|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  23rd Meeting: San Diego, USA, 19–26 February 2016 | Document: JCTVC-W0031 m37536 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Description of the reshaper parameters derivation process in ETM reference software** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | K. Minoo (Arris) T. Lu, P. Yin (Dolby) L. Kerofsky (InterDigital) D. Rusanovskyy (Qualcomm) E. François (Technicolor) | Tel: Email: | [Koohyar.Minoo@arris.com](mailto:Koohyar.Minoo@arris.com) [tlu@dolby.com](mailto:tlu@dolby.com) [Louis.Kerofsky@interdigital.com](mailto:Louis.Kerofsky@interdigital.com) [dmytror@qti.qualcomm.com](mailto:dmytror@qti.qualcomm.com) [edouard.francois@technicolor.com](mailto:edouard.francois@technicolor.com) |
| *Source:* | Arris, Dolby, InterDigital, Qualcomm, Technicolor | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This document describes the reshaping process algorithms implemented in the reference ETM software.

# Introduction

MPEG document N15792 describes the Exploratory Test Model (ETM) for HDR extension of HEVC. This document aims at describing in more details the HDR video analysis and processing algorithms developed in the ETM reference software, applied to derive the reshaping parameters.

Section 2 provides an overview of the ETM framework, and a short description of the different reshaping parameters derivation modes. Section 3 describes in more details these different derivation processes.

# ETM framework overview

A functional diagram of the ETM is shown in Figure 1. The system uses HEVC Main 10 profile for the bitstream generation and bitstream decoding, and uses meta-data provided by the decoder to control decoder side processing used to reconstruct an HDR and WCG representation. The input HDR signal is pre-processed to produce a modified HDR signal that is provided to the HEVC main 10 encoder. The HEVC main 10 decoder output is used to reconstruct the HDR signal. The pre-processing and post-processing steps primarily aim at improving the coding efficiency of HDR content, and at providing support of SDR backward compatibility.



Figure . System diagram.

In the current ETM implementation, the input HDR signal is actually first converted in the HDR analysis and processing step to 10 bits 4:2:0 YCbCr signal derived from the input linear-light content. The reshaping functions therefore apply to this YCbCr signal. At the decoder side, the HDR reconstruction process, as implemented in the HM decoder, generates 10 bits 4:2:0 YCbCr samples that can be further converted to linear light (e.g. using HDRConvert). The initial transfer function for converting the input linear-light signal to YCbCr 10 bits is the PQ. Howver another transfer function could also be considered.

## Luma and chroma reshaping models

The inverse reshaper for luma is based on a piece-wise polynomial (PWP) model, with 8 pieces. At the decoder side, a reconstructed luma sample Yinvr is derived using the following equation from the decoded luma sample Ydec:

Yinvr = a0j + a1j . Ydec + a2j . Ydec2 ( 1 )

where j is the index of the piece-wise segment to which Ydec belongs, and aij is the ith-order polynomial coefficient for the j-th segment.

For chroma, the inverse reshaper is based on a piece-wise linear (PWL) model, with up to 32 pieces. Two methods are supported for inverse chroma reshaping.

* For method 0 (auto-plane chroma reshaping), input sample value Cdec is mapped to an output sample value Cinvr as following:

Cinvr = Cdec \* Scale[ i ] + Offset[ i ] ( 2 )

with i being the index of the piece-wise segment where Cdec belongs to, Scale[i] and Offset[i] being the linear model parameters of the i-th segment.

* For method 1 (cross-plane chroma reshaping), input chroma sample Cdec is mapped to an output sample value Cinvr as following. A scaling factor sc dependent on the co-located luma sample Ydn is derived as follows:

sc = Ydn\* Scale[ i ] + Offset[ i ] ( 3 )

with i being the index of the piece-wise segment where Ydn belongs to, Scale[i] and Offset[i] being the linear model parameters of the i-th segment. Then Cinvr is obtained by scaling Cdec using the scaling factor sc:

Cinvr = offset1 + sc \* ( Cinvr – offset2 ) ( 4 )

## Basics of HDR analysis and processing in ETM software

HDR analysis and processing applies prior to the HEVC encoding. It maps the input HDR signal to a format adapted to the HEVC Main 10 profile.

The reshaping parameters are derived in the ETM reference software according to three possible configuration modes (defined by parameter Reshape\_mode).

