HEVC-compatible. One of the views is coded in HEVC syntax (texture only) while for remaining data (texture and depth) additional syntax structures have been proposed. For both texture and depth hierarchical view coding structure similar to MVC is used: the already coded views are used as references for prediction of the subsequent views. There are three main inter-view prediction types: view-synthesis (DIBR-based), disparity-compensation (MVC-like), and depth-based motion prediction (DBMP).

The main idea of the proposed coding technology is to exploit view-synthesis prediction as much as possible. The base view (HEVC-compatible view) and its depth are coded directly i.e. without any inter-view prediction. The side views (textures and depths) are synthesized using the base view as a reference. Then, in the side views, disoccluded regions (hidden in the base view) are identified. Only the disoccluded regions from the side views are coded. Coding of the side views takes advantage of other inter-view prediction modes: disparity compensation and DBMP.

The cameras parameters are compressed and transmitted in SEI messages in a single bitstream along with the video.

2.1 Coder overview

The block scheme of an encoder is shown in Fig. 1. The input data format (multi-view video plus depth representation) is converted to the single-view plus disocclusion representation. Then, converted data is encoded with the use of five sub-encoders that produce separate sub-streams. These encoders cooperate by mutually providing the inter-view prediction data and the camera information. The resultant sub-streams are finally multiplexed and encapsulated into base view bitstream in order to produce the output bitstream.

There are five sub encoders:

- HEVC-compatible encoder, used for coding of a single base view (Section 3.3),
- Camera parameters encoder (section 3.4),
- HEVC-based depth encoder, used for coding of depth data (Section 3.5),
- HEVC-based texture encoder, used for coding of side views (Section 3.6),
- HEVC-based residual encoder, used for coding of residual layer (Section 3.7).

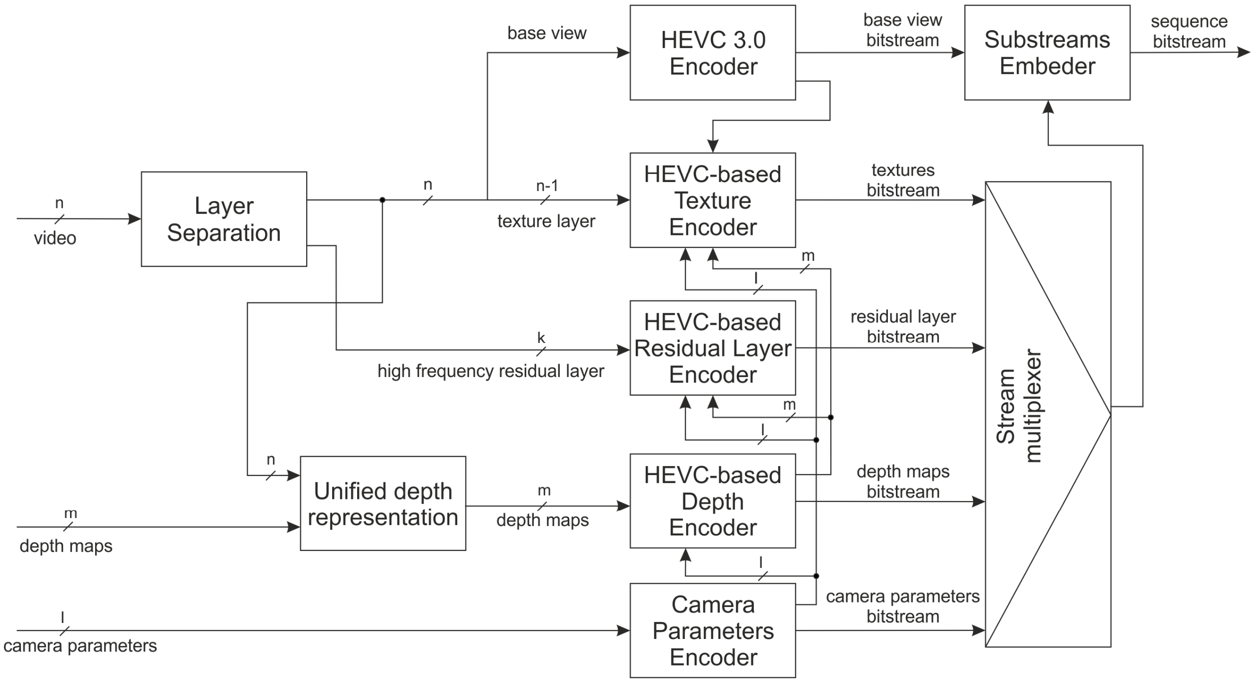


Fig. 1. Scheme of the proposed coder.

The number of coded texture views *n* can be different from the number of coded depth maps *m*. Also the number of coded camera parameters sets *l* can be different from the number of coded texture views *n*, and depth maps *m*. The number of coded texture views *n*, depth maps *m*, camera parameters sets *l* and residuals *k* are independently settable in the coder configuration file.

2.2 Decoder overview

The 3D video bitstream consists of 5 sub-streams. Each sub-stream represents one of the following types of data:

- texture of base view, compatible with HEVC syntax, contains sequence and picture parameter sets etc.,
- textures of the side views,
- depth maps corresponding to individual views,
- residual layer of individual views,
- camera parameters.

Each of the sub-streams can be independently extracted from the 3D video bitstream.

Decoder consists of a sub-stream extractor, a demultiplexer, and 5 sub-decoders. Base view sub-stream is decoded independently from all others by the original (unmodified) HEVC-compatible decoder. In order to decode other sub-streams, the camera parameters sub-stream needs to be decoded first in the camera parameters decoder. The decoded camera parameters are fed to other decoders except abovementioned base-view decoder. Depth maps sub-streams are decoded in the HEVC-based depth decoder with the use of the camera parameters decoded earlier, in advance. The depth maps are then fed into the HEVC-based texture decoder along with the decoded base view. The texture sub-stream are decoded in the HEVC-based texture decoder with the use of previously decoded camera parameters, depth maps, and the base view. At the end, the reconstructed residual layers of individual views are added to textures of all views.

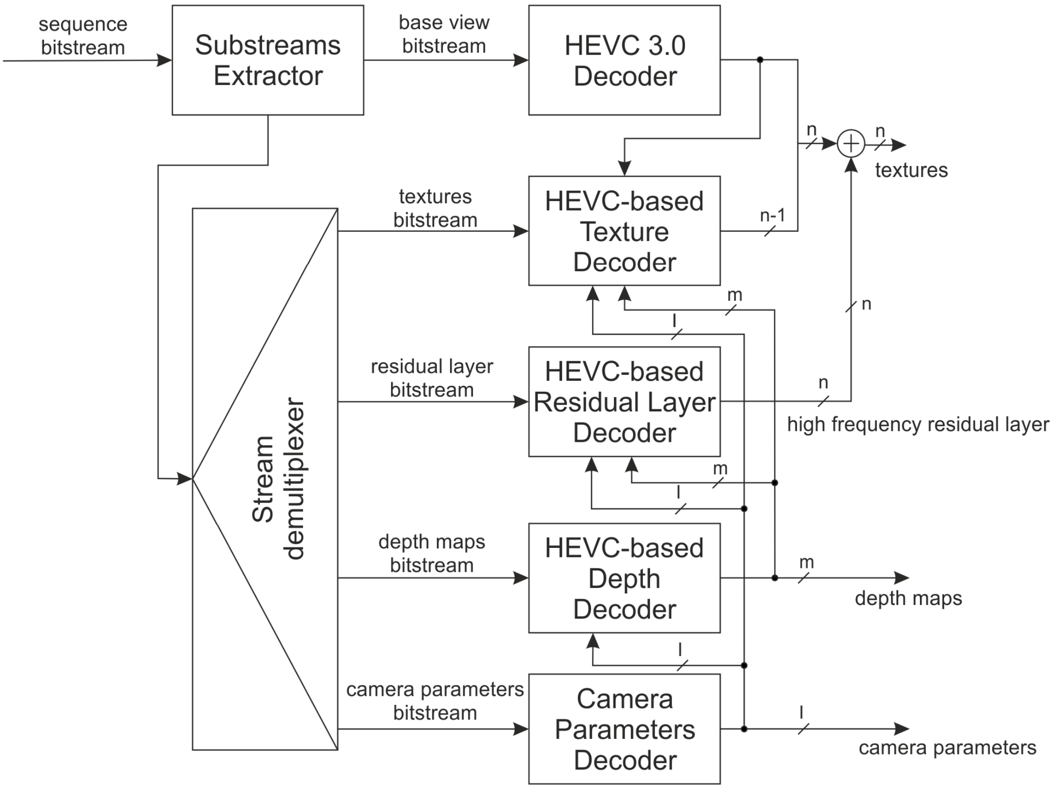


Fig. 2. Scheme of the proposed decoder.

Final view synthesis

View synthesis is performed after the decoding. In the implementation it was done with the use of standard VSRS 3.5 [3]. In the case of GT-Fly sequence batch file provided by Nokia has been used.

3 Algorithm Description

3.1 Texture layer separation

The proposed technology use an approach, similar to SVC (Scalable Video Coding) or to wavelet coding, in which input video is spitted into layers in the spatial frequency domain. Each layer presents different level of details, and all layers represent the input video.

