Robust 3D LUT Estimation Method for SHVC Color Gamut Scalability

Yuwen He [1], Yan Ye [2], Jie Dong [3]

InterDigital Communications
9710 Scranton Road, San Diego, CA 92121, USA
[1]yuwen.he@interdigital.com, [2]yan.ye@ interdigital.com, [3]jdong@avsx.org

***Abstract*— Color gamut scalability (CGS) in scalable extensions of High Efficiency Video Coding (SHVC) supports scalable coding with multiple layers in different color spaces. Base layer conveying HDTV video in BT.709 color space and enhancement layer conveying UHDTV video in BT.2020 color space is identified as a practical use case for CGS. Efficient CGS coding can be achieved using a 3D Look-up Table (LUT) based color conversion process. This paper proposes a robust 3D LUT parameter estimation method that estimates the 3D LUT parameters globally using the Least Square method. Problems of matrix sparsity and uneven sample distribution are carefully handled to improve the stability and accuracy of the estimation process. Simulation results confirm that the proposed 3D LUT estimation method can significantly improve coding performance compared with other gamut conversion methods.**

***Index Terms*— scalable video coding, color gamut scalability, inter-layer prediction, color gamut, 3D LUT**

I. Introduction

The scalable extension of a single layer video coding standard is aimed at providing efficient and robust support for video applications with heterogeneous characteristics such as error prone networks, varying channel bandwidth, and heterogeneous device capabilities (temporal/spatial resolutions, 2D/3D, and backward compatibility). Various scalable coding extensions were developed since MPEG-2. High Efficiency Video Coding (HEVC) [1] is the latest video coding standard developed by the Joint Collaborative Team on Video Coding (JCT-VC), and it was finalized in early 2013. The use cases and requirements of the HEVC scalable extension (SHVC) were extensively discussed [2]. Support for view, spatial, temporal, quality, standard, bit-depth, color gamut, chroma format, aspect ratio scalability, etc., was considered. Among these scalability types, color gamut scalability provides an essential path for HD to UHD migration with improved user experience. This is because UHDTV, as defined by the recommendations ITU-R BT.2020 [5], uses an expanded color space to enhance high fidelity color rendering. Fig. 1 shows the comparison between BT.709 (HDTV) and BT.2020 (UHDTV) in CIE color definition. As shown, the volume of visible colors that can be rendered in BT.2020 is much broader than that in BT.709.

L. Kerofsky et al. [6] first proposed color gamut scalable coding methods for SHVC. They compared the chromaticity between the BT.709 and BT.2020 color spaces, and proposed two color gamut conversion methods: (1) gain-offset predictor, and (2) cross-color predictor. The first method is a linear model based prediction without cross color component support, that is, each color component is predicted independently. The second method is also a linear model based prediction, but allows cross color component support. The second method gives better coding performance in their initial tests. Since then the JCT-VC set up an ad-hoc group (AHG 14) to further investigate color gamut scalability. At the 14th JCT-VC meeting, Technicolor [7] provided five test sequences for color gamut scalability tests. A snapshot of each of these sequences (the BT.709 version) is shown in Fig. 2. For each video sequence, two versions, one in BT.709 color primaries and the other in BT.2020 color primaries, were provided. The BT.709 sequences were professionally color graded with BT.709 reference displays; and the BT.2020 sequences were professionally color graded with DCI-P3 [20] reference displays and then converted to BT.2020 color primaries. These test sequences are used to develop the color gamut scalability feature in SHVC. In addition to the color gamut conversion methods proposed in [6], P. Bordes et al. also proposed the 3D LUT based color gamut conversion method [8][12]. Additionally, X. Li et al. [14] and A. Aminlou et al. [15] investigated using Weighted Prediction as already supported in the single layer HEVC standard to improve the enhancement layer coding efficiency for CGS. Core Experiments (CE) [11] were set up to investigate CGS coding efficiency under test conditions suitable for the practical use case of HD to UHD migration, in which the base layer format is 8-bit/10-bit BT.709 1080p and the enhancement layer is 10-bit BT.2020 3840x2160p. Various color gamut conversion methods were tested in this CE, including weighted prediction [14][15], gain offset [16][18], piece-wise linear [17], and 3D LUT [12][13].