* Reshape mode 0 is focused on HDR-only feature, and aims at improving HDR compression performance. In this mode, chroma reshaping is achieved using chroma method 0 (auto-plane chroma reshaping).
* Reshape mode 1 is addressing the SDR-backward compatibility feature, while also targeting good HDR compression performance. In this mode, chroma reshaping is achieved using chroma method 1 (cross-plane chroma reshaping), using two separate piece-wise linear model parameters for Cb and Cr chroma components.
* Reshape mode 2 is addressing the SDR-backward compatibility feature, while also targeting good HDR compression performance. In this mode, chroma reshaping is achieved using chroma method 1 (cross-plane chroma reshaping), using one or two piece-wise linear model parameters (decided by the input parameters) for Cb and Cr chroma components.

These three reshaping parameters derivation modes are described on more details in the next sections.

# Algorithms for reshaping look-up-tables derivation

## Mode 0 – HDR-only

The reshaping look-up-table derivation process in mode 0 uses the following steps:

* Image analysis and metadata update mechanism
* Luma reshaping parameters derivation
* Chroma reshaping parameters derivation

### Image analysis and metadata update mechanism

**Image analysis**: This process performs basic analysis for an input HDR image. Basic image analysis referrers to the computation of a set of image statistics, such as the maximum and minimum values of each component, and block based average component values and standard deviations. These statistics are stored for usage of the next steps.

**Metadata update mechanism**: This process makes decisions when new reshaping metadata should be derived. In current ETM software, the metadata update is sent at every scene change. In ongoing CE2 experiment, the process would automatically make decisions for necessary updates.

**Note:** In current ETM software, the scene information is read from an external cfg file for each test clips. The cfg file contains information for each scene (per line) in the following format:

scene\_start\_POC, actual\_scene\_gamut (0:709, 3:P3), fading\_flag (0:non-fading) ( 5 )

For each scene, the first frame is analyzed to find the original source min/max value for luma channel in the normalized range [0.0 1.0], i.e., org\_L\_CL and org\_L\_CH.

This is an interim implementation from m37267 based on ETM integration plan. The external scene information cfg file will be eliminated in planned CE2 experiments and fully automatic metadata updating mechanism would be introduced.

### Derivation of luma reshaping parameters

For luma component, a power function is used as forward luma reshaping model:

y = a \* (x + b)α + c

The luma reshaping algorithm is described as in the following ordered steps:

1. Based on the image statistics, a fixed heuristically determined table and threshold values are used to decide α value and L\_factor. The targeted min/max value L\_CH and L\_CL are computed as:

* L\_range = L\_factor \* 1.0
* L\_CH = 0.5 + L\_range/2.0 ( 6 )
* L\_CL = 0.5 - L\_range/2.0

1. Generate inverse reshaping LUT using the inverse of forward reshaping function.

LUT(x) = (((x/1023- I\_CL)/(I\_CH - I\_CL))1.0/α \*   
 (org\_I\_CH - org\_I\_CL) + org\_I\_CL) \* 1023 ( 7 )

where x is the input luma value in the range of [0 1023] (full range 10 bit). If x is the input luma value in standard range, the weight and offset values for fixed to float and float to fixed conversion is applied based on the definition of standard range conversion.

1. Approximate PWP coefficients for inverse reshaping LUT for luma channel, as described in section 4. The PWP coefficients are to be transmitted in HEVC bitstream.
2. Based on PWP coefficients from 3), build luma inverse reshaping LUT using luma PWP function from section 2.1.
3. Build forward reshaping LUT by inverting the inverse reshaping LUT from 4).

### Derivation of chroma reshaping parameters

For chroma samples, reshaper is modeled with a piece-wise linear model. Depending on the application, derivation of PWL’s parameters can vary, below we provided an example of PWL model derivation for video signals that are being compressed in a target color container, which is different from a native color gamut signals were captured in. Example of such cases are video signals captured with a color gamut of BT.709 or P3 and being coded with color primaries of BT.2020.

To compensate the difference in color volumes between BT.709/P3 and BT.2020 color representation at the encoder side, an independent forward reshaper consisting a single-range PWL model is applied to each of chroma components:

yCb = Scb \* (xcb - Ocb) ( 8 )

yCr = Scr \* (xcr - Ocr)

where xcr and xcb are input chorma samples and ycb and ycr are output of the forward chroma reshaper.