In case of our proposal, the input video texture is split into two layers:

- the so called **texture layer** (similar to base layer in SVC), which contains content that can be efficiently coded with classic predictive coding.

- the so called **residual layer**, which contains high frequency residual content that can be represented jointly for several views.

Both layers are transmitted to the decoder and after decoding are summed together in order to produce reconstructed video.

The separation of layers occurs at the very beginning of the processing as a result of motion-compensated temporal filtering [5].

Each frame of each view is processed independently. Block-based motion estimation is performed in order to find motion vectors pointing to frames neighboring in time (3 previous and 3 next frames). Basing on matched blocks low-pass filtering is performed.

The process yields low-frequency texture layer which is fed to the texture encoder, while the remaining high frequency residual part of the input video is fed to the residual layer encoder.

The layer separation process is entirely automatic.

3.2 Unified depth representation

As mentioned before, the idea behind the proposed technology is that only the base view (texture and depth) is coded directly as a whole. In side views only the disoccluded regions are coded, while the remaining parts are reconstructed from the available views using DIBR (Depth Image Based Rendering). In such an approach, the amount of depth information in side-views is considerably reduced. Because the amount of coded data is limited, it is necessary to adjust the input set of depth maps in such a way, that the single depth map related to the base view contains as much information as possible. To attain that, the depth information represented by the depth maps is merged into a unified depth representation and then projected back onto the views, so that a refined set of depth maps is produced. This step is a necessary part of data format conversion and is fully automatic. Of course, only legal input textures and depth maps are taken into account: 2 views for 2-view case and 3 views for 3-view case, but the process is not limited only to such cases.

The first stage is to improve depth map smoothness by using Mid-Level Hypothesis algorithm [6]. The further processing is based on merging depth information into an unified scene depth representation by depth synthesis (Fig. 3).

Each of the input depth maps $D_1...D_n$ is used to synthesize virtual view D_i' in i-th view's position. For each virtual depth map, fragments with no information (disocclusions) are filled with information from other virtual depth maps, where the information is present. Then, a new unified depth map D_i'' is computed as a result of weighted median filtration of all virtual depth maps across all views.

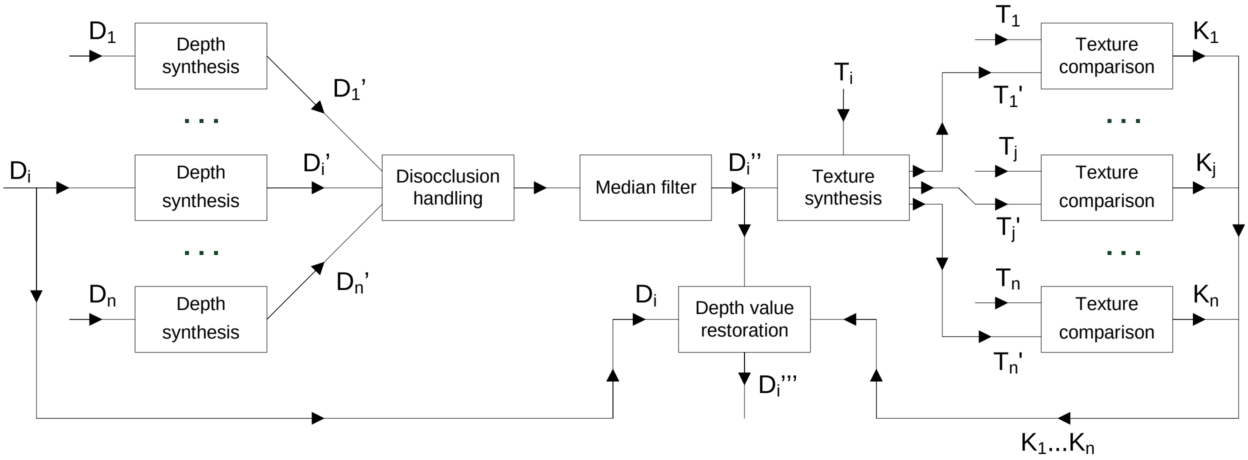


Fig. 3. Block diagram of depth unification process. T_1...T_n are input textures, D_1...D_n are input depth maps, D_1'...D_n' are depth synthesized in position of i-th view, T_1'...T_n' are synthesized textures, and D_1''' is resultant unified depth map.

To prevent virtual texture quality loss due to unsupervised depth map modification, virtual texture quality is assessed. Using depth map D_i' and input texture T_i, virtual textures T_1'...T_n' are synthesized. These virtual textures are then compared with the original textures T_1...T_n and for each a difference cost image K_j is computed. In regions, where any of difference costs values in K_1...K_n exceed given threshold, depth value is replaced with an arithmetic average of D_i'' and D_i . This yields resultant unified depth map D_i'''.

The above-mentioned algorithm is repeated for each of the input views, so that a refined set of depth maps is produced.

3.3 Base view texture layer coding - HEVC-compatible codec

The syntax of the base view is compliant with HEVC, because no modifications has been introduced. The original HM encoder has been enhanced to support GOP sizes that are not necessarily powers of 2 in order to better suit random access requirement. Exemplary allowed GOP sizes allowed are 12 and 15, which were used for coding of test material.

3.4 Camera parameters coding

The intrinsic and extrinsic camera parameters are encoded into the bitstream in Multiview SEI messages. Those messages are sent once per a GOP and provide the camera parameter set for all frames and views in a GOP.

Three types of SEI messages are used:

- Multiview Acquisition Info SEI - that transports intrinsic and extrinsic camera parameters as described in [8]. In our proposal the original syntax and semantics were slightly modified in order to efficiently encode frame-to-frame changes of the camera parameters,

- Multiview Translation Info SEI - that transports the translation parameter, was especially designed for the case when the value of translation parameter is modified from frame to frame,

- Multiview Depth Info SEI is used to encode z_{near} and z_{far} depth parameters and also provides efficient an prediction mechanism for the case when values of z_{near} and z_{far} parameter are modified from frame to frame.

3.5 Depth coding

Depth layer coding is based on HEVC codec with some 3D improvements:

- in side views, view-synthesis prediction is used and thus only disoccluded regions are coded,

- inter-view disparity compensation (MVC-like) is used,

- inter-view depth-based motion prediction (DBMP) is used,

and some depth-specific improvements:

- Depth is internally represented non-linearly, so that closer objects are represented more accurately than distant objects - see 3.5.1. for details,

- 64x64 transform (not available in HEVC 3.0) is used, see JCTVC-D224 for details,

- when used as a prediction reference, depth values are firstly compensated with respect to z_{near}-z_{far} range, which can be different among frames - see 3.5.2 for details,

- edge ringing artifacts in depth are reduced with specially tuned RD-opt.

3.5.1 Depth Map non-linear representation

The human perception of depth depends on absolute distance of viewed objects, therefore the internal depth representation is non-linear. Closer objects are represented more accurately than distant ones. Internal depth sample values are defined by the following power-law expressions, similar as in the case of well known gamma correction:

$$\mathit{depth\ value\ internal} = \left(\frac{\mathit{depth\ value\ external}}{\mathit{maximum\ value\ external}}\right)^{\mathit{exponent}} \cdot \mathit{maximum\ value\ internal}$$

$$\mathit{depth\ value\ external} = \left(\frac{\mathit{depth\ value\ internal}}{\mathit{maximum\ value\ internal}}\right)^{1/\mathit{exponent}} \cdot \mathit{maximum\ value\ external}$$

Exponent is automatically chosen by encoder and sent to decoder in the encoded bitstream. Depth map samples are represented on increased number of bits with use of IBDI (Internal Bit Depth Increase) tool.

3.5.2 Z near - Z far compensation (ZZC) tool

Proposed z_{near}-z_{far} compensation (ZZC) is a new coding tool for multiview video, designed especially for inter-frame depth map coding.

The concept of ZZC exploits the observation that frames from different views and time instances of encoded depth sequence may have different z_{near} and z_{far} parameters. The mentioned z_{near} and z_{far} parameters describe range of depths represented in a gray-scale depth map. If z_{near} and z_{far} parameters are different for two frames, then given depth value is represented with different gray-scale values in those depth maps. Consequently, using one of such depth maps as a reference for the other one will result in a poor prediction.

To overcome this problem, a new ZZC coding tool is proposed. Prior to any inter-frame depth map prediction, each depth map that resides on the codec reference picture list is scaled, so that gray-scale depth values in scaled image and currently coded image refer to the same depth. As a result, depth maps with compensated z_{near} and z_{far} range are used for prediction.

3.6 Texture layer coding

Here, we describe the texture layer coding that is used for all views except for the base view (which is coded with HEVC-compatible coder).

The texture layer coding is based on HEVC codec with some improvements related to the 3D video:

- view-synthesis is used and only disoccluded regions are coded,
- inter-view disparity compensation (MVC-like) is used,
- inter-view depth-based motion prediction (DBMP) is used,

and some texture-specific improvements:

- QP parameter is locally adjusted with respect to depth, so that closer objects are coded with higher quality than distant objects - see Section 3.6.1 for details.

