|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, CH, 21-30 November, 2011 | Document: JCTVC-G1036d03 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Common conditions for reference picture marking and list construction proposals** | | |
| *Status:* | Input document to JCT-VC | | |
| *Purpose:* | Information | | |
| *Author(s) or Contact(s):* | Ye-Kui Wang Miska M. Hannuksela TK Tan (editors) Rickard Sjöberg | Email: | [yekuiw@qualcomm.com](mailto:yekuiw@qualcomm.com) [miska.hannuksela@nokia.com](mailto:miska.hannuksela@nokia.com) [tktan.p@rd.nttdocomo.co.jp](mailto:tktan.p@rd.nttdocomo.co.jp) [rickard.sjoberg@ericsson.com](mailto:rickard.sjoberg@ericsson.com) |
| *Source:* | AHG21 | | |

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

# Abstract

This document describes common test conditions for the following types of input contributions:

* Signaling of picture identification in slice header (similar to picture order count)
* Signaling of reference picture set
* Signaling of reference picture lists (including reference picture list modification commands, if present), including lists 0, 1, and the combined list.
* Signaling of decoded reference picture marking, if present

Note that the common conditions may also be applied for other high-level syntax proposals, such as coding of syntax elements in sequence parameter set (including VUI), picture parameter set, adaptation parameter set, and slice header (including syntax elements other than picture identification).

# Procedure for using the common conditions

This document specifies the common test conditions for the following types of input contributions:

* Signaling of picture identification in slice header (similar to picture order count)
* Signaling of reference picture set
* Signaling of reference picture lists (including reference picture list modification commands, if present), including lists 0, 1, and the combined list.
* Signaling of decoded reference picture marking, if present

The input contributions using these common conditions shall report the bit count used for all related syntax elements and structures in sequence parameter set, picture parameter set, adaptation parameter set, and slice header.

The subsequent sections specify the prediction structures for which the bit counts shall be reported. Many of the prediction structures are described below in a table form. One row in the table corresponds to one frame. The rows are in decoding order.

In addition, the input contributions using these common conditions

* should verify that the proposed technology also operates on temporal subsets of the bitstreams corresponding to the prediction structures in the following sections
* (for low-delay prediction structures) should analyze the error robustness, for example assuming a maximum of two consecutive entirely lost pictures having the same temporal\_id, and to ensure that loss of reference pictures used for reference by received pictures is detectable

# High-delay prediction structures

The reference picture lists and pictures marked as "used for reference" for the high-delay prediction structures differ from the first “random access cycle” of a coded video sequence (from the beginning of the sequence starting with an IDR access unit to the next IDR/CRA access unit, exclusive) and for each subsequent random access cycle (from a CRA access unit, inclusive, to the next IDR/CRA access unit, exclusive) in the same coded video sequence. The bit count overhead shall be reported separately for the first random access cycle and a subsequent random access cycle.

Each picture shall be coded as one slice, and the lowest QP as required to be tested per the common conditions as specified in JCTVC-G1200 shall be used.

## Random access common conditions

GOP size 8 is used. RAPs are coded in roughly every one second. For example, if the frame rate is 15 fps, then a CRA picture is coded for every 2 GOPs; if the frame rate is 30 fps, then a CRA picture is coded for every 4 GOPs; if the frame rate is 60 fps, a CRA picture is coded for every 8 GOPs.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - |  |
| 8 | B | 1 | 0 | 0 | 0 |  |
| 4 | B | 1 | 0 | 0 8 | 8 0 |  |
| 2 | B | 1 | 0 | 0 4 | 4 8 |  |
| 1 | b | 0 | 0 | 0 2 | 2 4 | 8 |
| 3 | b | 0 | 0 | 2 0 | 4 8 |  |
| 6 | B | 1 | 0 | 4 2 | 8 4 | 0 |
| 5 | b | 0 | 0 | 4 0 | 6 8 |  |
| 7 | b | 0 | 0 | 6 4 | 8 6 | 0 |
| 16 | B | 1 | 0 | 8 6 4 0 | 8 6 4 0 |  |
| 12 | B | 1 | 0 | 8 6 | 16 8 |  |
| 10 | B | 1 | 0 | 8 6 | 12 16 |  |
| 9 | b | 0 | 0 | 8 10 | 10 12 | 16 |
| 11 | b | 0 | 0 | 10 8 | 12 16 |  |
| 14 | B | 1 | 0 | 12 10 | 16 12 | 8 |
| 13 | b | 0 | 0 | 12 8 | 14 16 |  |
| 15 | b | 0 | 0 | 14 12 | 16 14 | 8 |
| 24 | B | 1 | 0 | 16 14 12 8 | 16 14 12 8 |  |
| … |  |  |  |  |  |  |