PWL parameters Scr, Scb and Ocr, Ocb are derived from correspondence of color primaries for a native color gamut of input signal and color primaries of a target color container.

Colorimetric information of BT.709, BT.2020 and P3 specifications is provided in Table 1.

Table 1 RGB color space parameters

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Color space** | **White point** | | **Primary colors** | | | | | |
| **xW** | **yW** | **xR** | **yR** | **xG** | **yG** | **xB** | **yB** |
| DCI-P3 | 0.314 | 0.351 | 0.680 | 0.320 | 0.265 | 0.690 | 0.150 | 0.060 |
| ITU-R BT.709 | 0.3127 | 0.3290 | 0.64 | 0.33 | 0.30 | 0.60 | 0.15 | 0.06 |
| ITU-R BT.2020 | 0.3127 | 0.3290 | 0.708 | 0.292 | 0.170 | 0.797 | 0.131 | 0.046 |

Scales for Cb Cr components are computed as following.

For target color container derive variables tDR, tDG and tDB from its colorimetric parameters as following:

( 9 )

For native color gamut derive variables nDR, nDG and nDB from its colorimetric parameters as following:

( 10 )

Derive variables Scr, Scb as following:

( 11 )

( 12 )

If forward reshaper to chroma samples is applied to normalized CrCb samples which are belong to the range [-0.5, 0.5], variables Scr, Scb are set equal to zero:

. ( 13 )

If forward reshaper to chroma samples is applied in 10b quantized codewords domain, CrCb samples belong to the range [0, 1023], variables Ocr, Ocb for normalized offset are derived as following:

. ( 14 )

Inverse reshaper is applied to decoded CbCr samples at the decoder side as follows:

yCb = iScb \* xcb + iOcb ( 15 )

yCr = iScr \* xcr + iOcr

where xcr and xcb are decoded chorma samples input to the inverse reshaper and ycb and ycr are output of the inverse chroma reshaper.

Parameters of inverse PWL reshaper for Cr and Cb components consist of iScr, iScb and iOcr, iOScb are derived as follows:

( 16 )

( 17 )

Reshaper parameters iScr, iScb and iOcr, iOScb are signaled to the decoder and utilized to initialized Scale[i] and Offset [i] and further used as in Eq.(1).

## Mode 1 – SDR-backward compatible mode

The tuning process in mode 1 uses the following steps:

* Image analysis and temporal stabilization
* Luma reshaping parameters derivation
* Chroma reshaping parameters derivation



Figure . Derivation of reshaping parameters in reshaping mode 1.

Note: the description below is made for full range signals. The adaptation for standard range is straightforward and can be identified by analyzing the corresponding source code in the ETM reference software.

### Image analysis and temporal stabilization

The image analysis is based on the estimated average linear-light luminance Lmean for each picture. Only luminance samples above a minimum threshold Lmin are taken into account. In practice, Lmin is set to 0.1 nits.

The luma reshaping in mode 1 is based on the following mapping function (applying to linear-light input luminance L):

F(L) = log( 1. + (L / Ba)g ) / log( 1. + (P / Ba)g ) ( 18 )

where P is the mastering display peak luminance, Ba and g two control parameters. In current ETM, g is fixed and set to 2.8. The Ba value can vary per picture.

The parameter Ba is derived from Lmean as follows:

Ba = a \* Lmean2 + b \* Lmean + c ( 19 )

Parameters a, b and c have been heuristically determined, based on a various set of content, as a = –0.0003, b = 0.558, c = 4.6767.

Temporal smoothing of Ba value is performed based on the value ratio = ( Ba / temporalBa ), with temporalBa being the previous smoothed Ba value, as follows.

if (ratio > 0.6666 && ratio < 1.5) // stabilize Ba

temporalBa = 0.025\*Ba + 0.975\*temporalBa

else if (ratio > 3 || ratio < 0.3333) // abrupt change

temporalBa = Ba

else

temporalBa = 0.1\*Ba + 0.9\*temporalBa

In addition to the temporal smoothing of Ba value, a detection module decides if new reshaping parameters need to be signaled or not in the PPS. The detection test is positive as soon as the temporalBa value evolves by more than a given threshold (set in practice to 0.15) compared to the previous value for which reshaping parameters were signaled.