3.6.1 Depth dependent adjustment of QP for texture layer

In order to improve perceptual quality of coded texture, a tool for bit assignment in the texture layer was developed. The basic idea is to increase texture quality of objects in the foreground and to increase compression factor (decrease texture quality) for objects in the background. The quality is adjusted in coding units (CUs) with use of quantization parameter QP that depends on the corresponding depth values. The QP adjustment is done simultaneously in coder and decoder so that no additional information is send. Described tool is disabled in the base view to preserve HEVC compatibility.

3.7 Residual layer coding

The content of the high frequency residual layer is usually not compressed efficiently with classic predictive coding. Sample values of this layer are not correlated and resemble noise. Thus, the content of the residual layer is modeled as a non-stationary random process which can be coded jointly among the views. The only parameters of this process: spatial energy distribution and spectral envelope are coded.

Spatial energy distribution of the residual layer is estimated with use of block-based processing. The residual video is divided into rectangular non-overlapping blocks. In each of those blocks, energy is measured. Energy values, associated with respective blocks, constitute an image of spatial energy distribution, whose resolution is smaller than resolution of the input video. Spatial energy distribution is coded with use of HEVC-based coder.

Spectral envelope is estimated from energy-normalized residual layer with use of technique similar to LPC. The result is a set of IIR filter coefficients (in horizontal and vertical direction) which are coded with use of LAR (log-area-ratio) 8-bit representation. A set of filter coefficients is sent in slice header (likewise to SAO/ALF filters) once per picture, and can be predicted through GOP structure.

The proposed technology allows for coding of residual layer for all views or only for one selected view. In the latter case, residual layer in the missing views is synthesized.

3.8 Inter-view prediction by view synthesis

View synthesis is used as a primary inter-view prediction mechanism. The encoder and the decoder use the same synthesis algorithm, similar to VSRS 3.5. Basing on all already coded views, a new virtual view is synthesized in the position of the current view. Some regions of newly synthesized image are not available because they were occluded in previously coded views. Those disoccluded regions are identified and marked on a binary map, named availability map, which controls coding and decoding process. Coder and decoder simultaneously use this map to determine, whether given CU is coded or not. Because in a typical case most of the scene is the same in all of views, only small parts are disoccluded in subsequently coded views, and thus only small amount of CUs is coded.

A final step of view-synthesis prediction is reduction of artifacts in synthesized view. This post-processing consists of Depth-Gradient-based Loopback Filterer (DGLF) and Availability Deblocking Loopback Filter (ADLF).

The first one (DGLF), reduces texture artifacts introduced by DIBR technique in the areas of a sudden depth changes. In order to cope that the synthesized image is adaptively filtered with respect to depth gradient strengths. Large depth edges impose strong low-pass filtering of the synthesized texture, while flat depth regions are not filtered at all.

The latter (ADLF), reduces artifacts that are generated as a result of block CU-based coding. Shape of coded region not necessarily matches shape of binary availability map. This discrepancy is a source of artificial edges between those regions. The ADLF provides smooth transition between coded and synthesized regions by interpolating between them.

The tool of prediction by view-synthesis is used in texture layer codec, depth layer codec and high frequency residual layer codec.

3.9 MVC toolset implemented in HEVC

The proposed technology exploits inter-view disparity compensation mechanism in a way similar to the one used in the MVC extension of the AVC. Inter-view prediction works by adding inter-view references to the reference lists used for texture and depth image inter prediction. Similarly to MVC, there is a distinction between anchor and non anchor frames and the use of inter-view references can be controlled separately for them. Moreover texture and depth image reference lists are also managed separately. The number of reference views for inter prediction, as well as their IDs for textures and depths, can be chosen independently. No change in the structure of the reference lists is done for the base view in order to preserve its compatibility with the single view HEVC decoder.

3.10 Depth-Based Motion Prediction (DBMP)

Depth-Based Motion Prediction (DBMP) is a new coding tool for multiview video coding which originates from the idea that motion fields of neighboring views in multiview sequence are highly correlated. The concept of DBMP was previously described in [9, 10] under the name of inter-view direct. DBMP provides an efficient representation of motion data in multiview video bitstreams that carry also depth/disparity maps. In the proposed method, the motion information, such as motion vectors and reference indices, for each pixel of encoded coding unit (CU) is directly inferred from already encoded CUs in the neighboring views at the same temporal instance (Fig. 4). This procedure is repeated independently for every pixel of encoded CU. Consequently, motion vectors and reference indices for CU are not transmitted in the bitstream but are obtained from the reference view.

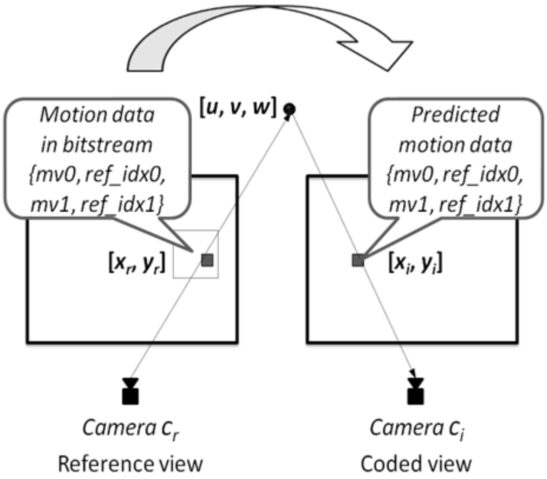

Motion data in bitstream {mv0, ref_idx0, mv1, ref_idx1}
[u, v, w]
Predicted motion data {mv0, ref_idx0, mv1, ref_idx1}
[x_r, y_r]
[x_i, y_i]
Camera c_r
Reference view
Camera c_i
Coded view

Fig. 4. Independent derivation of motion information for each point of encoded CU from corresponding point in reference view [9].

3.11 Stream multiplexing

The bitstream consists of 4 types of sub-streams (see Fig. 1):

- texture of the base view,
- texture of a side view (more than one such sub-stream may exist),
- depth map of a view (more than one such sub-stream may exist),
- residual of a view (one or more such sub-stream may exist).

An encoder produces a bitstream in the form of a sequence of standard NAL units. The bitstream of the base view is compliant with HEVC syntax.

Other streams, that are not HEVC-compatible, are encapsulated in transparent NAL units, so that they can be skipped by a basic HEVC decoder. Full 3D decoder can use them to decode all of the input views or only some of them.

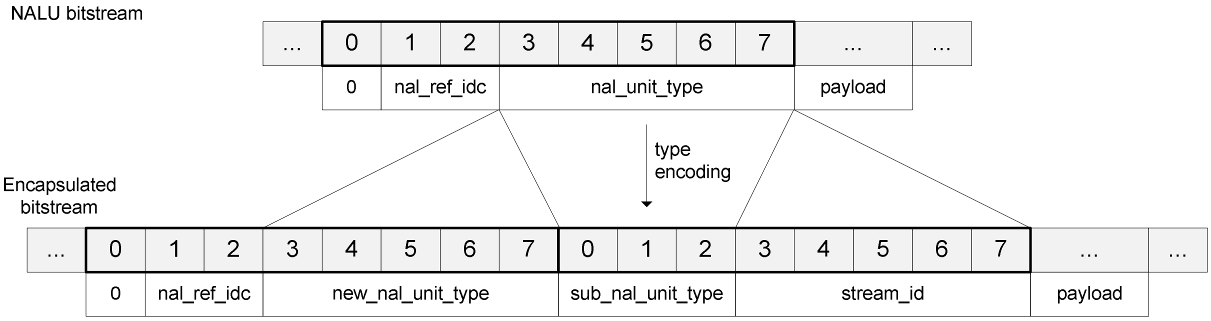


Fig. 5. Sub-stream encapsulation in NALU.

The encapsulation process (see Fig. 5) exploits values of nal_unit_type field (see Table 1) to carry information about encapsulated stream type. Those nal_unit_type values that are used, hitherto were marked as undefined in the Working Draft [7].

Table 1. Contents of new NAL unit types.

new_nal_unit_type	Content
24	Side view textures
25	Depth maps
26	High frequency residuals

All remaining data related with incoming NALU: original nal_unit_type, view number and payload, all are transported inside the encapsulated NALU and (after extraction) transparently delivered to the sub-decoders.

4 Fulfillment of the conditions defined in "Call for Proposal"

The proposed technology and contributed material fulfill conditions described in CfP [1]. In particular:

- the contribution was made in HEVC-Compatible & Unconstrained category and the bitstream is HEVC-compatible,
- complete results for all test cases were submitted,
- random access requirement, though usage of GOP size 12 (for class A sequence) and 15 (for class C sequences),
- automatic quantization adjustment based on depth was used and is described in Section 3.6.1,
- all processing before the coding is related to data format conversion - this includes the texture layer separation and the processing required for depth unified representation,
- multi-pass encoding is limited to the picture level.

5 Fulfillment of the conditions defined in "Requirements"

The matter of requirements imposed in Requirements document [2] is discussed below:

Ad 3.1.1. Video Data

The proposed technology supports both stereo and multiview input video data. The number of transmitted views is fully settable in configuration file.

The proposed uncompressed data format includes samples from left and right views, which are input and output of the codec. Depending on the mode used, it can be all input samples from left and right views or only samples used by the view synthesis algorithm to produce high quality intermediate views.

