



Title	Hierarchical-P reference picture selection based error resilient video coding framework for high efficiency video coding transmission applications
Author(s)	Maung, Htoo Maung; Aramvith, Supavadee; Miyanaga, Yoshikazu
Citation	Electronics, 8(3), 310 https://doi.org/10.3390/electronics8030310
Issue Date	2019-03-11
Doc URL	http://hdl.handle.net/2115/74496
Rights(URL)	https://creativecommons.org/licenses/by/4.0/
Type	article
File Information	electronics-08-00310.pdf



[Instructions for use](#)

Article

Hierarchical-P Reference Picture Selection Based Error Resilient Video Coding Framework for High Efficiency Video Coding Transmission Applications

Htoo Maung Maung ¹, Supavadee Aramvith ^{1,*} and Yoshikazu Miyanaga ²

¹ Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10330, Thailand; hmaungmaung@gmail.com

² Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan; miya@ist.hokudai.ac.jp

* Correspondence: supavadee.a@chula.ac.th

Received: 31 January 2019; Accepted: 5 March 2019; Published: 11 March 2019



Abstract: In this paper, a new reference picture selection (RPS) is proposed for a high efficiency video coding (HEVC) framework. In recent studies, HEVC has been shown to be sensitive to packet error which is unavoidable in transmission applications especially for wireless networks. RPS is an effective error resilient technique for video transmission systems where a feedback channel with short round trip delay time is available. However, its procedure cannot directly apply to the HEVC framework and thus this paper expands it. In RPS, error propagation can still happen during round trip delay time. To alleviate the effect of error propagation for better quality, the proposed algorithm considers both the RPS technique and the region-based intra mode selection method by using some novel features of HEVC. Experimental results demonstrate that the proposed method outperforms the hierarchical-P RPS algorithm in terms of PSNR and other metrics. The average PSNR improvement of the proposed algorithm over the reference algorithm under 10% packet error rate is 1.56 dB for 1080p sequences, 2.32 dB for 720p sequences and 1.01 dB for wide video graphics array (WVGA) sequences, respectively. The performance of proposed method is also tested for applications where feedback information is not available. The proposed method shows noticeable improvement for video sequences that contain low or moderate level of motions.

Keywords: HEVC; video coding; error resilient coding; error propagation; wireless video transmission; video communications; reference picture selection

1. Introduction

Video transmission over mobile devices has considerably increased during the past years. Due to bandwidth limitations and network packet errors, providing acceptable quality of video to mobile users is a challenging task. The high compression ratio of high efficiency video coding (HEVC) [1] helps for reducing the amount of the network traffic load and mitigates bandwidth demand. However, according to recent studies about the error robustness of HEVC in loss prone networks [2–4], HEVC encoded bit streams are very sensitive to packet errors and the quality of the decoded video is unacceptable for packet loss rates higher than 1%. Therefore, an effective error resilience framework is essential for HEVC video transmission.

The work in Reference [5] uses discrete wavelet transform based coding approach and achieves significant gain at low bit rates compare to HEVC. This method can reduce the output bit rate lower than that of HEVC at the same visual quality. But for applications under the loss prone environment, this method requires further investigation for error robustness. The major video coding standards like

H.264 [6] and HEVC mainly relies on discrete cosine transform. So, this method cannot easily adopt by such kind of standards. Reference picture selection (RPS) [7–9] is an effective error resilient tool if feedback information from the decoder is available. This method is mainly suitable for applications such as video telephony, video conferencing and so forth. An illustration of RPS concept called NEWPRED [8,9] which has been adopted in H.264/AVC is shown in Figure 1.

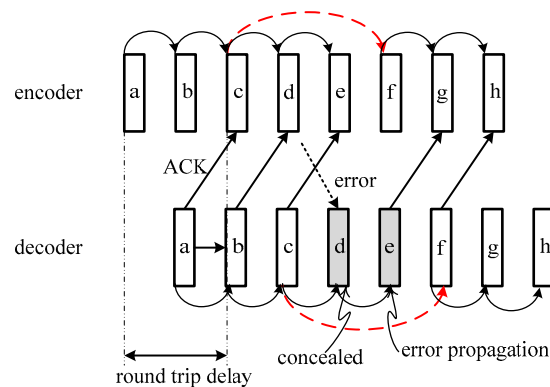


Figure 1. Acknowledgement (ACK) based NEWPRED for a round trip delay of 2 frame intervals.

The feedback signal can be either a negative acknowledgement (NACK) signal or an acknowledgement (ACK) signal. For ACK-based RPS, the encoder will not receive an acknowledgement (ACK) signal if a frame is lost or corrupted. Then, the encoder will select the last correctly received frame as the reference frame so that the effect from error propagation can be suppressed. The operation of NACK-based RPS is similar with the operation of ACK-based RPS. The RPS method can effectively stop error propagation and improve quality by stopping temporal dependencies between frames. The proxy-based reference picture selection method for mobile video telephony scenario was proposed in Reference [10]. In that method, the adaptive reference selection method using H.264/AVC was proposed for wireless uplink transmission which is an extended version of the NEWPRED method with extensions on slice level reference selection. Liu et al. proposed an RPS method for H.264/AVC by using a long-term reference picture [11].

However, previous studies on RPS mainly focused on the IPPP coding structure because this structure is suitable for constrained bit rate applications that require ultra-low delay and/or low complexity. But HEVC uses a hierarchical coding structure to achieve temporal scalability and high compression efficiency [12]. Moreover, HEVC introduces a new reference picture management concept called the reference picture set which is different from the sliding window process and memory management control operation (MMCO) of H.264 [13]. Thus, the above-mentioned RPS techniques cannot immediately be adopted for HEVC. We propose a new RPS algorithm for HEVC, called hierarchical-P RPS, by using a hierarchical coding structure and a reference picture set for low delay video conversational applications. Since the target applications have both bit rate constraint and delay constraint, the proposed algorithm is designed for a hierarchical-P coding structure. Although the hierarchical-B coding structure is well-known for random access applications, the hierarchical-P structure is more appropriate for low delay cases. The pros and cons of the hierarchical-P structure are well presented in Reference [14].

The hierarchical-P RPS method generates a control picture upon receiving a feedback signal from the decoder. Under certain packet loss conditions, there is a high potential that some portions of this control picture will be lost. Thus, error propagation will continue until the next control picture is received. If the feedback delay is long or the feedback signal is lost, there is a significant degradation in the quality of the decoded picture. To reduce this effect, the proposed method inserts intra-coded blocks in hierarchical-P RPS. This work has been partially presented in References [15–17] by using some initial results. But detail discussion of hierarchical-P RPS method for HEVC and the impact of feedback delay on the performance of the proposed algorithm have not yet been provided.

In addition, there are some RPS methods that do not require feedback information. Checkerboard pattern reference picture selection method is one of these techniques [18]. This method stored two reference pictures (default is four reference pictures for HEVC low delay P configuration) in the buffer. The current frame to be encoded is divided into smaller blocks called largest coding units (LCUs). These LCUs are categorized into two groups based on the checkerboard pattern. The blocks in one group are encoded using only one reference picture while the blocks in another group are encoded using remaining reference picture. Thus, if one reference picture contains error, then it will only propagate to only one group. This method can reduce the prediction mismatch at the decoder due to frame loss. Our proposed method requires the packet loss information in order to select the members of reference picture set and to decide the coding mode of region of interest (ROI). For applications where feedback information is not available, our proposed method can still apply by using error estimation model. Gilbert Elliot Model [19], a simple packet loss estimation method, is used in this work to obtain the packet loss information. According to the estimated packet loss information, the proposed algorithm selects the reference pictures and the coding mode of ROI. The performance of the proposed algorithm is compared with that of checkerboard pattern RPS.

This paper is organized as follows. In Section 2, some important background is presented. The details of the proposed algorithm are discussed in Section 3. Section 4 provides the experimental results and Section 5 concludes this paper.

2. Background

In this section, some necessary backgrounds for this work such as two novel features of HEVC and the Lambda domain rate control method are briefly reviewed.

2.1. Novel Features of HEVC

The improvement in coding efficiency of HEVC is obtained by accumulating several small improvements from almost all parts of the encoder. In this section, two most important features related with this research are briefly reviewed. They are the coding tree structure and the reference picture set.

2.1.1. Coding Tree Structure

In the block-based hybrid video coding, each input video frame is firstly split into several blocks and each block is served as the basic coding unit for the whole encoding process. The size of basic unit is typically chosen as 16×16 pixels in almost all previous ITU-T and ISO/IEC video coding standards. However, the fixed block size of basic unit is difficult to adapt the various video contents at different resolutions. In order to get the required flexibility, HEVC introduces the coding tree unit (CTU) [1,13] that can be configure from 8×8 to 64×64 . Depending on the content, the CTU can also be divided into four smaller blocks called coding units (CU). This splitting process repeats recursively till the block size reaches the minimum allowed CU size. After splitting process, each CU serves as the basic unit for the coding process. It should be noted that not every CU becomes the smallest sized CU because the splitting process uses rate-distortion optimization (RDO) to make the decision for partitioning.

2.1.2. Reference Picture Set of HEVC

The only information that required for controlling the reference pictures in H.264/AVC is about the changes in a decoded picture buffer (DPB). The DPB updating is carried out after encoding of a frame is completed.

The new reference picture set concept of HEVC adds the information about the entire reference picture list into each slice header [13]. The DPB updating is carried out by extracting the information from slice header of a frame before decoding it. Thus, other information from earlier pictures in decoding order is not necessary. The usage of an incorrect reference picture can be avoided by using this concept and as a result, better error robustness is achieved.