### Derivation of luma reshaping parameters

When the detection module indicates that new reshaping parameters need to be signaled, PWP modeling of the inverse of luma reshaping function is performed, using the following steps (also illustrated in Figure 3):

* Derivation of forward reshaping luma LUT from function F(.) and related control parameters temporalBa and g
* Inversion of the LUT to generate the backward reshaping luma LUT
* Modeling of the backward reshaping luma LUT using PWP model, and generation of the corresponding PWP backward reshaping luma LUT (backwardLutY)
* Inversion of the PWP backward reshaping luma LUT to get the final forward reshaping luma LUT (forwardLutY)



Figure . Derivation of luma reshaping parameters in reshaping mode 1.

The luma reshaping function is expressed as

F(L) = log( 1. + (L / temporalBa)g ) / log( 1. + (P / temporalBa)g ) ( 20 )

where L corresponds to linear-light luminance values.

Actually, since the input signal is considered to be 4:2:0 10 bit PQ samples, an intermediate mapping look-up-table is built as follows, for each PQ luma value YPQ:

loggLutY[ YPQ ] = floor[ 1023 \* F( PQ-1(YPQ) ) ] ( 21 )

This look-up-table is inversed, and the resulting inverse look-up-table is modeled using a PWP model, as described in section 4.1. The corresponding PWP parameters are signaled as new luma reshaping metadata.

The resulting inverse LUT (backwardLutY) from these PWP model parameters corresponds to the inverse reshaping LUT that will apply at decoder side. Its inverse LUT is the forward LUT (forwardLutY) that applies in the luma reshaping module.

### Derivation of chroma reshaping parameters

Once luma reshaping parameters have been derived, the chroma reshaping parameters can be deduced. The goal is to compute the two LUTs betaU and betaV that will apply at the reshaping side as follows:

Ureshape = OC + (UPQ – OC ) / betaU[ YPQ ] ( 22 )

Vreshape = OC + (VPQ – OC ) / betaV[ YPQ ] ( 23 )

with OC equal to 512 (29) for 10 bits content.

The derivation process of these two LUTs uses the following steps (illustrated in Figure 4):

* Generation of reference SDR chroma pictures
* Derivation of the LUTs by LMS to match as far as possible the reshaped chroma pictures to the reference SDR chroma pictures (chroma LUTs fitting)
* Modeling of the LUTs by PWL
* Generation of final LUTs from the PWL parameters.



Figure . Derivation of luma reshaping parameters in reshaping mode 1.

The generation of reference SDR chroma signal applies as follows. First the input RGB linear-light signal is downsampled by 2 in horizontal and vertical dimensions. For each pixel, the SDR chroma reference samples are derived as follows:

* The normalized linear-light R,G,B samples are converted to luminance using conventional RGB-to-Y conversion coefficients

LHDR = m0 \* R + m1 \* G + m2 \* B ( 24 )

* The reshaped luma sample Yreshape resulting from the luma reshaping step (described in previous sub-section) is converted to linear light by applying an inverse gamma function:

Lsdr = ( Yreshape / 1023 ) (1 / 0.45) ( 25 )

* The ratio w = Lsdr / LHDR is derived, and used to rescale each sample R,G,B in order to obtain an SDR estimation of these samples:

Rscaled = w \* R Gscaled = w \* G Bscaled = w \* B ( 26 )

* These samples are converted as follows:

Rsdr = 1023 \* (Rscaled)0.45 Gsdr = 1023 \* (Gscaled)0.45 Bsdr = 1023 \* (Bscaled)0.45 ( 27 )

* Then a conversion to Cb Cr applies using the conventional RGB-to-YCbCr conversion matrix coefficients (in the considered target color gamut). The resulting samples are the CbCr samples corresponding to the reference SDR (we note them UrefVref)

The next steps aim at deriving initial betaU and betaV LUTs in order to fit as much as possible the reference SDR chroma pictures from the input PQ chroma signal. This is achieved by minimizing the following functions:

( 28 )

( 29 )

where p=(x,y) are the pixel coordinates in the chroma pictures.

Actually, in order to get enough stability and robustness, the steps between different samples are quantized. For each sample Yreshape(x,y), a quantized index is derived as :

idx(x,y) = step \* floor(Yreshape(x,y) / step )

where step is a predefined parameter (typically equal to 64).

The optimization aims at minimizing :

( 30 )

( 31 )

For a given value of index idx, the result of this optimization is derived as follows:

* Let Nidx be the set of pixels p=(x,y) such that ( step \* floor(Yreshape(x,y) / step ) ) = idx

( 32 )

( 33 )

This results in sparse LUTs that are further filled by linear interpolation.

