Low-Area Scalable Hardware Architecture for DMM-1 Encoder of 3D-HEVC Video Coding Standard

Gustavo Sanchez
Pontifical Catholic University of Rio Grande do Sul
IF Farroupilha
Brazil
gustavo.sanchez@acad.pucrs.br

Luciano Agostini
Federal University of Pelotas
Brazil
agostini@inf.ufpel.edu.br

Filipo Mór
Pontifical Catholic University of Rio Grande do Sul
Brazil
filipo.mor@ acad.pucrs.br

César Marcon
Pontifical Catholic University of Rio Grande do Sul
Brazil
cesar.marcon@pucrs.br

ABSTRACT

This work presents a low-area scalable architecture for the Depth Modelling Mode 1 (DMM-1) encoder of the 3D High Efficiency Video Coding (3D-HEVC) standard, removing the refinement stage. This simplification causes a small BD-rate increase (0.09%) but a significant reduction in memory usage of 30%. The scalable architecture can support different block sizes. Synthesis results for ST 65 nm Standard Cells technology show that the designed structure is capable of reaching real-time processing of HD 1080 p videos for all block sizes.

CCS CONCEPTS

• **Hardware → Application specific integrated Circuits; Computer systems organization → Real-time system architecture**.

KEYWORDS

3D-HEVC, DMM-1, Scalable Architecture, Low-Area Hardware Design

ACM Reference format:

G. Sanchez, L. Agostini and C. Marcon. 2017. Low-Area Scalable Hardware Architecture for DMM-1 Encoder of 3D-HEVC Video Coding Standard. In Proceedings of Symposium on Integrated Circuits and Systems Design, Fortaleza, Ceará Brazil, August 2017 (SBCCI 2017), 5 pages.

http://dx.doi.org/10.1145/3109984.3109986

1 INTRODUCTION

The Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) is a group of experts from ITU-T and ISO/IEC, established to work on multi-view and 3D video coding extensions of HEVC. JCT-3V spends significant effort in research and development to extend the High-Efficiency Video Coding (HEVC) standard to 3D video applications [1]. In this direction, JCT-3V finalized the 3D video coding (3D-HEVC) standard in February of 2015. It uses the most advanced features provided by HEVC and proposes many new features to explore 3D video characteristics.

The 3D-HEVC adopts the Multi-View plus Depth (MVD) [2] to encode and transmit an enormous amount of data required by 3D video applications. In MVD, each texture view is associated with a depth map. The motivation for MVD usage is to reduce the bandwidth for a 3D video transmission because techniques such as Depth Image Based Rendering (DIBR) [2] allows minimizing the number of transmitted texture views and synthesizing virtual views at the decoder.

The same camera captures the texture images and depth maps, which represent the distance between the camera and the objects. 8-bits samples in gray shades compose these maps, where the closer the object is from the camera, the lighter the shade of gray will be represented. Besides, depth maps contain distinct characteristics in contrast to texture frames: these maps are characterized by containing large areas of constant values (background or body of objects) and sharp edges (border of objects) [3].

The original HEVC algorithms were designed to explore texture characteristics, and they can be applied in depth maps coding. However, the coding process can be highly inefficient if depth maps characteristics, such as the presence of sharp edges, are not considered. Then, the JCT-3V proposed new coding tools, called bi-partition modes, to encode better depth maps in the 3D-HEVC intra prediction: four Depth Modeling Modes – DMM-1 to DMM-

SBCCI '17, August 28-September 01, 2017, Fortaleza - Ceará, Brazil
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5106-5/17/08…$15.00
http://dx.doi.org/10.1145/3109984.3109986

4; and the Region Boundary Chain – RBC [4]. The final version of 3D-HEVC excluded the DMM-2, DMM-3, and RBC because they added high complexity with small coding efficiency gains.

On the one hand, these new coding tools are capable of increasing the encoded 3D video quality, but on the other hand, their insertion increases the 3D-HEVC computational complexity and hinder to deal with real-time applications constraints. These applications require at least a processing rate of 30 frames per second, which is a hard task even for 2D high-definition videos. Moreover, many 3D applications should be running on embedded devices, allowing limited energy consumption and limited area. Therefore, hardware accelerators are mandatory to meet performance, energy consumption and area requirements.