In this paper we propose a robust parameter estimation method for 3D LUT based color gamut conversion. The proposed method estimates the color conversion parameters between two different color spaces using global optimization based on the Least Square method. The problem of sample sparsity and uneven distribution of input data in the 3D color space is solved by a novel matrix compaction method. Simulation results show that the proposed method can produce more accurate 3D LUT parameters compared to the existing parameter estimation methods [12], and improve the inter-layer reference picture quality for CGS coding.

The remainder of this paper is organized as follows. Section 2 gives an overview of color gamut conversion with 3D LUT. Section 3 explains in detail the proposed global

978-1-4799-6139-9/14/$31.00

parameter estimation for 3D LUT. Section 4 gives simulation results by comparing 3D LUT with other color gamut conversion methods, and comparing different 3D LUT estimation methods. Section 5 concludes the paper.

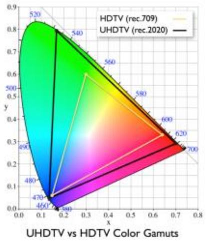

Fig. 1 BT.2020 and BT.709 color primaries comparison in CIE space

Fig. 2 Thumbnails of the BT.709 version of five CGS test sequences

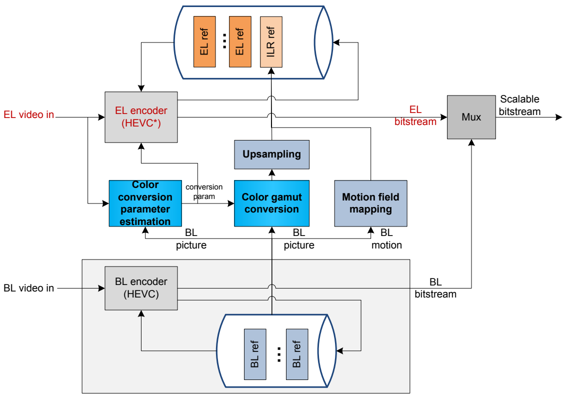

Fig. 3 Diagram of an SHVC CGS encoder

II. OVERVIEW OF 3D LUT BASED COLOR GAMUT SCALABILITY

SHVC [3][21] adopts a layered coding framework with inter-layer processing. The enhancement layer codec is kept the same as the single layer HEVC codec except some high level (that is, slice header level and above) syntax changes. The enhancement layer can reference the reconstructed sample and motion information from the reference layer via the inter-layer reference (ILR) picture. When there is a spatial resolution difference between the two layers, the ILR picture is generated by texture upsampling and motion field mapping using the reconstructed picture from the reference layer. A color gamut conversion process is added if there is a color space difference between the two layers. The ILR picture is inserted into the decoded picture buffer of the enhancement layer as a long-term reference picture. Fig. 3 shows a diagram of an SHVC CGS encoder. The color conversion parameters are estimated using the pair of enhancement layer input picture and base layer reconstructed picture, and coded in the bit-stream. The color gamut conversion process is performed using those estimated parameters. Color gamut conversion is placed before upsampling in order to reduce the complexity.

In the film industry, 3D Look-up Table (LUT) [9] is widely used for color gamut conversion, such as color grading or color management. 3D LUT based CGS coding was first proposed by Bordes et al. [8]. Large coding improvements were reported over the simple gain-offset model because 3D LUT can accurately represent the color conversion process during content generation. Denote (y, u, v) as the sample triplet in the BL color gamut, and (Y, U, V) as the sample triplet in EL color gamut. Using 3D LUT, the input color space is partitioned into small cubes in a symmetric or asymmetric way. Fig. 4 shows an example where the entire 3D color space is partitioned into 64 (4x4x4) cubes, or octants, of equal size: if the input bit depth is 8-bit, then each octant will have a size of 64x64x64. The 3D LUT thus generated will store the mapped colors for all 125 (5x5x5) vertices. For samples not located at one of the 125 vertices, interpolation is applied using values from the nearest vertices. For example, trilinear interpolation [10] or tetrahedral interpolation can be used. Trilinear interpolation uses 8 nearest vertices, and tetrahedral interpolation uses 4 vertices of the tetrahedron containing the input sample. Both interpolation methods are linear. Fig. 5 shows the tetrahedral interpolation method. Assume the input sample P in Fig. 5 is located inside the tetrahedron whose vertices are P_0, P_1, P_5, P_7. Tetrahedral interpolation is calculated as in Equation (1).