Finally, the resulting LUTs are modeled using PWL models, using the approach described in section 4.2. The LUTs resulting from these PWL models are used for the chroma reshaping at encoder side and chroma inverse reshaping at decoder side.



Figure . Illustration of chroma reshaping LUT derivation in mode 1.

## Mode 2 – alternate SDR-backward compatible mode

The basic idea behind operation of mode 2 is explained in the document m37092. In that work the HDR-YCbCr signal which was assumed to be generated from linear HDR-RGB samples, using the inverse of a power/gamma TF, could be reshaped to an SDR-viewable-YCbCr signal if the HDR-gamma and the SDR display gamma values would be reciprocal.

In this work the process of reshaping the input HDR-YCbCr signal to an SDR-viewable signal is expanded to cover the cases where input HDR-YCbCr is generated with an arbitrary TF (e.g. PQ, etc.).

To cover a generic case the input HDR-TF is approximated with a power TF with a power value, alpha and an appropriate black level lift. The mismatch between the HDR TF and the SDR display TF is compensated by the reshaping process by taking the following steps under the Mode 2 of encoder operation:

* Adjustment of dynamic range
* Optimizing the reshaping parameters
* Coding of inverse reshaping parameters
* Applying the reshaping

The following inputs are provided to the forward reshaping process at the encoder side:

* Equivalent of the alpha value for a power function which estimates the input TF, denoted by
* Hypothetical Black Level lift of input TF, denoted by
* Gamma value of the target SDR display, denoted by
* Black Level Lift of the target SDR display, denoted by

### Adjustment of dynamic range

At the outset the input signal is normalized so the rest of processes will be conducted in the same manner regardless of the sample range type of the input signal. This approach also allows that on the encoder side the bit-depth and sample ranges of input to the reshaper would be different than the bit-depth and sample range of the output of the reshaper which is the input to the legacy encoder. Note that this is just an encoder flexibility and all the decoder side operations will be conducted with the constrained bit-depth and sample range of the decoder and inverse reshaper.

### Optimizing the reshaping parameters

The reshaping is responsible for compensating the difference between the HDR-TF power factor and the inverse of display gamma function. To accomplish this goal the value of power factor which needs to be compensated by the reshaping process is calculated by the process outlined in this section.

First each picture is analyzed to calculate the value of gamma which needs to be compensated by a power function in the Y channel of the reshaper.

Picture analysis is conducted at block, picture and temporal level as follows:

* Block level: Each picture is divided to blocks of certain size (by default 16x16). For each block three statistical data of (min, max and mean) based on approximated linear brightness are derived.
* Picture level: Overall black-level, peak-brightness and average Intensity for each picture are derived by considering the corresponding statistics for all blocks in the picture and considering the masking effect of average local brightness on the minimum black-level perceived.
* Temporal level: Temporal black level, temporal peak brightness and temporal average luminance values are updated based on the corresponding values for each new picture and the past pictures.

The temporal values derived in the above, are used to assign an intrinsic system-gamma based on:

( 34 )

In the current ETM GammaLift is 0.5. Brightness is the temporal Peak brightness and BlackLevel is the temporal picture black level, considering the masking effect of average temporal brightness.

The power value of the luma reshaping function is determined by:

( 35 )

The temporal system-gamma (and hence the reshaper power value) gets updated if the change in the new value of temporal system gamma is greater than a certain threshold (currently 10%).

### Coding of inverse reshaping parameters

Every time the power value of the reshaping process is updated, the parameters associated with the inverse reshaper need to be recalculated and transmitted in the PPS.

Reshaper parameters are calculated first for Luma and then for each of the two chroma components, by creating a LUT for each of the three components as follows:

* Luma: Inverse Reshaping of Luma component Y is applied according to the following:

( 36 )

where f(.) is a reshaping function used for enhancing the contrast of Luma component, while g(.) is the reshaping function responsible for gamma compensation of the HDR TF. Currently ETM only applies the g(.) function which has the following description:

( 37 )

, is the normalization function for reconstructed samples by the decoder.

In contribution m37091, when the input TF matches the target SDR display gamma f(.) was chosen to be the inverse of a log function to improve the contrast for the lower brightness.