When the hierarchical-P coding structure is used, each picture in a group of pictures (GOP) is encoded using a different value of the quantization parameter (QP) depending on the hierarchical level of the picture. Pictures at a higher hierarchical level use higher QP values. Hence, the pictures in the lowest hierarchical level have the highest quality and they are referred to as core frames in this paper. Other pictures are referred to as common frames. Since core frames have a better quality than common frames, they are usually kept in the DPB longer than common frames for using as reference pictures.

2.2. Lambda Domain Rate Control Method

The lambda domain rate control [20,21], which is adopted in the HEVC reference software (HM.15.0) [22], uses the Hyperbolic rate distortion model. This method includes bit allocation, quantization parameter computation and updating of model parameters. Bit allocation at the GOP level is computed based on the target bitrate and the current buffer status. This GOP level bit budget and hierarchical level of picture are used for calculating picture level bit budget. Then bit budget for the largest coding unit (LCU) of a frame is computed. For LCU level bit allocation, generated header bits are also taken into account.

After getting the bit budgets, bit per pixel (bpp) value for a given target bit rate is obtained. This value is then used for computing the slope of the rate-distortion curve (λ) by using (1)

$$\lambda = \alpha \cdot bpp^\beta, \quad (1)$$

where α and β are two model parameters. Once the λ value is obtained, the QP can be computed by using (2).

$$QP = 4.2005 \ln \lambda + 13.7122, \quad (2)$$

After encoding a frame or a CTU, the corresponding parameters values are updated. For updating parameters, bpp_{real} is computed from the actual generated bits and the actual λ value, λ_{real} , are used. New parameters can be obtained by using (3) to (5) as follows.

$$\lambda_{comp} = \alpha_{old} \cdot bpp_{real}^{\beta_{old}}, \quad (3)$$

$$\alpha_{new} = \alpha_{old} + \delta_\alpha \cdot (\ln \lambda_{real} - \ln \lambda_{comp}) \cdot \alpha_{old}, \quad (4)$$

$$\beta_{new} = \beta_{old} + \delta_\beta \cdot (\ln \lambda_{real} - \ln \lambda_{comp}) \cdot \ln bpp_{real}, \quad (5)$$

where λ_{comp} is the computed λ , α_{old} is the previous α value and β_{old} is the previous β value. δ_α and δ_β are two constant values. Detail implementation of the lambda domain rate control can be found in References [20,21].

3. Proposed Method

The overall block diagram of the proposed feedback-based error resilient framework for the HEVC video transmission system is shown in Figure 2. Before encoding a frame, the moving region (MR) extraction procedure generates MR-map. Then, the hierarchical-P RPS algorithm manages the reference picture set according to the feedback information. The MR-map is used in CTU-level coding mode decision and rate control. The CTU-level coding mode decision can further improve the quality of the video by selecting intra mode for some important blocks. Rate control is modified to exploit the MR-map and maintain the generated bits to meet the target bit rate while keeping a good quality for blocks in the MR.

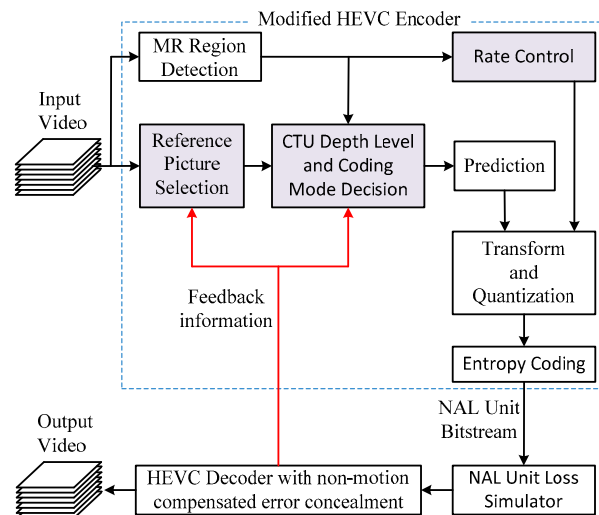


Figure 2. Block diagram of proposed feedback-based error resilient video codec.

3.1. Proposed Hierarchical-P RPS Algorithm

The hierarchical-P coding structure can be obtained by using “low-delay-P” configuration. Under common test conditions for HEVC [23], this configuration defines four pictures in a GOP and four active reference pictures (*ref_pics_active*). When using “low-delay-P” configuration, generally, three core frames and one immediately encoded previous frame serve as reference pictures for the current frame. The immediate previous frame can be either a core or a common frame. For the first few frames of the sequence, more than one common frame is used as a reference since the number of encoded core frames is less than three. The design of the proposed hierarchical-P RPS algorithm is based on this configuration of HEVC. Regarding the feedback channel, the ACK-based system was chosen because it provides better error robustness than the NACK-based system especially when the feedback channel has some errors [3,8].

The proposed hierarchical-P RPS algorithm has two steps: error handling and DPB updating. If an ACK signal is not detected, the index of error frame is firstly determined. Picture order count (POC) is used as the frame index. If the POC of a reference picture in DPB is the same as or higher than that of the error frame, this reference picture is considered as an unreliable picture. If the POC of the reference picture is same as POC of the error frame, this picture contains some errors and any prediction from this picture will cause error propagation. If the POC of the reference picture is higher than that of the error frame, that picture is encoded using the error frame as a reference and error propagation has already happened in this frame. Thus, these kinds of frames are removed from the DPB. This step is to stop temporal dependencies between the error frame and future encoded frames.

The next step includes reference picture selection and DPB updating. Once the error related reference frames are removed from the buffer, the error control process is completed. But the proposed algorithm always tries to keep the number of reference pictures in the reference picture list as close as possible to the *ref_pics_active* value from the configuration file. Thus, the encoder searches new possible reference pictures for current frame in the DPB. Both core and common frames can be used as new reference picture but they must be available in the DPBs of both the encoder and the decoder. If no error is detected before encoding the current frame but the number of reference pictures in the DPB is less than the *ref_pics_active*, all reference pictures in the DPB are used for encoding the current frame.

To better explain the concept, an example from Figure 3 is used. Suppose the detected POC of the error frame is 8. The next frame to be encoded is frame 10 (POC 10). In error free condition, reference pictures kept in DPB for encoding POC 10 are: POC 9, POC 8, POC 4 and POC 0. Since frame 8 has an error, it is unreliable and needs to be removed from the DPB. Frame 9 is also unreliable because frame 8 is one of its reference frames. Therefore, frame 9 also needs to be removed from the DPB.

Hence, the DPB contains frame 4 and frame 0 only (reliable frames). It should be noted that the POCs of reliable frames, that is, POC 4 and POC 0, are less than the POC of the error frame which is POC 8. What the algorithm did here is just remove the reference frame from the DPB where its POC value is the same as or higher than the POC value of the error frame. If there is a reliable previous reference frame which is still available in both encoder and decoder, it will be added to the DPB for encoding frame 10.

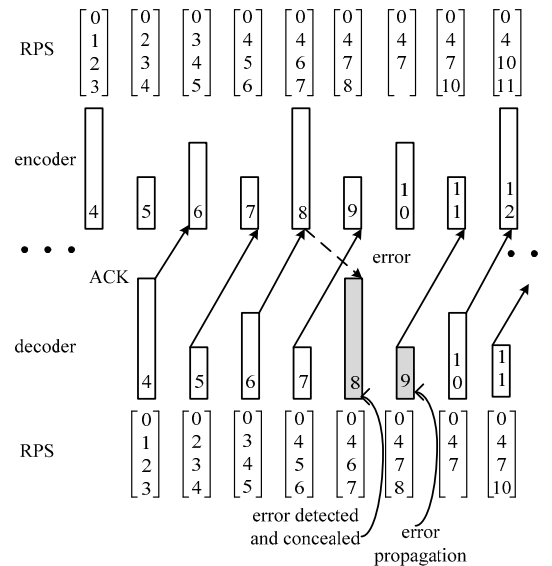


Figure 3. Example of hierarchical-P reference picture selection (RPS) where picture order count (POC) 8 (frame 8) contains error.

In this example, POC 7 is possible. Because POC 7 serves as a reference picture for encoding POC 9 and still remains in the buffer until the DPB update and RPS process for POC 10 is completed. If no error occurs, POC 7 will be removed during the DPB update process for POC 10. But in case of error, it is added to the reference picture list for encoding POC 10. So, the reference pictures for encoding POC 10 become POC 7, POC 4 and POC 0. Using POC 7 as a reference picture can be beneficial because the temporal distance between POC 7 and POC 10 is shorter than that for other reference pictures; there is a high probability it will contain more redundancies and can achieve better motion prediction. The detailed procedure of reference selection and DPB updating of the proposed hierarchical-P RPS is shown in Algorithm 1.

Algorithm 1 Proposed Hierarchical-P RPS Algorithm

- 1: **if** ACK signal is detected **then**
 - 2: **go to** DPB_update
 - 3: **else**
 - 4: **detect** error frame index
 - 5: **remove** unreliable reference picture from DPB
 - 6: DPB_update:
 - 7: **if** active reference picture > *ref_pics_active* **then**
 - 8: **remove** frame with lowest POC from DPB
 - 9: **if** active reference picture < *ref_pics_active* **then**
 - 10: **find** available reference picture in both DPBs
 - 11: **if** found **then**
 - 12: **make** this picture as active reference picture
 - 13: **process** next frame
-

3.2. Proposed CTU-Level Coding Mode Decision

The hierarchical-P RPS algorithm is very responsive to handle error as it modified DPB based on the error status. However, error propagation can still happen in some situations such as long back channel delay or error occurs in the control picture.

