Multimedia Tools and Applications (2021) 80:27385–27405
https://doi.org/10.1007/s11042-021-11005-9

Refinement of the recovered motion vectors for error concealment in HEVC

Mohammad Kazemi [1]

Received: 10 August 2020 / Revised: 7 January 2021 / Accepted: 4 May 2021 /
Published online: 18 May 2021

Abstract
For error concealment of the relatively large corrupted areas in High Efficiency Video Coding (HEVC), the available spatial information is far from the corrupted region and cannot be directly exploited for error concealment. In this paper, a method is proposed to use the spatial information in a new manner to refine the already recovered Motion Vectors (MVs). The refinement method works based on boundary matching and adaptively selection among three approaches for fine tuning of the temporal MVs. The experiments show that the refinement leads to a significant improvement, 2–7 dB in PSNR, for some frames, and the highest MS-SSIM against the state of the art methods. Another important feature of the proposed method is its generality; which can be added on top of other MV recovery methods.

Keywords Video transmission · Error/loss concealment · Motion vector recovery · HEVC

1 Introduction

Nowadays, real-time digital video transmission over packet networks is very popular. Due to the tremendous volume of the raw video data, the compression is inevitable. Delivering compressed data over the wired/wireless channels is challenging since the underlying networks are not always reliable and some data loss during transmission is usually experienced. The compressed video is too sensitive to data loss so that a bit error may cause a significant corruption. The more compression on the data, the more sensitivity to data loss; for this reason, High Efficiency Video Coding (HEVC) is less error resilient than H.264/AVC [1].

There is no emphasis on the type of the network in this paper. In both wired and wireless networks, the networking hardware might drop the packets due to the congestion. Because of several reasons, data loss might occur in the physical medium, too. A packet might be

✉ Mohammad Kazemi
m.kazemi@eng.ui.ac.ir

[1] Department of Electrical Engineering, University of Isfahan, Isfahan 8174673441, Iran

corrupted partially, but due to the predictive and variable length coding, the video packet might be completely unusable. If there was a back-channel for informing the missing packets, the problem is solved in some cases or the actual loss rate becomes very small. But in the case of long round-trip time, the retransmitted packets might arrive too late and not usable for live videos. Furthermore, in multicast applications, the sender is not able to respond the retransmission requests of all clients timely. Therefore, packet loss is inevitable in many video communication systems, wired or wireless, with or without back channel. With channel coding, some bit errors can be corrected [5, 43], but the losses beyond the correction capability of the channel codes cannot be resolved.

In live video communication, the lost packets must be concealed [12, 22, 39, 41, 44]. Using error concealment techniques, the lost data are replaced with some temporally or spatially adjacent data. Note that error concealment is not a standardized part of the HEVC video codec. Actually, HEVC and the other video coding standards have no definition/restriction for error concealment. The companies may embed one or more appropriate error concealment algorithms into their HW/SW codec solutions. Error propagation is also inevitable if the lost data are not concealed correctly; since the error concealed frame could be used as the inter prediction reference of the other frames. For this reason, fully intra coding of frames might be an appropriate solution for HEVC video transmission over lossy channels [19, 20]; since it mitigates the error propagation problem. However, error concealment is still required.

Error concealment techniques can be divided into spatial and temporal domain techniques. In the spatial domain, the lost area of the frame is concealed using the spatially neighboring pixels. These methods exploit the correlations among the spatially neighboring pixels and known as Spatial Error Concealment (SEC) techniques. In the temporal processing techniques, the available contents from the previous and/or future frames are used for loss concealment; these techniques are known as Temporal Error Concealment (TEC). Spatio-Temporal Error Concealment (STEC) is a hybrid of the above two approaches where both spatial and temporal pixels are used for filling the lost area.

The Coding Tree Unit (CTU) size in HEVC can be as large as 64 × 64 pixels. An integer number of CTUs are regarded as one slice and an integer number of slices are encapsulated into a single transmission packet. Therefore, a lost packet in HEVC streams affects several CTUs which can be a significant amount of picture area. Because of the relatively large loss areas, the spatial available pixels are very far from the lost pixels; and except for static and uniform scenes, the middle part of the loss area has a little correlation with the spatial neighboring blocks. Therefore, the SEC techniques are not suitable for HEVC losses. The multi-directional interpolation of [6] and the sparsity based method presented in [30] are only applicable and examined for 16 × 16 pixel blocks. The method presented in [3] works based on rearranging the wavelet coefficients into independent partitions at the encoder side. This approach is not applicable in DCT based codecs such as HEVC. The different techniques such as [29, 51] which try find a model to map and filter the reference frame pixels to the lost area are not successful for the large loss areas. Therefore, a promising approach is to use TEC techniques for concealment of loss areas in HEVC [22]. These techniques which are reviewed in Section 2 try to find Motion Vector (MV) for the missing blocks of the loss area and fill them with motion compensated replacement from the previous frame. The MVs can be obtained with a variety of methods, since both spatial and temporal information can be exploited for MV recovery.

In H.264/AVC, the loss areas are not too large and spatial information can be used for MV recovery; but it is not the case in HEVC. We can say that the existing approaches have not any idea to effectively use the spatial pixels for error concealment in HEVC; while the surrounding available pixels are valuable information. This is in fact the main contribution of the current paper; efficiently exploiting the boundary pixels of the missing area with the aim of higher loss concealment quality. As already mentioned and will be shown experimentally, boundary pixel information is not appropriate for MV recovery, but it can be used efficiently for refinement of the already recovered MVs. Three methods for refinement are proposed, and based on the prior frame behavior with applying each refinement method, one or neither of them is finally applied for fine tuning of the MVs. As the experiments show, the proposed method gives generally higher quality compared to other methods. Even though the improvement is not significant for frames of the test sequences, there are some frames which are concealed with 2-7 dB higher PSNR with the adaptive refinement. This point is not well reflected in the average objective scores, but it is important to the subjects viewing the video carefully. Since the additional complexity of the adaptive method is still below the processing power of the devices used for live video communication, the proposed method is worth implementing even though the improvement is not significant for all frames.

The paper organization is as follows. The related works and their differences with the proposed method are provided in Section 2. The proposed refinement method is explained and discussed in Section 3, and the performance comparison with the other error concealment techniques are provided in Section 4. Finally, the paper is concluded in Section 5.

2 Related work

In viewpoint of MV recovery, TEC techniques can be categorized into four major groups:

2.1 TEC with no MV recovery

A homography-based registration and using it for error concealment is presented in [11]. The available pixels around the corrupted area are registered to their counterparts in the reference frame, and then the registration is used for mapping the matching points into the loss area. This algorithm is suitable for small loss areas where the sufficient correlation between the lost and available pixels needing for registration exists. An error concealment with Principle Component Analysis (PCA) is proposed in [10] where the information of a trained PCA is used. The image vector is mapped to the constructing Eigen vectors which are used for corrupted image reconstruction. In the case of scene changes, the information must be updated. However, similar to [11], the success of this method depends on the correlation of the lost area with the available surrounding pixels which is only the case for the blocks close to the available data. Therefore, for many blocks in HEVC, this algorithm is not applicable.

There are some methods which use Deep Neural Networks (DNNs) for error concealment [34]–[45]. In [34], a combination of convolutional Long Short Term Memory (LSTM) layers and a simple convolutional layer is used for predicting the optical flow of the lost area. A two-stream version of this method is presented in [36] where extracting the horizontal and vertical motion fields are extracted by separate networks. However, working based only on the optical flow is not reliable. For example when the pixels with close values and close locations have different motions, optical flow does not work properly. In another method, a capsule of three frames is used as the input of the network and the fourth frame of the video sequence is applied as the label [37]. After training the network, output of the network can be used for error concealment for the fourth frame if it is corrupted. Generative Adversarial Network (GAN) can be trained for video error concealment [45]. In this work, the temporally and spatially surrounding pixels of the loss area are the inputs, and the trained generator is used for recovery of the lost area. Generally, DNN based solutions are time consuming even with state of the art GPUs; therefore these algorithms are hardly usable for on the fly error concealment.

2.2 TEC using the spatially available neighboring MVs (TEC-SMV)

These methods exploit spatially neighboring MVs for MV recovery. In Macro-Block (MB) based codecs, one method is to use the average or median of motion vectors of surrounding available blocks of the lost MB. The spatial MVs can be averaged for each pixel of the lost MB which is then used for pixel-wise MV recovery. The averaging weights are inversely proportional to distance of each pixel of the lost block with the surrounding available pixels, with Euclidean distance measure in [28] and Mahalanobis distance in [2]. In another work presented in [24] a plane is fitted to the MVs (x and y components separately) of the available blocks around the lost MB. By interpolating in this plane, an MV for the missing MB is recovered.