## Dyadic and nested temporal scalability with 1 reference frame per list

GOP size 8 is used. Random access points (RAPs) are coded in roughly every one second. For example, if the frame rate is 15 fps, then a CRA picture is coded for every 2 GOPs; if the frame rate is 30 fps, then a CRA picture is coded for every 4 GOPs; if the frame rate is 60 fps, a CRA picture is coded for every 8 GOPs.

This structure may be used by encoders which are unable to estimate motion from more than one reference frame per list, when temporal scalability is desired, and temporal layer switching is desired at any access unit.

Two structures are possible. In the first structure, pictures with the same temporal\_id in one GOP may be interleaved in decoding order (as shown in Table 1), while in the second structure all pictures with the same temporal\_id in one GOP are continuous in decoding order (as shown in Table 2).

Simulation results for both structures, as shown in Table 1 and Table 2, shall be provided. In Table 1 and Table 2, it is assumed that the frame rate is 15 fps. For other frame rates, the prediction structures shall be adapted accordingly, as described above.

**Table 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - | - |
| 8 | B | 1 | 0 | 0 | 0 | - |
| 4 | B | 1 | 1 | 0 | 8 | - |
| 2 | B | 1 | 2 | 0 | 4 | 8 |
| 1 | B | 0 | 3 | 0 | 2 | 4, 8 |
| 3 | B | 0 | 3 | 2 | 4 | 8 |
| 6 | B | 1 | 2 | 4 | 8 | - |
| 5 | B | 0 | 3 | 4 | 6 | 8 |
| 7 | B | 0 | 3 | 6 | 8 | - |
| 16 | CRA | 1 | 0 | - | - | 8 |
| 12 | B | 1 | 1 | 8 | 16 | - |
| 10 | B | 1 | 2 | 8 | 12 | 16 |
| 9 | B | 0 | 3 | 8 | 10 | 12, 16 |
| 11 | B | 0 | 3 | 10 | 12 | 16 |
| 14 | B | 1 | 2 | 12 | 16 | - |
| 13 | B | 0 | 3 | 12 | 14 | 16 |
| 15 | B | 0 | 3 | 14 | 16 | - |
| 24 | B | 1 | 0 | 16 | 16 | - |
| … |  |  |  |  |  |  |

**Table 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - | - |
| 8 | B | 1 | 0 | 0 | 0 | - |
| 4 | B | 1 | 1 | 0 | 8 | - |
| 2 | B | 1 | 2 | 0 | 4 | 8 |
| 6 | B | 1 | 2 | 4 | 8 | 0 |
| 1 | B | 0 | 3 | 0 | 2 | 4, 6, 8 |
| 3 | B | 0 | 3 | 2 | 4 | 6, 8 |
| 5 | B | 0 | 3 | 4 | 6 | 8 |
| 7 | B | 0 | 3 | 6 | 8 | - |
| 16 | CRA | 1 | 0 | - | - | 8 |
| 12 | B | 1 | 1 | 8 | 16 | - |
| 10 | B | 1 | 2 | 8 | 12 | 16 |
| 14 | B | 1 | 2 | 12 | 16 | 8 |
| 9 | B | 0 | 3 | 8 | 10 | 12, 14,16 |
| 11 | B | 0 | 3 | 10 | 12 | 14, 16 |
| 13 | B | 0 | 3 | 12 | 14 | 16 |
| 15 | B | 0 | 3 | 14 | 16 | - |
| 24 | B | 1 | 0 | 16 | 16 | - |
| … |  |  |  |  |  | - |