To reduce this effect and to improve the quality, the ROI based intra mode selection method is proposed. MR information, which serves as ROI in this research, is extracted by using the frame differencing method as discussed in References [24,25]. If an error is detected, the coding mode for CTUs in the MR region is set as intra mode. However, using large amount of intra-coded blocks can cause bit fluctuation and can affect the quality of the reconstructed video. The decision-making parameter for intra mode selection is the temporal distance between the current frame and the last intra refresh frame (*lidst*). Since the GOP size of hierarchical coding structure with three temporal layers is four, the minimum *lidst* value for the coding mode (CM) decision is set as 4-frame distance in this work. The idea is that each GOP contains one intra refresh frame under the worst-case condition. Hence, intra mode is selected only when *lidst* is 4-frame distance or higher and an error message is received. CM of each CTU can be determined by using (6).

$$CM(p) = \begin{cases} \text{Intra,} & \text{if } p \in MR \text{ and } err_{found} \\ & \text{and } lidst \geq 4, \\ \text{Inter,} & \text{otherwise.} \end{cases}, \quad (6)$$

where, p is the p^{th} LCU in a frame.

3.3. Proposed Rate Control Scheme

Under the packet loss condition, the quality of MR is protected by using intra coding mode because MR region gets more attention than non-MR region. Since intra mode introduces a large number of output bits, rate control algorithm is required to adjust its parameters in order to meet the target bit rate. The lambda domain rate control proposed in References [20,21] considers both frame level and slice level intra mode selection cases. But the blocks in MR normally exit across multiple slices in a frame. Hence, a slice may contain few blocks in MR and these blocks are required to encode with intra mode while the others are still using inter mode. For such condition, block level intra selection is needed and this paper extends the lambda domain rate control to meet the requirement.

The bit budget of the current frame, $T_{CurrPic}$, can be considered as the combination of the bit budget of MR, T_{MR} and the bit budget of non-MR, T_{NMR} , as shown in (7).

$$T_{CurrPic} = T_{MR} + T_{NMR}, \quad (7)$$

From the frame level bit allocation process, $T_{CurrPic}$ is obtained. Since the total number of pixels in the whole frame, N_{Pic} and the number of pixels in MR, N_{MR} , are known from the MR-map information, T_{MR} can be computed by using (8).

$$T_{MR} = T_{CurrPic} \times \frac{N_{MR}}{N_{Pic}} \quad (8)$$

Both T_{MR} and T_{NMR} are then calculated. The T_{MR} is for inter mode and a new T_{MR} is required for intra coding. The bit refinement process is carried out by using computed T_{MR} as an input. Then *bpp* value of each region is obtained.

To get the *QP* of MR, α and β values assigned for the intra frame are used. On the other hand, α and β values assigned for the current frame are still used for computing the *QP* of non-MR. After encoding the entire frame, parameter updating process is carried out for each region separately. Based on the actual generated bits of MR and non-MR, bpp_{real_MR} and bpp_{real_NMR} are computed. λ_{comp_MR} is then computed by using bpp_{real_MR} , α_{old_MR} and β_{old_MR} , as in (9). New parameter values for MR are

then computed by (10) and (11). These values are used for updating α and β values of the intra frame. A similar procedure is used for non-MR and new parameter values are used for updating α and β values of the inter frame that has the same hierarchical level as that of the current frame.

$$\lambda_{comp_MR} = \alpha_{old_MR} \cdot bpp_{real_MR}^{\beta_{old_MR}} \quad (9)$$

$$\alpha_{new_MR} = \alpha_{old_MR} + \delta_{\alpha} \cdot (\ln \lambda_{real_MR} - \ln \lambda_{comp_MR}) \cdot \alpha_{old_MR} \quad (10)$$

$$\beta_{new_MR} = \beta_{old_MR} + \delta_{\beta} \cdot (\ln \lambda_{real_MR} - \ln \lambda_{comp_MR}) \cdot \ln bpp_{real_MR} \quad (11)$$

If MR is encoded with inter mode, a single bpp is computed by using $T_{CurrPic}$. QP is calculated by using α and β values assigned for the current frame. After encoding, the normal parameters updating process is applied by using (3) to (5).

3.4. Summary of the Proposed Error Resilient Algorithm

Algorithm 2 summarizes the proposed algorithm. Based on feedback information and the MR-map, the proposed algorithm combines the hierarchical-P RPS and CTU-level coding mode decision to mitigate network errors. The region-based rate control also enhances the quality of important region while maintaining the desired target bit rate.

Algorithm 2 Proposed Error Resilient Algorithm

```

1: extract MR and create MR-map
2: read feedback signal
3: if ACK signal is detected then
4:    $err\_found = false$ 
5: else
6:    $err\_found = true$ 
7: apply proposed hierarchical-P RPS
8: compute  $T_{MR}$  and  $T_{NMR}$ 
9: for all CU in frame do
10:  if  $CU \in MR$  AND  $err\_found$  AND  $lidst \geq 4$  then
11:    set CU coding mode to intra
12:    apply bit budget refinement for MR
13:  else
14:    set CU coding mode to inter
15:    compute QP value
16:    encode CU
17:    update rate control parameters
18: process next frame

```

4. Experimental Results and Discussion

The proposed algorithm is primarily designed for systems in which feedback information is available; however, it is also possible to apply in non-feedback system with the help of a packet loss estimation model or distortion model at the encoder. Thus, in the experiments, the proposed algorithm is tested under both conditions. All the sequences used in the experiments are standard HEVC test sequences. Three different resolutions are included in the experiments. They are WVGA resolution (832 × 480), 1080p resolution (1920 × 1080) and 720p resolution (1280 × 720).

Since the target application is delay sensitive video transmission over wireless networks such as Wi-Fi network or mobile network, all experiments were carried out by using the “low-delay-P”

configurations of HEVC. The reason is that the “random-access” and the “low-delay-main” configurations require more computation time than the “low-delay-P” configuration. Each row of LCUs is composed as a slice in the experiments. In order to do so, slice mode 1 that is a slice structure based on the number of blocks per slice is used. For 64×64 LCU size, slice argument value is set as 13 for WVGA sequences, 20 for 720p sequences and 30 for 1080p sequences, respectively. If smaller LCU size is chosen, the number of slices per frame is increased. This will increase the number of packets and also the amount of overhead data. But the effect of single packet loss is less severe than 64×64 LCU case because each packet contains fewer amounts of data than 64×64 LCU case. In the experiments, the GOP size is chosen as 4 frames. Changing the GOP size will have impact on the decoder refresh period which should be integer multiple of GOP size to make sure the decoder refresh frame is at the beginning of a GOP. Five different target bit rates for each video resolution, as recommended in Reference [26], are tested in the experiments.

For packet loss simulation, loss simulation software from [27] is hired. Three different packet loss rates, 3%, 5% and 10% are available with the software. Co-located block copying technique was added to the decoder for error concealment. For the video quality measurement, three different metrics are used namely Peak Signal to Noise Ratio for luminance component (Y-PSNR), Structural Similarity Index (SSIM) [28] and Feature Similarity Index (FSIM) [29].

4.1. Experimental Results for Feedback Available Case

In the experiments where feedback is available, the performance of the proposed algorithm is compared with that of the two configurations of the hierarchical-P RPS algorithm, HP_RPS_I and HP_RPS_II. In the HP_RPS_I configuration, only one intra-frame is included at the beginning of the sequence but in the HP_RPS_II configuration, an intra-coded frame was regularly inserted at every second. The strength and weakness of each HP_RPS configuration are also explored. Both hierarchical-P RPS and the proposed algorithm are implemented by using HM15.0. Twelve test sequences are involved in this experiment: three WVGA sequences, six 720p sequences and three 1080p sequences. Firstly, a 3-frame feedback delay under 10% PER case is explored.

In the HP_RPS_I configuration, the temporal distance between the reference frame and the current frame can be very long for some error conditions. This long temporal distance can affect encoder performance. In contrast, the HP_RPS_II configuration can reduce this temporal distance by using the last intra-coded frame as a reference frame. However, adding too many intra frames can also reduce encoder performance due to bit fluctuation. According to the experimental results, the HP_RPS_I configuration performed better than HP_RPS_II for most test cases. This is because HP_RPS_II introduces too many intra-coded blocks that consume a large portion of the available bit budget and as a result, large QP values are used to meet the bitrate constraint. Due to its better performance, HP_RPS_I was selected for comparing to the proposed algorithm.

Generally, for all test sequences, the proposed algorithm achieved better PSNR, SSIM and FSIM values than HP_RPS_I and HP_RPS_II. The detail results of all test sequences for five different target bit rates are shown in Tables 1 and 2. The FSIM metric computes by using luminance values of input videos whereas the FSIMc metric takes into account both luminance and chrominance values.

For 1080p sequences, the average PSNR improvement of the proposed algorithm over the HP_RPS_I configuration of hierarchical-P RPS is 1.56 dB. The minimum improvement is 0.43 dB and the maximum one is 3.24 dB. For 720p sequences, the proposed algorithm achieves an average PSNR improvement about 2.32 dB over HP_RPS_I. For WVGA sequences, the average PSNR improvement of the proposed algorithm is 1.01 dB. Rate-distortion curves of selected sequences are shown in Figure 4.

In terms of average SSIM improvement, the proposed algorithm achieves 5% improvement for 1080p sequences, 2% for 720p sequences and 4% for WVGA sequences, respectively. Rate-SSIM curves of selected sequences are shown in Figure 5.