With the method proposed in [31], for each 8 × 8 lost block, the previous frame is checked to find the most correlated MV of the co-located block with its neighbours. Based on the MVs behaviour in the past frames, the motion trajectory in the few past frames is used for MV recovery of the 8 × 8 block.

Some neighbouring MVs have stronger correlations to the MV of the lost MB than the others, and hence they should have stronger impact in MV recovery, as presented in [17]. In this paper, with the processing of MVs in co-location of correctly available MB in the previous frame, a tendency of the MV in each 4 × 4 block to its spatial MVs is estimated. Then an interpolation between the two MVs to which the block has the highest tendency is used as the recovered MV for the 4 × 4 block. MV recovery for 4 × 4 blocks inside the MBs was also presented in [49]. In this method, one adjacent MV is selected for 12 outer 4 × 4 blocks, and a modified boundary matching for four 4 × 4 inner blocks.

The middle blocks of the losses in HEVC coded frame is not close enough to the available blocks; therefore, the above algorithms are not suitable.

2.3 TEC using the temporally available neighboring MVs (TEC-TMV)

In these methods temporally available neighboring MVs are used for error concealment. The simple yet an efficient method is to use MVs of the co-located block for motion compensated replacement which is known as Motion Copy, abbreviated to MV-C for addressing in this paper. For the intra coded co-located blocks, Zero MV is used in MV-C. Averaging of MVs of the co-located blocks in the prior and next frames is presented in [8], this approach leads to higher PSNR but imposes one frame delay. By a recursive approach, the MV-C can be refined as presented in [9] where the differences of MVs in the successive frames are used to refine the MVs.

Another method is Motion Vector Extrapolation (MVE), where the previous frame block MVs are projected into the current frame blocks, the MV whose pointed block has the maximum overlapping area with the current block is selected as the recovered MV for this block [33]. By using MVs of a few past frames, an approximation of the optical flow is estimated and it is then used for pixel based error concealment [4]. It is actually similar to pixel-wise MVE but with half-pixel accuracy. Based on the work presented in [47], for each pixel, a set of extrapolated MVs are collected and their mean value is used for motion compensated pixel replacement. Tuning the extrapolated MVs for each 4 × 4 pixel blocks with spatially neighboring MVs are presented in [26, 50]. These algorithms exploit both TEC-TMV and TEC-SMV approaches.

2.4 TEC with decoder side MV estimation (TEC-DMVE)

In this group of loss concealment techniques, the MVs of the spatially and temporally adjacent blocks are not used; and instead the MV is estimated at the decoder similar to the Motion Estimation (ME) carried out at the encoder.

With boundary matching, one or more bands of the surrounding pixels of the lost MB are used as the pixels to which we are looking for the best matched pixels surrounding the candidate blocks. These pixels can be at the outer boundary or at the inner boundary of candidate blocks. In the literature, matching of the outer boundary is known as Outer Boundary Matching Algorithm (OBMA) or DMVE and for referring to inner boundary matching, Boundary Matching Algorithm (BMA) is usually used. The number of lines of pixels in OBMA could be as large as eight lines while for BMA, one line is used for matching.

A combination of OBMA (DMVE) and BMA is presented in [18] for H.264/AVC in Switching and Blending modes. Using the partitioning of the co-located CTU and finding the MV of the Prediction Units (PUs) with BMA was presented in [32]. However, this method is applicable where only one CTU is lost. Motion estimation at the decoder with the aim of object shape preservation is introduced in [48]. It is applicable for small enough objects.

Some methods are specifically proposed for error concealment of a sequence of successive MBs in H.264/AVC [35]–[25]. These methods might be more useful for HEVC, since in HEVC with a lost packet, at least one missing CTU is equivalent to loss of sixteen MBs. The challenge is in the fact that the error concealment of a block will affect the error concealment of the following blocks as well. One solution known as AECOD is ordering the MBs for error concealment as presented in [35]. In AECOD, based on the texture of the available MBs in surrounding the corrupted area, the order of error concealment is determined in an adaptive manner. A missed MB with higher texture around it will be error concealed with higher priority. The other solution is to use Dynamic Programming (DP) for considering the interaction of the loss concealed MBs [25]. This algorithm is used for 16 × 16 pixel blocks (MBs) and can be iteratively applied for 8 × 8 pixel blocks, as well. DP approach is also used in [15] where the lost area is partitioned parallelogram-wise and then an appropriate MV is obtained for each partition. The best partitioning is selected with a blind image quality assessment method.

2.5 The proposed method compared to the above groups

In the previous sub-sections, various methods for MV recovery were presented which have been mostly proposed for H.264/AVC. The works of TEC-SMV group are not applicable to HEVC losses; since in CTU losses, the spatially neighboring MVs are far from the inner blocks of the corrupted area, and hence they cannot be efficiently interpolated for the inner blocks' MV recovery. However, in normal loss rates, if a block is lost, the MV in the co-located block is available with a high probability; therefore, TEC-TMV techniques are easily applicable in HEVC error concealment. The next group is TEC-DMVE; its success depends on the availability of the spatially neighboring pixels which are not provided for many blocks of the lost CTUs. However, with the algorithms presented in [25, 35], the boundary matching DMVE can be exploited in CTUs loss concealment, since they are introduced for error concealment of successively connected MBs, and not individual MBs.

In comparing TEC-TMV with TEC-DMVE, our preliminary experiments show that TEC-TMV techniques provide generally higher quality than TEC-DMVE techniques for most (and not all) of the test sequences; since exploiting the available surrounding pixels does not lead to good MV recovery of the relatively large missing areas. Nonetheless, there are also some cases that TEC-DMVE shows comparable performance. Furthermore, the available surrounding pixels provide valuable information which is completely ignored in TEC-TMV methods. Therefore, our proposed method tries to efficiently combine TEC-TMV and TEC-DMVE approaches. In particular, in the proposed method, the MVs in TEC-TMV are refined according to TEC-DMVE; that is, the MVs are refined with boundary matching. This means boundary matching is not used for MV recovery, but it is used for refinement of the already temporally recovered MVs.

In this work, the first version of the MVs is obtained by MV-C. It provides good loss concealment quality and also has no additional cost and even has lower computational complexity than normal decoding. Three methods for refinement of these MVs are applied and their performances in terms of quality improvement are analyzed. Finally, with an adaptive approach, one of them or none is chosen to be finally applied to error concealment. The proposed refinement leads to significantly higher quality in some cases, compared to the case of not employing refinement. It is also shown that the proposed algorithm outperforms the other error concealment algorithms for most sequences and generally offers more trustable behavior for various test sequences. Furthermore, the proposed refinement method is general and can be used to refine the MVs recovered by any TEC-TMV algorithm, one example has been presented at the end of Section 4.

3 Proposed method: MV refinement exploiting boundary matching

Boundary information plays an important role in MV recovery in Flexible MB Ordering (FMO) coded H.264/AVC streams. However, in the case of HEVC and CTU losses, its performance is not promising. On the other hand, the methods of TEC-TMV, as already mentioned, uses only the prior frame's MV information without taking into account this boundary. In the methods presented in this section, these two provisions (the prior frame's MVs and the boundary of the corrupted area) are exploited for higher quality error concealment. The MVs are first recovered by MV-C and they are then fine-tuned using one of the following three methods. In all methods, OBMA with three lines of pixels is used for boundary matching.

3.1 Method1: Refinement with starting from the corners (R-SC)

In this method, blocks of the lost area are rank ordered based on the available boundaries. That is, the blocks at the corners have the highest priority for MV refinement, since they have two lines of correct boundaries. MV refinement and error concealment is performed for these blocks. The next level of priority is the blocks adjacent to the blocks at the corners, since they have one correct boundary and one

Fig. 1 The order of 16 × 16 pixel blocks for MV refinement of two lost CTUs, the darker block has higher priority

1	2	3	4	4	3	2	1
2	5	6	7	7	6	5	2
2	5	6	7	7	6	5	2
1	2	3	4	4	3	2	1

error concealed boundary in the highest priority. With this approach, the order of blocks for MV refinement of the whole lost area is determined as shown in Fig. 1.

3.2 Method2: Refinement with order of AECOD presented in [35] (R-AECOD)

As already explained, in [35] a method was presented in which the error concealment order is determined by the texture of the neighboring blocks. The texture is measured by standard deviation of the luma pixel values. All blocks are examined for the highest texture in the adjacently received or already error concealed blocks. Based on outer boundary matching, a MV is recovered for the selected block. This idea was used for MV recovery in [35], but for refinement of the MVs in this paper. For example, Fig. 2 shows an instance of the refinement order of the blocks in two consecutively lost CTUs.

3.3 Method 3: Refinement with Vectorized dynamic programming (R-VDP)

In the previous two methods, if the adjacent blocks are not filled by loss concealment, the corresponding boundaries cannot be used during matching process. For example, for the blocks at the corners, only two of four boundaries are available and are used for matching measure. With DP, all four boundaries are taken into account.