Ad 3.1.2 Supplementary data

The proposed technology supports generation of high quality intermediate views by transmitting depth maps along with textures. In order to produce intermediate views, DIBR is performed with use of the reconstructed data. All required camera parameters are transmitted along with the video, and support random access feature as well.

Ad 3.1.3 Data volume

The total amount of uncompressed video and supplementary data strongly depends on the particular structure of the coded scene, because in all side views only disoccluded regions are processed. In usual case, the total amount of uncompressed video and supplementary data is about 2 to 3 times of a single uncompressed video data.

Ad 3.1.4 Metadata

The proposed technology supports efficient camera parameter coding. Both the intrinsic and the extrinsic camera parameters along with z_{near} and z_{far} value can be send in efficient frame-to-frame manner in the bitstream for each depth map.

Camera parameters are sent along with other types of data and random access is provided with the same period as for the rest of the bitstream.

Ad 3.1.5 Low complexity for editing

The proposed codec can work in all-I mode in which all frames in all views are coded as I frames, so that each time frame can be acessed separately. In case of other GOP structures, the editing capability is the same as in MVC.

Ad 3.1.6 Applicability

The proposed technology can be used for both natural and synthetic scenes.

Ad 3.2.1 Compression efficiency

The proposed technology is capable of coding video and supplementary data in a bitstream with bitrate not exceeding twice the bit rate of a single video compressed with HEVC. The proposed technology is based on HEVC and offers higher compression performance than MVC [11].

Ad 3.2.2 Synthesis accuracy

The compressed data format employs disocclusion detection, which is based on view synthesis. Therefore, regions that are not used in the view synthesis are not coded at all, and thus, remaining regions are coded with higher quality. The proposed technology supports independent control of compression strength for both texture and depth data.

Proposed technology is not directly related to any rendering technology. Only the estimation of disoccluded regions of the image is done with use of synthesis algorithm similar to VSRS 3.5.

Ad 3.2.3 Forward compatibility

Base view can be decoded with use of HEVC monoscopic decoder. Syntax of other substreams is based on HEVC syntax.

Proposed technology supports also stereo compatible mode where two view are coded with HEVC in simulcast mode without inter-view prediction, and can be putted together in one bitstream.

Proposed technology also supports MVC extension of HEVC for transmitting stereo pair.

Ad 3.2.4 Stereo/Mono compatibility

Proposed technology supports simple mono and stereo bitstream extraction simply by NALU filtering.

Ad 3.2.5 View scalability

Proposed technology supports view scalability simply by NALU filtering.

Ad 3.3.1 Rendering capability, 3.3.4 Variable baseline, 3.3.5 Depth range, 3.3.6 Adjustable depth location

Proposed technology uses state-of-the-art multivideo plus depth (MVD) representation of the data Proposed technology is not directly related to any rendering technology. Only the detection of disoccluded regions of the image is performed with use of synthesis algorithm, similar to VSRS 3.5. Any state-of-the-art DIBR algorithm can be used instead.

Ad 3.3.2 Low complexity

Fast and reliable rendering of intermediate views is possible, because uncompressed data format is composed of only the base view and disoccluded regions of the side views.

Ad 3.3.3 Display types

Proposed technology is independent from display type. Stereoscopic and autostereoscopic displays are supported.

6 Software implementation description

This software is written in C++ programming language. The implementation is based on HEVC codec version HM3.0-dev. The software was retrieved from HEVC repository server on May 10th 2011. The software is identified as revision 866. In addition to base software, some bug-fixes from HEVC repository has been merged. Bug-fixes for #128 (changeset 1008) and #174 (changeset 1009) has been added.

Also some additional libraries has been used, like [4], [5].

The software has been compiled and run successfully under Microsoft Windows using the Microsoft Visual Studio 2005, 2008 and 2010 C++ compiler in their 32-bit and 64-bit variants. Coding of Full-HD resolution sequences may require a 64-bit system.

7 Coding parameters of submitted materials

7.1 Coding order

The proposed technology supports various view configurations. All sequences submitted in response to CfP [1] were encoded with the following coding order.

In 3-view case the middle view is used as base view, whereas in 2-view case the right view is used as the base view. The texture for the base view is the first to be encoded, then the depth map for the base view is encoded. Next, the dependent views are encoded, but for a dependent views depth map always precede the texture. Lastly, the high frequency residual layer for the base view is encoded.

E.g. for Poznan Street sequence in 3-view case, the following coding order was used:

- Base view – view 4,
- Depth map for view 4,
- Depth map for view 3,
- Texture for view 3,
- Depth map for view 5,
- Texture for view 5,
- Residual layer for view 3.

7.2 View number convention

In order to conveniently identify input stream, the following naming convention is used. Views containing textures are numbered 0-99. Views containing depth maps are numbered 100-199. Views containing residual layer are numbered 200-299. This allows an easy identification of a position and type of a view (depth, texture, residual layer) throughout the entire codec.

7.3 Configuration file

Table 2 presents comparison of the configuration file for the anchor HEVC 2.0 codec and for the proposed one. Newly introduced parameters were highlighted red. Some of them are related to progress made in HEVC development, and some are related to the proposed tools. Parameters related to coding order and inter-view prediction scheme are presented in Table 3.

Table 2. Comparison of anchor HEVC 2.0 configuration file
with configuration file of the proposed codec.

Parameter	Proposed Codec	Anchor HEVC Codec	New parameter description
File I/O			
InputFile	depends on sequence	depends on sequence	
BitstreamFile	depends on sequence	depends on sequence	
ReconFile	depends on sequence	depends on sequence	
InputBitDepth	8	8	
OutputBitDepth	8	8	
FrameRate	depends on sequence	depends on sequence	
FrameSkip	0	0	
SourceWidth	depends on sequence	depends on sequence	
SourceHeight	depends on sequence	depends on sequence	
FrameToBeEncoded	depends on sequence	depends on sequence	
MultiviewSEIcfg	depends on sequence	n/a	file containing camera parameters to be encoded
ResidualCoeffsInputFile	depends on sequence	n/a	file containing residual layer filter coefficients
ResidualBlock	30	n/a	Block size for residual energy modelling
ResidualFactor	40	n/a	Normalization factor for residual energy modelling
Unit definition			
MaxCUWidth	64	64	
MaxCUHeight	64	64	
MaxPartitionDepth	4	4	
QuadtreeTULog2MaxSize	5	5	
QuadtreeTULog2MinSize	2	2	
QuadtreeTUMaxDepthInter	3	3	
QuadtreeTUMaxDepthIntra	3	3	
Coding Structure			
IntraPeriod	12	8	
DecodingRefreshType	1	1	
GOPSize	12	8	
RateGOPSize	12	8	
NumOfReference	4	4	

Parameter	Proposed Codec	Anchor HEVC Codec	New parameter description
NumOfReferenceB_L0	2	2	
NumOfReferenceB_L1	2	2	
HierarchicalCoding	1	1	
LowDelayCoding	0	0	
GPB	1	1	
NRF	0	1	
BQP	0	0	
ListCombination	1	1	
Motion Search			
FastSearch	1	1	
SearchRange	64	64	
BipredSearchRange	4	4	
HadamardME	1	1	
FEN	0	0	
Quantization			
QP	depends on sequence	depends on sequence	
depthQP	depends on sequence	n/a	QP value for depth map
residualQP	35	n/a	QP value for residual layer
MaxDeltaQP	0	0	
DeltaQpRD	0	0	
RDOQ	1	1	
DepthPower	-1	n/a	power factor used to encode depth map, -1 is default
Entropy Coding			
SymbolMode	1	1	
Deblock Filter			
LoopFilterDisable	0	0	
LoopFilterAlphaC0Offset	0	0	
LoopFilterBetaOffset	0	0	
Misc.			
InternalBitDepth	10	10	
Coding Tools			
MRG	1	1	
ALF	1	1	
SAO	1	n/a	new in HM 3.0
ALFEncodePassReduction	0	n/a	new in HM 3.0
Slices			
SliceMode	0	n/a	new in HM 2.2
SliceArgument	1500	n/a	new in HM 2.2
LFCrossSliceBoundaryFlag	1	n/a	new in HM 2.2
EntropySliceMode	0	n/a	new in HM 2.2
EntropySliceArgument	180000	n/a	new in HM 2.2
PCM			
PCMLog2MinSize	7	n/a	new in HM 3.0-dev

Table 3. Prediction structure configuration parameters in the proposed codec.