Table 1. PSNR and SSIM Results Comparisons for 10% PER (feedback available case).

Sequence	Target Bit Rate (kbps)	Average Y-PSNR (dB)			Average SSIM		
		HP_RPS_I	HP_RPS_II	Proposed	HP_RPS_I	HP_RPS_II	Proposed
BasketballDrillText (832 × 480) 50fps	384	26.71	25.45	27.31	0.81	0.77	0.82
	512	27.31	26.07	28.02	0.83	0.79	0.84
	768	28.40	27.12	28.88	0.86	0.83	0.87
	1200	29.47	28.27	29.81	0.88	0.87	0.89
	2000	30.46	29.19	30.74	0.91	0.90	0.92
BQMall (832 × 480) 60fps	384	24.14	24.30	25.76	0.70	0.73	0.77
	512	25.03	25.16	26.44	0.74	0.76	0.80
	768	25.97	25.96	27.31	0.78	0.79	0.83
	1200	26.97	26.76	28.19	0.82	0.82	0.86
	2000	28.12	27.72	28.94	0.86	0.86	0.88
PartyScene (832 × 480) 50fps	384	23.35	23.54	24.90	0.61	0.64	0.70
	512	24.14	24.30	25.63	0.66	0.68	0.74
	768	25.22	25.41	26.57	0.72	0.73	0.78
	1200	26.51	26.64	27.64	0.78	0.79	0.82
	2000	27.96	28.01	28.77	0.83	0.84	0.86
Johnny (1280 × 720) 60fps	256	32.97	32.53	36.57	0.90	0.89	0.93
	384	35.53	35.08	38.01	0.92	0.92	0.94
	512	36.62	36.40	38.80	0.93	0.93	0.95
	850	38.13	37.86	39.87	0.94	0.94	0.96
	1500	39.12	38.93	40.62	0.95	0.95	0.96
FourPeople (1280 × 720) 60fps	256	29.79	28.78	31.96	0.89	0.87	0.90
	384	31.82	30.70	33.85	0.91	0.89	0.92
	512	33.17	32.19	35.04	0.92	0.91	0.94
	850	35.48	34.65	36.91	0.94	0.93	0.95
	1500	37.41	36.39	38.70	0.95	0.95	0.96
KristenAndSara (1280 × 720) 60fps	256	32.07	29.93	34.33	0.91	0.88	0.93
	384	34.14	31.90	35.87	0.92	0.90	0.94
	512	35.25	32.00	36.91	0.93	0.90	0.95
	850	37.11	34.78	38.41	0.95	0.93	0.96
	1500	38.74	37.24	39.61	0.96	0.95	0.96
Vidyo1 (1280 × 720) 60fps	256	28.55	28.16	32.83	0.88	0.87	0.92
	384	30.79	29.61	34.75	0.90	0.89	0.94
	512	32.35	31.74	35.94	0.92	0.91	0.95
	850	35.05	34.69	37.58	0.94	0.94	0.96
	1500	36.82	36.56	38.93	0.95	0.95	0.97
Vidyo3 (1280 × 720) 60fps	256	27.57	27.40	31.29	0.86	0.86	0.92
	384	29.93	28.81	33.16	0.89	0.88	0.94
	512	31.25	30.63	34.26	0.91	0.90	0.95
	850	33.39	33.02	35.75	0.94	0.93	0.96
	1500	35.26	35.09	37.25	0.95	0.95	0.97
Vidyo4 (1280 × 720) 60fps	256	29.39	28.31	32.52	0.88	0.87	0.91
	384	31.81	31.13	34.40	0.90	0.89	0.93
	512	33.08	32.57	35.33	0.91	0.91	0.93
	850	34.97	34.64	36.60	0.93	0.93	0.95
	1500	36.66	36.46	37.89	0.94	0.94	0.96
BasketballDrive (1920 × 1080) 60fps	2000	25.14	24.95	28.38	0.73	0.72	0.81
	3000	26.30	26.33	29.06	0.76	0.76	0.83
	4500	27.76	27.76	29.60	0.79	0.79	0.85
	7000	29.27	29.31	30.06	0.84	0.83	0.87
	10,000	29.95	30.00	30.40	0.86	0.86	0.90
BQTerrace (1920 × 1080) 50fps	2000	25.15	25.11	27.50	0.70	0.70	0.79
	3000	26.27	26.30	28.14	0.74	0.74	0.81
	4500	27.34	27.25	28.60	0.78	0.77	0.82
	7000	28.30	28.30	29.00	0.81	0.81	0.83
	10,000	28.83	28.79	29.26	0.83	0.83	0.84
Cactus (1920 × 1080) 50fps	2000	27.79	27.50	30.20	0.79	0.78	0.83
	3000	28.78	28.53	31.05	0.81	0.80	0.86
	4500	30.27	30.12	31.72	0.84	0.83	0.87
	7000	31.43	31.31	32.37	0.86	0.86	0.89
	10,000	32.06	31.97	32.74	0.88	0.87	0.89

Table 2. FSIM Results Comparisons for 10% PER (feedback available case).

Sequence	Target Bit Rate (kbps)	FSIM			FSIMc		
		HP_RPS_I	HP_RPS_II	Proposed	HP_RPS_I	HP_RPS_II	Proposed
BasketballDrillText (832 × 480) 50fps	384	0.9237	0.9032	0.9254	0.9200	0.8988	0.9224
	512	0.9327	0.9131	0.9362	0.9296	0.9093	0.9336
	768	0.9461	0.9303	0.9481	0.9437	0.9273	0.9459
	1200	0.9576	0.9470	0.9579	0.9557	0.9447	0.9560
	2000	0.9664	0.9593	0.9662	0.9647	0.9572	0.9645
BQMall (832 × 480) 60fps	384	0.8958	0.9041	0.9224	0.8907	0.8992	0.9189
	512	0.9123	0.9158	0.9318	0.9081	0.9116	0.9288
	768	0.9253	0.9272	0.9414	0.9218	0.9237	0.9389
	1200	0.9387	0.9389	0.9500	0.9359	0.9360	0.9478
	2000	0.9527	0.9520	0.9571	0.9506	0.9496	0.9552
PartyScene (832 × 480) 50fps	384	0.8934	0.8961	0.9075	0.8881	0.8909	0.9038
	512	0.9049	0.9060	0.9177	0.9004	0.9015	0.9146
	768	0.9198	0.9207	0.9290	0.9163	0.9172	0.9265
	1200	0.9341	0.9351	0.9386	0.9316	0.9324	0.9366
	2000	0.9482	0.9499	0.9503	0.9464	0.9480	0.9488
Johnny (1280 × 720) 60fps	256	0.9784	0.9741	0.9863	0.9775	0.9733	0.9860
	384	0.9848	0.9825	0.9896	0.9843	0.9819	0.9894
	512	0.9881	0.9864	0.9913	0.9877	0.9860	0.9911
	850	0.9911	0.9897	0.9934	0.9909	0.9895	0.9932
	1500	0.9927	0.9921	0.9949	0.9925	0.9919	0.9948
FourPeople (1280 × 720) 60fps	256	0.9662	0.9626	0.9743	0.9645	0.9607	0.9733
	384	0.9762	0.9716	0.9818	0.9750	0.9703	0.9813
	512	0.9815	0.9775	0.9855	0.9807	0.9766	0.9851
	850	0.9881	0.9853	0.9892	0.9876	0.9848	0.9890
	1500	0.9915	0.9893	0.9928	0.9912	0.9889	0.9926
KristenAndSara (1280 × 720) 60fps	256	0.9692	0.9539	0.9751	0.9685	0.9528	0.9746
	384	0.9783	0.9663	0.9829	0.9778	0.9656	0.9826
	512	0.9826	0.9668	0.9860	0.9822	0.9661	0.9858
	850	0.9881	0.9796	0.9901	0.9879	0.9792	0.9899
	1500	0.9921	0.9884	0.9930	0.9919	0.9882	0.9929
Vidyo1 (1280 × 720) 60fps	256	0.9592	0.9561	0.9786	0.9573	0.9540	0.9780
	384	0.9714	0.9651	0.9849	0.9701	0.9635	0.9845
	512	0.9773	0.9751	0.9873	0.9764	0.9740	0.9870
	850	0.9853	0.9844	0.9905	0.9848	0.9838	0.9903
	1500	0.9886	0.9882	0.9927	0.9883	0.9878	0.9926
Vidyo3 (1280 × 720) 60fps	256	0.9476	0.9458	0.9786	0.9446	0.9427	0.9774
	384	0.9674	0.9600	0.9850	0.9654	0.9576	0.9843
	512	0.9731	0.9701	0.9871	0.9715	0.9684	0.9865
	850	0.9811	0.9802	0.9900	0.9801	0.9791	0.9896
	1500	0.9868	0.9864	0.9925	0.9861	0.9857	0.9922
Vidyo4 (1280 × 720) 60fps	256	0.9602	0.9528	0.9798	0.9582	0.9504	0.9790
	384	0.9734	0.9696	0.9856	0.9721	0.9682	0.9852
	512	0.9786	0.9767	0.9881	0.9776	0.9756	0.9878
	850	0.9853	0.9843	0.9908	0.9847	0.9837	0.9905
	1500	0.9896	0.9893	0.9930	0.9892	0.9889	0.9928
BasketballDrive (1920 × 1080) 60fps	2000	0.8942	0.8889	0.9551	0.8894	0.8839	0.9531
	3000	0.9166	0.9163	0.9628	0.9127	0.9125	0.9609
	4500	0.9384	0.9380	0.9678	0.9355	0.9351	0.9661
	7000	0.9578	0.9577	0.9715	0.9557	0.9557	0.9699
	10,000	0.9674	0.9672	0.9740	0.9655	0.9655	0.9724
BQTerrace (1920 × 1080) 50fps	2000	0.9397	0.9404	0.9681	0.9371	0.9379	0.9668
	3000	0.9540	0.9551	0.9719	0.9520	0.9531	0.9707
	4500	0.9640	0.9634	0.9745	0.9624	0.9618	0.9733
	7000	0.9727	0.9727	0.9771	0.9715	0.9715	0.9761
	10,000	0.9757	0.9759	0.9786	0.9746	0.9748	0.9776
Cactus (1920 × 1080) 50fps	2000	0.9538	0.9510	0.9758	0.9517	0.9487	0.9749
	3000	0.9610	0.9595	0.9801	0.9592	0.9576	0.9793
	4500	0.9710	0.9704	0.9827	0.9697	0.9691	0.9821
	7000	0.9786	0.9781	0.9851	0.9777	0.9772	0.9845
	10,000	0.9821	0.9816	0.9865	0.9813	0.9808	0.9859