In DP, a large problem is broken into a series of overlapping sub-problems. In a recursive manner, the optimizations of sub-problems lead to the optimization of the original large problem. The result of each sub-problem is then stored to be used for the next sub-problems. For example, for optimizing the MV of the first column in Fig. 3, its boundary in the second column must be known, but they should be error concealed first, which in turn needs to know their MVs, and it is the next sub-problem. In the Vectorized DP (VDP) algorithm, the possible MVs of blocks in the second column are considered, and according to each possible quadruple of these MVs, an optimal quadruple of MVs is found for the first column. Figure 3 and the formulations below show the flow of the sub-problems.

Fig. 2 The priority order of blocks for MV refinement in Method 2 for error concealment of two adjacently lost CTUs

3	13	17	22	21	19	20	10
4	15	32	30	25	23	24	5
2	14	31	29	26	18	12	9
1	11	16	28	27	6	8	7

Fig. 3 The loss area concealment by VDP for two adjacently lost CTUs

A	*E*	*I*					
B	*F*	*J*					
C	*G*	*K*					
D	*H*	*L*					

The cost function is based on boundary matching and defined as follows:

$$\boldsymbol{SAD_A} = \boldsymbol{SAD_{A,left}} + \boldsymbol{SAD_{A,up}} + \boldsymbol{SAD_{A,right}} + \boldsymbol{SAD_{A,down}} \tag{1}$$

where

$$SAD_{A,left} = \sum_{left\ boundary} \left(I^n_{A,left} - I^{n-1}_{A,left}(MV_A)\right) \; SAD_{A,up} = \sum_{up\ boundary} \left(I^n_{A,up} - I^{n-1}_{A,\mathrm{up}}(MV_A)\right)$$
$$SAD_{A,right} = \sum_{right\ boundary} \left(I^n_{A,right}(MV_E) - I^{n-1}_{A,right}(MV_A)\right) SAD_{A,down} = \sum_{down\ boundary} \left(I^n_{A,down}(MV_B) - I^{n-1}_{A,down}(MV_A)\right) \tag{2}$$

where $I^n_{A,left}$ are the Y-components of the pixels in the left outer boundary of block A in the current frame; $I^{n-1}_{A,left}(MV_A)$ are the pixels in the inner boundary in the left side of the block in the reference frame to which the candidate motion vector MV_A points. It can be seen that in $SAD_{A,\ right}$ and $SAD_{A,\ down}$, the pixels in the current frame depend on the MV_E and MV_B. Therefore, SAD_A is a function of MV_E and MV_B. Similarly, SAD_B is a function of MV_A, MV_F and MV_C; and so on for the SADs of blocks C and D. We can write that:

$$\boldsymbol{SAD_{ABCD}(MV_A, MV_B, \ldots MV_H) = SAD_A + SAD_B + SAD_C + SAD_D} \tag{3}$$

Minimizing SAD_{ABCD} is the first sub-problem of the VDP approach:

$$\boldsymbol{SAD_{ABCD}}^*(\boldsymbol{MV_E, MV_F, MV_G, MV_H}) = \min_{\{\boldsymbol{MV_A, MV_B, MV_C, MV_D}\}} \{\boldsymbol{SAD_{ABCD}(MV_A, MV_B, \ldots MV_H)}\} \tag{4}$$

$$\boldsymbol{MV_{ABCD}}^*(\boldsymbol{MV_E, MV_F, MV_G, MV_H}) = \underset{\{\boldsymbol{MV_A, MV_B, MV_C, MV_D}\}}{\mathbf{arg}} \left\{\boldsymbol{SAD_{ABCD}}^*(\boldsymbol{MV_E, MV_F, MV_G, MV_H})\right\} \tag{5}$$

where $\boldsymbol{MV^*_{ABCD}}$ is the vector of MVs of the blocks in the first column. This means that if the quadruple (MV_E, MV_F, MV_G, MV_H) is known, the MVs of the first column are directly obtained from $\boldsymbol{MV^*_{ABCD}}(MV_E, MV_F, MV_G, MV_H)$.

We can do the same formulation for the second and third columns as well. According to each quadruple of MVs in the third column, the SADs of blocks in the second column are computed but the left boundaries are excluded from the SADs (the reason is explain later):

$$\boldsymbol{SAD_{EFGH}(MV_E}, \mathrm{MV}_{\boldsymbol{F}}, \boldsymbol{\ldots MV_L) = SAD_E + SAD_F + SAD_G + SAD_H} \tag{6}$$

For the optimization, the accumulated cost function is taken into account:

$$\begin{aligned}&\boldsymbol{SAD_{accum}\text{–}EFGH}\ (\boldsymbol{MV_E}, \boldsymbol{MV_F}, \ldots \boldsymbol{MV_L})\\&= \boldsymbol{SAD_{EFGH}}(\boldsymbol{MV_E}, \boldsymbol{MV_F}, \ldots \boldsymbol{MV_L})/\boldsymbol{BL_{EFGH}}\\&+ \boldsymbol{SAD_{ABCD}}^{*}(\boldsymbol{MV_E}, \boldsymbol{MV_F}, \boldsymbol{MV_G}, \boldsymbol{MV_H})/\boldsymbol{BL_{ABCD}}\end{aligned} \tag{7}$$

where BL_{EFGH} and BL_{EFGH} are the number of the pixels used in the SAD measures; this is done for normalization. As (7) shows, for each $(MV_E, MV_F, \ldots MV_L)$, SAD^{*}_{ABCD} is added to the accumulated cost function; that is, the quality of the error concealment in the first column is considered.

$$\begin{aligned}&\boldsymbol{SAD_{accum}\text{–}EFGH}^{*}(\boldsymbol{MV_I}, \boldsymbol{MV_J}, \boldsymbol{MV_K}, \boldsymbol{MV_L})\\&= \min_{\{\boldsymbol{MV_E}, \boldsymbol{MV_F}, \boldsymbol{MV_G}, \boldsymbol{MV_H}\}} \{\boldsymbol{SAD_{accum}\text{–}EFGH}(\boldsymbol{MV_E}, \boldsymbol{MV_F}, \ldots \boldsymbol{MV_L})\}\end{aligned} \tag{8}$$

$$\begin{aligned}&\boldsymbol{MV^{*}_{EFGH}}(\boldsymbol{MV_I}, \boldsymbol{MV_J}, \boldsymbol{MV_K}, \boldsymbol{MV_L})\\&= \underset{\{\boldsymbol{MV_E}, \boldsymbol{MV_F}, \boldsymbol{MV_G}, \boldsymbol{MV_H}\}}{\textbf{arg}} \{\boldsymbol{SAD_{accum}\text{–}EFGH}^{*}(\boldsymbol{MV_I}, \boldsymbol{MV_J}, \boldsymbol{MV_K}, \boldsymbol{MV_L})\}\end{aligned} \tag{9}$$

This says that if the optimal combination of MVs in the third column is known, we can immediately find those for the second column, and then for the first column. Note that, when minimizing, the terms related to the SADs of the left boundaries of the second column blocks are considered by $SAD^{*}_{ABCD}(MV_E, MV_F, MV_G, MV_H)$; this is the reason we already excluded them for SAD_{EFGH}.

The above procedure is continued up to reaching the last column of the lost area, where there is no unknown boundaries on its right. That is, with the minimization of the accumulated cost function, the MVs of the blocks in the last column are found, and with going to the left, column by column, the best MVs of the whole lost area are optimally found.

3.4 Comparison of the refinement methods

In this sub-section, performances of the above introduced refinement methods are examined experimentally. The test sequences are HEVC encoded with the reference software HM16.20, the Quantization Parameter (QP) is set to 22 leading to high picture quality; this way, the error concealment distortion is more notable and affects the corrupted frames' PSNR more significantly. We have applied a hypothesis form of loss shown in Fig. 4 on the frames of the Full-HD and HD video test sequences publically available in [16]. Several lines of CTUs in the middle of the frames and on regions of interest encounter a complete loss. Every loss is applied with the assumption that no loss has occurred in the previous frames. That is, only one frame has a loss pattern of Fig. 4 in each run. This assumption assists to conceal the erroneous frames not infected by error propagation of the prior concealment distortion. This may not be the case in real scenarios, but this way, the performances of the error concealment techniques are compared more concisely.