## Dyadic and nested temporal scalability with 2 reference frames per list

GOP size 8 is used. RAPs are coded in roughly every one second. For example, if the frame rate is 15 fps, then a CRA picture is coded for every 2 GOPs; if the frame rate is 30 fps, then a CRA picture is coded for every 4 GOPs; if the frame rate is 60 fps, a CRA picture is coded for every 8 GOPs.

This structure may be used by encoders which are unable to estimate motion from more than two reference frames per list, when temporal scalability is desired, and temporal layer switching is desired at any access unit.

Two structures are possible. In the first structure, pictures with the same temporal\_id in one GOP may be interleaved in decoding order (as shown in Table 3), while in the second structure all pictures with the same temporal\_id in one GOP are continuous in decoding order (as shown in Table 4).

Simulation results for both structures as shown in Table 3 and Table 4 shall be provided. In Table 3 and Table 4, it is assumed that the frame rate is 15 fps. For other frame rates, the prediction structures shall be adapted accordingly, as described above.

**Table 3**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - | - |
| 8 | B | 1 | 0 | 0 | 0 | - |
| 4 | B | 1 | 1 | 0, 8 | 8, 0 | - |
| 2 | B | 1 | 2 | 0, 4 | 4, 8 | - |
| 1 | B | 0 | 3 | 0, 2 | 2, 4 | 8 |
| 3 | B | 0 | 3 | 2, 0 | 4, 8 | - |
| 6 | B | 1 | 2 | 4, 0 | 8, 4 | - |
| 5 | B | 0 | 3 | 4, 0 | 6, 8 | - |
| 7 | B | 0 | 3 | 6, 4 | 8, 6 | - |
| 16 | CRA | 1 | 0 | - | - | 8 |
| 12 | B | 1 | 1 | 8, 16 | 16, 8 | - |
| 10 | B | 1 | 2 | 8, 12 | 12, 16 | - |
| 9 | B | 0 | 3 | 8, 10 | 10, 12 | 16 |
| 11 | B | 0 | 3 | 10, 8 | 12, 16 | - |
| 14 | B | 1 | 2 | 12, 8 | 16, 12 | - |
| 13 | B | 0 | 3 | 12, 8 | 14, 16 | - |
| 15 | B | 0 | 3 | 14, 12 | 16, 14 | - |
| 24 | B | 1 | 0 | 16 | 16 | - |
| … |  |  |  |  |  |  |

**Table 4**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | Reference picture list 0 | Reference picture list 1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - | - |
| 8 | B | 1 | 0 | 0 | 0 | - |
| 4 | B | 1 | 1 | 0, 8 | 8, 0 | - |
| 2 | B | 1 | 2 | 0, 4 | 4, 8 | - |
| 6 | B | 1 | 2 | 4, 0 | 8, 4 | 2 |
| 1 | B | 0 | 3 | 0, 2 | 2, 4 | 6, 8 |
| 3 | B | 0 | 3 | 2, 0 | 4, 8 | 6 |
| 5 | B | 0 | 3 | 4, 0 | 6, 8 | - |
| 7 | B | 0 | 3 | 6, 4 | 8, 6 | - |
| 16 | CRA | 1 | 0 | - | - | 8 |
| 12 | B | 1 | 1 | 8, 16 | 16, 8 | - |
| 10 | B | 1 | 2 | 8, 12 | 12, 16 | - |
| 14 | B | 1 | 2 | 12, 8 | 16, 12 | 10 |
| 9 | B | 0 | 3 | 8, 10 | 10, 12 | 14, 16 |
| 11 | B | 0 | 3 | 10, 8 | 12, 16 | 14 |
| 13 | B | 0 | 3 | 12, 8 | 14, 16 | - |
| 15 | B | 0 | 3 | 14, 12 | 16, 14 | - |
| 24 | B | 1 | 0 | 16 | 16 | - |
| … |  |  |  |  |  |  |

