Imagination
TECHNOLOGIES

JCTVC-J0225 : Restrictions to the maximum motion vector range

Alistair Goudie

July 2012

AVC vs HEVC motion vector restrictions

- **In the AVC specification, the Profiles/Limits section defined a limit on maximum motion vector range;-**
  - Horizontal vector limited to range -2048 to 2047.75 luma samples
  - Vertical vector limited varies based on level, but maximum range -512 to +511.75 luma samples
- **The only restriction which appears in the HEVC draft is an optional field in the VUI (defined in Annex E).**
  - Optionally defines the fields 'log2_max_mv_length_horizontal' and 'log2_max_mv_length_vertical'
  - If these fields aren't defined, the vector range default to -2^{16} to 2^{16}-1 (in quarter sample resolution, which is equivalent to -16384.0 to +16383.75 luma samples)
- **This implies spatial/temporal motion vector components need to be stored as 17-bit signed components**
- **Which also implies coded 'motion vector difference' coded in the bitstream could be 18-bit signed components**

HEVC motion vector calculation and clipping

- **In a P/B slice, when prediction mode is MODE_INTER and merge_flag = 0;-**
  - Reference picture index and motion vector difference are encoded in the bitstream
  - A motion vector predictor to the left and from above the current prediction unit are selected
    - If the selected vector was from a different reference picture, each component is scaled and clipped to a 16-bit signed component
  - The 'motion vector difference' is then added to the selected motion vector predictor (without clipping, result may be 17-bit signed component)

- **In a P/B slice, when prediction mode is MODE_SKIP or prediction mode is MODE_INTER and merge_flag = 1;-**
  - A list of spatial neighbouring candidates is generated, plus a co-located temporal vector
    - The temporal vector may be scaled, in which case it is clipped to 16-bit signed components.
  - If MaxNumMergeCand > 1, a merge_idx is coded in the bitstream to select the candidate to use as the motion vector for the current prediction unit (without clipping)

- **In the HM software, motion vectors and 'motion vector differences' are stored as 16-bit signed components**

Proposal 1 – limit MV to 16-bit signed component

- **The simplest restriction is to limit both motion vectors and motion vector differences to 16-bit signed components (in quarter sample resolution)**
  - MinMotionVector[HOR] = -32768 (equivalent to -1 << 15)
  - MinMotionVector[VER] = -32768 (equivalent to -1 << 15)
  - MaxMotionVector[HOR] = 32767 (equivalent to (1 << 15)-1)
  - MaxMotionVector[VER] = 32767 (equivalent to (1 << 15)-1)

  - 32767/4 = 8191.75 luma samples

- **Advantages;-**
  - It doesn’t require changes to the existing clipping of scaled motion vector predictors/temporal motion vectors
  - Hardware implementation of clip to fixed power of 2 is small

- **Disadvantage;-**
  - A decoder designed for lower profile (e.g. SD resolution) still needs to store 16-bit motion vectors

Proposal 2 – limit MV to picture size

- **If absolute motion vector is greater than the picture size then inter prediction will be from boundary replicated pixels from the edge of the picture, so the second proposal is to limit motion vector to picture resolution (with an extra clip to ensure they are not greater than 16-bit signed)**
  - MinMotionVector[HOR] = -4 * min(8192, (PicWidthInCtbs<<Log2CtbSize))
  - MinMotionVector[VER] = -4 * min(8192, (PicHeightInCtbs<<Log2CtbSize))
  - MaxMotionVector[HOR] = (4 * min(8192, (PicWidthInCtbs<<Log2CtbSize))) - 1
  - MaxMotionVector[VER] = (4 * min(8192, (PicHeightInCtbs<<Log2CtbSize))) - 1

- **Advantages;-**
  - It allows a hardware decoder designed for lower levels to reduce the size of the motion vector store and/or pack additional flags with the motion vector into a 32-bit word

- **Disadvantage;-**
  - The clip applied to motion vector scaling would need to change to use these clip values

Proposal 3 – limit vertical MV to 1024 luma samples

- Similar to proposal 2, but with additional constraint on vertical motion to limit to -1024 to +1023.75 luma samples
  - MinMotionVector[HOR] = -4 * min(8192, (PicWidthInCtbs<<Log2CtbSize))
  - MinMotionVector[VER] = -4 * min(1024, (PicHeightInCtbs<<Log2CtbSize))
  - MaxMotionVector[HOR] = (4 * min(8192, (PicWidthInCtbs<<Log2CtbSize))) - 1
  - MaxMotionVector[VER] = (4 * min(1024, (PicHeightInCtbs<<Log2CtbSize))) - 1

- Advantages;-
  - Improves the ability of a decoder to pack additional flags with the motion vector into a 32-bit word (e.g. reference picture index)
  - It introduces the possibility of using a non-normative approach to frame level parallelism in a multicore decoder (where a 2nd core could start predicting from a frame still being generated by1 1st core, provided the 1st core has finished in-loop filtering of 1024+CTB size rows below core 2)

- Disadvantage;-
  - It could limit the decoders ability to do inter prediction of a fast vertical pan

Method of applying constraint

- **The preferred method of applying the constraint is an encoder restriction defined in Annex A which would apply to all Profiles and Levels**

- **At previous meetings there were concerns about encoder restrictions not being obeyed, so a modification of the decoding process by adding additional clip operations is also presetented**

Imagination
TECHNOLOGIES

JCTVC-J0225 : Restrictions to the maximum motion vector range

Alistair Goudie

www.imgtec.com