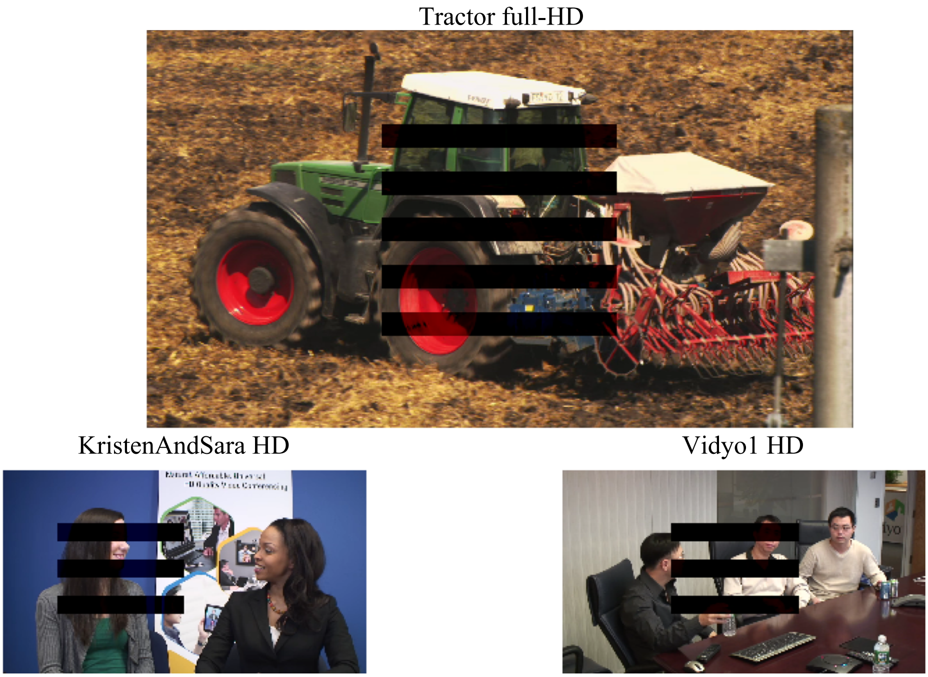

Fig. 4 The intentionally lost CTU bands for Full-HD and HD sequences

With the refinement area of ±1 pixel in up, down, left and right direction; the refinement algorithms are implemented in HEVC decoder and used to conceal the intentionally lost CTUs. The decoder gives the frame-wise quality in PSNR. The resultant PSNR curves corresponding to each refinement method and for every 10th frame of the HD and Full-HD sequences are shown in Fig. 5. The improvement achieved by the three refinement methods is observable in this figure. However, the performances of the refinement methods are actually different for various frames and various sequences. One can easily find cases where each method performs the best quality. Nonetheless, for some cases, all or some refinement methods degrade the error concealed quality. For this reason, an adaptive refinement aiming for a more trustful refinement outcome is examined in the next sub-section.

3.5 Adaptive refinement

As shown in Fig. 5, the performance of the refinement methods is not always satisfactory. In some cases, the MV refinement leads to lower loss concealment quality. The aim of this section is to improve the efficiency of the refinement methods even further.

Varying nature of the motion refinements in the previous experiments indicates that the texture of the frame and the behavior of the moving objects would be helpful for predicting whether refinement should be carried out or not; and then, which refinement method to be used. A simple and yet very effective policy is to check the temporally adjacent frame behavior if it is employed in the refinement process. In fact, with the assumption that the prior frame of the corrupted frame is intact in the location of loss (and it is a valid assumption for common loss rates), it is possible to examine which refinement method can lead to higher quality for it. Then the selected refinement method is applied on the corrupted frame; even, it may be decided to apply no refinement. That is, at the corrupted area,

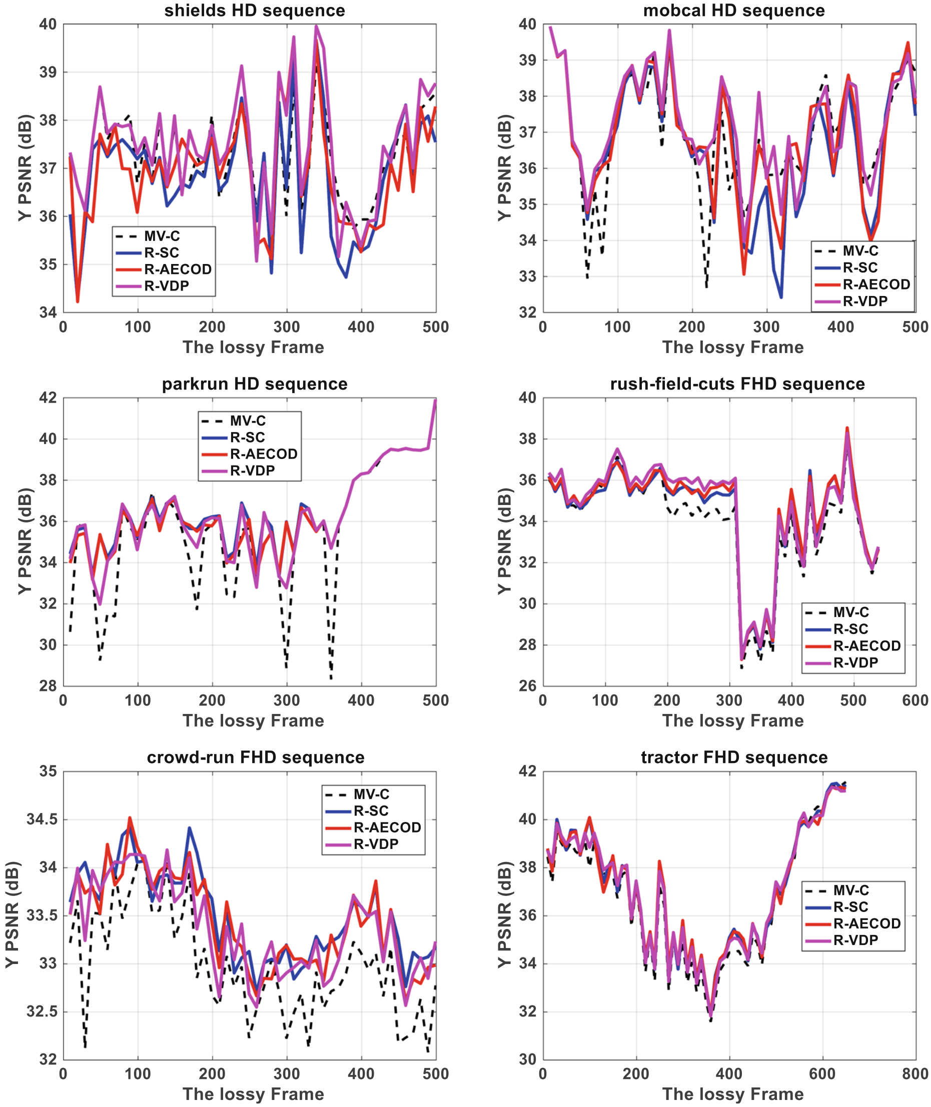

Fig. 5 Comparison of the error concealment quality achieved from the three refinement methods for test sequences

each CTU of the prior frame can be assumed lost and it is loss concealed four times, three times for three refinement methods and one without refinement. The case that leads to the highest quality in the prior frame is recommended for the corrupted frame. This policy is named *Adaptive* refinement from now on; Fig. 6 shows the pseudo-code of this algorithm.

In this pseudocode, $B(X)$ denotes a block of loss area at coordinate X. In order not to complicate the pseudocode, details of the refinement methods are not included. As explained in sub-sections 3.1, 3.2 and 3.3, in a search window of 3 × 3 pixel, the best position with minimum boundary mismatch is selected for fine tuning of MVs, i.e. obtaining α_{Mi}s in Fig. 6.

```
Map the connected missing area in the current frame (frame $n$) to the previous frame (frame $n-1$)

// Performing error concealment on the previous frame with four methods and save the outputs
$B_{n-1}(X) \leftarrow B_{n-2}(X + MV_c(X))$ for all blocks of the loss area //$MV_c(X)$ is the recovered MV with Motion Copy for block at X
                                                                              // coordinate
$P_0 \leftarrow PSNR_{n-1}$ //Saving PSNR of the error concealed loss area

Execute sub-program of R-SC method to find $\alpha_{M1}(X)$ (refinement of MVs of the blocks)
$B_{n-1}(X) \leftarrow B_{n-2}(X + MV_c(X) + \alpha_{M1}(X))$ for all blocks of the loss area
$P_1 \leftarrow PSNR_{n-1}$

Execute sub-program of R-AECOD method to find $\alpha_{M2}(X)$ (refinement of MVs of the blocks)
$B_{n-1}(X) \leftarrow B_{n-2}(X + MV_c(X) + \alpha_{M2}(X))$ for all blocks of the loss area
$P_2 \leftarrow PSNR_{n-1}$

Execute sub-program of R-VDP method to find $\alpha_{M3}(X)$, (refinement of MVs of the blocks)
$B_{n-1}(X) \leftarrow B_{n-2}(X + MV_c(X) + \alpha_{M3}(X))$ for all blocks of the loss area
$P_3 \leftarrow PSNR_{n-1}$

If $P_0 = max\ \{P_0, P_1, P_2, P_3\}$ then apply no refinement to frame $n$ //use Motion Copy for error concealment
If $P_1 = max\ \{P_0, P_1, P_2, P_3\}$ then apply R-SC refinement to frame $n$ //use Motion Copy +R-SC for error concealment
If $P_2 = max\ \{P_0, P_1, P_2, P_3\}$ then apply R-AECOD refinement to frame $n$ //use Motion Copy +R-AECOD for error concealment
If $P_3 = max\ \{P_0, P_1, P_2, P_3\}$ then apply R-VDP refinement to frame $n$ //use Motion Copy +R-VDP for error concealment
```