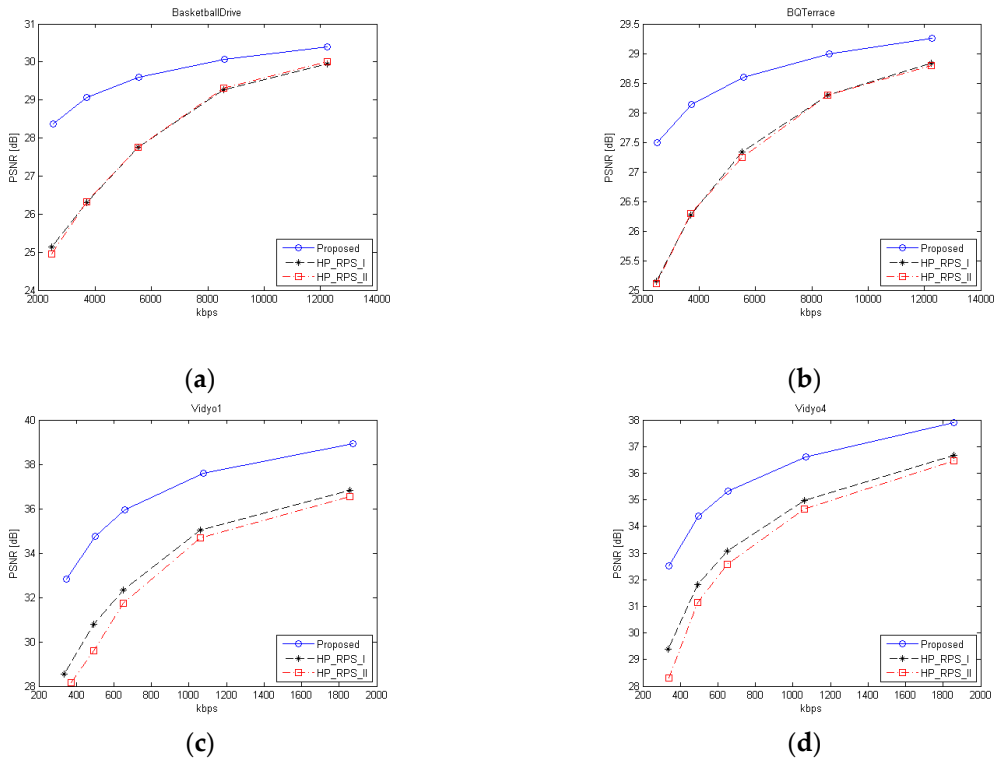


Figure 4. Rate-distortion performance between hierarchical-P RPS and proposed method for 10% PER. (a) BasketballDrive; (b) BQTerrace; (c) Vidyo1; (d) Vidyo4.

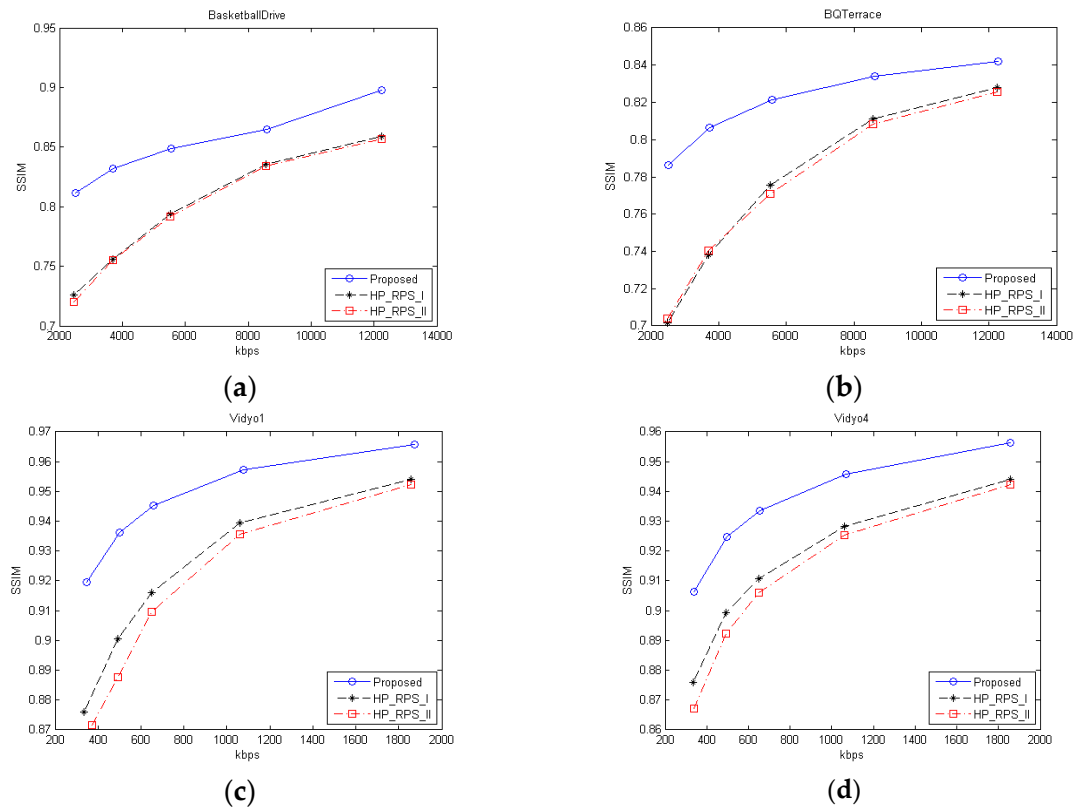


Figure 5. Rate-SSIM performance between hierarchical-P RPS and proposed method for 10% PER. (a) BasketballDrive; (b) BQTerrace; (c) Vidyo1; (d) Vidyo4.

From the view point of the FSIM metric, the average improvement of the proposed algorithm over HP_RPS_I is 1.9% for 1080p sequences. The maximum and minimum values are 2.84% and 0.29%, respectively. For 720p sequences, the maximum improvement is 3.09% whereas the minimum value is 0.09%. The average value for 720p resolution is 0.79% and that for WVGA resolution is 0.8%. The maximum and minimum improvement values for WVGA sequences are 2.66% and -0.02% , respectively. Rate-FSIM curves are described in Figure 6.

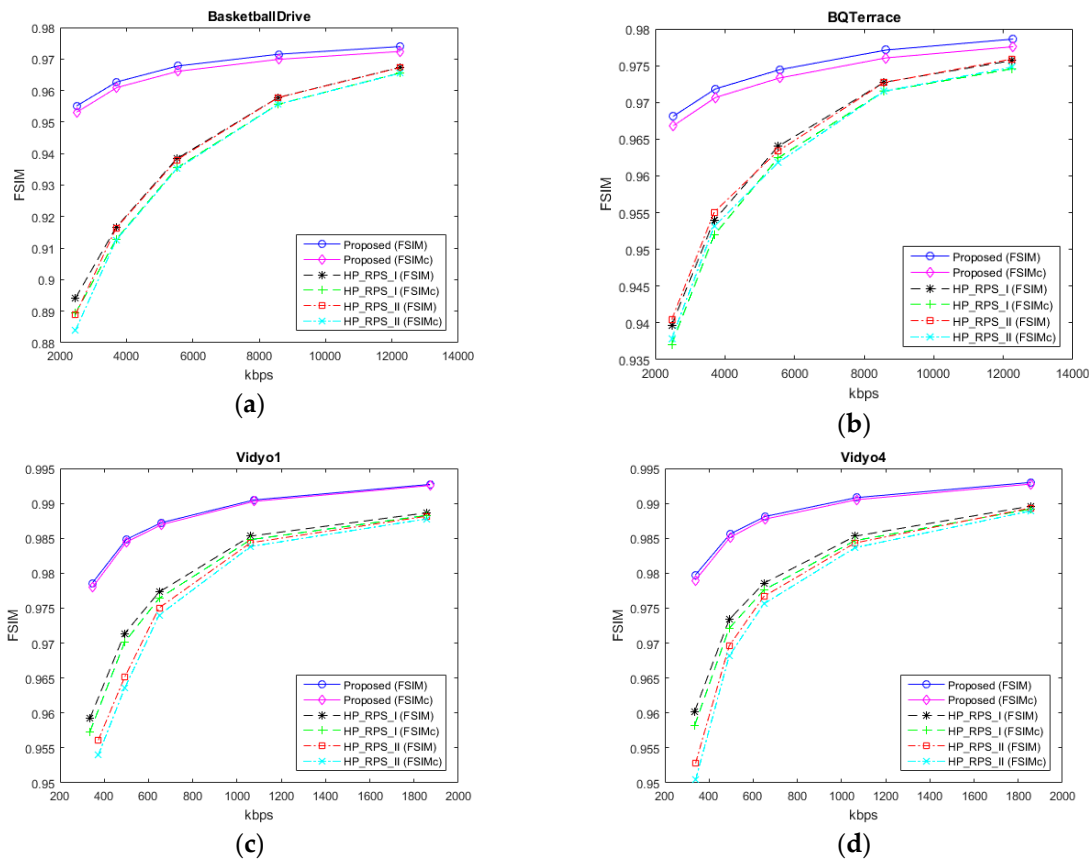


Figure 6. Rate-FSIM performance between hierarchical-P RPS and proposed method for 10% PER. (a) BasketballDrive; (b) BQTerrace; (c) Vidyo1; (d) Vidyo4.

