|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  9th Meeting: Geneva, CH, 27 April – 7 May 2012 | Document: JCTVC-I0518 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **HVS-based Generalized Quantization Matrices** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Sangoh Jeong  Hendry  Byeongmoon Jeon  Jungsun Kim  #221 Yangjae-dong, Seocho-gu,  Seoul 137-130,  Korea | Tel: Email: | [sangoh.jeong@lge.com](mailto:sangoh.jeong@lge.com)  [hendry.hendry@lge.com](mailto:hendry.hendry@lge.com)  [bm.jeon@lge.com](mailto:bm.jeon@lge.com)  [jungsun.kim@lge.com](mailto:jungsun.kim@lge.com) |
| *Source:* | LG Electronics | | |

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

# Abstract

This contribution proposes a new method of generating scaling matrices or quantization matrices (QM’s), based on a generalized version of an HVS model. When decoder receives parameters from encoder, the decoder can generate a QM having key characteristics of an HVS model – having lower QM values in lower frequency components than those in higher frequencies.

# Introduction

In the current HEVC specification [1], a flat QM or a default QM is decided to used at the decoder according to the “scaling\_list” value from the encoder. Although the results of using the two QM’s are visually similar for certain tested sequences, devices, and applications, there is still a chance that other/future sequences, devices, and applications may need different QM’s to serve their objective goals, due to the vagueness of subjective quality.

In [4], an intra 8x8 QM is proposed as, for i = 1, 2, …, 8 and j = 1,2, …, 8,

intra 8x8 QM (i, j)= Round[16 / MTF(i, j)]

, where is the parameter to control the QM values.

The current default QM’s [1] depends on the MTF(modulation transfer function) based on an HVS model [3], as introduced in [2]. The MTF for the default intra 8x8 QM is as follows.

MTF(8x8) =

1.0000 1.0000 1.0000 1.0000 0.9599 0.8746 0.7684 0.6571

1.0000 1.0000 1.0000 1.0000 0.9283 0.8404 0.7371 0.6306

1.0000 1.0000 0.9571 0.8898 0.8192 0.7371 0.6471 0.5558

1.0000 1.0000 0.8898 0.7617 0.6669 0.5912 0.5196 0.4495

0.9599 0.9283 0.8192 0.6669 0.5419 0.4564 0.3930 0.3393

0.8746 0.8404 0.7371 0.5912 0.4564 0.3598 0.2948 0.2480

0.7684 0.7371 0.6471 0.5196 0.3930 0.2948 0.2278 0.1828

0.6571 0.6306 0.5558 0.4495 0.3393 0.2480 0.1828 0.1391

The MTF has a long history in backgrounds, but the parameter selections may still have room for adjustment to meet the needs of diverse modern and future applications.

This contribution proposes a unified structure of providing diverse QM’s for different sequences, devices and applications.

# Proposed Generalized QM Generation

## Original Concept

The proposed generalized intra 8x8 QM can be generated by, for i = 1, 2, …, 8 and j = 1,2, …, 8,

8x8 QM (i, j) = Round[ / MTF(i, j)] = Round[MTF(i, j)]

, where a real number is the parameter to control MTF-based intra QM values and an integer is the scaling factor.

When and, the intra 8x8 QM becomes the flat QM. When  and, the intra 8x8 QM becomes the (current) default intra QM. The new intra QM’s are based on a variation of the MTF.

The generalized QM’s are aimed at serving a variety of objective quality vs. bitrate trade-offs, while maintaining key properties of using the current QM’s – quantizing low frequency components more than high frequency components.

The inter QM’s can be made using the linear relationship between intra QM values and inter QM values explained in [2], but a different set of and can also be defined for the inter QM’s. Figure 1 shows the simple structure for the proposed model.



Encoder

Decoder

Figure 1. Proposed structure for generalized QM’s

At the encoder side, the values of can be decided based on, for example, RDOQ for specific application purposes. The decoder will need to have a module to reconstruct the QM’s occasionally. The module should have some memory to keep a form of values related to the MTF, when the values are approximated or modified. These values should be integers, and the -th power of a modified value of the MTF should be simple and has to be integer-arithmetic, too.

## Computation-Friendly Concept

Since the MTF and are real numbers, the following computation-friendly versions are needed.