Fig. 6 Pseudo-code of the adaptive refinement method

The results given in Fig. 7 show the PSNRs of the MV-C concealment, without and with adaptive refinement. This figure shows that, for some sequences and some frames, this gain is not significant and for rare cases it is negative. Note that the refinement process relies on the

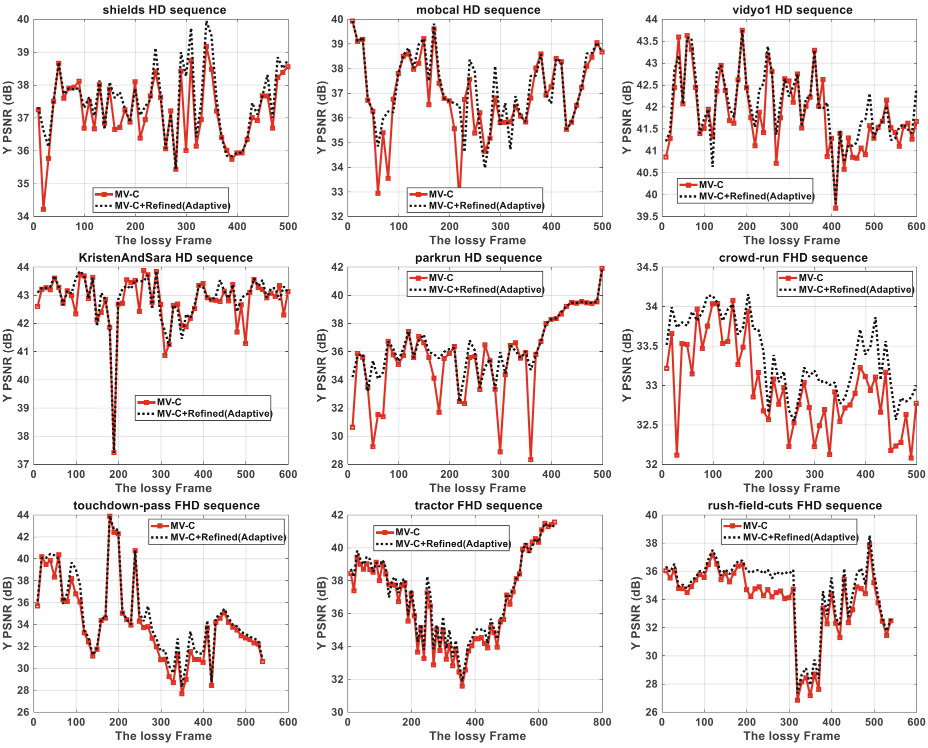

Fig. 7 Comparative curves showing the achieved PSNR improvement due to Adaptive refinement

boundary matching. It can be said that with refinement we have better matched boundary, but this actually does not mean the higher quality or higher PSNR. Therefore, the negative gains are possible. However, for some frames of *Prakrun* sequence, the improvement is as much as 4 or 7 dB, or for rush-field-cuts there is 2 dB improvement for frames 200–300. Even though, the improvement is not significant if it is averaged over frames and test sequences, but this fact that some frames show considerable improvement justifies the effectiveness of the proposed refinement method. The frames which are error concealed significantly better by refinement can provide much better quality perception for the persons viewing the video. One heavily corrupted frame in 500 frames does not effectively change the average score of the objective metrics, but it is very annoying for the subjects. Unfortunately, this might not be reflected in the average objective scores.

It is true that the proposed method is more complex than the others, but it is not problematic and it is easily manageable. First, the refinement methods are independent and can be executed completely parallel. Second, error concealment is not a computationally complex algorithm compared to Motion Estimation (ME) and Mode Decision processes at the encoder [22]. Some error concealment algorithms (e.g. Motion Copy) are less complex than standard decoding; since they have no residual data processing and MV recoveries do not need complicated processes. Boundary matching is much less complex that block matching used for ME at the encoder. And the important point is that the error concealment program is not called always, but only in the case of losses, for example 10% of times for loss rate of 10%. Therefore, if one device is able to real-time encode the video in HEVC standard; it is also able to execute the error concealment procedures without difficulty.

4 Performance comparison

In this section, the performance of the MV-C with refinement is compared with the other error concealment techniques relevant to this study. Most of the recent works are not applicable to HEVC losses, for example [6]–[30] deal with spatial error concealment and are not efficient for CTU loss concealment, [29] is for MB error concealment when its neighboring MBs are available, and [42] is for the case when all frames are encoded as I-frame for cloud-based mobile video streaming.

Some algorithms are designed for H.264/AVC but they can be extended for HEVC losses and we compare them with our proposed algorithm. These works (explained in Section 2) are named as Iter-DP [25], AECOD [35] and MV-Refi [9]. The search range in all these works is 32 pixels, we also checked 64 pixels for them but the variations were not significant. Some other relevant algorithms introduced in [7, 23, 27] are presented and examined for HEVC losses. In [7], MV-C is used for loss concealment, but if the residuals of the co-located blocks are larger than a threshold, or if it is an intra coded block, its MV is unreliable. These blocks, if possible are merged together. Then the MV of the merged block is set as the average of the surrounding blocks' MVs. In [23], similar to [7], MV-C is used for error concealment but unreliable MVs go under further processing. In that work, the loss concealment is performed in two stages; firstly, the corrupted area is replaced using MV-C, and Zero MV is used if the co-located block was intra coded. Then for the unreliable MVs, MVs are again obtained using DMVE. The search widow size and the threshold are 64 pixels and 3000, respectively. In [27], the MVE is refined to preserve the object shapes and to consider the shape information, the partitioning in the previous frame is exploited. The recovered MV is now modified such that if the projected MV comes from a partition with a larger size, it has correspondingly more chance to be selected as the recovered MV. In that work, there is a parameter which is used to weight the overlapped area; it is set as 0.3 in the experiments as mentioned by the authors. We also compared our work with the method presented in [46] where the order of concealment is determined firstly CTU-wise and then PU-wise. In that work, based on the dominant depth in the co-located CTU and the available 4 × 4 blocks around it, the CTUs are sorted such that the CTU with the smallest associated weighted depth is error concealed firstly. In the selected CTU, the PUs are sorted based on the texture randomness index introduced in [40]. The algorithms presented in [7, 23, 27, 46] are labeled as “MV-C Merge”, “Partition Weight”, “MV-C + DMVE” and “Weighted BM” respectively.

The sub-programs for the above error concealment algorithms have been implemented in HM16.20 decoder. The experimental settings are the same as before, but the loss scenario is realistic. As recommended in [13, 14], slices are encapsulated into the packets where the maximum packets size is 1400 bytes in order not to exceed Ethernet network Maximum