* Chroma: Inverse Reshaping for component C (Cb or Cr) is applied according to the followings:

( 38 )

where is the collocated Luma sample corresponding to location and,

( 39 )

Note that the value of corresponds to the chroma scaling factor which is derived from input parameter and the value of and are derived from black level lift values for HDR-TF and the reference SDR display.

When the LUTCb and LUTCr are the same based on the selection of the reshaper choice of parameters, then only one of the two sets get transmitted.

The Luma and chroma LUTs will be coded by invoking processes in 4.1 and 4.2, respectively.

### Applying the reshaping

To apply the forward reshaping on the encoder side, either a LUT can be generated which inverses the inverse-reshaper LUT generated by the coded parameters of the inverse reshaper, or it can directly operate on the normalized sample values, with the formulas which are inverse of decoder process as explained above.

In the current version of the ETM software, the latter approach is used. When direct formula on the normalized data is used, at the last step, the appropriate conversion to a desired sample range for the given bit-rate is needed.

# Piece-wise models parameters derivation

## Piece-wise polynomial parameters derivation

The PWP model parameters consist of the description of segments of the model and the coefficients of the model on each segment. Given a target function, the set of segments and coefficients must be determined. For signalling a reduced set of parameters sufficient to specify a smooth second order model are used. The full set of model parameters is reconstructed using the values signalled and the conditions for smoothness. The PWP parameter derivation has two major components. Selection of the partitioning into segments and the derivation of model parameters for each segment.

### Derivation of PWP segments

A search technique is used to determine the segments used for the PWP model. An upper bound N on the number of segments is given i.e. N=8. For each value in the range 1 to N, two candidate sets of segments are selected. The first candidate set partitions the x-axis uniformly into N segments. The second candidate set partitions the y-axis uniformly into N segments and uses the target function to determine the corresponding partitioning of the x-axis. For each candidate set of segments, a set of model coefficients is constructed and the fitting error is determined using the procedure in 4.1.2. The set of segments with minimal fitting error is selected. The fitting error consists of the MSE between the target LUT and the model and may include an offset to account for the number of segments.

### Derivation of PWP coefficients

The procedure for fitting a second order PWP model to a given target LUT with a given set of segments is described. Let S denote the number of segments of the model. Each segment is described by a length which may be used to signal the segment partitioning. The endpoints of the intervals, called pivot points, are determined from the set of signalled interval lengths. A set of intervals defined by S+1 points {pj} for j=0 to S where the jth interval is defined by pj<=x<pj+1. Model coefficients on each interval are defined by ci,j where i denotes the order of the coefficient and j denotes the interval.

#### Constraints for smooth PWP modelling

For purposes of signalling, the model is described relative to the endpoints of each interval. The interpolated function on the jth interval becomes:

y = c0,j + c1,j\*(x-pj) + c2,j\*(x-pj)2 for x in the range pj<=x<pj+1 ( 40 )

A piecewise second order model with S segments has 3S degrees of freedom. The constraints of smooth continuity impose relations between the model parameters at the internal points. When constraints of smooth continuity at the interior points are imposed 2(S-1) degrees of freedom are removed leaving S+2 degrees of freedom. To describe these S+2 degrees of freedom, the constant term at each point of S+1 points and the linear term of the first point are used. The full set of model parameters is determined from this set of value using the constraints of smooth continuity and recursion.

|  |  |  |  |
| --- | --- | --- | --- |
| c0,j + c1,j\*(pj+1 -pj) + c2,j\*(pj+1 -pj)2 | = | c0,j+1 | (continuity at pj+1) |
| c1,j+ 2\*c2,j\*(pj+1 -pj) | = | c1,j+1 | (smoothness at pj+1) |

These systems of equations can be solved to give a recursive definition of the first order coefficients that depends on the S+1 zeroth order coefficients c0,0 through c0,S. The initial first order value c1,0 is used along with this recursion allows calculation of all other first order values.

c1,j+1 = 2\*(c0,j+1 - c0,j)/(pj+1 - pj) - c1,j ( 41 )

The second order coefficients are determined by the first order coefficients as follows:

c2,j = (c1,j+1 – c1,j)/(2\*(pj+1 - pj)) ( 42 )

These relations allow reconstructing the full set of 3S model parameters from the reduced set of S+2 parameters signalled in the bitstream.