## Dyadic and nested temporal scalability with 2 reference frames per list and higher quality references (semi-long term)

GOP size 8 is used. A CRA picture is coded for every 8 GOPs. Higher quality reference are kept in the buffer for a longer time and used for predicting the key pictures to reduce the rate of decay in the quality of the pictures.

**Table 5**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - | - |
| 8 | B | 1 | 0 | 0 | 0 | - |
| 4 | B | 1 | 1 | 0, 8 | 8, 0 | - |
| 2 | B | 1 | 2 | 0, 4 | 4, 8 | - |
| 1 | B | 0 | 3 | 0, 2 | 2, 4 | 8 |
| 3 | B | 0 | 3 | 2, 0 | 4, 8 | - |
| 6 | B | 1 | 2 | 4, 0 | 8, 4 | - |
| 5 | B | 0 | 3 | 4, 0 | 6, 8 | - |
| 7 | B | 0 | 3 | 6, 4 | 8, 6 | - |
| …. |  |  |  |  |  |  |
| 48 | B | 1 | 0 | 40, 32, 24, 16 | 40, 32, 24, 16 | - |
| 44 | B | 1 | 1 | 40, 32 | 48, 40 | 24 |
| 42 | B | 1 | 2 | 40, 32 | 44, 48 | 24 |
| 41 | B | 0 | 3 | 40, 32 | 42, 44 | 24, 48 |
| 43 | B | 0 | 3 | 42, 40 | 44, 48 | 24, 32 |
| 46 | B | 1 | 2 | 44, 40 | 48, 44 | 24, 32 |
| 45 | B | 0 | 3 | 44, 40 | 46, 48 | 24, 32 |
| 47 | B | 0 | 3 | 46, 44 | 48, 46 | 24, 32, 40 |
| … | B | 1 | 0 |  |  |  |
| 64 | CRA | 1 | 0 | - | - | 56, 48, 40 |
| … | B | 1 | 0 |  |  |  |

## Dyadic and nested temporal scalability for high frame rate

Modern digital still and video cameras can capture at high frame rates, such as 240 Hz, thus enabling impressive slow motion capturing. It is advantageous if such high-frame-rate sequences are coded in a temporally scalable manner in order to enable sub-bitstream extraction and playback at different speeds and with different decoder capabilities.

Dyadic and nested temporal scalability and reference picture lists are as in Section 2.2, but GOP size is 64 and temporal\_id values from 0 to 7, inclusive, are allocated. GOP size 64 at frame rate 240 Hz provides that same interval for consecutive pictures having temporal\_id equal to 0 than GOP size 8 at frame rate 30 Hz.

## Long-term referenece picture use in high-delay prediction structures

This test simulates interviewing scenarios, wherein two or more sides are talking, the video to be shown switches between the sides, from a few tens of seconds to a few minutes.

Simulation results of the following two sets of tests in this category should be reported.

The RA coding structure without CRA pictures as specified in JCTVC-G1200 shall be applied.

In the first test set, the following configurations apply:

* The first scene (the interviewer) lasts 20 seconds, followed by the second scene (the interviewee) that lasts 60 seconds, in output order, and the pattern repeats in the same manner.
* When encoding the first picture of one scene after the period of another scene, in output order, the last picture, in output order, of the same scene with the same or lower temporal\_id is used for reference, e.g., as a long-term reference picture. However, it is also allowed to use only short-term reference pictures, at the discretion of the proponents.

In the second test set, the same configuration as in the first test set, but with the first scene period being 10 seconds and the second scene being 30 seconds.