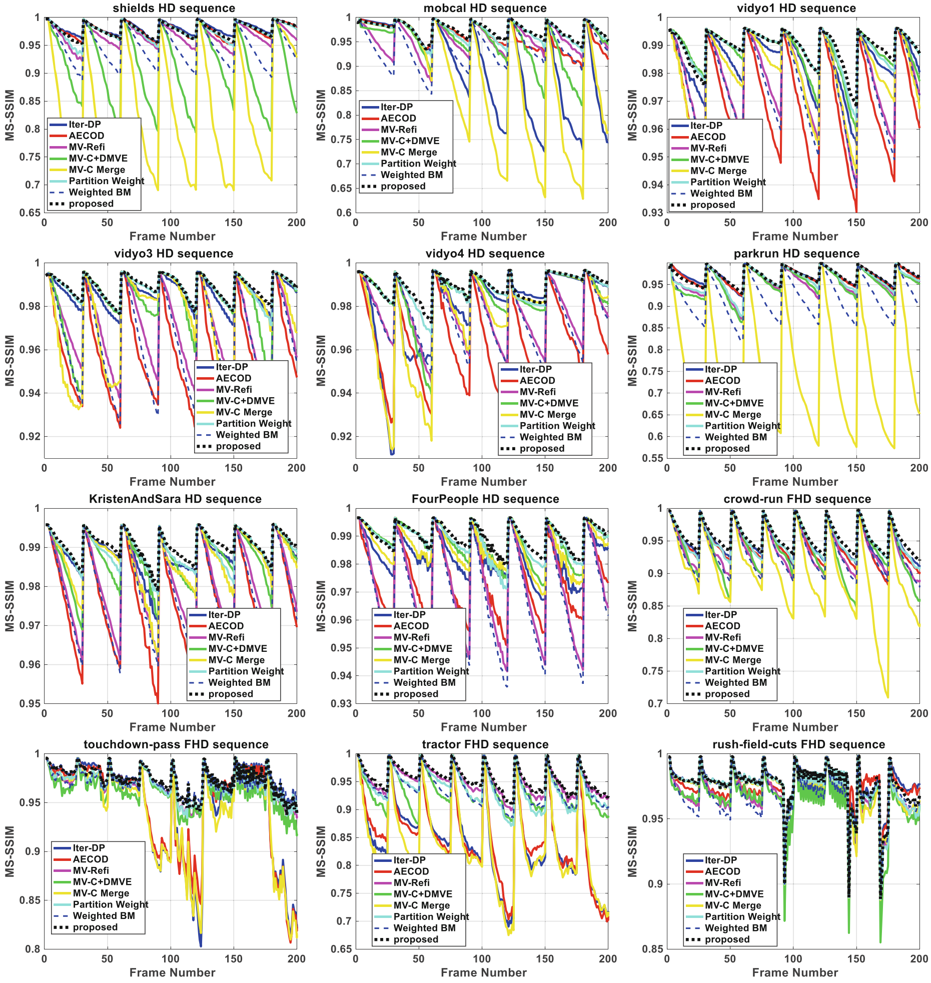

Fig. 8 MS-SSIM Quality comparison of the adaptive refinement method and other error concealment works, PLR = 12%

Table 1 The average MS-SSIM gain (improvement) obtained from our proposed error concealment compared to the other methods

Sequences	Iter- DP [25]	AECOD [35]	MV-Refi [9]	MV-C+ DMVE [23]	MV-C Merge [7]	Partition Weight [27]	Weighted BM [46]
PLR=2%							
Shields HD	−0.0004	0.0012	0.0021	0.0051	0.0187	0.0005	0.0050
Mobcal HD	0.0156	0.0043	0.0022	0.0032	0.0182	0.0003	0.0054
Vidyo1 HD	0.0005	0.0030	0.0013	0.0002	0.0013	0.0002	0.0019
Vidyo3 HD	0.0005	0.0060	0.0022	0.0007	0.0012	0.0000	0.0032
Vidyo4 HD	0.0012	0.0039	0.0016	0.0007	0.0017	0.0003	0.0022
Parkrun HD	−0.0001	0.0002	0.0033	0.0026	0.0305	0.0014	0.0091
KristenAndSara HD	0.0004	0.0020	0.0014	0.0004	0.0007	0.0002	0.0021
FourPeople HD	0.0023	0.0029	0.0025	0.0001	0.0006	0.0006	0.0035
Crowd-run FHD	0.0020	0.0035	0.0032	0.0021	0.0165	0.0009	0.0037
Touchdown-pass FHD	0.0087	0.0085	0.0008	0.0017	0.0098	0.0006	0.0002
Tractor FHD	0.0323	0.0321	0.0011	0.0037	0.0318	0.0030	0.0030
Rush-field-cuts FHD	−0.0003	0.0003	0.0010	0.0016	0.0006	0.0003	0.0011
PLR=6%							
Shields HD	−0.0013	0.0024	0.0064	0.0253	0.0685	0.0009	0.0161
Mobcal HD	0.0458	0.0141	0.0069	0.0121	0.0609	0.0013	0.0164
Vidyo1 HD	0.0015	0.0110	0.0050	0.0006	0.0040	0.0005	0.0062
Vidyo3 HD	0.0013	0.0192	0.0089	0.0032	0.0064	0.0003	0.0119
Vidyo4 HD	0.0030	0.0140	0.0050	0.0024	0.0053	0.0007	0.0065
Parkrun HD	−0.0005	0.0004	0.0117	0.0085	0.0982	0.0048	0.0301
KristenAndSara HD	0.0011	0.0064	0.0048	0.0013	0.0024	0.0006	0.0062
FourPeople HD	0.0037	0.0075	0.0090	0.0005	0.0017	0.0005	0.0109
Crowd-run FHD	0.0048	0.0098	0.0100	0.0085	0.0448	0.0032	0.0119
Touchdown-pass FHD	0.0208	0.0197	0.0023	0.0051	0.0237	0.0016	0.0007
Tractor FHD	0.0846	0.0825	0.0025	0.0117	0.0870	0.0077	0.0083
Rush-field-cuts FHD	−0.0007	0.0009	0.0035	0.0050	0.0020	0.0012	0.0039
PLR=12%							
Shields HD	−0.0020	0.0035	0.0144	0.0640	0.1299	0.0033	0.0331
Mobcal HD	0.0806	0.0209	0.0158	0.0271	0.1123	0.0032	0.0301

Table 1 (continued)

Sequences	Iter- DP [25]	AECOD [35]	MV-Refi [9]	MV-C + DMVE [23]	MV-C Merge [7]	Partition Weight [27]	Weighted BM [46]
Vidyo1 HD	0.0040	0.0211	0.0123	0.0023	0.0088	0.0006	0.0138
Vidyo3 HD	0.0020	0.0320	0.0198	0.0077	0.0149	0.0006	0.0250
Vidyo4 HD	0.0065	0.0257	0.0123	0.0057	0.0116	0.0010	0.0139
Parkrun HD	−0.0002	0.0008	0.0232	0.0147	0.1650	0.0078	0.0544
KristenAndSara HD	0.0020	0.0141	0.0111	0.0033	0.0059	0.0012	0.0131
FourPeople HD	0.0061	0.0145	0.0197	0.0017	0.0043	0.0011	0.0220
Crowd-run FHD	0.0119	0.0185	0.0172	0.0188	0.0797	0.0044	0.0214
Touchdown-pass FHD	0.0359	0.0347	0.0047	0.0110	0.0419	0.0034	0.0015
Tractor FHD	0.1234	0.1172	0.0043	0.0277	0.1358	0.0136	0.0160
Rush-field-cuts FHD	−0.0013	0.0013	0.0070	0.0109	0.0041	0.0021	0.0080

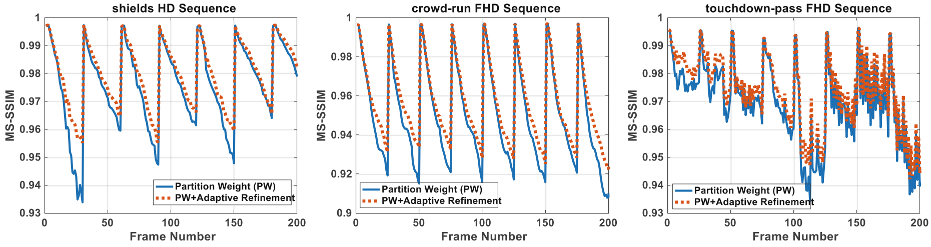

Fig. 9 The performance of adaptive refinement applied on "Partition Weight" method

Transfer Unit (MTU) size. The channel loss is simulated such that the packets experience burst losses modeled by two-state Morkov chain [38]. The average PLRs are 2%, 6% and 12% with an average burst length of three packets. These are repeated for 20 randomly generated patterns and the results are then averaged. The video qualities are measured with MS-SSIM; as this metric shows usually trustable measurement for error concealed videos as investigated in [21]. The sub-program for computing MS-SSIM has been implemented in HM16.20 decoder, by which the quality of each video frame can be easily measured.

