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

ISO/IEC JTC1/SC29/WG11 MPEG2012/ M23783
February 2012, San Jose, USA

Source **Poznań University of Technology,**
Chair of Multimedia Telecommunications and Microelectronics, Poznań, Poland

Status **Report**

Title **Poznan University of Technology tools for 3DV coding integration into 3D-HTM**

Author **Olgierd Stankiewicz** (ostank@multimedia.edu.pl),
Krzysztof Wegner (kwegner@multimedia.edu.pl),
Marek Domański (domanski@et.put.poznan.pl),
Jacek Konieczny (jkonieczny@multimedia.edu.pl),
Jakub Siast (jstast@multimedia.edu.pl)

Table of contents

1 Introduction

This documents reports an integration of coding tools originate from compression technology prepared at Poznan University of Technology [1] in response to Call for Proposals on 3D Video Coding Technology[2]. The tools, marked using the label "integration tool" in document [3], were decided to be integrated into software used for the submissions m22570, m22571, and m22668 and published by HHI at [4].

2 Overview

All integrated tools are marked in the source code with define labels with "POZNAN_" prefix. Integration process for those tools is discussed in further sections:

- View synthesis based inter-view disocclusion prediction (Section 3.1),
- Depth-based motion parameter prediction (Section 3.2),
- Adjustment of QP of texture based on depth data (Section 3.3),
- Non-linear depth representation (Section 3.4),
- Z-near z-far compensated weighted prediction (Section 3.5),
- Depth edge-based r-d optimization tuning (Section 3.6),

3 Integrated tools description

3.1 View synthesis based inter-view disocclusion prediction

View synthesis is used as a primary inter-view prediction mechanism. The encoder and the decoder use the same synthesis algorithm, already existing in 3D-HTM software. Basing on all already coded views, a new virtual view is synthesized in the position of the currently coded view. Some regions of newly synthesized image are not available because they were occluded in the others views. Those disoccluded regions are identified and marked on a binary map, named availability map, which controls coding and decoding process. The 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 with Availability Deblocking Loopback Filter (ADLF).The 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.

3.1.1 Tool activation

The tool can be enabled or disabled in "TypeDef.h" file with `POZNAN_CU_SKIP` and POZNAN_CU_SYNTH flag.

3.1.2 Tool dependences

The view synthesis based inter-view prediction tool requires no additional tools activation.

3.1.3 Implementation details

Additional buffers to keep availability maps were implemented in `TEncCU` class and in `TComPic` class. Availability maps are computed in `TRenTop::extrapolateAvailabilityView (...)` function which is slightly modified `TRenTop::extrapolateView(...)` function from 3D-HTM base software.

In encoding/decoding process encoder/decoder determine if block should be encoded into bitstream/decode from bitstream using `TComPic::checkSynthesisAvailability(...)` function.

3.1.4 Configuration parameters

No parameters for this tool needs to be specified in configuration file.

3.2 Depth-based motion parameter prediction

Integration process for this tool is still in progress.

3.3 Adjustment of QP of texture based on depth data

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 at coding units (CUs) level 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 are send. Described tool can be disabled in the base view to preserve HEVC compatibility. Because 3D-HTM use Depth after Texture coding order, implemented tool are using synthesized depth map from previously coded depth maps.

3.3.1 Tool activation

The tool can be enabled or disabled in "TypeDef.h" file with `POZNAN_TEXTURE_TU_DELTA_QP_ACCORDING_TO_DEPTH` flag.

Moreover as mentioned before this tool can be disabled in the base view to preserve HEVC compatibility by `POZNAN_TEXTURE_TU_DELTA_QP_NOT_IN_BASE_VIEW` flag in “TypeDef.h”

3.3.2 Tool dependences

In current software version adjustment of QP of texture blocks is based on synthesized depth map values. In order to enable this tool synthesis tool needs to be enabled too (`POZNAN_CU_SYNTH` flag in "TypeDef.h" file).