Sanchez et al. [5] propose the design of a scalable DMM-1 and DMM-4 architecture sharing resources among modes and the work [6] focuses only in DMM-4 mode. Since DMM-1 is much more complex than DMM-4, the insertion of extra resources to allow computing DMM-4 algorithm as performed in [5] requires a considerable high area. This work proposed a similar architecture compared with [5], however, we remove the computation of DMM-4 from that structure to obtain an efficient low-area hardware architecture for the DMM-1 encoder, which scales to support all 3D-HEVC block sizes.

The remaining of this paper is organized as follows. Section 2 describes the 3D-HEVC depth intra-frame prediction algorithm and explains the DMM-1 mode. Section 3 details the DMM-1 architecture. Section 4 discusses the achieved results, and Section 5 renders the conclusions of this paper.

2 3D-HEVC DEPTH MAPS INTRA-FRAME PREDICTION

Fig. 1 shows the main blocks and flow of the depth maps intra-frame prediction implemented in the version 16.0 of 3D-HEVC Test Model (3D-HTM) [7].

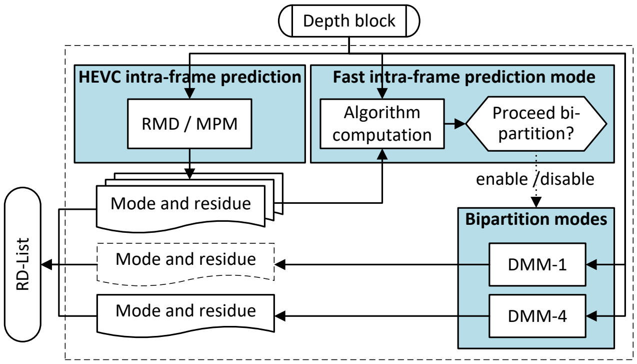

Figure 1: Main blocks and flow of 3D-HTM depth map intra-frame prediction.

Two steps compose the algorithm: the (i) HEVC intra-frame prediction that implements the same intra-frame algorithms used for texture videos, and the (ii) bi-partition modes, which explore depth maps edges during the encoding process. The HEVC intra-frame prediction uses the Rough Mode Decision (RMD) and the Most Probable Modes (MPM) algorithms to build a Rate-Distortion (RD) list. Additionally, it includes few intra-frame prediction modes, among the 35 available, with high probability to obtain efficient coding [7]. Then, only the modes inside the RD-list will be thoroughly evaluated in the HEVC RD calculation, implying a complete coding process for these modes to get the real compression rate and distortion and then, choosing the best encoding mode.

RMD evaluates all possible modes using the Sum of Absolute Transformed Differences (SATD) between the original block and the predicted one, searching for the modes that obtain the lowest SATDs, which are added to RD-list. This process simplifies the HEVC intra-frame prediction because it avoids the insertion of all available prediction modes in the RD-list. Additionally, the MPM algorithm considers the information of the previous neighbor encoded blocks, which adds new important modes into the RD-list.

The bipartition modes should be evaluated in parallel with the original HEVC intra-frame prediction step. However, as the bipartition modes are mainly designed to encode edges, the algorithm proposed in [9] only enables the evaluation of them if the first mode in RD-list is not the planar and if the encoding block variance is higher than a pre-defined threshold, otherwise, it is disabled. If the bipartition modes should be evaluated, the DMM-1 and DMM-4 are processed, and their results are added in the RD-list. The DMM-1 implements its block partitioning using a wedgelet, which algorithm is presented next.

2.1 Depth Modeling Mode 1 (DMM-1)

The DMM-1 algorithm is based on wedgelets, where a wedgelet is a straight line that separates a block in two regions. Fig. 2 shows an example of encoding a 4×4 depth block along with the evaluation of three patterns.

The DMM-1 prediction process encodes the block samples to indicate the region where each pixel belongs. Subsequently, it generates the predicted block using the average value of each region and then computes the Sum of Absolute Differences (SAD) between the original encoding block and the predicted one. Finally, the algorithm selects the pattern with lowest SAD as the best wedgelet (e.g., in Fig. 2, the pattern b is selected).