The frame-wise MS-SSIM measures are shown in Fig. 8. The PLR for these curves is 12%. It can be seen from these figures that the proposed method leads to higher quality most of the times. In order to refresh the decoder from the error, for HD (FHD) sequences, every 30 (25) frames, an I-frame is transmitted and we assumed that with channel protection, this I-frame is correctly delivered. The reason for 25 frames for FHD is that the FHD sequences are more

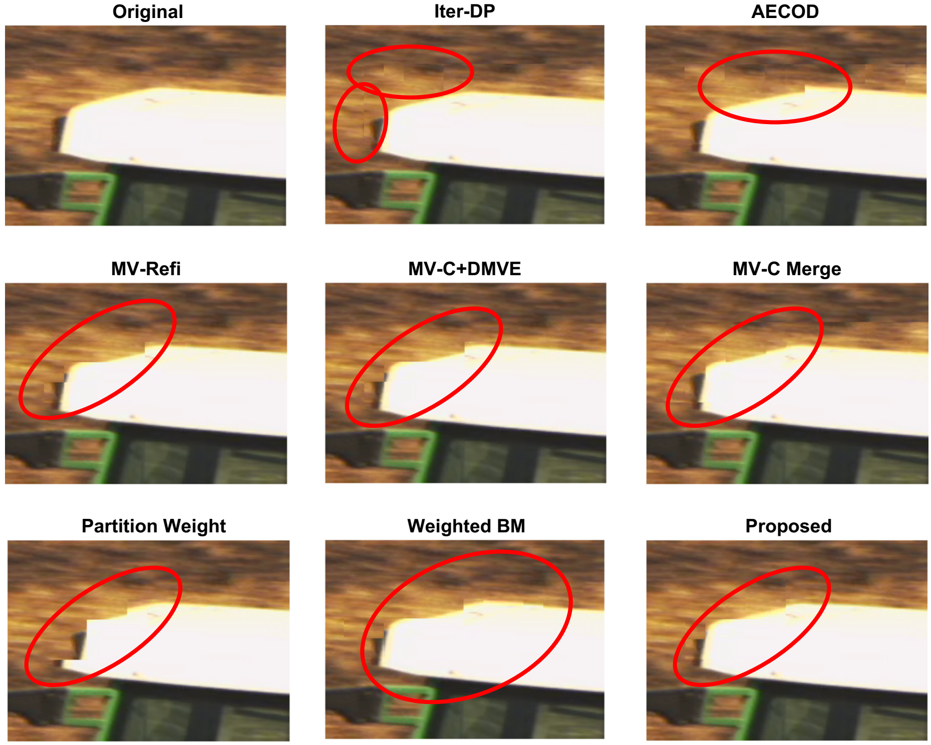

Fig. 10 Visual quality comparison of the proposed method against the other methods for Frame 53 of *Tractor* sequence

sensitive to losses than HD ones. The spiky peaks in the curves are due to the quality of these loss-free I-Frames.

It is worth noting that for all sequences, the curves of our proposed method are on the top. As already mentioned, this is due to exploiting both temporal (MVs) and spatial (boundaries) information in a constructive manner.

We averaged the MS-SSIM scores of the frames for the proposed method as well as for the other methods. The results in Table 1 shows the gain achieved by the proposed method compared to the others for PLRs of 2%, 6% and 12%. The larger positive numbers in the table show more improvement through the proposed method. The negative number show that those methods perform better than the proposed method for that sequence; for example "Iter-DP" method and *Shields*, *Parkrun* and *Rush-field-cuts* sequence; however, the differences are small and also "Iter-DP" does not show better performance for the other sequences as shown by the table.

As already mentioned the proposed refinement method is general and is usable for many temporal error concealment methods. In particular, it can be appended to any algorithm of TEC-TMV group introduced in Section 2.3. For example, we can apply our Adaptive refinement on "Partition Weight" method which is more successful than "MV-C Merge" and "MV-C + DMVE" as shown in performance figures and table. The results given in Fig. 9 show that the achieved MS-SSIM is higher with the refined MV. This experiment confirms the performance and the generality of the proposed refinement method.

As the last part, the visual quality of the outputs of the error concealment methods is presented. Two sample frames of *Tractor* and *Vidyo1* are shown in Figs. 10 and 11,

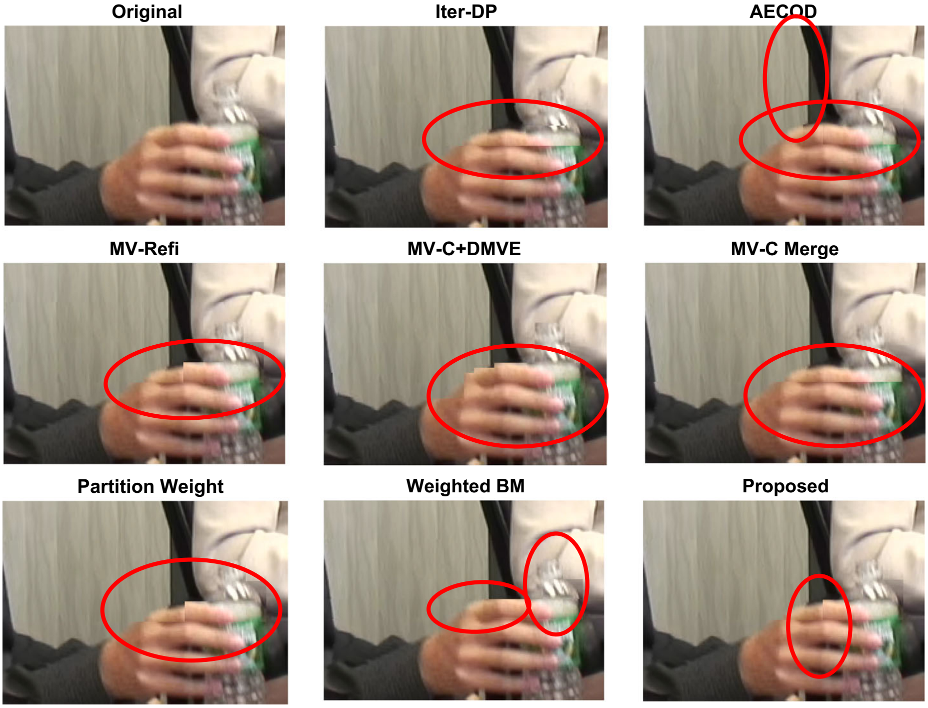

Fig. 11 Visual quality comparison of the proposed method against the other methods for Frame 4 of *Vidyo1* sequence

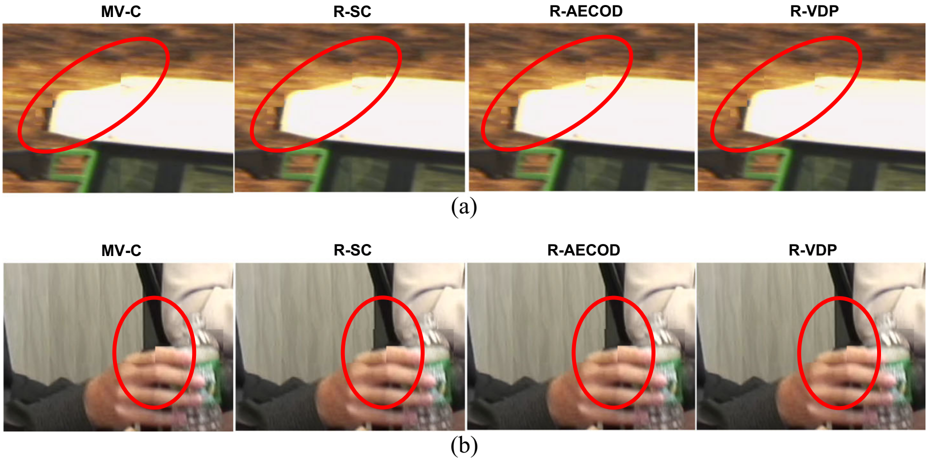

Fig. 12 Visual quality gain achieved by the refinement methods for (a) frame 53 of *Tractor* sequence and (b) frame 4 of *Vidyo1* sequence

respectively. In Fig. 10, the borders have high quality in "Iter-DP" method, but this is not the case for the background. Or in Fig. 11, recovery of the finger by "Weighted BM" is better than our method, but with the close look, one can see that the bottle top is not correct for that. Furthermore, in order to show the effect of the refinements, the picture outputs by the three refinement methods are also shown in Fig. 12 for those two frames, respectively.

5 Conclusion

In this paper, a method which blends two main approaches of temporal error concealment, TEC-TMV and TEC-DMVE, was presented. Generally, TEC-DMVE approaches are not as successful as TEC-TMV is for HEVC loss concealment; even though it may outperform TEC-TMV for some video sequences. The successes of TEC-TMV and TEC-DMVE techniques are highly content dependent. In this paper, it was proposed to exploit the techniques of TEC-DMVE in order to refine the MVs recovered by TEC-TMV. In other words, the weak points of TEC-TMV are covered by TEC-SMV refinement. In this approach, the spatially neighboring pixels in the boundary of the corrupted area are used for refinement of the recovered MVs. Three refinement techniques were applied on Motion Copy error concealment algorithm. It was shown that even though boundary information may not be so helpful for MV recovery, but it is successful for fine tuning of the already recovered MVs. With an adaptive manner, one can decide which refinement method or even none should be applied.

With experimental results, the proposed method shows promising performance compared to the other works presented or extendable for loss concealment of the HEVC streams. It happens for some sequences that one specific error concealment algorithm gives slightly higher quality than the proposed method, but for other sequences this algorithm gives lower quality. However, our proposed method results in either the best or near the best, for all test sequences. The visual quality provided by the proposed method was also verified. Another important aspect of our refinement method is that it can easily be applied on the MVs recovered with any TEC-TMV algorithm. This feature was tested on one method and over several test sequences.