#### Determination of reduced set of model parameters

The description above describes how smoothness constraints allow reconstruction of a full set of PWP model coefficients from a reduced set of parameters using the smoothness constrains. The problem of how to determine this reduced set of parameters is described here. Given the pivot point values {pj} and input values {xi}, the vector of model outputs Ymodel is a linear function of the reduced set of parameters, V. A matrix M defining this linear relation is computed.

Ymodel = M\*V ( 43 )

Given a vector of target (inverse) reshaping function Ytarget, the parameter values giving a minimal MSE fit are determined solving the MSE fit problem Ytarget = M\*V by conventional means. Analytically the vector of optimal parameters is given by Voptimal = inv(MT\*M)\* MT\* Ytarget

We note the parameters determined from this optimization process are constrained to give a smooth model but may not equal the value of the target LUT at the pivot points. Lower MSE results by allowing some difference at the pivot points rather than forcing the model to exactly fit the target LUT at the pivot points.

#### Bit-depth of reduced parameter values

The bit-depth needed for the signaled parameters is determined as follows. Each parameter is signaled as an integer and separate fraction component. The integer part of the constant coefficients uses the same number of bits as the model output since these are simply the values of the model at one of the pivot points. The value of the single linear coefficient uses at ranges between +- the maximum sample value and thus needs at most one more bit than the number of bits used for the model output. The number of necessary fraction bits is determined as follows. For each x values, the model is a sum of three components rounded to an integer. Each component requires 2 bits of fraction. The constant coefficients are signaled with 2 bits of fraction. The single linear coefficient is scaled by the difference in x-value and the pivot point value in the application of the model. This difference is no more than the size of the input implying that to have 2 bits of fraction in the product the linear coefficient should have two plus the source bit depth in fraction bits i.e. 12 bits for 10 bit source.

#### Modifying PWP coefficients

The model coefficients used in section 4.1.2.1, {ci,j}, are described with offsets relative to the pivot points. The model coefficients use for reconstruction in section2.1, {ai,j}, are not offset relative to the pivot point locations. An algebraic identity on each segment is used to compute the coefficients used in section 2.1 from those reproduced form the bitstream in section 4.1.2.1.

The expression defining the modified coefficients, {ai,j}, in relation to the coefficients reproduced from the bitstream, {ci,j}, is given by the equality of the model output for any x in the jth interval. Namely a0,j + a1,j\*x + a2,j\*x2 = c0,j + c1,j\*(x-pj) + c2,j\*(x-pj)2. Basic algebra implies the following relations.

|  |  |  |
| --- | --- | --- |
| a0,j | = | c0,j - c1,j\* pj + c2,j\* pj2 |
| a1,j | = | c1,j -2\* c2,j\* pj |
| a2,j | = | c2,j |

These relations define the PWP model coefficients used in reconstruction from the model coefficients reconstructed from the bitstream.

## Piece-wise linear parameters derivation

The input of this process is a 1D-LUT, typically of 1024 elements when the signal is represented with 10 bits. The outputs are the PWL parameters modeling this LUT.

The main PWL parameters are:

* the length of each interval Vi,
* the slope on each interval Si,
* the initial offset F0.

The encoding of each parameter is performed in fixed point representation, with N bits for the integer part, and M bits for the fractional part.

The PWL parameters estimation is split into 2 parts:

* Full precision PWL model estimation which allows to obtain the intervals for each segment;
* Refined estimation which takes into account the model encoding.

### Full precision PWL model estimation

The inputs of this process are:

* a maximum number of intervals N,
* an initial maximum fitting error threshold t,
* the LUT to model, yi = LUT[i], for i=0 .. (lutSize-1).

The algorithm, illustrated in Figure 6, iteratively computes each segment as follows:

* First an interval is initialized to length 1 (or to minimal segment length)
* Then a segment estimation is done using Least Mean Squares (as described in section 4.2.3).
  + It is an unconstrained segment fitting for the first segment (as there is no constraint of continuity).
  + It is a constrained segment fitting for the reminding segments (as the PWL mode is implicitly continuous). The start of the current segment is constrained to be the end of the previous one.
  + The segment estimation can take into account the number of points contributing to the chroma correction factor (i.e. LMS is weighted by the number of contributing points in the content).
* The fitting error is computed. If the error is below the threshold t, the interval length is increased and the segment is re-estimated, if not, the next segment is computed.
* At the end, if the number of segments is below the maximum allowed number of segments, the intervals computed are provided as output of the process. Otherwise, the threshold t is increased, and the complete process is re-launched.