$$\begin{aligned} X = {} & LUT_X[P_0] + T_y \times dy \times (LUT_X[P_1] - LUT_X[P_0]) \\ & + T_u \times du \times (LUT_X[P_5] - LUT_X[P_1]) \\ & + T_v \times dv \times (LUT_X[P_7] - LUT_X[P_5]) \end{aligned} \qquad (1)$$

Where "X" can be the Y, U, or V component. $LUT_X[P]$ is the output of 3D LUT of component X at the vertex P, (dy, du, dv) are the relative distances between P and the vertex P_0 in each color dimension, and T_y, T_u and T_v are calculated as follows. For each color component, a separate 3D LUT is estimated and transmitted.

$$T_y = \frac{1}{(y_1 - y_0)},\ T_u = \frac{1}{(u_1 - u_0)},\ T_v = \frac{1}{(v_1 - v_0)}$$

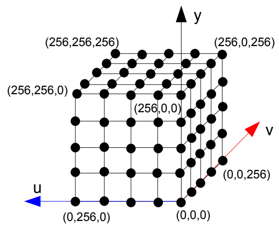

Fig. 4 An example of 3D LUT for 8-bit YUV input

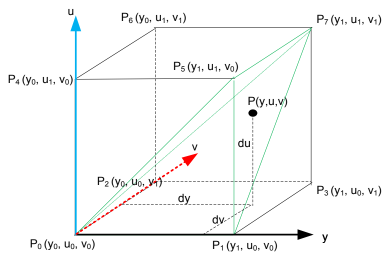

Fig. 5 Tetrahedral interpolation used in 3D LUT interpolation ([13])

The problem of 3D LUT estimation is to estimate, for each color component X (X can be Y, U, or V), the output values $LUT_X[P]$ of all octant vertices P. P. Bordes et al. [8][12] proposed the local parameter estimation method with the following three steps: 1) For each color component and each octant, parameters of a non-vertex based cross-color linear model is estimated with the Least Square (LS) method; 2) The estimated parameters of the non-vertex based cross-color linear model from 1) are translated to the 8 vertices of each octant; 3) Those vertices shared among neighboring octants are merged using weighted averaging, where the weights are calculated based on the number of input samples in each neighboring octant.

Two things affect the estimated parameter's accuracy using the local estimation method in [8][12]. Firstly, for ease of implementation, the local method of [8][12] uses a non-vertex based cross-color linear model to solve the LS problem in step 1). It then takes another step 2) to translate the model parameters into vertex values. This causes a mismatch between the model used in parameter estimation and the actual model used in 3D LUT interpolation during actual color gamut conversion process. Secondly, because the local estimation process cannot guarantee parameter continuity at the boundaries of neighboring octants, step 3) is necessary to merge the parameters at shared vertices among neighboring octants. Such separated steps are not optimal, especially given the merging process in [8][12] considers the number of samples instead of the actual distortion caused by merging. To improve parameter estimation accuracy, we propose the global parameter estimation method, which estimates all parameters for all vertices jointly with the LS method. The proposed method uses the same model during estimation and during actual color gamut conversion. It also automatically guarantees parameter continuity among neighboring octants and can eliminate the need for a merging step.

III. ROBUST 3D LUT ESTIMATION METHODOLOGY

The color gamut conversion process with 3D LUT can be modeled as a linear process with either tetrahedral or trilinear interpolation. There are three challenges in global 3D LUT estimation: (1) The number of 3D LUT parameters to be estimated is large. For 8 bit video, if the unit octant size is 16x16x16, then there are 17x17x17 entries in the 3D LUT, or equivalently 4913 (17x17x17) unknown parameters for each component. Such large scale linear system estimation needs a large amount of memory and computation. (2) Not all entries of the 3D LUT are fully used for a given video input. For some CGS test sequences, if a LUT with 17x17x17 entries is used, the number of used entries in the 3D LUT is less than 1000, or only about 20%. Least Square cannot be directly applied when there are insufficient input samples to estimate all entries. (3) Input base layer samples are not evenly distributed in the 3D color space. They mainly cluster around a limited number of major colors and are distributed sparsely elsewhere. This unbalanced characteristic will cause instability of the LS estimation process.