## Dyadic coding structure with 2 reference frames

This picture structure is identical to the random access common condition settings as in HM-4.1 but with temporal\_id turned on. GOP size 8 is used with a CRA distance of 32. Note that the AHG21 setting for the version of the code at the Geneva meeting is like this.

This test set includes the use of reference pictures with the same temporal\_id for pictures with temporal\_id greater than 0.

# Type POC QPoffset QPfactor temporal\_id ref\_buf\_size ref\_pic #ref\_pics reference pictures predict deltaRIdx-1 deltaRPS #ref\_idcs reference idcs

Frame1: B 8 1 0.442 0 4 1 2 -8 -10 0

Frame2: B 4 2 0.3536 1 2 1 3 -4 -6 4 1 0 4 3 1 2 1

Frame3: B 2 3 0.3536 2 2 1 4 -2 -4 2 6 1 0 2 4 1 1 1 1

Frame4: B 6 3 0.3536 2 2 1 5 -2 -4 -6 -8 2 1 0 -4 5 1 1 1 1 1

Frame5: B 1 4 0.68 3 2 0 6 -1 -3 1 3 5 7 1 0 5 6 1 1 1 1 1 1

Frame6: B 3 4 0.68 3 2 0 5 -1 -3 1 3 5 1 0 -2 7 1 0 1 1 1 1 0

Frame7: B 5 4 0.68 3 2 0 4 -1 -3 1 3 1 0 -2 6 1 0 1 1 1 0

Frame8: B 7 4 0.68 3 2 0 4 -1 -3 -5 1 1 0 -2 5 1 1 1 1 0

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - |  |
| 8 | B | 1 | 0 | 0 | 0 |  |
| 4 | B | 1 | 1 | 0, 8 | 8, 0 |  |
| 2 | B | 1 | 2 | 0, 4 | 4, 8 |  |
| 6 | B | 1 | 2 | 4, 2 | 8, 4 | 0, |
| 1 | b | 0 | 3 | 0, 2 | 2, 4 | 6, 8 |
| 3 | b | 0 | 3 | 2, 0 | 4, 6 | 8 |
| 5 | b | 0 | 3 | 4, 2 | 6, 8 |  |
| 7 | b | 0 | 3 | 6, 4 | 8, 6 | 2 (optional) |
| 16 | B | 1 | 0 | 8 | 8 |  |
| 12 | B | 1 | 1 | 8, 16 | 16, 8 | 6 |
| 10 | B | 1 | 2 | 8, 6 | 12, 16 |  |
| 14 | B | 1 | 2 | 12, 10 | 16, 12 | 6, 8 |
| 9 | b | 0 | 3 | 8, 6 | 10, 12 | 14, 16 |
| 11 | b | 0 | 3 | 10, 8 | 12, 14 | 16 |
| 13 | b | 0 | 3 | 12, 10 | 14, 16 |  |
| 15 | b | 0 | 3 | 14, 12 | 16, 14 | 10 (optional) |
| 24 | B | 1 | 0 | 16 | 16 |  |
| … |  |  |  |  |  |  |

# Low-delay prediction structures

For all low-delay prediction structures, the 1500 bytes per slice encoding configuration (using SliceMode=2 and SliceArgument=1500) shall be applied.

For a particular test sequence, the highest QP as required to be tested per the common conditions as specified in JCTVC-G1200 shall be used.