Figure . Full precision PWL model estimation process.

### Closed loop PWL model estimation

The inputs are in previous section. In addition, the computed intervals from previous step are considered.

The steps are shown in Figure 7. The algorithm iteratively computes each segment as follows:

* Parameters for all intervals are coded using the fixed point representation.
* For each interval, the interval parameters are decoded.
* Then a segment estimation is done as in the previous step.
* The resulting segment slope is coded, i.e. using a fixed point representation (and possibly an inverse function).
* The coded slope is decoded.
* Using the decoded value, the end point of the interval is computed and will be used to constrain the estimation of the next segment.



Figure . Closed loop PWL model estimation process.

An illustration of the impact of the close loop process compared to directly coding the PWL parameters is shown in figure xx.

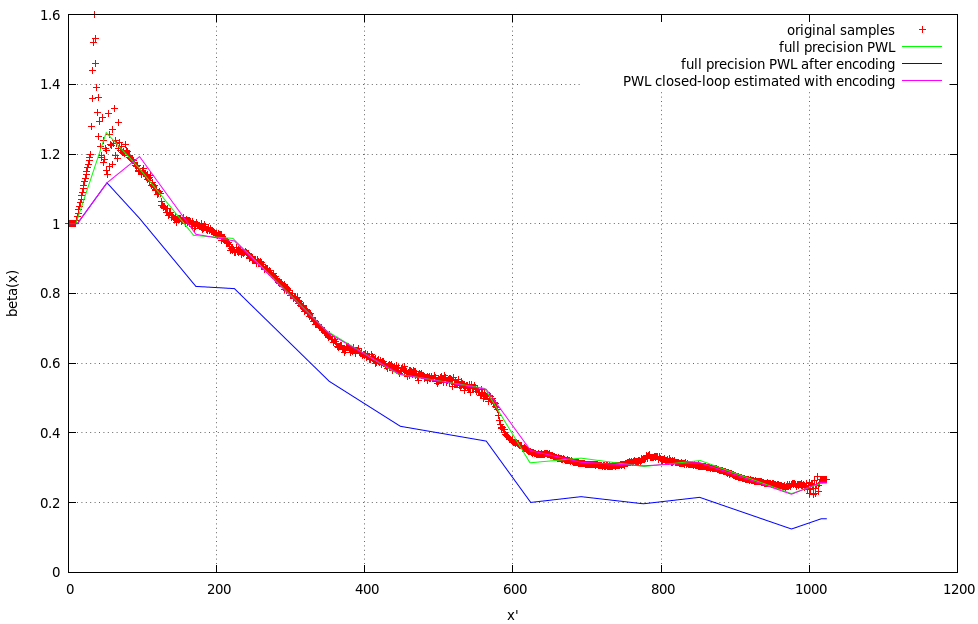
****

Figure . Example of PWL fitting with closed loop coding.

### LMS estimation of a segment parameters

Let’s consider a look-up-table LUT[.] to model using a Piece Wise Model with non-uniform intervals. The following notations are used:

* yi = LUT[ xi ], i=0..(lutSize-1)
* M(x) = c0 + c1.x is the PWL model used to fit the data of interval k.
* [Ik, Ik+1] is the kth interval running from Ik to Ik+1, corresponding to segment Sk

To estimate the first segment S0, an unconstrained model estimation is applied:

( 44 )

where are the coefficients of the model for interval 0.

To estimate further segments Si, for i > 0, the same problem is solved, by enforcing the C0 continuity, then only the scale coefficient c1k of the PWL model is estimated:

( 45 )

Then the c0k coefficient is deduced as follows:

( 46 )

**Note:** when the data are coming from a previous step using an optimization based on the content as this is the case when encoding the cross-channel LUTs in reshape mode 1, the fitting can take into account the number of samples used to compute each value of the LUT in order to minimize the global reconstruction error. That is, the LMS estimation consists in solving the following equation:

( 47 )

where ωi is the number of samples used to compute the data yi and e(a,b) is the quadratic error between a and b.

# Patent rights declaration(s)

**Arris may have current or pending patent rights 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 ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

**Dolby may have current or pending patent rights 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 ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

**InterDigital may have current or pending patent rights 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 ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

**Qualcomm may have current or pending patent rights 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 ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

**Technicolor may have current or pending patent rights 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 ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**