Take BT.709 to BT.2020 color gamut conversion for example. There are two inputs to 3D LUT estimation: the BT.709 compressed video and the BT.2020 video (the estimation target). We use Equation (2) to describe the color gamut conversion process with 3D LUT.

$$z_i(c) = f_{P(c)}(x_i), i \in [0, N-1] \quad (2)$$

Where x denotes the input signal in the form of a triplet (y,u,v) in BT.709. z(c) is the output signal of component c, with c being Y, U or V in BT.2020, P(c) is the 3D LUT vertex parameters of component c to be estimated, $f_{P(c)}$ is the tetrahedral interpolation function, which is linear, i is the index of input sample, and N is the total number of input samples. We rewrite (2) in the matrix form:

$$z_i(c) = \vec{w}_i(c) * P(c), i \in [0, N-1] \quad (3)$$

Where "*" is matrix multiplication, $\vec{w}_i(c)$ is the weighting vector for the i-th input sample, and $P(c)$ is the parameter vector to be estimated. The weighting vector $\vec{w}_i(c)$ consists of entries $w_{i,j}$, which is the weight of the j-th output entry of the 3D LUT for the i-th input sample, and can be calculated according to tetrahedral interpolation in Equation (1).

$$\vec{w}_i(c) = [w_{i,0} \quad \cdots \quad w_{i,M-1}], i \in [0, N-1]$$

The parameter vector to be estimated, $P(c)$, is written as:

$$P(c) = [p_0 \quad \cdots \quad p_{M-1}]$$

Where M is the number of 3D LUT output entries, for example, M is 4913 for a 3D LUT with 17x17x17 entries. For simplicity, we omit the component c in the following equations because the 3D LUT of each component can be estimated independently. By aggregating Equation (3) for all samples, we get:

$$Z = W * P \quad (4)$$

$$Z = [z_0 \quad \cdots \quad z_{N-1}]^T$$

$$W = \begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N-1,0} & w_{N-1,1} & \cdots & w_{N-1,M-1} \end{bmatrix} \quad (5)$$

With Least Square estimation, we can get the solution as in Equation (6), where H is the auto-correlation matrix in (7).

$$P = H^{-1} * (W^T * Z) \quad (6)$$

$$H = (W^T * W) \quad (7)$$

When only small portions of the 3D LUT entries are used in color conversion, the matrix W in Equation (5) is sparse with a lot of zero elements. Therefore H in Equation (7) is also sparse and may not be invertible. Thus the solution in Equation (6) may not be available. Our solution is to compact the parameter vector P by only considering those referenced 3D LUT entries. First we scan all input samples (y,u,v) of the BL reconstructed video, and mark all those vertices in the 3D LUT that will be used in the interpolation process. After that, we can get the compact parameter vector P' by removing all those unused vertices.

$$P' = compact(P)$$

We can calculate W' and H' accordingly by removing the corresponding rows/columns that are removed from the compacted P'. Then the solution will be defined as:

$$P' = H'^{-1} * (W'^T * Z) \quad (8)$$

$$H' = (W'^T * W') \quad (9)$$

With this method, the sparsity problem of the matrix W is solved. Additionally, the memory needed to store the auto-correlation matrix H is greatly reduced after compaction, since the size of H' is much smaller than H.

However, colors in the input video usually are not evenly distributed in the 3D color space. A few major colors have very high occurrence rates, whereas other colors rarely occur. This causes H' to be unbalanced: the elements in H' that correspond to major colors have very large values, whereas other elements in H' have very low values. The result is that the dynamic range of elements in H' is very large, which may cause the inversion of H' and the estimation process of P' to become unstable. To solve this problem, we introduce a constraint to establish a trade-off between accuracy and stability of the estimation process.

$$H' = (W'^T * W') + \lambda I, \ \ \lambda \geq 0 \qquad (10)$$

Where I is the unary matrix, and λ is the factor to balance between accuracy and stability. Larger λ puts more bias on the stability of the process. λ is set to 0.01 in our implementation.