Parameter name	Exemplary value for GT_Fly sequence	New parameter description
Multiview Coding Parameters		
NumViews	7	number of encoded views
ViewOrder	5-105-101-1-109-9-205	views encoding order
View prediction structure parameters		
ViewNumber	5	view number
AnchorRefTextureL0	x	reference view number for anchor frame (list 0) or "x" if none
AnchorRefTextureL1	x	reference view number for anchor frame (list 1) or "x" if none
NonAnchorRefTextureL0	x	reference view number for non-anchor frame (list 0) or "x" if none
NonAnchorRefTextureL1	x	reference view number for non-anchor frame (list 1) or "x" if none
ViewNumber	1	Same as above
AnchorRefTextureL0	5	
AnchorRefTextureL1	x	
NonAnchorRefTextureL0	5	
NonAnchorRefTextureL1	x	
ViewNumber	9	Same as above
AnchorRefTextureL0	5	
AnchorRefTextureL1	x	
NonAnchorRefTextureL0	5	
NonAnchorRefTextureL1	x	
ViewNumber	105	Same as above
AnchorRefTextureL0	x	
AnchorRefTextureL1	x	
NonAnchorRefTextureL0	x	
NonAnchorRefTextureL1	x	
ViewNumber	101	Same as above
AnchorRefTextureL0	105	
AnchorRefTextureL1	x	
NonAnchorRefTextureL0	105	
NonAnchorRefTextureL1	x	
ViewNumber	109	Same as above
AnchorRefTextureL0	105	
AnchorRefTextureL1	x	
NonAnchorRefTextureL0	105	
NonAnchorRefTextureL1	x	
ViewNumber	205	Same as above
AnchorRefTextureL0	x	
AnchorRefTextureL1	x	
NonAnchorRefTextureL0	x	
NonAnchorRefTextureL1	x	
ViewNumber	201	Same as above
AnchorRefTextureL0	x	
AnchorRefTextureL1	x	
NonAnchorRefTextureL0	x	
NonAnchorRefTextureL1	x	
ViewNumber	209	Same as above
AnchorRefTextureL0	x	
AnchorRefTextureL1	x	
NonAnchorRefTextureL0	x	
NonAnchorRefTextureL1	x	

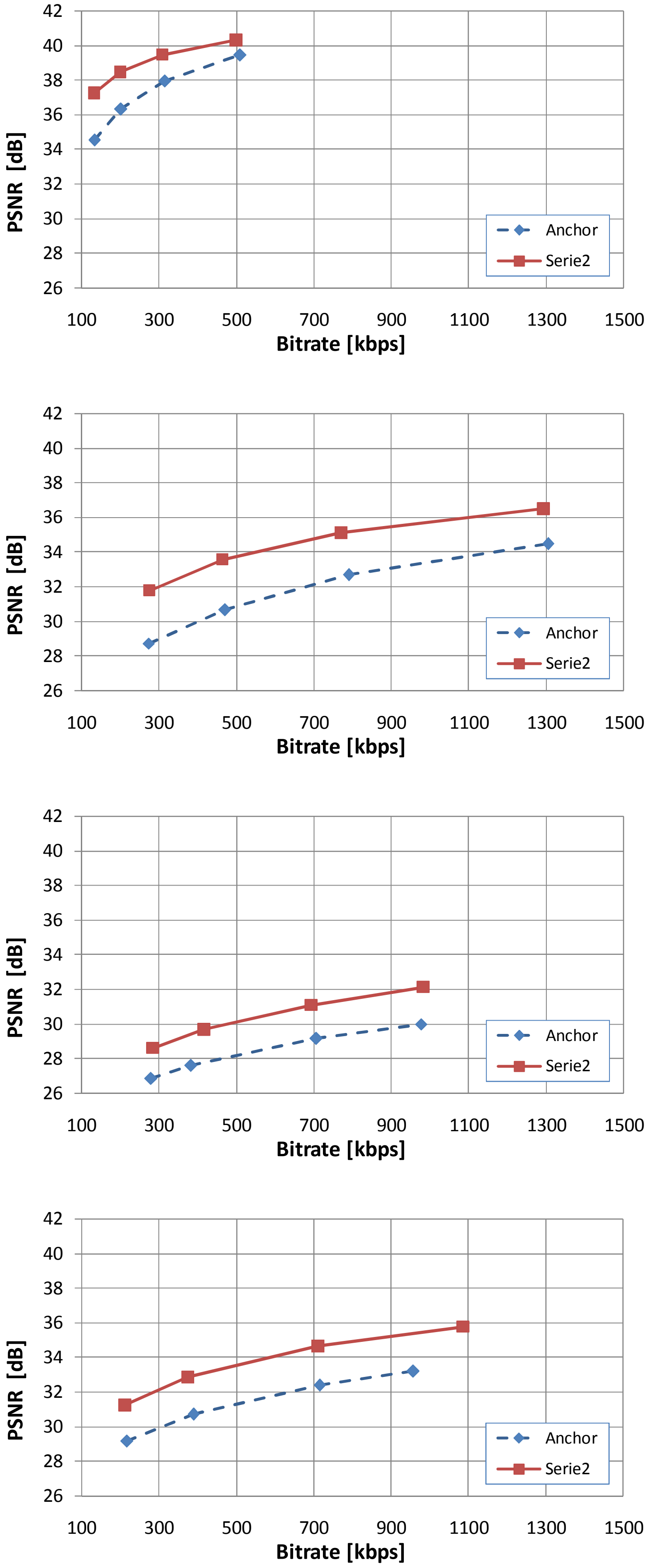
8 Compression performance

The proposed technology, implemented on basis of HM software (see Section 6), was used to encode the test materials with parameters described in Section 7. The PSNR values of the decoded 3D video has been compared to those of anchor sequences. Other views that the base has not been compared, because these views are reconstructed with view-synthesis technique, chich implies different types of artifacts than coding and thus cannot be adequatly compared with PSNR. Bjontegaard [12] metric results, shown in Table 4, reveal average 57,9% gain (2-view case) and 69,8% gain (3-view-case) over anchor coding. Figures 7-10 show RD-curves for the base view for all sequences in 2-view and 3-view case separately. Average percentage of the overall bitstream consumed by given substreams is shown in Figures 11-14.

Please note that PSNR is not a good tool for quality assessment of video coded with statistical tools, like high frequency residual layer coding. Other experiments performed by authors, show that subjective quality assessment implies higher bitrate reduction as for PSNR evaluation.

Table 4. Bjontegaard [12] metric results for the base view: 2-view and 3-view case.

Sequence	2-view case		3-view case	
	ΔPSNR [dB]	ΔBitrate [%]	ΔPSNR [dB]	ΔBitrate [%]
Poznan_Hall2	1,8	-45,3	2,5	-59,3
Poznan_Street	2,7	-54,6	3,9	-69,6
Undo_Dancer	1,9	-51,0	2,7	-62,6
GT_Fly	2,2	-55,5	3,1	-67,0
Kendo	4,2	-58,8	6,2	-73,3
Balloons	5,2	-64,6	6,7	-74,3
Lovebird1	4,2	-66,7	5,8	-76,4
Newspaper	5,1	-66,5	6,3	-76,1
Average	**3,4**	**-57,9**	**4,7**	**-69,8**



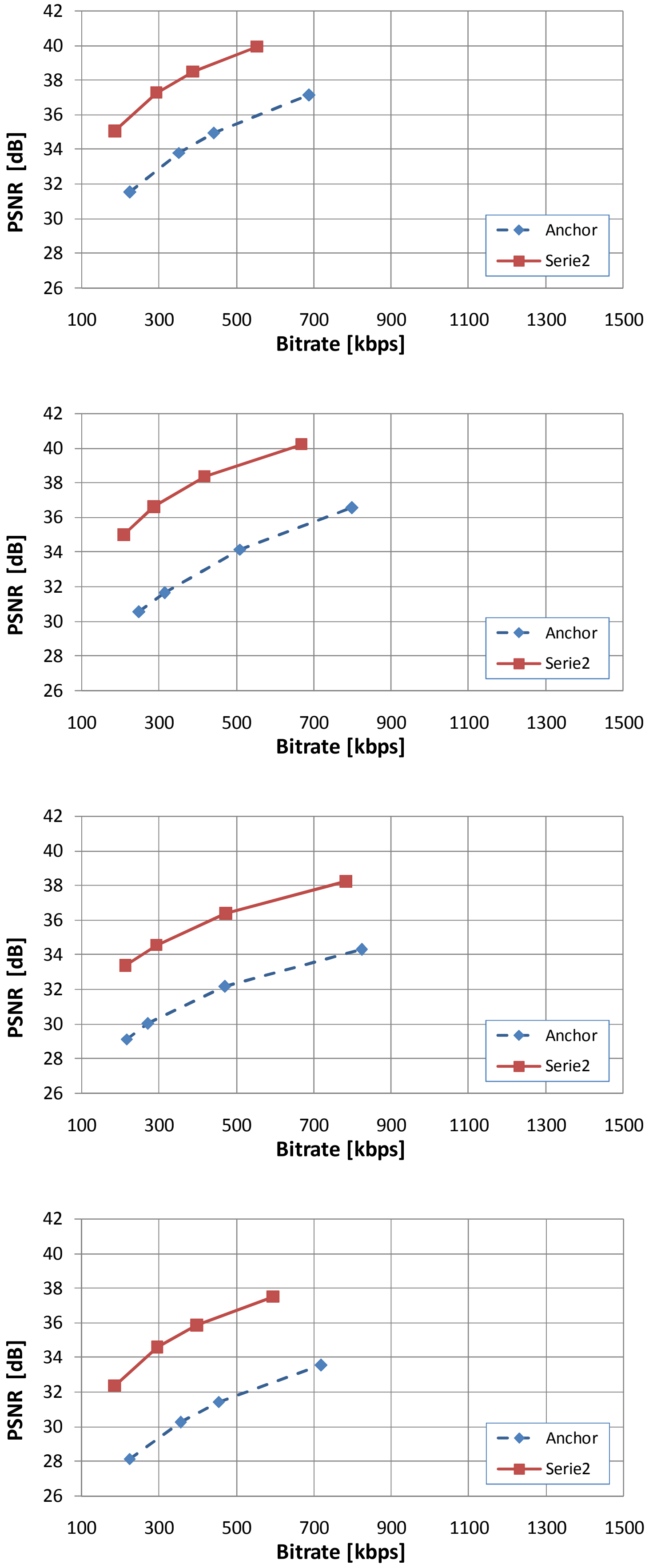
a) Poznan_Hall2 sequence

b) Poznan_Street sequence

c) Undo_Dancer sequence

d) GT_Fly sequence

Fig. 7. Compression performance - 2-view-case - class A sequences.



