|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  15th Meeting: Geneva, CH, 23 Oct. – 1 Nov. 2013 | Document: JCTVC-O0055 |

|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  6th Meeting: Geneva, CH, 25 Oct. – 1 Nov. 2013 | Document: JCT3V-F0032 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **MV-HEVC/SHVC HLS: Skipped slice and use case** | | |
| *Status:* | Input Document to JCT-VC and JCT-3V | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Tomoyuki Yamamoto Takeshi Tsukuba Tomohiro Ikai  1-9-2 Nakase, Mihama-ku, Chiba-shi, Chiba 261-8520 JAPAN | Tel: Email: | +81-43-299-8526 [yamamoto.tomoyuki@sharp.co.jp](mailto:yamamoto.tomoyuki@sharp.co.jp) |
| *Source:* | SHARP Corporation | | |

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

# Abstract

This contribution introduces an ROI-capable scalable video applications using skipped tile. In the application, 1) media box transfers modified EL bitstream corresponding to the region requested by the users, 2) EL data is modified so that the outside-of-region part of the bitstream includes less data by replacing the outside-of-region with skipped tile, and 3) SHVC decoder could receive whole BL bitstream and partial EL sub bitstream corresponding to the requested ROI.

This contribution proposes modified syntax for signalling skipped slice/tile:

* SPS-level syntax to indicate whether all tiles included in CVS are motion constrained.
* Slice-level non-significant slice/tile indication flag signalled when multiple tiles are used and when either all tiles are motion constrained or all tiles are inter-layer constrained.
* Slice-level syntax for the number of skipped CTUs in skipped slice/tile based on maximum number of CTUs included in the slice/tile
* Bitstream constraint on conformance cropping window to include only non-skipped tiles.

# Introduction

In recent meetings, there are several adoptions and contributions relating to the ROI functionality in HEVC or SHVC. One potential solution to achieve ROI is to apply independently decodable tiles or tile sets. In that scenario, so-called skipped tile is useful. With skipped tile, tiles outside of ROI could be indicated and associated slice data could be omitted. There are several approaches for indication of skipped tiles such as skipped slice flag in slice segment header (JCTVC-M0269, M0046), skipped tile set flag in constrained tile set SEI (JCTVC-M0383), and new NAL unit type for VCL associated with skipped tile (JCTVC-M0277).

In this contribution, we present tile skip flag signalled in slice segment header in enhancement layer when independent tiles are enabled. In order to signal independent decodability of tiles, new indicator syntax is also introduced in SPS extension. With these modifications, SHVC could support ROI-capable video. In the rest of this contribution, we firstly introduce ROI-capabled video application using SHVC, then we will introduce details of proposed changes to support skipped tiles in SHVC.

# Application: ROI-capable video using SHVC

In video applications, some region could be more important compared to the other region for users as well as content provider. Such more important region is called ROI (Region of Interest). Users tend to expect better quality for ROI compared to non-ROI area. ROI could be displayed picture in picture form, or it could be displayed on a secondary display (e.g. tablets or smart phones) while full picture displayed on the primary display.

Fig. 1 illustrates ROI-capable multi-layer video distribution system based on SHVC. SHVC bitstream is generated by encoding video with higher and lower quality/resolutions. If display receives whole SHVC bitstream, it could decode and display higher quality video. When a user indicates ROI by some means, intelligent network element would extract part of SHVC bitstream and send it to the user. The user could decode and display ROI of the higher quality video.