For the BasketballDrive sequence that contains several running players and camera movement, the proposed algorithm achieves up to 3.24 dB PSNR improvement and 9% SSIM improvement over HP_RPS_I. In terms of FSIM, the improvement of the proposed algorithm is considerably higher than that of HP_RPS_I at low target bit rates.

For the BQTerrace sequence that contains several small moving objects, significant changes in background and camera movement, the PSNR improvement of the proposed algorithm is 2.35 dB but SSIM improvement varies from 1% to 9% based on target bit rate. Up to 2.84% FSIM improvement is achieved for this sequence. It should be noted that co-located block copying for error concealment greatly affects the quality of such kind of sequence. For Cactus sequence, the average PSNR and SSIM improvement are 1.55 dB and 3.5% respectively. This sequence has a still background and moving objects with moderate speed in the foreground. Noticeable improvements of PSNR, SSIM and FSIM can be seen especially at the lower bit rates for all 1080p sequences. For Johnny, Vidyo1, Vidyo3 and Vidyo4 sequences, the significant PSNR improvement is achieved for all target bit rates. The major contents in these videos are located around the middle region of the frame. The MR region extraction method used in the proposed algorithm set more weight values to the middle region. Thus, the major contents of the above sequences are well within the important region and as a result, significant improvements are obtained for these sequences. In terms of SSIM, up to 6% improvement is obtained for these

sequences. The maximum FSIM improvement of these sequences is 3.09% and the minimum is 0.21%. For KristenAndSara sequence and FourPeople sequence, PSNR improvement is more noticeable at low bit rates but SSIM improvement is only about 1%. According to SSIM values, the performance of the proposed algorithm and that of HP_RPS_I are about the same for these two sequences since the major contents of these sequences are not only located within the middle region but also located in some areas which are closed to the boundary region and the motions of these contents are very low. Thus, MR region extraction method cannot extract some important regions of that sequences and the encoded quality of that regions when using the proposed algorithm is same as or lower than using HP_RPS_I. Therefore, the average improvement of the proposed algorithm over HP_RPS_I is small. The average FSIM improvement is about 0.37%. If the bit rate is increased, the performance of HP_RPS_I is closed to that of proposed system. For PartyScene and BQMall sequences, above 1 dB improvement in PSNR is achieved for all bit rates except the highest one. Up to 5% SSIM improvement in BQMall sequence and 9% SSIM improvement for PartyScene sequence are obtained. In terms of FSIM, the average improvement of BQMall is about 1.56% whereas that of PartyScene is 0.86% respectively. These sequences include moving persons with moderate speed. It can be seen that the proposed algorithm performs quite well for that sort of sequences.

For the BasketballDrillText sequence, which includes several running players and camera movements, both average PSNR improvement and average SSIM improvement are small. Only less than 1 dB improvement in PSNR and about 1% SSIM improvement are achieved for all bit rates. Average FSIM improvement is only 0.15% for this sequence. The MR region extraction method used the same block size as LCU for its computation. Since the BasketballDrillText sequence contains a lot of motions, the size of the MR region is quite large. The proposed algorithm inserts more intra coded blocks than other sequences. Consequently, inter coded blocks must use the highest QP value to meet the target bit rate, hence the overall performance is small for that sequence. Selected frames of the BasketballDrive sequence and the Vidyoy1 sequence are shown in Figures 7 and 8, respectively.



Figure 7. Frame 150 of BasketballDrive sequence under 10% PER. (a) HP_RPS_I; (b) Proposed Method.

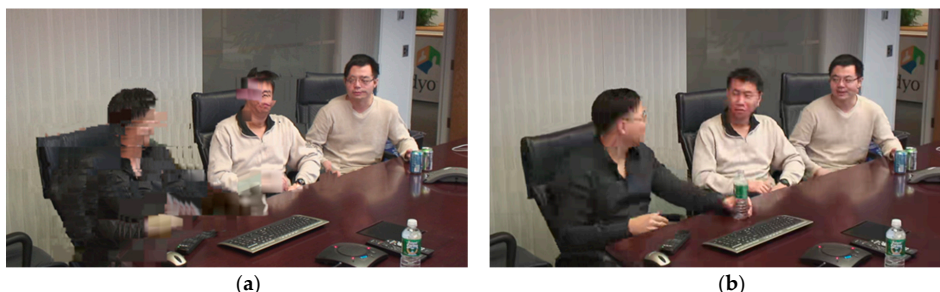


Figure 8. Frame 350 of Vidyoy1 sequence under 10% PER. (a) HP_RPS_I; (b) Proposed Method.

For other packet error rates, the average PSNR improvement was 0.72 dB for WVGA sequences and 1.35 dB for 720p sequences under 5% PER and 0.73 dB for WVGA sequences and 1.33 dB for 720p sequences under 3% PER, respectively. According to the experimental results, the proposed algorithm outperforms the HP_RPS_I for lower packet error rates.

4.2. Impact of Feedback Delay

Moreover, the impact of feedback delay is also explored by conducting additional experiments. For this experiment, six test sequences are used. These sequences include two 1080p sequences, two 720p sequences and two WVGA sequences. All sequences are encoded with three target bit rates by using same configuration as in the first experiment. Feedback delays used in this experiment are 3-frame delay, 4-frame delay and 5-frame delay. The detail results are presented in Table 3. According to the results, it can be seen that the longer the delay, the higher the quality degradation. When the delay is long, it is possible that all available reference pictures in the buffer are encoded using the prediction from the error frame. For that case, the coding mode for current frame must select intra mode because there is no reliable reference picture in the buffer. Frequent insertion of I-frame occurs in long feedback delay case. These intra frame cause bit fluctuation and quality drop. In the experiment, 5-frame delay of the proposed algorithm can produce less than 1 dB improvement in average PSNR for BasketballDrive, BQTerrance, BQMall and PartyScene sequences. For Vidy4 and FourPeople sequences, the performance of the proposed algorithm is lower than that of HP_RPS_I especially at low bit rates. This is the effect of too much intra coded frames as discussed above.

Table 3. PSNR Comparison under Different Feedback Delays (10% PER).

Sequence	Target Bit Rate (kbps)	3-frame Delay		4-frame Delay		5-frame Delay	
		HP_RPS_I	Proposed	HP_RPS_I	Proposed	HP_RPS_I	Proposed
BQMall (832 × 480) 60fps	384	24.14	25.76	24.01	25.46	23.36	24.31
	768	25.97	27.31	25.60	26.88	25.18	25.87
	2000	28.12	28.94	27.57	28.32	26.93	27.41
PartyScene (832 × 480) 50fps	384	23.35	24.90	23.44	24.82	21.96	22.56
	768	25.22	26.57	25.18	26.34	24.03	24.65
	2000	27.96	28.77	27.66	28.43	26.97	27.43
FourPeople (1280 × 720) 60fps	256	29.79	31.96	29.99	32.17	30.03	27.39
	512	33.17	35.04	33.24	35.14	30.39	30.03
	1500	37.41	38.70	37.07	38.36	33.62	33.89
Vidy4 (1280 × 720) 60fps	256	29.39	32.52	29.61	33.07	31.05	29.80
	512	33.08	35.33	33.21	35.45	31.32	32.42
	1500	36.66	37.89	36.49	37.65	35.05	35.55
BasketballDrive (1920 × 1080) 60fps	2000	25.14	28.38	24.88	27.63	26.34	27.05
	4500	27.76	29.60	27.21	28.78	27.28	27.96
	10,000	29.95	30.40	28.98	29.41	28.32	28.44
BQTerrance (1920 × 1080) 50fps	2000	25.15	27.50	24.96	26.65	25.72	26.19
	4500	27.34	28.60	26.93	27.96	26.88	27.03
	10,000	28.83	29.26	28.32	28.63	28.01	28.05

4.3. Experimental Results for No Feedback Case

In the experiments where feedback is not available, the checkerboard pattern RPS (chkRPS) method is used as the reference method. Since the proposed method requires packet loss information, a simple packet loss estimation model called Gilbert Elliot model is used in this experiment. Only 10% PER case is considered in this work. The estimated packet error information used in encoding process has same packet error rate with the packet error trace file of network abstraction layer (NAL) unit loss software but the packet error locations of these two files are not identical. Nine test sequences (i.e., three WVGA sequences, three 720p sequences and three 1080p sequences) are encoded by using both proposed method and chkRPS method. Each sequence is encoded for five different target bitrates. Regular intra frame is added every one second for both methods. PSNR and FSIM metrics are used for performance evaluation.