Table 1 depicts that there are many possible wedgelets in a depth map block. However, 3D-HEVC defines a search with a refinement over the complete wedgelet set, consequently reducing DMM-1 complexity.

Table 1: Number of evaluated wedgelets in DMM-1

Block size	Total possible wedgelets	Evaluated wedgelets	Percentage of reduction
4×4	86	58	32.5%
8×8	802	314	60.8%
>= 16×16	510	384	24.7%

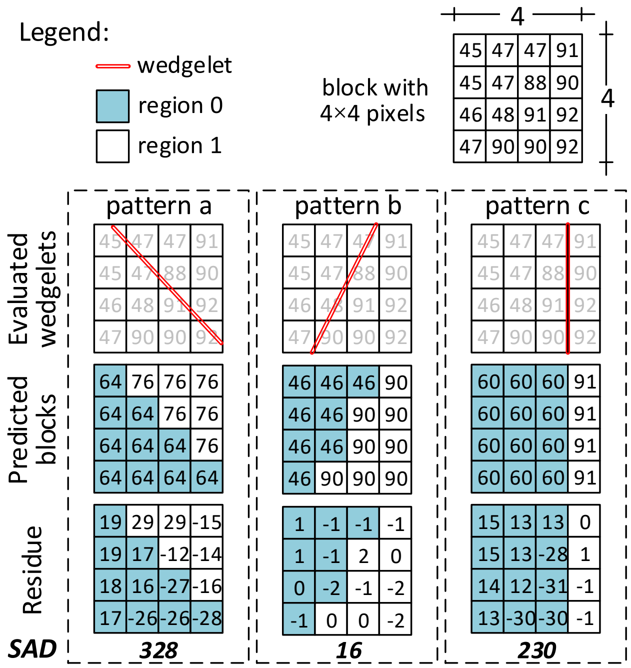
Legend:
wedgelet
region 0
region 1
block with 4×4 pixels
4
4
45 47 47 91
45 47 88 90
46 48 91 92
47 90 90 92
pattern a
pattern b
pattern c
Evaluated wedgelets
Predicted blocks
Residue
SAD
pattern a predicted: 64 76 76 76 / 64 64 76 76 / 64 64 64 76 / 64 64 64 64
pattern b predicted: 46 46 46 90 / 46 46 90 90 / 46 46 90 90 / 46 90 90 90
pattern c predicted: 60 60 60 91 / 60 60 60 91 / 60 60 60 91 / 60 60 60 91
pattern a residue: 19 29 29 -15 / 19 17 -12 -14 / 18 16 -27 -16 / 17 -26 -26 -28
pattern b residue: 1 -1 -1 -1 / 1 -1 2 0 / 0 -2 -1 -2 / -1 0 0 -2
pattern c residue: 15 13 13 0 / 15 13 -28 1 / 14 12 -31 -1 / 13 -30 -30 -1
328
16
230

Figure 2: Example of encoding a block with the DMM-1 algorithm.

Table 1 also presents the number of wedgelets evaluated before the refinement stage and the corresponding reduction. The complete DMM-1 process signalizes the selected wedgelet and sends the residue between the original depth block and the predicted one to the next encoder modules.

Fig. 3 illustrates the DMM-1 encoding algorithm that covers three stages: (i) Main Stage, (ii) Refinement Stage and, (iii) Residue Stage.

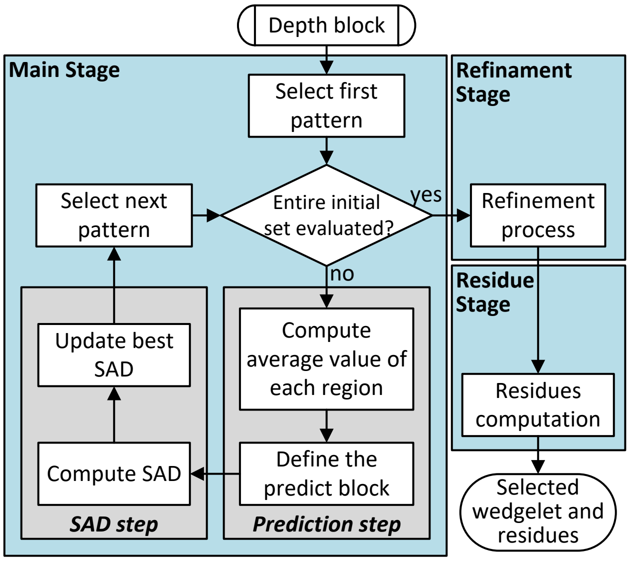

Figure 3: Main blocks of the DMM-1 algorithm.