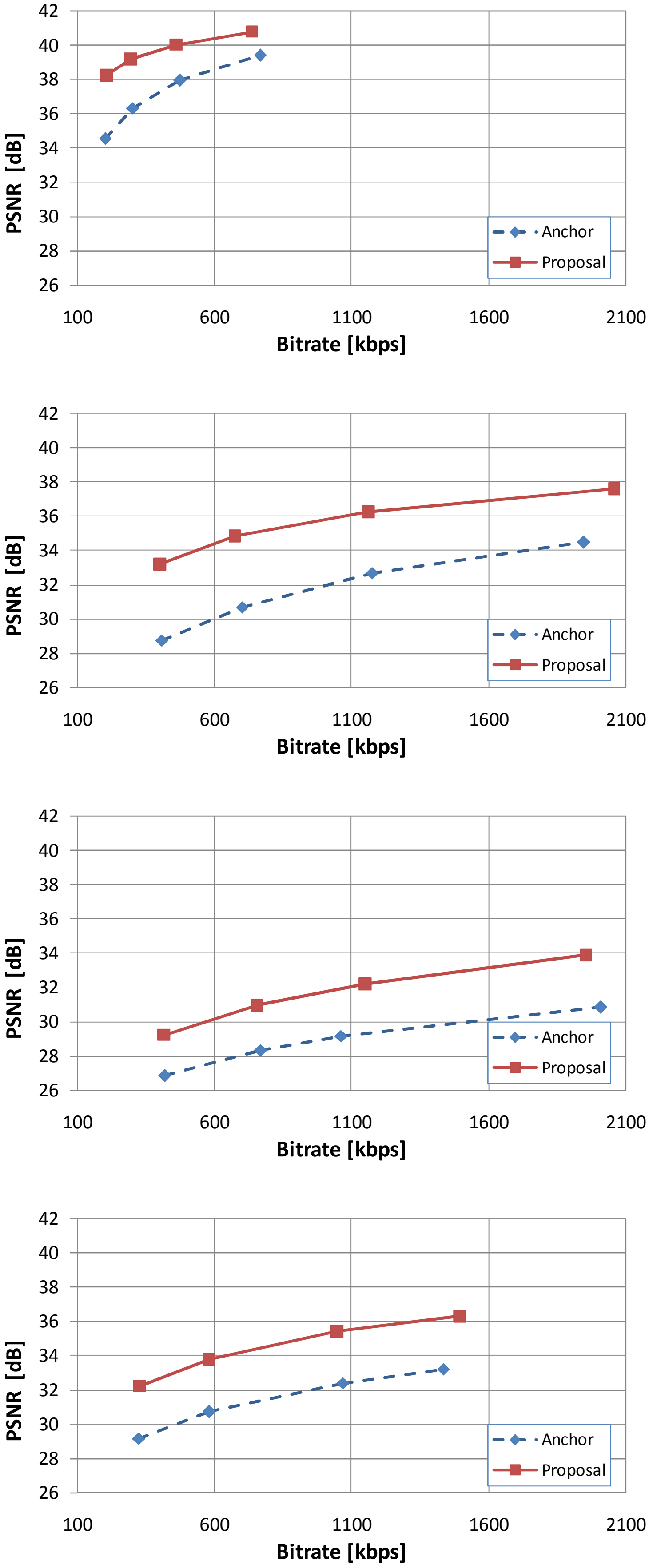
a) Kendo sequence

b) Balloons sequence

c) Lovebird1 sequence

d) Newspaper sequence

Fig. 8. Compression performance - 2-view-case - class C sequences.



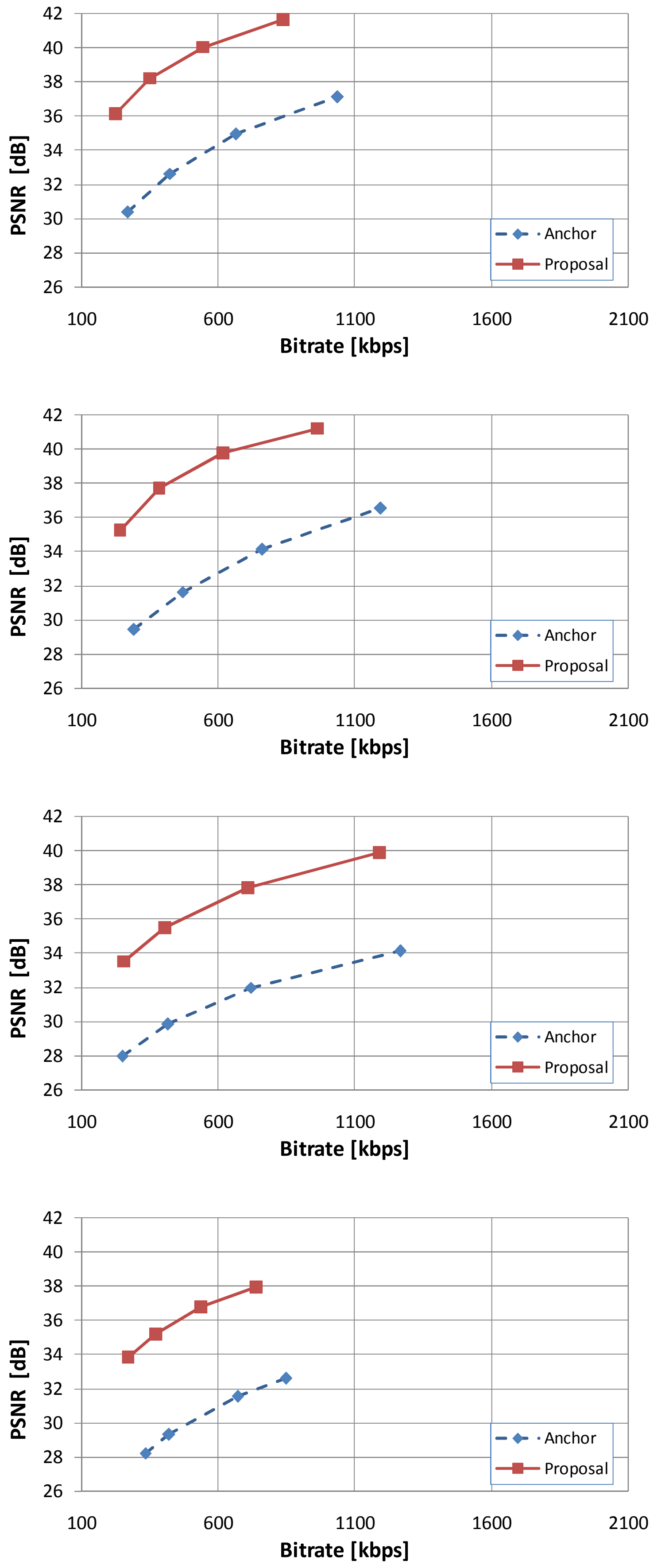
a) Poznan_Hall2 sequence

b) Poznan_Street sequence

c) Undo_Dancer sequence

d) GT_Fly sequence

Fig. 9. Compression performance - 3-view-case - class A sequences.



a) Kendo sequence

b) Balloons sequence

c) Lovebird1 sequence

d) Newspaper sequence

Fig. 10. Compression performance - 3-view-case - class C.