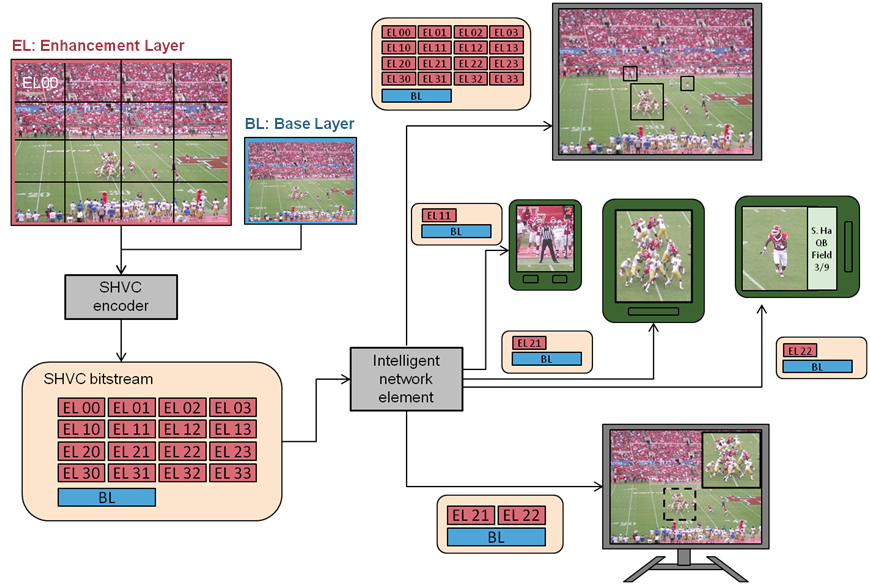


Fig. 1: ROI-capable video application using SHVC

**User benefits:**

* Users can select ROI and can enjoy higher quality ROI.
* Users with limited bandwidth can still enjoy ROI.
  + Since only part of EL data will be needed to display higher-quality ROI, transferred data is relatively small.
* Users do not need to wait when changing ROI.
  + Since BL data is always available, lower-quality ROI (scaled BL decoded video) can be displayed immediately after request of changing ROI. When EL data for new ROI is transferred, higher-quality ROI video will be displayed.

**System components and data requirements:**

* SHVC encoder / SHVC bitstream
  + Part of EL bitstream need to be decoded independent of rest parts of EL. Potentially, EL encoded with independently decodable tiles can be used for the purpose. When encoding such EL tile, non-collocated part of BL region could be referenced via inter-layer prediction, although disallowing such referencing may bring additional benefits to some more bandwidth-sensitive applications.
* Intelligent Network Element (INE)
  + INE receives ROI requests from a user.
  + INE extracts the sub bitstream corresponding to the requested ROI from the whole bitstreams.
    - Extraction process should be easy. INE could discard NAL units based on NAL unit header, VPS, and SPSs. INE overwrite syntax values in VPS, SPS and PPS parameters. INE might overwrite some slice header syntax corresponding to the ROI but should not modify slice data corresponding to the non-ROI. INE should not use information acquired by decoding slice data (e.g. decoded picture, syntax elements below slice data, etc.).
  + INE transfers the extracted sub bitstream to the user.
    - Regions corresponding to the extracted sub-bitstream is not necessarily match with the requested ROI. However, if INE could extract smaller area including ROI, it would be better.
* Client terminal
  + Client terminal shall have SHVC decoder to decode sub bitstream corresponding to ROI.
  + Client terminal shall be equipped with some means to select ROI and to signal the selected ROI to the INE. Client terminal might display whole regions of video (using decoded BL for example) to help users to select ROI.
  + Note that client terminal might be composed of several devices (e.g. primary display for showing whole region and for selecting ROI, secondary display for showing ROI).

# Proposal

Proposed modifications to support skipped tiles in SHVC is as follows.

## Proposed modification in SPS extension

New syntax all\_tiles\_decoding\_dependency\_flag is added. It indicates independent decodability of tiles included in CVS. Spcifically, it indicates whether all tiles depend only on pixel values or syntax values of the collocated tiles when a picture in the same layer is referenced (motion constraint).

Yellow indicates proposed changes.

|  |  |
| --- | --- |
| sps\_extension ( ) { | Descriptor |
| … |  |
| **all\_tiles\_decoding\_dependency\_flag** | u(1) |
| } |  |

**all\_tiles\_decoding\_dependency\_flag** equal to 1 specifies that all tiles in CVS are motion constrained. all\_tiles\_decoding\_dependency\_flag equal to 0 specifies that all tiles in CVS are not motion constrained.

## Proposed modification in Slice header

Syntax non\_significant\_tile\_flag and num\_ctb\_in\_slice\_segment\_minus1 are added to indicate skip tile and number of luma CTBs in a slice in skipped tile. They are signalled when multiple tiles are used and tiles can be decoded independently in temporal prediction.

|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| … |  |
| … |  |
| if( slice\_segment\_header\_extension\_present\_flag ) { |  |
| **slice\_segment\_header\_extension\_length** | ue(v) |
| if ( tiles\_enabled\_flag && all\_tiles\_decoding\_dependency\_flag ) |  |
| { |  |
| **non\_significant\_tile\_flag** | u(1) |
| if( non\_significant\_tile \_flag ) |  |
| **num\_ctu\_in\_slice\_segment\_minus1** | ue(v) |
| } |  |
| **slice\_segment\_header\_extension2\_present\_flag** | u(1) |
| if (slice\_segment\_header\_extension2\_present\_flag) { |  |
| **slice\_segment\_header\_extension2\_length** | ue(v) |
| for( i = 0; i < slice\_segment\_header\_extension2\_length; i++) |  |
| **slice\_segment\_header\_extension\_data\_byte**[ i ] | u(8) |
| } |  |
| } |  |
| byte\_alignment( ) |  |
| } |  |

**non\_significant\_tile\_flag** equal to 1 specifies that the tile containing the slice is the non-significant tile. non\_significant\_tile\_flag equal to 0 specifies that the tile containing the slice is not non-significant tile. When not present, non\_significant\_tile\_flag is inferred to be equal to 0.