### Alpha() Selection

Although the can be any real number larger than or equal to 0, but we can choose among , for computational simplicity. It is not likely that  will be used due to a large amount of quality degradation. Then, an integer value of can be defined so that  can be used as an index instead of the floating value, .

### Approximations on QM (8x8) matrix

From the original definition,

8x8 QM (i, j) = Round[ / MTF(i, j)] = Round[MTF(i, j)]

= Round[(MTF(i, j))]

= Round[(MTF(i, j) )]

= Round[(MTF(i, j) ) ]

 Round[(Round[MTF(i, j)]) ]

Here, Round[MTF(i, j) ] =

1 1 1 1 1 1 1 2

1 1 1 1 1 1 1 2

1 1 1 1 1 1 2 2

1 1 1 1 1 2 2 2

1 1 1 1 2 2 3 3

1 1 1 2 2 3 3 4

1 1 2 2 3 3 4 5

2 2 2 2 3 4 5 7



1 1 1 1 1 1 1 2

1 1 1 1 1 1 1 2

1 1 1 1 1 1 2 2

1 1 1 1 1 2 2 2

1 1 1 1 2 2 4 4

1 1 1 2 2 4 4 4

1 1 2 2 4 4 4 8

2 2 2 2 4 4 8 8

=     

    





     

Then,

Round[MTF(i, j) ]      

   





    

Then,

(Round[MTF(i, j)])  

    

   





    

Then,

(Round[MTF(i, j)]) 

    

   





    

With, the values of and 



Finally, when ,

8x8 QM (i, j)  Round[(Round[MTF(i, j)]) ]

 {16, **qm1**, **qm2**, **qm3** }

where**, qm1 =** Round(), **qm2 =** Round(), **qm3 =** Round()

### Possible Realizations of the Approximated QM (8x8) ()

***Option 1:*  Transmission of all 3 QM values (qm1, qm2, qm3) from Encoder**

**🡪 Only the encoder has the following table, and the decoder map the received values to**

**make a QM.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| **qm1** | 16 | 17 | 19 | 21 | 23 | 25 | 27 | 29 | 32 | 35 | 38 | 41 | 45 | 49 | 54 | 59 | 64 |
| **qm2** | 16 | 19 | 23 | 27 | 32 | 38 | 45 | 54 | 64 | 76 | 91 | 108 | 152 | 166 | 181 | 215 | 256 |
| **qm3** | 16 | 21 | 27 | 35 | 45 | 59 | 76 | 99 | 128 | 166 | 215 | 279 | 362 | 470 | 609 | 790 | 1024 |

***Option 2:*  Transmission of** **from Encoder**

**🡪 Both encoder and decoder have the table above, and map the values to make a QM.**

***Option 3:* One QM value (qm1) transmission from Encoder. Approximation of other QM values**

**(qm2, qm3) from qm1 at the Decoder. The encoder has the following table.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| **qm1** | 16 | 17 | 19 | 21 | 23 | 25 | 27 | 29 | 32 | 35 | 38 | 41 | 45 | 49 | 54 | 59 | 64 |

Since **qm1 =** Round(), **qm2 =** Round(), **qm3 =** Round(),

approximated values of **qm2** and **qm3** can be derived as follows.

**🡪 qm2 =** Round()  ( / 16)

 (**qm1)**  / 16

**🡪qm3 =** Round( )   / 256

 (**qm1)**  / 256

***Option 4:* Transmision of** ,**then reconstruction from computation such as Talyor Series.**

* **Computationally expensive, complexity of realization is high.**

### Signaling

To signal the  used to generate QM, the followings are suggested:

* Modify scaling\_list\_enable\_flag in SPS to scaling\_list\_mode coded with 2 bits (i.e., u(2).
  + scaling\_list\_mode equals 0 means flat QM is used
  + scaling\_list\_mode equals 1 means that depending on the preferred options above, some other parameters will be signalled
  + scaling\_list\_mode equals 2 means that the current user defined QM signalled in APS is used
* The value of should be fixed to 16.

With the above suggestion, the necessary change to current signalling is as follow:

Originally,

|  |  |
| --- | --- |
| seq\_parameter\_set\_rbsp( ) { | Descriptor |
| **…** | u(1) |
| **scaling\_list\_enable\_flag** |  |
| **…** |  |
| } |  |

**scaling\_list\_enable\_flag** equal to 1 specifies that scaling list is used for scaling process for transform coefficients in . scaling\_list\_enable\_flag equal to 0 specifies that scaling list is not used for scaling process for transform coefficients in .