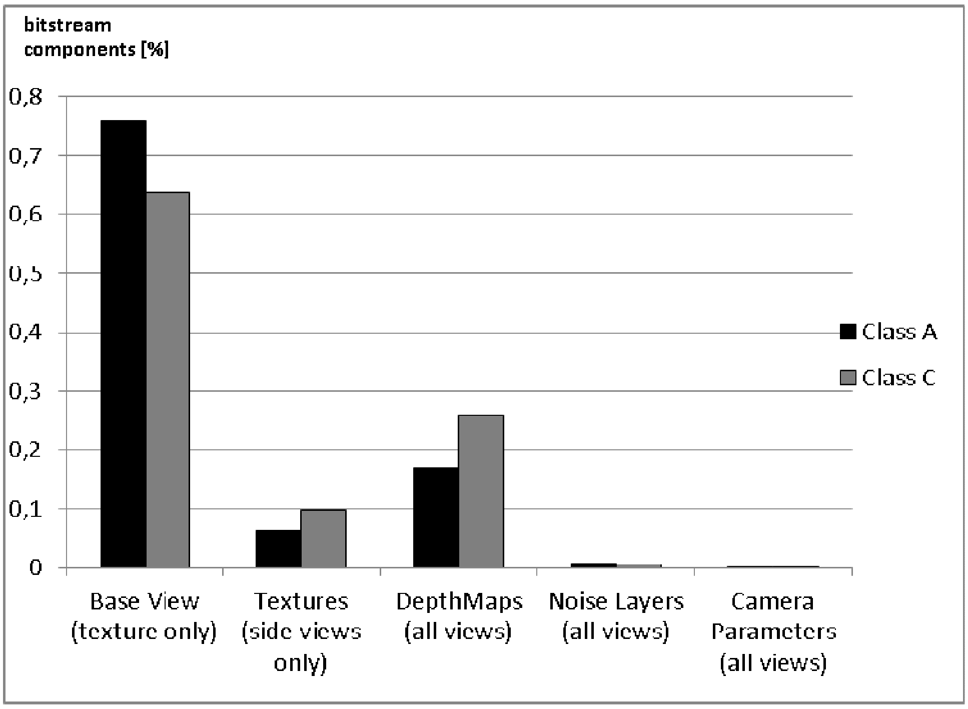


Fig. 11. Percentage of substreams bitrate in the overall bitstream, averaged over class A and class C sequences, 2 view case.

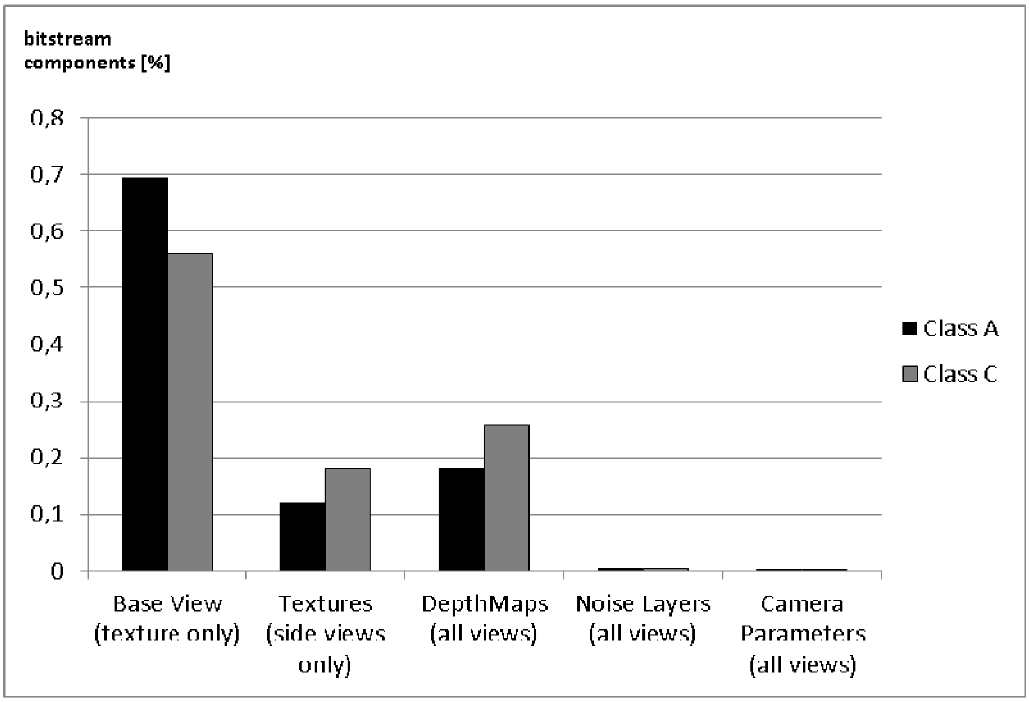


Fig. 12. Percentage of substreams bitrate in the overall bitstream, averaged over class A and class C sequences, 3 view case.

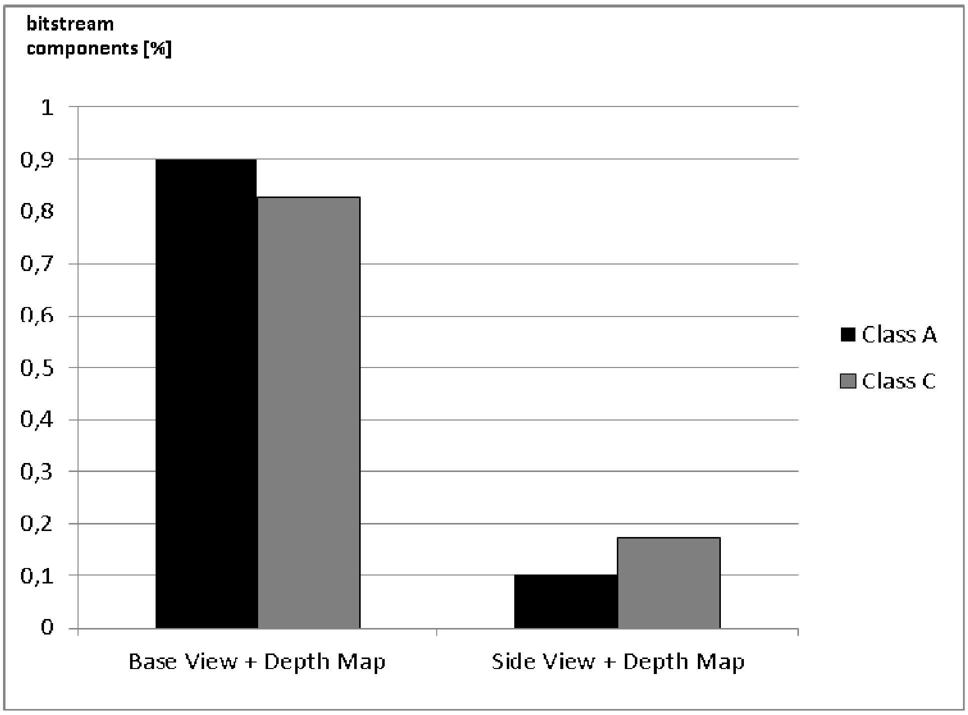


Fig. 13. Percentage of bitrate in the overall bitstream - comparison between a single base view (plus depth) and a single side view (plus depth) - 2 view case. Other substreams than textures and depths are ommited.

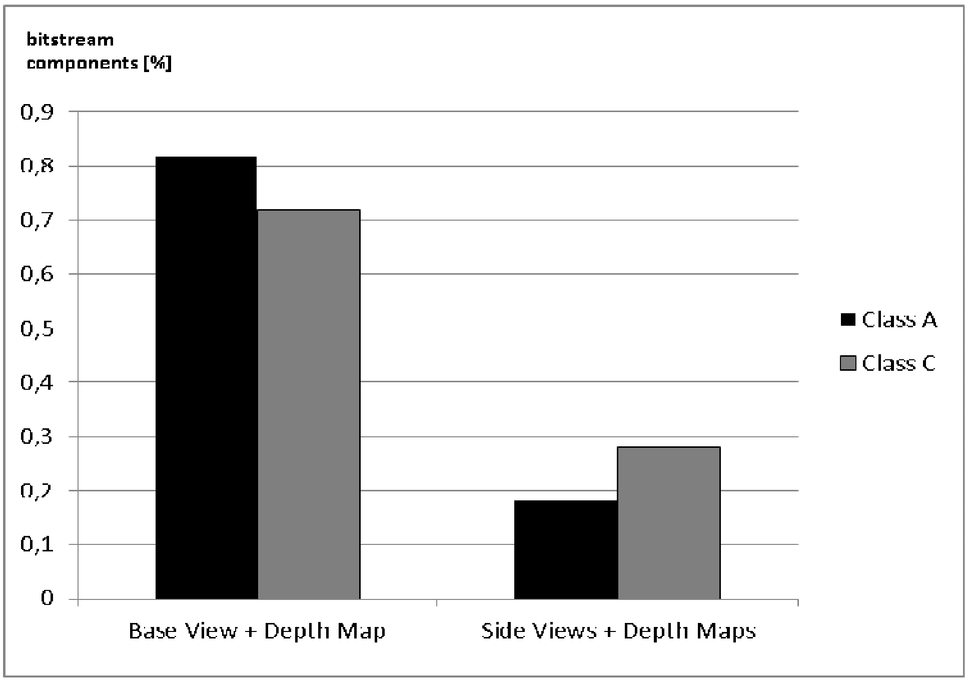


Fig. 14. Percentage of bitrate in the overall bitstream - comparison between a single base view and a single side view - 3 view case. Other substreams than textures and depths are ommited.

9 Complexity analysis

The computational complexity of the proposed technology has been assessed with use of the current software implementation. Its performance has been compared to anchor technology, which is HM codec. The experiment has been done on 64-bit Intel i7 machine with 4GB of memory.

Processing time of a single frame of video, averaged over sequences in classes A and C separately, for 2-view and for 3-view case is shown in Fig. 15, including data format conversion (Layer separation and Depth Unified Representation), encoding (all views, all layers), decoding (overall time) and also encoding and decoding time of anchor HM codec.

Fig. 16 presents the same decoding time as Fig. 15 but in greater detail.

Processing time required to encode a single frame of given substream, averaged over sequences in classes A and C separately, for 2-view and for 3-view case is presented in Fig. 17.