According to the experimental results, proposed method shows higher performance than chkRPS for six test sequences out of nine sequences. For the remaining three sequences, chkRPS is better than

the proposed algorithm. These three sequences are BasketballDrillText sequence, BasketballDrive sequence and Cactus sequence.

For BasketballDrillText sequence, chkRPS obtains an average PSNR improvement of 0.45dB over proposed method. In FSIM metric as shown in Figure 9f, the two lowest target bit rates still better than chkRPS about 0.25%. Similarly, for the BasketballDrive sequence, the two lowest target bit rates of the proposed algorithm are higher in FSIM than chkRPS about 0.9% as presented in Figure 9e. It is noted that the proposed algorithm is not suitable for sequences that contain fast moving objects if the estimated error location is not accurate enough. For Cactus sequence, chkRPS achieves about 0.6dB improvement over the proposed algorithm. This sequence has still background but rotating foreground objects. Similar to the above two sequences, if the contents of the video scene are moving with considerably fast pace and the error location estimation is not accurate, the performance of the proposed algorithm is not good. Again, in the Cactus sequence, the FSIM value of the lowest target bit rate is higher than that of chkRPS. Although the results of these three sequences are not good, it is still demonstrated that the proposed algorithm is more suitable for bit rate constraint applications.

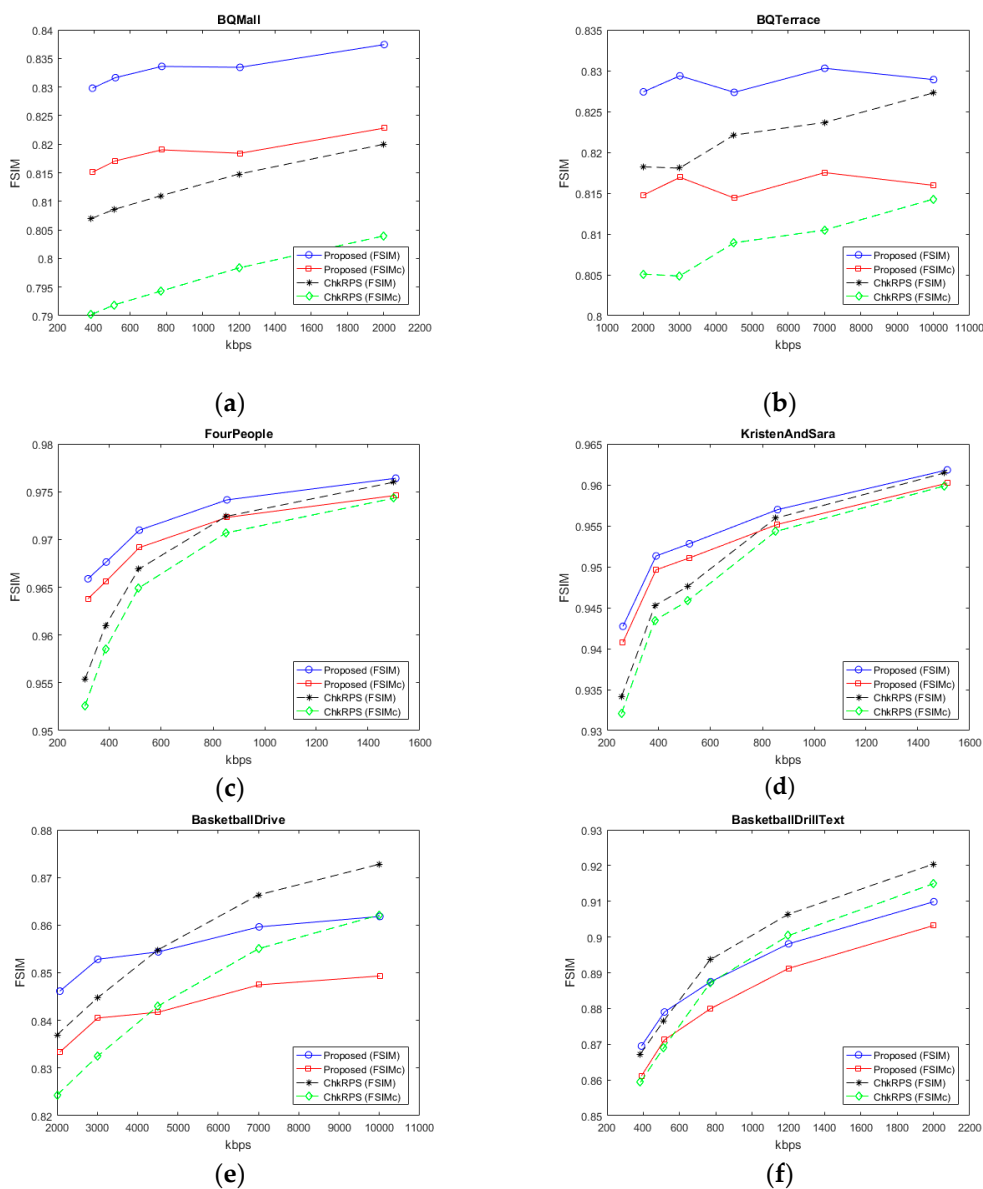


Figure 9. Rate-FSIM performance between checkerboard pattern RPS and proposed method for 10% PER. (a) BQMall; (b) BQTerrace; (c) FourPeople; (d) KristenAndSara; (e) BasketballDrive; (f) BasketballDrillText.

For three 720p sequences, the proposed method outperforms the chkRPS method. The average PSNR improvement is about 0.62dB. With FSIM metric, the average improvement over chkRPS is 0.47%. The maximum value 1.05% and the minimum one is 0.04%. The lower the target bit rate, the larger the improvement value. The contents of these three videos are located mainly in the central region and only facial expressions and hand gestures are involved. The same background is appeared throughout the scene. Such kind of sequences is the most suitable one for the proposed algorithm. For BQMall and PartyScene sequences, the performance of proposed system is significantly better than that of chkRPS as shown in Figure 9a,b. For these two sequences, the minimum PSNR improvement is 0.28 dB and the maximum is 1.32 dB. In terms of FSIM, the minimum improvement value is 0.57% and the maximum is 2.3%. The BQMall sequence includes walking people with moderate speed and the camera movement whereas the PartyScene sequence includes still background with moving foreground objects. For BQTerrace sequence that contains slow camera movement and scene changes, proposed method is slightly better than chkRPS especially for low target bit rate. Table 4 summarizes the FSIM results of the proposed method and the chkRPS method for all test sequences in the experiment. According to the results, the proposed method is mainly suitable for sequences that contains moderate or few motions. For sequences that contain fast moving objects, chkRPS is more appropriate than the proposed method.

Table 4. FSIM results for 10% PER (feedback not available case).

Sequence	Target Bit Rate (kbps)	FSIM		FSIMc	
		chkRPS	Proposed	chkRPS	Proposed
BasketballDrillText (832 × 480) 50fps	384	0.8673	0.8695	0.8593	0.8611
	512	0.8765	0.8790	0.8690	0.8712
	768	0.8936	0.8875	0.8871	0.8800
	1200	0.9064	0.8981	0.9004	0.8912
	2000	0.9204	0.9099	0.9149	0.9032
BQMall (832 × 480) 60fps	384	0.8070	0.8298	0.7902	0.8151
	512	0.8086	0.8316	0.7919	0.8171
	768	0.8110	0.8336	0.7943	0.8190
	1200	0.8148	0.8334	0.7984	0.8184
	2000	0.8200	0.8374	0.8040	0.8228
PartyScene (832 × 480) 50fps	384	0.8504	0.8664	0.8401	0.8576
	512	0.8541	0.8668	0.8440	0.8578
	768	0.8591	0.8707	0.8491	0.8616
	1200	0.8679	0.8735	0.8582	0.8642
	2000	0.8743	0.8806	0.8648	0.8715
Johnny (1280 × 720) 60fps	256	0.9612	0.9685	0.9594	0.9672
	384	0.9664	0.9730	0.9648	0.9719
	512	0.9679	0.9740	0.9662	0.9728
	850	0.9729	0.9754	0.9714	0.9742
	1500	0.9714	0.9751	0.9698	0.9738
FourPeople (1280 × 720) 60fps	256	0.9554	0.9659	0.9526	0.9638
	384	0.9610	0.9676	0.9586	0.9656
	512	0.9669	0.9710	0.9649	0.9692
	850	0.9724	0.9741	0.9707	0.9723
	1500	0.9760	0.9764	0.9743	0.9746
KristenAndSara (1280 × 720) 60fps	256	0.9342	0.9427	0.9321	0.9408
	384	0.9452	0.9513	0.9434	0.9496
	512	0.9476	0.9528	0.9459	0.9511
	850	0.9559	0.9570	0.9543	0.9552
	1500	0.9614	0.9618	0.9599	0.9602

Table 4. Cont.