After the compact parameter vector P' is estimated, we reconstruct the original parameter vector P by filling the unused (and un-estimated) vertices in P using the neighboring vertices in P' using an appropriate interpolation method (e.g. tetrahedral or trilinear): $P = decompact(P')$

IV. SIMULATION RESULTS

The CGS core experiments [11] defined two performance tests. The EL format is 10-bit 3840x2160p in BT.2020 for both tests. The BL format is 8-bit 1920x1080p in BT.709 for Test 1, and 10-bit 1920x1080p in BT.709 for Test 2. Detailed test descriptions are given in [11]. In TABLE I and TABLE II, results of the proposed 3D LUT estimation method are reported in comparison with different color gamut conversion methods, including the local 3D LUT estimation method proposed in JCTVC-O0159 [12]. For the proposed method, two 3D LUT sizes, 9x9x9 and 17x17x17, are tested. The Bjontegaard Delta rate (BD-rate) reduction considering the overall (BL+EL) bitrate is used to measure the CGS coding efficiency. The anchors in TABLE I and TABLE II are based on SHM-3.0.1 [19]. In O0159 and in this paper, 3D LUT is derived and signaled once per sequence in the Picture Parameter Set extension, and is used for the whole sequence. Compared to the other CGS methods using weighted prediction (O0194, O0180), gain offset (O0195, O0201), and piecewise linear (O0196), 3D LUT based methods (O0159, proposed method) can achieve much bigger coding gains (7~10%). For 3D LUT based methods, global estimation method can additionally improve coding efficiency by ~2% compared with local estimation method (O0159).

TABLE I
AVERAGE BD-RATE COMPARISON OF DIFFERENT CGS METHODS: TEST 1

	SCE4	AI 2x 8-bit base			RA 2x 8-bit base		
Proposals	**Technology**	**Y(%)**	**U(%)**	**V(%)**	**Y(%)**	**U(%)**	**V(%)**
O0194 [15]	WP (adp.)	-6.6	-5.5	-9.5	-3.4	-1.0	-4.4
	WP (adp+fixed)	-6.6	-5.5	-9.5	-3.7	-1.3	-4.7
O0201 [18]	GO	-4.6	-3.3	-7.3	-2.9	-0.2	-3.7
O0196 [17]	Piecewise-linear	-4.4	-3.0	-7.9	-3.3	-0.1	-4.6
O0195 [16]	Region adaptive GO	-6.9	-5.6	-10.1	-3.0	-0.4	-3.7
O0180 [14]	WP	-6.0	-4.7	-9.0	-3.5	-1.1	-4.9
O0159 [12]	LUT (9x9x9)	-12.3	-9.9	-16.0	-8.2	-3.0	-9.9
Proposed method	LUT 10-bit (9x9x9)	-14.6	-14.1	-21.0	-9.7	-7.0	-14.9
Proposed method	LUT 10-bit (17x17x17)	-15.3	-15.7	-22.9	-10.0	-8.7	-16.6

TABLE II
AVERAGE BD-RATE COMPARISON OF DIFFERENT CGS METHODS: TEST 2

	SCE4	AI 2x 10-bit base			RA 2x 10-bit base		
Proposals	**Technology**	**Y(%)**	**U(%)**	**V(%)**	**Y(%)**	**U(%)**	**V(%)**
O0201 [18]	GO	-4.0	-2.3	-6.0	-2.5	-0.1	-3.7
O0196 [17]	Piecewise-linear	-4.1	-2.8	-6.8	-3.1	-0.6	-4.7
O0159 [12]	LUT (9x9x9)	-12.2	-9.6	-14.9	-8.5	-3.4	-10.1
Proposed method	LUT 10-bit (9x9x9)	-14.7	-13.8	-20.2	-10.2	-7.5	-15.5
Proposed method	LUT 10-bit (17x17x17)	-15.4	-15.5	-22.2	-10.5	-9.1	-17.2

V. CONCLUSIONS

In this paper, a robust 3D LUT estimation method is proposed to address the challenges in global 3D LUT parameter estimation. Simulations confirm that significant coding performance gain can be achieved with the proposed 3D LUT estimation method for color gamut scalability.