The Main Stage evaluates the initial wedgelet set and finds the best wedgelet partition. For each wedgelet in the evaluation set, the encoding block is mapped into the binary pattern of that wedgelet. The average values of all samples mapped in region 0 and region 1 are computed and the predicted block is defined as the average value of each region (Prediction step). Next, for each wedgelet pattern, the SAD is computed between the original block and the predicted one and finally, all SADs are compared. The wedgelet with the lowest SAD is chosen (SAD step). In the Refinement Stage, up to eight wedgelets are evaluated around the selected one in the main stage. The Residue Stage subtracts the best-predicted block from the original one and adds the best wedgelet to RD-list. Finally, the wedgelet will be evaluated according to their RD-cost.

2.2 Removal of DMM-1 Refinement Analysis

Sanchez et al. [5] propose a simplification in the DMM-1 algorithm that removed the DMM-1 Refinement Stage. This work maintains this simplification. The impact of RD was evaluated using the HTM-16.0 considering 3-views case, the videos defined in the Common Test Conditions (CTC) for 3D videos [10] for both random access and all-intra conditions. Table 2 shows that this simplification causes an average BD-rate degradation of 0.09% and 0.25% in the synthesized views under random access and all-intra modes, respectively.

Table 2: BD-rate impact when removing DMM-1 refinement

Video	BD-rate random access	BD-rate all-intra
Balloons	0.16%	0.21%
Kendo	0.00%	0.24%
Newspaper_CC	0.19%	0.40%
GT_Fly	0.02%	0.20%
Poznan_Hall2	0.03%	0.23%
Poznan_Street	0.10%	0.09%
Undo_Dancer	0.08%	0.25%
Shark	0.15%	0.41%
Average	**0.09%**	**0.25%**

Table 3: Memory saving by removing the DMM-1 refinement

Block size	Memory (bits)		Reduction (%)
	All wedgelets	Without refinement	
4×4	1,376	928	32.6
8×8	51,328	20,096	60.8
16×16	130,560	98,304	24.7
32×32	130,560	98,304	24.7
Total	313,824	217,632	30.6

Table 3 presents the required memory bits considering that all wedgelets patterns should be stored. The comparison between storing all wedgelets and removing the refinement shows memory savings higher than 30%. Due to the massive memory resources that should be spent to store all available refinements wedgelets, the removal of DMM-1 refinement in a hardware design presents a sound tradeoff between area, performance, and coding efficiency.

3 DMM-1 ARCHITECTURE

This section describes the proposed scalable architecture to implement the DMM-1 algorithm without the refinement stage. Section 3.1 presents the main core of the architecture, which can be replicated to reach the scalability levels. Section 3.2 presents the high-level structure of the designed architecture that is similar to the structure proposed in [5]. However, the core architecture is substituted by the D-Core proposed in 3.1. The D-Core used in this work is smaller than the core used in [5] since it does not include features to encode DMM-4.

3.1 DMM-1 Core (D-Core) Architecture

Fig. 4 illustrates the DMM-1 core architecture (D-Core). Each D-Core receives a pixel value (PIXEL_IN signal) and stores it in a register. In the prediction step (i.e., STAGE(1)=0), the average value of each region should be computed. In this step, the architecture selects the stored pixel to be added to the previous value of the region that the current pixel belongs (i.e., depending on signals PREVIOUS_0 or PREVIOUS_1).

In the SAD step (i.e., STAGE(1)=1 and STAGE(0)=0), the SAD of each wedgelet is computed. Thus, the stored pixel is subtracted from the average value of the region it belongs (i.e., AVG_0 or AVG_1). Later, the absolute value of this result is added to the values generated by previous D-Cores (i.e., PREVIOUS_0 and PREVIOUS_1).