**num\_ctb\_in\_slice\_segment\_minus1** plus 1specifies the number ofCTUs in the current slice segment. The length of the num\_ctu\_in\_slice\_segment\_minus1 syntax element is Ceil( Log2( ColWidth[tileIdx] \* RowHeight[tileIdx] ) bits, where tileIdx is equal to TileId[ CtbAddrRsToTs[ slice\_segment\_address ] ].

## Proposed modification to slice segment layer

When a slice segment is non-significant, the slice\_segment\_data( ) syntax structure is not present in the coded bitstream.

|  |  |
| --- | --- |
| slice\_segment\_layer\_rbsp( ) { | Descriptor |
| slice\_segment\_header( ) |  |
| if (!non\_significant\_slice\_segment\_flag) |  |
| slice\_segment\_data( ) |  |
| rbsp\_slice\_segment\_trailing\_bits( ) |  |
| } |  |

## Proposed decoding process for non-significant slice segment with tiles

All pixels in all CTU in a non-significant slice segment are not required to be decoded. When they are decoded, they shall have the value of 0.

Note: The region described above is not referenced due to all\_tiles\_decoding\_dependency\_flag and loop\_filter\_across\_tiles\_enabled\_flag setting. In addition, the region is not included in output due to conformance window setting.

## Proposed semantics change

**loop\_filter\_across\_tiles\_enabled\_flag** equal to 1 specifies that in-loop filtering operations may be performed across tile boundaries in pictures referring to the PPS. loop\_filter\_across\_tiles\_enabled\_flag equal to 0 specifies that in-loop filtering operations are not performed across tile boundaries in pictures referring to the PPS. The in-loop filtering operations include the deblocking filter and sample adaptive offset filter operations. When not present, the value of loop\_filter\_across\_tiles\_enabled\_flag is inferred to be equal to 1.

It is a requirement of bitstream conformance that, when nuh\_layer\_id associated with the PPS is greater than 0 and non-significant slice exists in the current CVS, loop\_filter\_across\_tiles\_enabled\_flag shall be equal to 0.

**conformance\_window\_flag** equal to 1 indicates that the conformance cropping window offset parameters follow next in the SPS. conformance\_window\_flag equal to 0 indicates that the conformance cropping window offset parameters are not present.

**conf\_win\_left\_offset**, **conf\_win\_right\_offset**, **conf\_win\_top\_offset**, and **conf\_win\_bottom\_offset** specify the samples of the pictures in the CVS that are output from the decoding process, in terms of a rectangular region specified in picture coordinates for output. When conformance\_window\_flag is equal to 0, the values of conf\_win\_left\_offset, conf\_win\_right\_offset, conf\_win\_top\_offset, and conf\_win\_bottom\_offset are inferred to be equal to 0.

The conformance cropping window contains the luma samples with horizontal picture coordinates from SubWidthC \* conf\_win\_left\_offset to pic\_width\_in\_luma\_samples − ( SubWidthC \* conf\_win\_right\_offset + 1 ) and vertical picture coordinates from SubHeightC \* conf\_win\_top\_offset to pic\_height\_in\_luma\_samples − ( SubHeightC \* conf\_win\_bottom\_offset + 1 ), inclusive.

The value of SubWidthC \* ( conf\_win\_left\_offset + conf\_win\_right\_offset ) shall be less than pic\_width\_in\_luma\_samples, and the value of SubHeightC \* ( conf\_win\_top\_offset + conf\_win\_bottom\_offset ) shall be less than pic\_height\_in\_luma\_samples.

When ChromaArrayType is not equal to 0, the corresponding specified samples of the two chroma arrays are the samples having picture coordinates ( x / SubWidthC, y / SubHeightC ), where ( x, y ) are the picture coordinates of the specified luma samples.

NOTE  – The conformance cropping window offset parameters are only applied at the output. All internal decoding processes are applied to the uncropped picture size.

It is a requirement of bitstream conformance that, when nuh\_layer\_id associated with the SPS is greater than 0 and non-significant slice exists in the current CVS, conformance\_window\_flag in the SPS shall be equal to 1 and the luma samples contained in the conformance cropping window shall not include any luma samples decoded from the non-significant slice.

# Conclusion

This contribution introduces an applications for skipped tile where media box convert scalable bitstream encoded with constraint tiles to ROI-oriented bitstream with skipped tiles. Relating to the application, this contribution proposes modified syntax for signalling skipped slice/tile. The proposed modification inclueds 1) SPS-level syntax regarding dependency between tiles in CVS, 2) slice-level non-significant slice/tile indication flag signalled when multiple tiles are used and when tiles are independently coded, 3) slice-level syntax for the number of skipped CTUs based on maximum number of CTUs in the slice/tile, and 4) bitstream constraint on conformance window to only include non-skipped tiles. Since skipped tile is needed to support the introduced ROI application, we recommend to agree on introducing skipped tile in SHVC and adopting proposed changes.

# Patent rights declaration(s)

**SHARP Corporation 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).**