VI. REFERENCES

[1] G. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, Issue 12, 2012.

[2] A. Luthra, E. François, W. Husak, "Draft requirements, test conditions and exploratory experiments for High Dynamic Range and Wide Color Gamut video coding for Broadcasting, OTT, and Storage Media," MPEG2014/M32716, Valencia, ES, Mar. 2014.

[3] J. Chen, J. Boyce, Y. Ye, M. Hannuksela, G. J. Sullivan, Y. -K. Wang, "High efficiency video coding (HEVC) scalable extension Draft 6," JCTVC-Q1008, Valencia, ES, Mar. 2014.

[4] ITU-R Rec. BT.709, "Parameter values for the HDTV standards for production and international programme exchange," Dec. 2010.

[5] ITU-R Rec. BT.2020, "Parameter values for UHDTV systems for production and international programme exchange," April 2012.

[6] L. Kerofsky, A. Segall, S.-H. Kim, K. Misra, "Color Gamut Scalable Video Coding," JCTVC-K0241, 11th JCTVC Meeting, Shanghai, CN, 10–19 Oct. 2012.

[7] P. Andrivon, P. Bordes, "AHG14: Wide Color Gamut Test Material Creation," JCTVC-N0163, Vienna, AT, Jul. 2013.

[8] P. Bordes, P. Andrivon, R. Zakizadeh, "AHG14: Color Gamut Scalable Video Coding using 3D LUT," JCTVC-M0197, 13th JCTVC Meeting, Incheon, KR, 18–26 Apr. 2013.

[9] 3D lookup table, http://en.wikipedia.org/wiki/ 3D_lookup_table

[10] Trilinear interpolation, http://en.wikipedia.org/wiki/ trilinear_interpolation

[11] A.Segall, P.Bordes, C.Auyeung, X.Li, E.Alshina, A.Duenas, "Description of Core Experiment SCE4: Color Gamut and Bit-Depth Scalability," JCTVC-N1104, Vienna, AT, July 2013.

[12] P. Bordes, P. Andrivon, F. Hiron, P. Salmon, R. Boitard, "SCE4: Results on 5.3-test1 and 5.3-test2," JCTVC-O0159, Geneva, Oct. 2013.

[13] Y. He, Y. Ye, J. Dong, "Non-SCE4/AHG14: Combined bit-depth and color gamut conversion with 3D LUT for SHVC color gamut scalability," JCTVC-O0161, Geneva, CH, Oct. 2013.

[14] X. Li, V. Seregin, J. Chen, K. Rapaka, Y. Chen, M. Karczewicz, "Non-SCE4: Weighted Prediction Based Color Gamut Scalability," JCTVC-O0180, Geneva, CH, Oct. 2013.

[15] A. Aminlou, K. Ugur, M. M. Hannuksela, E. Alshina, A. Alshin, "SCE4: Test 5.1 results on bit-depth and color-gamut scalability," JCTVC-O0194, Geneva, CH, Oct. 2013.

[16] C. Auyeung, "Non-SCE4: Picture and region adaptive gain-offset prediction for color space scalability," JCTVC-O0195, Geneva, CH, Oct. 2013.

[17] C. Auyeung, "SCE4: Results of test 5.4-model1 on piecewise linear color space predictor," JCTVC-O0196, Geneva, CH, Oct. 2013.

[18] J. Zhao, S. Deshpande, K. Misra, S.H. Kim, "SCE4 Test5.2: Color prediction with Gain-Offset model,"JCTVC-O0201, Geneva, Oct. 2013.

[19] SHVC reference software, SHM-3.0.1, https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/tags/SHM-3.0.1

[20] "Digital Cinema System Specification version 1.2," Digital Cinema Initiatives, October 2012.

[21] G.J. Sullivan, J.M. Boyce, Y. Chen, J-R. Ohm, C.A. Segall, A. Vetro, "Standardized Extensions of High Efficiency Video Coding (HEVC)", *IEEE Journal of Selected Topics in Signal Processing*, Dec. 2013.