## Low-delay common conditions

This low-delay coding structure is as follows.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - |  |
| 1 | B | 1 | 0 | 0 | 0 |  |
| 2 | B | 1 | 0 | 1 0 | 1 0 |  |
| 3 | B | 1 | 0 | 2 1 0 | 2 1 0 |  |
| 4 | B | 1 | 0 | 3 2 1 0 | 3 2 1 0 |  |
| 5 | B | 1 | 0 | 4 3 2 0 | 4 3 2 0 |  |
| 6 | B | 1 | 0 | 5 4 3 0 | 5 4 3 0 |  |
| 7 | B | 1 | 0 | 6 5 4 0 | 6 5 4 0 |  |
| 8 | B | 1 | 0 | 7 6 4 0 | 7 6 4 0 |  |
| 9 | B | 1 | 0 | 8 7 4 0 | 8 7 4 0 |  |
| 10 | B | 1 | 0 | 9 8 4 0 | 9 8 4 0 |  |
| 11 | B | 1 | 0 | 10 8 4 0 | 10 8 4 0 |  |
| 12 | B | 1 | 0 | 11 8 4 0 | 11 8 4 0 |  |
| 13 | B | 1 | 0 | 12 8 4 0 | 12 8 4 0 |  |
| 14 | B | 1 | 0 | 13 12 8 4 | 13 12 8 4 |  |
| 15 | B | 1 | 0 | 14 12 8 4 | 14 12 8 4 |  |
| 16 | B | 1 | 0 | 15 12 8 4 | 15 12 8 4 |  |
| 17 | B | 1 | 0 | 16 12 8 4 | 16 12 8 4 |  |
| … |  |  |  |  |  |  |

## Low-delay temporal scalability

This prediction structure enables low-delay and use of system-layer error robustness methods unequally per temporal\_id.

**Table 6**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - | - |
| 1 | P | 0 | 2 | 0 | - | - |
| 2 | P | 1 | 1 | 0 | - | - |
| 3 | P | 0 | 2 | 2 | - | 0 |
| 4 | P | 1 | 0 | 0 | - | - |
| 5 | P | 0 | 2 | 4 | - | - |
| 6 | P | 1 | 1 | 4 | - | - |
| 7 | P | 0 | 2 | 6 | - | 4 |
| 8 | P | 1 | 0 | 4 | - | - |
| … |  |  |  |  |  |  |

## Low-delay with feedback

In this prediction structure it is assumed that the encoder gets feedback from the far-end decoder on the success of decoding. Based on the feedback, the encoder maintains two “key” frames, which are marked as “used for reference”. In addition, the most recent frame is marked as “used for reference”.

The use case is simulated with the following prediction structure, where Ni = {[include a pseudo-randomly generated increasing series of POCs here, e.g. 6 items]} and Mi = {[ Ni - Ri where Ri is a pseudo-random value]}.

[Open issue: It was suggested to use fixed values for Ni and Mi values, instead of using pseudo-randomly generated values. However, there was no concrete proposal of the fixed values.]

**Table 7**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| … |  |  |  |  |  |  |
| Ni | P | 1 | 0 | Ni-1 |  | Mi, Mi-1 |
| Ni+1 | P | 1 | 0 | Ni |  | Mi, Mi-1 |
| Ni+2 | P | 1 | 0 | Ni+1 |  | Mi, Mi-1 |
| … |  |  |  |  |  |  |

The delay between the time when a particular picture is captured at the encoder side and when the encoder side receives the feedback is assumed to be equal to the round trip time (RTT) plus one frame during (this is the minimum, depending on other factors, the value can be higher, but using the minimum should be reasonable).

The distance between “key” pictures Mi and Mi-1 should be derived as follows:

M(i) - M(i-1) = Floor( RTT \* fps / 1000 ) + 1

Such a distance between the two “key” pictures means that the encoder will produce a new "key" picture when it receives feedback about the previous "key" picture.

Simulation results of two RTT values, 200 ms (representing a pretty good connection) and 1000ms (representing a not-so-good connection), should be reported. It is at the discretion of the proponents on whether it is desirable to use long-term reference pictures or not.

When the RTT is set to a very high value, this configuration can also be used to simulate the use case wherein a picture of a background of a telepresence conference room, which is marked as a long-term reference picture at some point (e.g., after the distance between this picture and the current picture is beyond the maximum POC distance for short-term reference pictures), and which stays a very long time in the DPB, for efficient coding of uncovered background regions, and for error recovery. Due to that the length of the sequences is relatively short (less than 10 minutes), for this use case, it is assumed that the “background” long-term reference picture, once available, is kept throughout the entire test sequence.