The expected memory usage of the current implementation of the encoder and the decoder does not exceed:

- 4 times the memory usage of the single view HM encoder/decoder for 3-view-case (which is about 3 GB for full HD sequences) and,
- 3 times the memory usage of the single view HM encoder/decoder for 2-view-case (which is about 2 GB for full HD sequences).

We expect that after optimizations, subsequent views can be processed in a sequential manner, so that only one view is processed in the codec at a time.

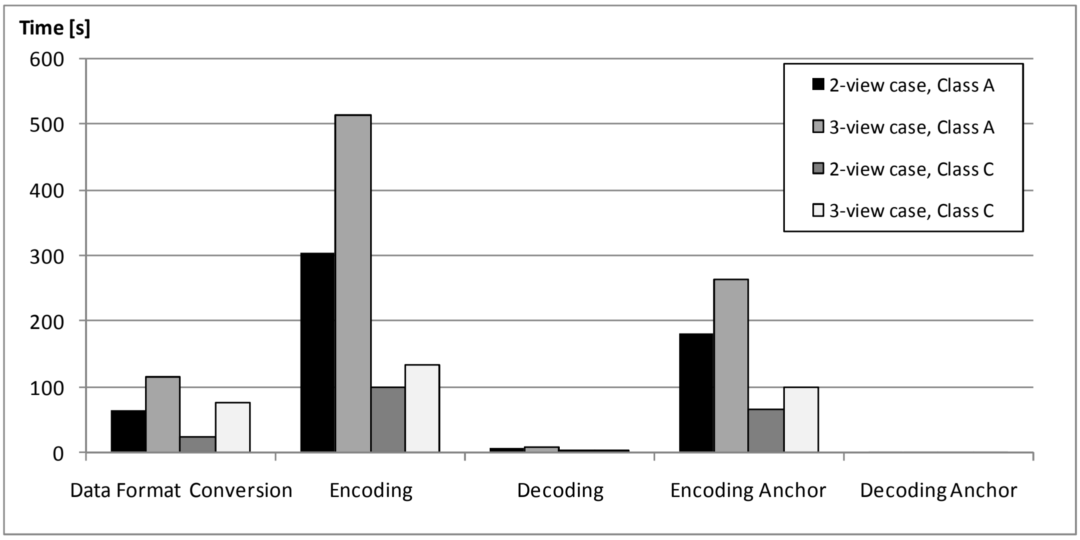


Fig. 15. Average frame processing time: data format conversion (Layer separation and Depth Unified Representation), encoding (all views, all layers), decoding (overall time) and also encoding and decoding time of anchor HM codec.

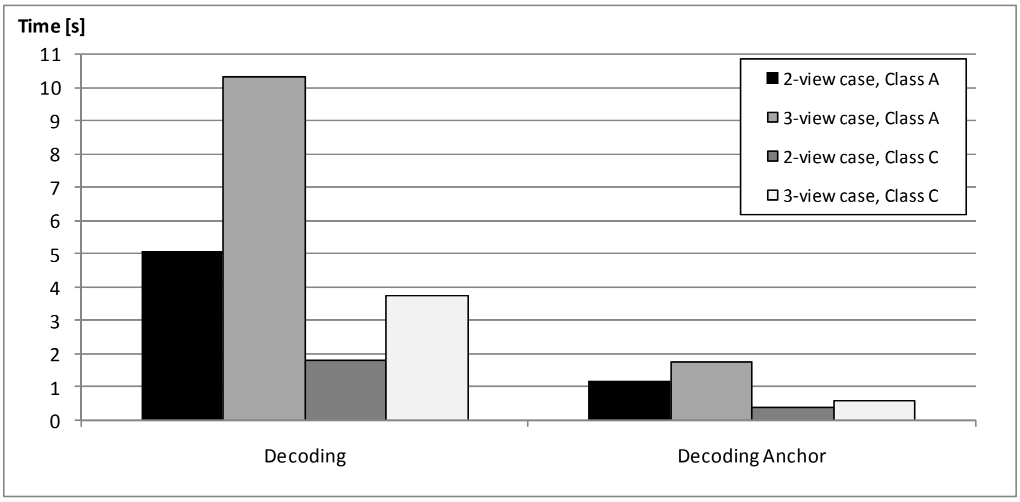


Fig. 16. Average frame processing time:
comparison between proposed technology and anchor (HM codec).

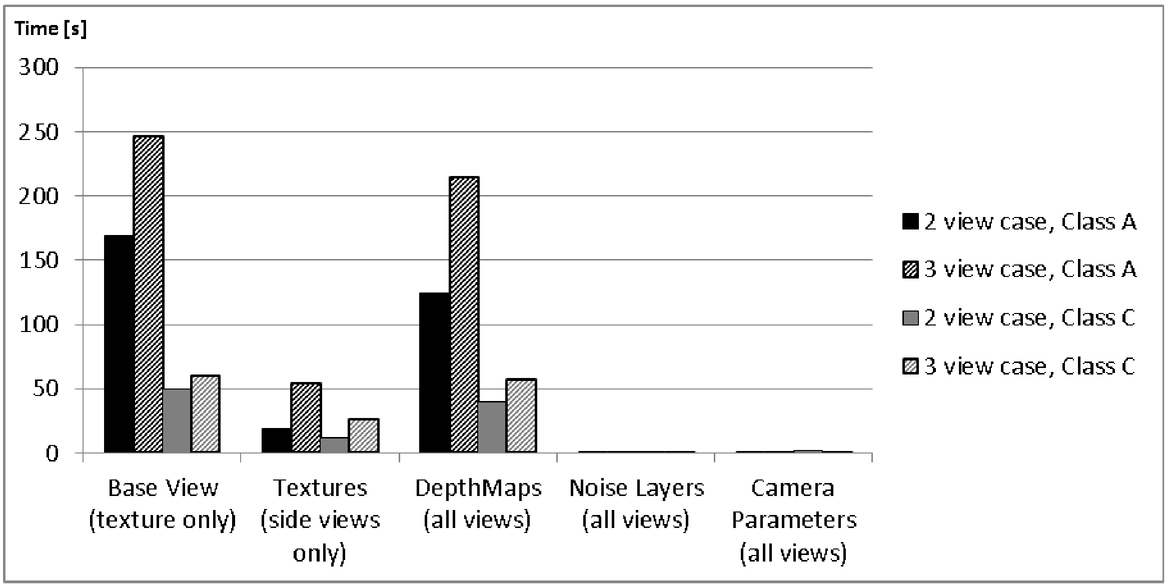


Fig. 17. Amount of time required to encode a single frame of given substreams, averaged over sequences in classes A and C separately, for 2-view and for 3-view case.

10 Conclusions

The contribution was made in HEVC-Compatible & Unconstrained category and the devised bitstream is compatible with HEVC syntax. However, it is worth to notice that the proposed 3D coding technology is in nature independent from single-view coding technology and with some minor changes it can be easily adapted to different coding standards, like AVC or MVC.

11 Acknowledgement

This work was supported by the public funds as a research project.

12 Patent rights

Poznan University of Technology may have IPR relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the UTI-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

13 References

[1] "Call for Proposals on 3D Video Coding Technology", ISO/IEC JTC1/SC29/WG11 MPEG2011/N12036, Geneva, Switzerland, March 2011

[2] ISO/IEC JTC1/SC29/WG11, “Applications and Requirements on 3D Video Coding”, Doc. N11829, Geneva, Switzerland, March 2011

[3] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, M. Tanimoto, "View generation with 3D warping using depth information for FTV". Signal Processing: Image Communication. Volume 24. Issue 1-265-72 (2009)

[4] http://avlib.multimedia.edu.pl

[5] http://avisynth.org.ru/mvtools/mvtools.html

[6] O. Stankiewicz, M. Domański, K. Wegner, "Stereoscopic Depth Refinement by Mid-Level Hypothisis", IEEE International Conference on Multimedia & Expo, Singapore, Singapore, July 2010

[7] T. Wiegand, W.-J. Han, B. Bross, J.-R. Ohm, G. J. Sullivan, "WD3: Working Draft 3 of High-Efficiency Video Coding", JCTVC-E603, Geneva, Switzerland, March 2011

[8] S. Yea, A. Vetro, A. Smolic, H. Brust, “Revised syntax for SEI message on multiview acquisition information”, ITU-T and ISO/IEC JTC1, JVT-Z038, Antalya, Turkey, January 2008

[9] J. Konieczny, M. Domański, “Inter-View Direct Mode for Multiview Video Coding”, ISO/IEC JTC1/SC29/WG11, MPEG2010/M17800, Geneva, Switzerland, July 2010

[10] J. Konieczny, M. Domański, “Extended Inter-View Direct Mode for Multiview Video Coding”, ICASSP 2011, Prague, Czech Republic, May 2011

[11] K. Wegner, O. Stankiewicz, K. Klimaszewski, M. Domański, "Comparison of multiview compression performance using MPEG-4 MVC and prospective HVC technology" ISO/IEC JTC1/SC29/WG11 MPEG 2010 / M17913, Geneve, Switzerland, 2010

[12] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves", ITU-T VCEG, Texas, USA, Proposal VCEG-M33, 2001