References

1. Adeyemi-Ejeye AO, Alreshoodi M, Al-Jobouri L, Fleury M (2019) Impact of packet loss on 4K UHD video for portable devices. Multimed Tools Appl 78:31733–31755
2. Ahmed AV, Khot UP (2020) An efficient generalized error concealment in video codec. International Journal of Computer Vision and Image Processing 10(4):1–28
3. Akbari A, Trocan M, Granado B (2017) Sparse recovery-based error concealment. IEEE Transactions on Multimedia 19(6):1339–1350
4. Belfiore S, Grangetto M, Magli E, Olmo G (2005) Concealment of whole-frame losses for wireless low bit-rate video based on multiframe optical flow estimation. IEEE Trans Multimed 7(2):316–329
5. Boussard V, Golaghazadeh F, Coulombe S, Coudoux FX, Corlay P (2020) Robust H.264 video decoding using CRC-based single error correction and non-desynchronizing bits validation. IEEE International Conference on Image Processing (ICIP)
6. Byongsu H, Jonghyon J, Cholsu R (2019) An improved multi-directional interpolation for spatial error concealment. Multimed Tools Appl 78:2587–2598
7. Chang YL, Reznick YA, Chen Z, Cosman PC (2013) Motion compensated error concealment for HEVC based on block-merging and residual energy. International Packet Video Workshop (PV)
8. Chen C, Liu Y, Yang Z, Bu J, Deng X (2008) Multi-frame error concealment for H.264/AVC frames with complexity adaptation. IEEE Trans Consum Electron 54(3):1422–1429
9. Chien JT, Li GL, Chen MJ (2010) Effective error concealment algorithm of whole frame loss for H.264 video coding standard by recursive motion vector refinement. IEEE Trans Consum Electron 56(3):1689–1695
10. Choe G, Nam C, Chu C (2018) An effective temporal error concealment in H.264 video sequences based on scene change detection-PCA model. Multimed Tools Appl 77(24):31953–31967
11. Chung B, Yim C (2019) Bi-sequential video error concealment method using adaptive Homography-based registration. IEEE Transactions on circuits and systems for video technology 30(6):1535–1549
12. Fleury M, Moiron S, Ghanbari M (2011) Innovations in video error resilience and concealment. Recent Patents Signal Process 1(2):1–11
13. Francisco NYZ, et. al. (2017) Effects of error concealment on HEVC standard over wireless channels designed using Rayleigh fading and Markov model on AODV routed slices. IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)
14. Ghanbari M (2011) Standard codecs: image compression to advanced video coding. IET publication, 3rd edition, appendix E
15. Hojati S, Kazemi M, Moallem P (2019) Error concealment with parallelogram partitioning of the lost area," Multimed Tools Appl, 79: 7449–7469
16. : https://media.xiph.org/video/derf/
17. Huang Z, Cai Q (2018) A video data recovery algorithm in wireless communication networks. IEEE 18th International Conference on Communication Technology: 727–731.
18. Hwang MC, Kim JH, Duong DT, Ko SJ (2008) Hybrid temporal error concealment methods for block-based compressed video transmission. IEEE Trans Broadcast 54(2):198–207
19. Kazemi M (2020) In favor of fully intra coding for HEVC video transmission over lossy channels. Signal, Image and Video Processing, doi 15:165–173. https://doi.org/10.1007/s11760-020-01735-y
20. Kazemi M, Ghanbari M, Shirmohammadi S (2020) Intra coding strategy for video error resiliency: behavioral analysis. IEEE Transactions on Multimedia 22(9):2193–2206
21. Kazemi M, Ghanbari M, Shirmohammadi S (2020) The performance of quality metrics in assessing error-concealed video quality. IEEE Trans Image Process 29:5937–5952
22. Kazemi M, Ghanbari M, Shirmohammadi S (2021) A review of temporal video error concealment techniques and their suitability for HEVC and VVC. Multimed Tools Appl 80:12685–12730. https://doi.org/10.1007/s11042-020-10333-6
23. Kim DH, Kwon YJ, Choi KH (2018) Motion-vector refinement for video error concealment using downhill simplex approach. ETRI J 40(2):266–274
24. Li Y, Chen R (2017) Motion vector recovery for video error concealment based on the plane fitting. Multimed Tools Appl 76(13):14993–15006
25. Lie W, Lee C, Yeh C, Gao Z (2014) Motion vector recovery for video error concealment by using iterative dynamic programming optimization. IEEE Transactions on Multimedia 16(1):216–227
26. Lin TL, Chen WC, Lai CK (2013) Recovery of lost motion vectors using encoded residual signals. IEEE Trans Broadcast 59(4):705–716

27. Lin T, Yang N, Syu R, Liao CC, Tsai WL (2013) Error concealment algorithm for HEVC coded video using block partition decisions. IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)
28. Lin TL, Ding TL, Yang NC, Wu PY, Tung KH, Lai CK, Chang TE (2016) Video motion vector recovery method using decoding partition information. J Disp Technol 12(11):1451–1463
29. Lin TL, Ding TL, Fan CY, Chen WC (2017) Error concealment algorithm based on sparse optimization. Multimed Tools Appl 76(1):397–413
30. Liu X, Zhai D, Zhou J, Wang S, Zhao D, Gao H (2017) Sparsity-based image error concealment via adaptive dual dictionary learning and regularization. IEEE Trans Image Process 26(2):782–796
31. Nam C, Chu C, Kim T, Han S (2020) A novel motion recovery using temporal and spatial correlation for a fast temporal error concealment over H.264 video sequences. Multimed Tools Appl 79:1221–1240
32. Peng YT, Cosman PC (2014) Weighted boundary matching error concealment for HEVC using block partition decisions. 48th Asilomar Conference on Signals, Systems and Computers: 921–925
33. Peng Q, Yang T, Zhu C (2002) Block-based temporal error concealment for video packet using motion vector extrapolation. IEEE Int Conf Commun, Circuits Syst West Sino Expo 1:10–14
34. Punjabi SA, Katsaggelos AK (2018) Video error concealment using deep neural networks. In Proceedings of the IEEE International Conference on Image Processing, Athens, Greece, pp 380–384
35. Qian X, Liu G, Wang H (2009) Recovering connected error region based on adaptive error concealment order determination. IEEE Trans Multimedia 11(4):683–695
36. Sankisa A, Punjabi A, Katsaggelos AK (2019) Optical flow prediction for blind and non-blind video error concealment using deep neural networks. International Journal of Multimedia Data Engineering and Management 10(3):27–46
37. Sankisa A, Punjabi A, Katsaggelos AK (2020) Temporal capsule networks for video motion estimation and error concealment. SIViP 14:1369–1377
38. Silva CAGD, Pedroso CM (2019) MAC-layer packet loss models for Wi-fi networks: a survey. IEEE Access 7:180512–180531
39. Suh JW, Ho YS (2002) Error concealment techniques for digital TV. IEEE Trans Broadcast 48(4):299–306
40. Tang CW, Chen CH, Yu YH, Tsai CJ (2006) Visual sensitivity guided bit allocation for video coding. IEEE Trans Multimedia 8(1):11–18
41. Usman M, He X, Xu M, Lam K (2015) Survey of error concealment techniques: research directions and open issues. IEEE Picture Coding Symposium: 233-238
42. Usman M, He X, Lam KM, Xu M, Bokhari SMM, Chen J (2016) Frame interpolation for cloud-based Mobile video streaming. IEEE Transactions on Multimedia 18(5):831–839
43. Usman MA, Seong CH, Lee MH, Shin SY (2019) A novel error detection & concealment technique for videos streamed over error prone channels. Multimed Tools Appl 78:22959–22975
44. Wang Y, Zhu QF (1998) Error control and concealment for video communication: a review. Proc IEEE 86(5):974–997
45. Xiang J, Xu C, Yan Q, Peng XW (2019) Generative adversarial networks based error concealment for low resolution video. IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)
46. Xu J, Jiang W, Yan C, Peng Q, Wu X (2018) A novel weighted boundary matching error concealment Schema for HEVC. IEEE International Conference on Image Processing (ICIP)
47. Yan B, Gharavi H (2009) A hybrid frame concealment algorithm for H.264/AVC. IEEE Trans Image Process 19(1):98–107
48. Yang SH, Chang CW, Chan CC (2015) An object-based error concealment technique for H.264 coded video. Multimed Tools Appl 74(23):10785–10800
49. W. Yu, H. Sun, G. He and Z. Zhang (2017) A multi-step temporal error concealment method. International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)
50. Zhou J, Yan B, Gharavi H (2001) Efficient motion vector interpolation for error concealment of H.264/ AVC. IEEE Trans Broadcast 57(1):57–80
51. Zhou Z, Dai M, Zhao R, Li B, Zhong H, Wen Y (2016) Video error concealment scheme based on tensor model. Multimed Tools Appl 76(14):16045–16061

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.