If Option 1 is the preferred method, then the necessary changes are:

|  |  |
| --- | --- |
| seq\_parameter\_set\_rbsp( ) { | Descriptor |
| **…** |  |
| **scaling\_list\_mode** | u(2) |
| **if (scaling\_list\_mode == 1)** |  |
| **{** |  |
| **scaling\_qm1** | ue(v) |
| **scaling\_qm2** | ue(v) |
| **scaling\_qm3** | ue(v) |
| **}** |  |
| **…** |  |
| } |  |

**scaling\_list\_mode** specifies the scaling list mode that shall be used. scaling\_list\_mode equal to 0 specifies that scaling list is not used for scaling process for transform coefficients in . Otherwise, specifies that scaling list is used for scaling process for transform coefficients in . When scaling\_list\_mode equals 1, then scaling\_qm1, scaling\_qm2, and scaling\_qm3 shall be used as component values of the quantization matrix. When scaling\_list\_mode equals 2, the QM matrix shall present in APS.

Else if Option 2 is the preferred method, then the necessary changes are:

|  |  |
| --- | --- |
| seq\_parameter\_set\_rbsp( ) { | Descriptor |
| **…** |  |
| **scaling\_list\_mode** | u(2) |
| **if (scaling\_list\_mode == 1)** |  |
| **{** |  |
| **scaling\_alpha** | ue(v) |
| **}** |  |
| **…** |  |
| } |  |

**scaling\_list\_mode** specifies the scaling list mode that shall be used. scaling\_list\_mode equal to 0 specifies that scaling list is not used for scaling process for transform coefficients in . Otherwise, specifies that scaling list is used for scaling process for transform coefficients in . When scaling\_list\_mode equals 1, then scaling\_alpha shall be used as the component value of the quantization matrix. When scaling\_list\_mode equals 2, the QM matrix shall present in APS.

Else if Option 3 is the preferred method, then the necessary changes are:

|  |  |
| --- | --- |
| seq\_parameter\_set\_rbsp( ) { | Descriptor |
| **…** |  |
| **scaling\_list\_mode** | u(2) |
| **if (scaling\_list\_mode == 1)** |  |
| **{** |  |
| **scaling\_qm1** | ue(v) |
| **}** |  |
| **…** |  |
| } |  |

**scaling\_list\_mode** specifies the scaling list mode that shall be used. scaling\_list\_mode equal to 0 specifies that scaling list is not used for scaling process for transform coefficients in . Otherwise, specifies that scaling list is used for scaling process for transform coefficients in . When scaling\_list\_mode equals 1, then scaling\_qm1 shall present whereas the value of qm2 and qm3 shall be derived. When scaling\_list\_mode equals 2, the QM matrix shall present in APS.

The authors would like to advocate option 2. Reason….

# Conclusion

With the proposed method, a wide range of different QM matrices can be utilized at the decoder by sending only theandfrom the encoder. The existing flat and default QM’s are also a part of the proposed generalized QM’s. By the proposed engineering simplifications, the complexity will not change much, since the QM generation is needed only once in every GOP at most.

The benefit of applying the proposed model is that it can serve a variety of rate and distortion constraints using a simpler unified model, still maintaining the key properties of the HVS - quantizing low frequency components more than high frequency components. Therefore, this structure should be considered for CE.

# Patent rights declaration(s)

**LG Electronics 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).**

# Reference

[1] JCTVC-H1003, “High efficiency video coding (HEVC) text specification draft 6,” 8th JCT-VC Meeting, San Jose, CA, USA, 1-10 February, 2012.

[2] M. Haque, A. Tabatabai, Y. Morigami, “HVS Model based Default Quantization Matrices”, JCTVC-G880, November 2011.

[3] Long- Wen Chang and Ching- Yang Wang and Shiuh-Ming Lee, “ Designing JPEG quantization tables based on Human Visual System., *Signal Processing: Image Communication*, Volume 16, Issue 5, pp 501-506, Jan 2001.

[4] S. Jeong and B. Jeon, “Newer Quantization Matrices”, JCTVC-I129, April 2012.