Sequence	Target Bit Rate (kbps)	FSIM		FSIMc	
		chkRPS	Proposed	chkRPS	Proposed
BasketballDrive (1920 × 1080) 60fps	2000	0.8370	0.8461	0.8243	0.8334
	3000	0.8447	0.8528	0.8324	0.8405
	4500	0.8548	0.8544	0.8430	0.8417
	7000	0.8664	0.8596	0.8550	0.8475
	10,000	0.8728	0.8618	0.8621	0.8493
BQTerrace (1920 × 1080) 50fps	2000	0.8182	0.8274	0.8051	0.8148
	3000	0.8181	0.8294	0.8049	0.8169
	4500	0.8221	0.8274	0.8089	0.8144
	7000	0.8237	0.8303	0.8105	0.8175
	10,000	0.8273	0.8289	0.8143	0.8160
Cactus (1920 × 1080) 50fps	2000	0.9168	0.9177	0.9118	0.9122
	3000	0.9205	0.9206	0.9155	0.9153
	4500	0.9278	0.9219	0.9232	0.9165
	7000	0.9322	0.9235	0.9278	0.9180
	10,000	0.9182	0.9237	0.9318	0.9182

5. Conclusions

In this paper, a feedback-based error resilient framework for HEVC video transmission is presented. The noticeable performance of the proposed algorithm is achieved for both high resolution videos that contain significant motion and sequences that contain foreground objects in central region. When feedback is available, the proposed algorithm achieved 1.9% average FSIM improvement for 1080p sequences, 3.09% for 720p sequences and 0.8% for WVGA sequences, respectively. More than 1dB average PSNR improvement is achieved for all resolutions. In terms of SSIM, 2%, 4% and 5% average improvement are obtained for WVGA, 720p and 1080p resolutions, respectively. The accuracy and size of MR is one of the main factors for the performance of the proposed algorithm. All three quality metrics used in this paper demonstrate that the proposed algorithm outperforms the HP_RPS method for all test sequences under 10% PER. Similar results are obtained for lower packet error rates such as 5% PER and 3% PER. Feedback delay is another important factor that can affect the quality of decoded video. The impact of feedback delay is also explored in the experiment. For 3-frame and 4-frame delay cases, more than 1 dB average PSNR improvement is achieved. But for 5-frame delay, about 0.3 dB improvement is obtained. It is noticed that the feedback delay higher than 5-frame delay is not suitable for the proposed algorithm.

The proposed algorithm is also tested for system with no feedback channel. Better performance is obtained for sequences that include moderate level or low level motion objects. For the sequences with fast motion objects, the performance of the proposed algorithm is not as good as reference method. This is due to the accuracy of packet loss estimation model. If the encoder hires more advanced distortion model or packet error estimation model, the performance can be improved.

The experimental results from both feedback available case and feedback not available case showed that the proposed method is mainly suitable for low bit rate applications. Further improvement in perceived quality can be achieved if the decoder uses the advanced error concealment technique than the simple co-located block copying technique that employs in this paper.

Author Contributions: Conceptualization, investigation, software, writing—original draft preparation, H.M.M.; supervision, writing—review and editing, S.A. and Y.M.

Funding: This research has been supported in parts by the Collaborative Research Project entitled Video Processing and Transmission, JICA Project for AUN/SEED-Net, Japan and the Ministry of Internal Affairs and Communications for SCOPE Program (185001003).

Acknowledgments: The authors would like to express their appreciation to João Carreira for allowing to use the checkerboard pattern RPS encoder.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [[CrossRef](#)]
2. Piñol, P.; Torres, A.; López, O.; Martínez, M.; Malumbres, M.P. Evaluating HEVC video delivery in VANET scenarios. In Proceedings of the 2013 IFIP Wireless Days (WD 2013), Valencia, Spain, 13–15 November 2013.
3. Aabed, M.A.; AlRegib, G. No-reference quality assessment of HEVC videos in loss-prone networks. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014), Florence, Italy, 4–9 May 2014.
4. Nightingale, J.; Wang, Q.; Grecos, C.; Goma, S. The impact of network impairment on quality of experience (QoE) in H. 265/HEVC video streaming. *IEEE Trans. Consum. Electron.* **2014**, *60*, 242–250. [[CrossRef](#)]
5. Ferroukhi, M.; Ouahabi, A.; Attari, M.; Habchi, Y.; Taleb-Ahmed, A. Medical Video Coding Based on 2nd-Generation Wavelets: Performance Evaluation. *Electronics* **2019**, *8*, 88. [[CrossRef](#)]
6. Wiegand, T.; Sullivan, G.; Bjøntegaard, G.; Luthra, A. Overview of the H. 264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 560–576. [[CrossRef](#)]
7. Girod, B.; Farber, N. Feedback-based error control for mobile video transmission. *Proc. IEEE.* **1999**, *87*, 1707–1723. [[CrossRef](#)]
8. Fukunaga, S.; Nakai, T.; Inoue, H. Error resilient video coding by dynamic replacing of reference pictures. In Proceedings of the 1996 IEEE Global Telecommunications Conference (GLOBECOM'96), London, UK, 18–28 November 1996.
9. Bjøntegaard, G. *An Error Resilience Method Based on Back Channel Signaling and FEC*; ITU-T/SG15/LBC-96-033; Telenor R&D: San Jose, CA, USA, 1996.
10. Tu, W.; Steinbach, E. Proxy-based reference picture selection for error resilient conversational video in mobile networks. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 151–164. [[CrossRef](#)]
11. Liu, C.; Wang, Y.K.; Hannuksela, M.M.; Chen, Y.; Sujeet, M.; Gabbouj, M. RTP/AVPF compliant feedback for error resilient video coding in conversational applications. In Proceedings of the 9th International Symposium on Communications and Information Technology (ISCIT 2009), Incheon, Korea, 28–30 September 2009.
12. Schierl, T.; Hannuksela, M.M.; Wang, Y.K.; Wenger, S. System layer integration of high efficiency video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1871–1884. [[CrossRef](#)]
13. Sjöberg, R.; Chen, Y.; Fujibayashi, A.; Hannuksela, M.M.; Samuelsson, J.; Tan, T.K.; Wang, Y.K.; Wenger, S. Overview of HEVC high-level syntax and reference picture management. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1858–1870. [[CrossRef](#)]
14. Hong, D.; Horowitz, M.; Eleftheriadis, A.; Wiegand, T. H.264 hierarchical P coding in the context of ultra-low delay, low complexity applications. In Proceedings of the 28th Picture Coding Symposium (PCS 2010), Nagoya, Japan, 7–10 December 2010.
15. Maung, H.M.; Aramvith, S.; Miyanaga, Y. Region-of-interest based error resilient method for HEVC video transmission. In Proceedings of the 15th International Symposium on Communications and Information Technologies (ISCIT 2015), Nara, Japan, 7–9 October 2015.
16. Maung, H.M.; Aramvith, S.; Miyanaga, Y. Improve region-of-interest based rate control for error resilient HEVC framework. In Proceedings of the 2016 International Conference on Digital Signal Processing (DSP), Beijing, China, 16–18 October 2016.
17. Maung, H.M.; Aramvith, S.; Miyanaga, Y. Error resilience aware rate control and mode selection for HEVC video transmission. In Proceedings of the IEEE International Conference on Consumer Electronics (ICCE 2017), Las Vegas, NV, USA, 8–10 January 2017.
18. Carreira, J.; Assunção, P.; Faria, S.; Ekmekcioglu, E.; Kondo, A.; Lim, H. Reference picture selection using checkerboard pattern for resilient video coding. In Proceedings of the IEEE Visual Communications and Image Processing (VCIP), Singapore, 13–16 December 2015.
19. Haßlinger, G.; Hohlfeld, O. The Gilbert-Elliott model for packet loss in real time services on the Internet. In Proceedings of the 14th GI/ITG Conference on Measurement, Modelling and Evaluation of Computer and Communication Systems (MMB 2008), Dortmund, Germany, 31 March–2 April 2008.

20. Li, B.; Li, H.; Li, L.; Zhang, J. Rate Control by R-lambda Model for SHVC. Document JCTVC-M0037, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC). 2013. Available online: http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=7288 (accessed on 2 January 2019).
21. Li, B.; Li, H.; Li, L.; Zhang, J. λ Domain Rate Control Algorithm for High Efficiency Video Coding. *IEEE Trans. Image Process.* **2014**, *23*, 3841–3854. [[CrossRef](#)] [[PubMed](#)]
22. McCann, K.; Bross, B.; Han, W.J.; Kim, I.K.; Sugimoto, K.; Sullivan, G.J. High Efficiency Video Coding (HEVC) Test Model 15 (HM 15) Encoder Description. Document JCTVC-Q1002. 2014. Available online: http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=9103 (accessed on 18 September 2018).
23. Bossen, F. Common Test Conditions and Software Reference Configurations. Document Rec. JCTVC-J1100. Stockholm, Sweden, 2012. Available online: http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=6469 (accessed on 2 January 2019).
24. Ren, G.; Li, P.; Wang, G. A novel hybrid coarse-to-fine digital image stabilization algorithm. *Inform. Technol. J.* **2010**, *9*, 1390–1396. [[CrossRef](#)]
25. Hu, H.M.; Li, B.; Lin, W.; Li, W.; Sun, M.T. Region-based rate control for H. 264/AVC for low bit-rate applications. *IEEE Trans. Circuits Syst. Video Technol* **2012**, *22*, 1564–1576. [[CrossRef](#)]
26. Joint Call for Proposals on Video Compression Technology. ITU-T SG16/Q6 document VCEG-AM91 and ISO/IEC MPEG Document N11113, ITU-T and ISO/IEC JTC 1. 2010. Available online: https://www.itu.int/wftp3/av-arch/jctvc-site/2010_04_A_Dresden/JCTVC-A114.doc (accessed on 2 January 2019).
27. Wenger, S. Nal Unit Loss Software. Document JCTVC-H0072, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC). 2012. Available online: http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=4373 (accessed on 25 December 2018).
28. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
29. Zhang, L.; Zhang, L.; Mou, X.; Zhang, D. FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Trans. Image Process.* **2011**, *20*, 2378–2386. [[CrossRef](#)] [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).