Quantization matrix entries as QP offsets

R. Joshi, J. Sole and M. Karczewicz

Qualcomm

Quantization Matrices in CD

- Each entry is interpreted as a scale factor
  - (QM entry / 16)
- The granularity of effective QP change is very different for different scale factors
  - 1 to 2 or 128 to 255 (effective QP change of 6)
- Typical quantization matrices have low scale factors at lower frequencies
  - Much more granularity needed

Other issues

- Different clipping before dequantization depending on
  - whether quantization matrices are being used
  - transform size
- Different dequantization paths depending on whether quantization matrices are being used.
- Additional storage to avoid multiplication
  - M[i][j]*levelScale[qP%6] for 6 possible values of qP%6

Notation

- B: internal bit depth
  - as specified by InternalBitDepth in the common config files
- N: transform size
- M: $\log_2(N)$
- levelScale[k]: { 40, 45, 51, 57, 64, 72 } with k = 0, 1, … , 5
- M[i][j]: 8-bit unsigned scaling list matrix entries
- c[i][j]: coefficient level
- d[i][j]: Dequantized level values

Proposed method

- Quantization matrix entries interpreted as QP offset
- Higher granularity than existing QPs
  - Example: 12 QP increments to double the step-size
- QP for coefficient (i,j) is derived as
  - $QP_{mod}[i][j] = 2 * QP + (M[i][j] - 64)$
  - Clip to [0, 119]
  - Granularity of half QP.
  - Higher granularities are possible
    - Limited by the amount of precision for levelScale array (6 bits).

Proposed method – *cont.*

- Unification of dequantization paths
  - Similar processing for no quantization matrix and quantization matrix case
  - For no quantization matrix case, set QP_{mod} to 2*QP

$$d[i][j] = ((c[i][j] * levelScale[QP_{mod}[i][j] \% 12] << (QP_{mod}[i][j] / 12)) + (1 << (shift - 1))) >> shift$$

  - levelScale[k]: { 40, 42, 45, 48, 51, 54, 57, 60, 64, 68, 72, 76 } with k = 0, 1, …, 11.

32-bit intermediate processing

- c[i][j]: 16-bit signed
- levelScale: 7-bit unsigned
- (QP_{mod}[i][j]/12): 9-bit unsigned
- No need to clip c[i][j] if 16-bit level restriction is applied on the encoder side.

Advantages

- Same granularity of QP change at high and low values of quantization matrix entries.
- Same processing path for dequantization with or without quantization matrices.
- Clipping before dequantization can be removed if 16-bit level restriction is applied on the encoder side.
- No need to store (M[i][j]*levelScale[QP%6]) for 6 QP%6 values. However
  - Bigger LUT needed for QP%12 and QP/12.
  - One addition, one clip and 2 table lookups.

Conclusion

- Proposed quantization matrices where the entries act as fractional QP offsets
  - Uniform granularity
  - Reduced storage
- Recommend adoption of the method