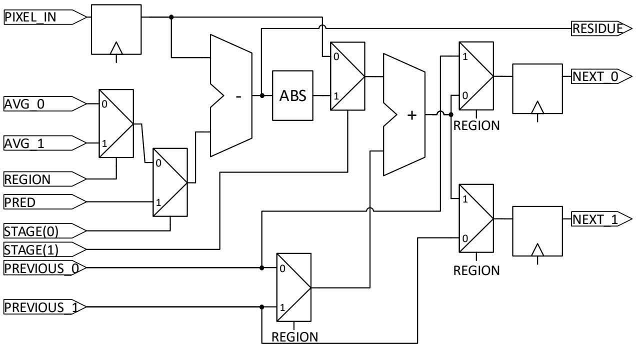

Figure 4: Schematic of the D-Core architecture.

During the Residue Stage (i.e., STAGE(0)=1), the core computes the residues between the predicted sample and the original signal subtracting PRED, which is the value of the predicted sample, from the stored pixel. The obtained results are sent to the output (RESIDUE).

3.2 The Scalable DMM-1 Encoding Architecture

Fig. 5 shows the proposed architecture for the DMM-1 coding tool. This regular and scalable structure comprises (i) N×N D-Cores array; (ii) two memories for storing the DMM-1 patterns (PATTERNS MEMORY) and the division values (DIVIDER MEMORY); (iii) two adder trees; (iv) two dividers; (v) a comparator; and (vi) a register bank. The architecture supports all HEVC block sizes of N×N bytes (i.e., 4×4 to 32×32).

During the first N cycles of execution, one line containing N D-Cores are filled with PIXEL_IN signal (1 byte); consequently, the architecture requires a bandwidth of N bytes. Then, the D-Cores start processing each stage of the DMM-1 algorithm. The signals of the N D-Cores positioned in the east column of each N×N D-Cores array are added in the adder trees. This result is sent to a divider in the prediction step, which is responsible for obtaining the average value of each region. Additionally, the result is sent to the comparator in the SAD step, which is responsible for finding the best SAD among all patterns. Then, the comparator stores the best SAD, the pattern and, the average value of each region.

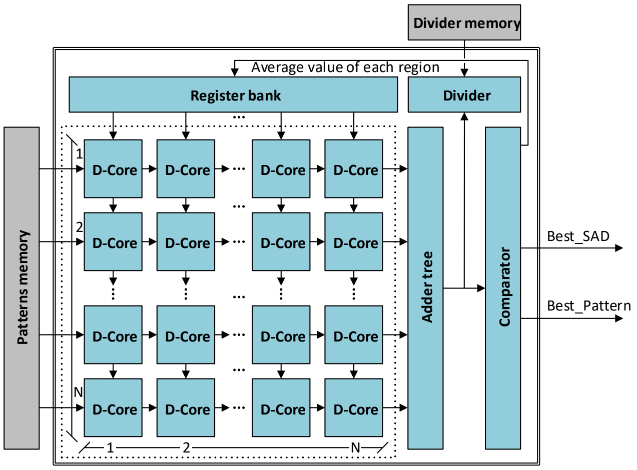

Figure 5: Architecture of DMM-1 coding tools for *N*×*N* blocks.

During both, prediction step and SAD step, the information of Region 0 and 1 of the previous cores (i.e., NEXT_0 and NEXT_1, respectively) are sent in pipeline stages to the next cores, which are connected to PREVIOUS_0 and PREVIOUS_1 of the following core to achieve higher performance. The east cores send their next values to the adder tree; thus, all values can be added in one pipeline stage. Additionally, the prediction step and the SAD step are performed in an interlaced way. When a pattern reached the end of the prediction step (i.e., the division is computed), the average value of each region is stored in the register bank, and this information is feedback to process the SAD of that pattern. When all SADs have been computed, the average value of the best pattern is feedback to the register bank according to the region the sample belongs, which will be inserted as the signal PRED in the D-Cores to compute the residues.

4 EXPERIMENTAL RESULTS

We designed the scalable architecture in VHDL and synthesized for ST 65 nm intending to evaluate the hardware performance. The architecture takes 132, 648, 816 and 876 clock cycles to process an entire block, according to the block size, being N for filling the D-Cores with pixels, N for the residues (which is the last stage) and the remaining for the prediction and SAD steps. This architecture computes N+2 patterns in parallel - N patterns are calculated in D-Cores, one in the adder trees and one in the dividers.