## Adaptive reference picture selection

This is a test case where the encoder adaptively selects which reference pictures to keep. A real-world encoder would select pictures in order to optimize compression efficiency for each sequence but here we use the same reference pictures to simplify testing.

The reference picture list for each picture is given by the following C-program:

#include <stdio.h> // for printf

#include <stdlib.h> // for qsort

#define NUM\_PICS\_IN\_SEQUENCE 600

#define NUMBER\_OF\_REF\_PICS 5

#define PROBABILITY\_KEEP\_MINUS\_2 95

/\* H.263 annex A pseudo-random generator \*/

long rand (long L, long H)

{

static long randx = 1; /\* long is 32 bits \*/

static double z = (double) 0x7fffffff;

long i,j;

double x; /\* double is 64 bits \*/

randx = (randx \* 1103515245) + 12345;

i = randx & 0x7ffffffe; /\* keep 30 bits \*/

x = ( (double)i ) / z; /\* range 0 to 0.99999 ... \*/

x \*= (L+H+1); /\* range 0 to < L+H+1 \*/

j = (long)x; /\* truncate to integer \*/

return(j - L); /\* range -L to H \*/

}

int comp(const void \* a, const void \* b)

{

int \*aa = (int\*)a;

int \*bb = (int\*)b;

if (\*aa==\*bb)

return 0;

else

{

if (\*aa > \*bb)

return -1;

else

return 1;

}

}

int main(const int argc, const char \*\*argv)

{

int ref[NUMBER\_OF\_REF\_PICS];

int poc;

int i;

for(i = 0 ; i < NUMBER\_OF\_REF\_PICS ; i++)

{

ref[i] = -i-1;

}

printf(" POC L0\n");

for(poc = 0 ; poc <= NUMBER\_OF\_REF\_PICS ; poc++)

{

printf("%3i ", poc);

for(i = 0 ; i < poc ; i++)

printf(" %i ", ref[i] + poc);

printf("\n");

}

for(poc = NUMBER\_OF\_REF\_PICS+1 ; poc < NUM\_PICS\_IN\_SEQUENCE ; poc++)

{

for(i = 1 ; i < NUMBER\_OF\_REF\_PICS ; i++)

ref[i]--;

if(rand(0,99) < PROBABILITY\_KEEP\_MINUS\_2)

{

int index = rand(0,10000) % (NUMBER\_OF\_REF\_PICS-1);

ref[index+1] = -2;

}

qsort(ref, NUMBER\_OF\_REF\_PICS, sizeof(int), comp);

printf("%3i ", poc);

for(i = 0 ; i < NUMBER\_OF\_REF\_PICS ; i++)

printf(" %i ", ref[i] + poc);

printf("\n");

}

}

The reference picture lists for the first pictures are given by table 11.

**Table 8**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| POC | pic type | nal\_ref\_flg | t\_id | RefPicList0 | RefPicList1 | Other pictures marked as “used for reference” |
| 0 | IDR | 1 | 0 | - | - | - |
| 1 | P | 1 | 0 | 0 | - | - |
| 2 | P | 1 | 0 | 1, 0 | - | - |
| 3 | P | 1 | 0 | 2, 1, 0 | - | - |
| 4 | P | 1 | 0 | 3, 2, 1, 0 | - | - |
| 5 | P | 1 | 0 | 4, 3, 2, 1, 0 | - | - |
| 6 | P | 1 | 0 | 5, 4, 3, 1, 0 | - | - |
| 7 | P | 1 | 0 | 6, 5, 4, 1, 0 | - | - |
| 8 | P | 1 | 0 | 7, 6, 5, 1, 0 | - | - |
| 9 | P | 1 | 0 | 8, 7, 5, 1, 0 | - | - |
| 10 | P | 1 | 0 | 9, 8, 7, 5, 1 | - | - |
| … |  |  |  |  |  |  |