3.3.3 Implementation details

No actual QP value update in coding unit object fields is done. QP value is adjusted through a parameter of class function `TComTrQuant::setQPforQuant(...).` Functions used for computing QP adjustment are implemented in `TComDataCU` class.

3.3.4 Configuration parameters

No parameters for this tool needs to be specified in configuration file.

3.4 Non-linear depth 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 the encoder with use of base QP for the depth and sent to decoder in the encoded bitstream (one parameter is added to i.e. SPS):

$$\mathrm{exponent} = \mathrm{clip}\left((\mathrm{QP}_{\mathrm{depth}} - 30) \cdot 0.0125 + 1.25 \ ;\ 1.0 \ ;\ 1.66\right)$$

Depth map samples at encoder/decoder side are represented on increased number of bits with use of IBDI (Internal Bit Depth Increase) tool.

3.4.1 Tool activation

The tool can be enabled or disabled in "TypeDef.h" file with `POZNAN_NONLINEAR_DEPTH` flag. Exponent can be send in a bitstream as a float number or as a byte if `POZNAN_NONLINEAR_DEPTH_SEND_AS_BYTE` flag in "TypeDef.h" file is active.

3.4.2 Tool dependences

This tool requires no additional tools activation.

3.4.3 Implementation details

Function `TComPicYuv::power(...)` implemented in `TComPicYuv` class is used to transform `TComPicYuv` object to internal or external representation. This function uses `TComPowerConverter` class object that convert individual internal and external values.

With the object of correct work of view synthesis algorithm, already implemented in the base software, minor changes were made. Look-Up Tables (LUTs) used in this synthesis algorithm were changed to obtain proper results with non-liner depth maps. It is done with modifications in `TAppComCamPara::xSetShiftParametersAndLUT(...)` function. LUTs precision was also increased (enabled with `POZNAN_LUT_INCREASED_PRECISION` flag in "CommonDef.h" file), because originally did not supported Internal Bit Depth Increase (IBDI) tool of the codec.

Everywhere in codec, depth maps are kept in transformed (non-linear) form. To obtain an original depth map values reverse transformation is needed. While writing a depth map to YUV file this reverse transformation is done automatically (function `TAppDecTop::xWriteOutput(...)` and `TAppEncTop::xWriteOutput(...)` were modified. View synthesis and View Synthesis Optimization (VSO) tools are patched with use of modified LUTs. In general, this tool has no computational complexity, because mainly consists of pre-/post-processing of the coded depth maps.

3.4.4 Configuration parameters

Exponent parameter is chosen automatically, so no parameters for this tool needs to be specified in configuration file. Although, it is possible to specify this parameter manually - this parameter should be float number and labeled as "`DepthPower`" in configuration file or as "`-dpow`" in command line.

3.5 Z-near z-far compensated weighted prediction

A similar tool is already implemented in the base software (weighted prediction),so integration was abandoned.

3.6 Depth edge-based r-d optimization tuning

Integration process for this tool is still in progress.

4 Conclusions

Not all required Poznan tools have been already integrated in the 3D-HTM software. About two more weeks after San Jose meeting are required to end the integration process.

5 Acknowledgement

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

6 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).

7 References

[1] M. Domański, T. Grajek, D. Karwowski, K. Klimaszewski, J. Konieczny, M. Kurc, A. Łuczak, R. Ratajczak, J. Siast, O. Stankiewicz, J. Stankowski, K. Wegner, "Technical Desciption of Poznan University of Technology proposal for Call on 3D Video Coding Technology", ISO/IEC JTC1/SC29/WG11 (MPEG) M22697, Geneva, Switzerland, November 2011.

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

[3] H. Schwarz, K. Wegner, "Test Model under Consideration for HEVC based 3D video coding", ISO/IEC JTC1/SC29/WG11 (MPEG) w12350, Geneva, Switzerland, November 2011.

[4] https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/