Table 4 shows the synthesis results of these experiments, where the architecture is capable of achieving real-time processing of HD 1080p videos for all available block sizes. Moreover, Table 4 presents a comparison against Sanchez et al. [5] showing that this new architecture is capable of reducing area, in average, by 21% with an average increase in power dissipation of 17%.

Table 4: Synthesis results targeting ST 65 nm

Element	***Bipartition* [5] *N*x*N* Configuration**			
	4x4	**8x8**	**16x16**	**32x32**
Gates	4,070	15,749	49,628	211,950
Freq. (MHz)	515,5	632.9	198.8	53.2
Cycles/block	134	650	818	878
1080p fps	30.1	30.0	30.0	30.0
Power (mW)	22.2	90.9	133.1	166.5
Element	***DMM-1* [This Work] *N*x*N* Configuration**			
	4x4	**8x8**	**16x16**	**32x32**
Gates	3,291	12,564	37,474	165,484
Freq. (MHz)	514.3	629.9	198.3	53.2
Cycles/block	132	648	816	876
1080p fps	30.0	30.0	30.0	30.0
Power (mW)	25.2	113.4	163.0	182.4

5 CONCLUSIONS

This paper presented a scalable architecture for the DMM-1 encoding algorithm based on the 3D-HEVC standard. The refinement stage was not included in this architecture, reducing in 30% the necessary memory size with a drawback of only 0.09% of the BD-rate increase in the synthesized views. Synthesizing the DMM-1 architecture with ST 65nm Standard Cells Technology, the experimental results shows that it was capable of processing HD 1080p real-time videos for all block sizes. Besides, the proposed architecture saves area for all block sizes when compared with related work.

ACKNOWLEDGMENTS

This paper was achieved in cooperation with Hewlett-Packard Brazil Ltda. using incentives of Brazilian Informatics Law (Law nº 8.248 of 1991). Authors also would like to thank CAPES (processes 88881135737/2016-01 and 88881119481/2016-01), CNPq (processes 309707/2015-3 and 486136/2013-2) and FAPERGS (process 16/2551-0000241-0) Brazilian research agencies to support the development of this work.

REFERENCES

[1] JCT-3V. Available at: <//phenix.int-evry.fr/jct2/>, access in Ago. 2015.

[2] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, R. Tanger. "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Image Communication*, v. 22, n. 2, pp. 217-234, Feb. 2007.

[3] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, T. Wiegand. "Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems," *IEEE International Conference on Image Processing (ICIP)*, pp. 2448-2451, 2008.

[4] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, H. Lakshman, P. Merkle, F. Rhee, G. Tech, M. Winken, T. Wiegand. "3D High-Efficiency Video Coding for Multi-View Video and Depth Data," *IEEE Transactions on Image Processing*, v. 22, n. 9, pp. 3366-3378, Sep. 2013.

[5] G. Sanchez, C. Marcon, L. Agostini. "Real-time scalable hardware architecture for 3D-HEVC bipartition modes," *Journal of Real-Time Image Processing*, 2016.

[6] G. Sanchez, B. Zatt, M. Porto, L. Agostini. "A real-time 5-views HD 1080p architecture for 3D-HEVC Depth Modeling Mode 4," *Symposium on Integrated Circuits and Systems (SBCCI)*, pp. 1-6, 2014.

[7] Y. Chen, G. Tech, K. Wegner, S. Yea. "Test Model 11 of 3D-HEVC and MV-HEVC", *document JCT3VK1003 of JCT-3V*, Geneva, CH, Feb. 2015.

[8] L. Zhao, L. Zhang, S. Ma, D. Zhao. "Fast Mode Decision Algorithm for Intra Prediction in HEVC," *IEEE Visual Communications and Image Processing (VCIP)*, pp. 1-4, 2011.

[9] Z. Gu et al. "Fast Intra Prediction Mode Selection for Intra Depth Map Coding," ISO/IEC JTC1/SC29/WG11, Vienna, Aug. 2013.

[10] D. Rusanovskyy, K. Muller, A. Vetro. "Common Test Conditions of 3DV Core Experiments," ISO/IEC JTC1/SC29/WG11 MPEG2011/N12745, Geneva, Jan. 2013.