Multimedia Tools and Applications (2024) 83:20997–21017
https://doi.org/10.1007/s11042-023-16310-z

PCA-based hierarchical clustering approach for motion vector estimation in H.265/HEVC video error concealment

Ali Radmehr[1] · Ali Aghagolzadeh[1] · Seyed Mehdi Hosseini Andargoli[1]

Received: 1 August 2022 / Revised: 9 June 2023 / Accepted: 11 July 2023 /
Published online: 5 August 2023

Abstract
Video error concealment (EC) is a challenging issue, and one of the key challenges is accurately modeling motion vectors (MVs) in corrupted frames. To address this problem, we propose a novel Principal Component Analysis (PCA)-based model for EC that uses hierarchical clustering for MV estimation. Our approach has two main steps: first, we obtain the MV fields represented in the 2D plane, and second, we conceal the erroneous regions of the frame using a PCA-based hierarchical clustering algorithm. This algorithm selects the best MVs among the field candidates while maximizing the correlation of MVs in a cluster. Our approach outperforms recent related techniques in terms of PSNR and SSIM for the H.265/HEVC compression standard. Specifically, our method improves average PSNR by up to 4.99 dB and average SSIM by 0.029. Furthermore, our method has slightly lower computational complexity compared to the compared techniques, especially for videos with relatively uniform motion over the missing areas.

Keywords Video error concealment · Principal component analysis · H.265/HEVC video coding

1 Introduction

The high demand for video communication leads to the provision of highly packetized video data [32]. Video communication has become increasingly popular in recent years due to advancements in technology and the proliferation of high-speed internet connections. As a result, there has been a surge in the demand for video content, which has led to the development of highly packetized video data. Packetization is the process of dividing data into smaller

✉ Ali Aghagolzadeh
aghagol@nit.ac.ir

Ali Radmehr
a.radmehr.ir@ieee.org

Seyed Mehdi Hosseini Andargoli
smh_andargoli@nit.ac.ir

[1] Faculty of Electrical and Computer Engineering, Noshirvani University of Technology, Babol, Iran

units, or packets, that can be transmitted over a network. The packetization of video data offers several benefits, such as improved network efficiency, reduced latency, and increased reliability. Moreover, packetization allows for the transmission of video data over a wide range of networks, including wireless and mobile networks. The recent advancements in video codecs, such as the H.265/HEVC, have further improved the packetization of video data. These codecs use advanced compression techniques to reduce the size of video data, resulting in higher compression ratios compared to older codecs. This makes them highly suitable for a wide range of networks, including those with limited bandwidth. The use of highly packetized video data in conjunction with advanced video codecs has led to significant improvements in video quality, reduced buffering times, and increased network efficiency. As a result, these technologies are becoming increasingly important for video communication applications, such as video conferencing, live streaming, and video-on-demand services [20].

However, the high compression used in video codecs, particularly in the H.265/HEVC coding standard, makes them vulnerable to channel errors [16]. These errors can significantly degrade the video quality, especially in inter-coded consecutive frames [15]. The loss of critical information in these frames can result in motion discontinuities, blockiness, and even complete frame freezing, severely impairing the overall viewing experience. Therefore, effective algorithms are essential to mitigate the impact of channel errors and restore the video quality by accurately reconstructing the lost or corrupted information [24].

To address this problem, efforts are being made to develop error-resilient and concealment methods. The former aims to strengthen the integrity of the video data, while the latter attempts to hide the erroneously decoded frames [1]. Furthermore, Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) are introduced in the literature. However, FEC increases video bandwidth, and ARQ is not suitable for interactive applications [11, 35, 38]. The Error Concealment (EC) methods, on the other hand, are post-processing techniques that try to recover a corrupted parts of a frame from the correctly decoded spatial and temporal information. These methods are categorized into spatial, temporal, and hybrid classes. Spatial Error Concealment (SEC) algorithms exploit the pixel information available in the current video frame. Also, Temporal Error Concealment (TEC) techniques benefit from motion data, consisting of motion vectors (MVs) for the nearest correctly received motion data. Moreover, hybrid EC methods use temporal and spatial correlated information to recover erroneous parts of the video frame [17, 33]. The EC algorithms in H.264/AVC aim to conceal damaged Macroblocks (MBs) within each erroneous slice. However, for H.265/HEVC, the large damaged region of a frame makes the EC techniques used in H.264/AVC less effective and causes several significant issues [17]:

- Traditional EC methods rely only on neighboring MVs, which can produce inaccurate results when a video frame exhibits non-uniform motion.
- In H.265/HEVC, there is no Flexible Macroblock Ordering (FMO) slicing available; therefore, the EC method cannot rely only on adjacent motion data. Moreover, the damaged region is far from the correctly received spatially neighboring blocks. Thus, there are insufficient available MVs for the concealment procedure. The reason is that the Coding Tree Unit (CTU) size can be as large as 64 × 64 pixels in H.265/HEVC. To account for transmission errors, an integer number of CTUs are grouped into one slice, and an integer number of slices are encapsulated into a single transmission packet. Consequently, a lost packet in an H.265/HEVC bitstream affects several CTUs, resulting in a significant amount of frame area being lost. Therefore, all the information associated with these CTUs will be corrupted. Also, in this case, SEC algorithms may not be effective in recovering the lost data since the pixels are too far apart from each other to be useful.

Therefore, alternative techniques such as TEC methods and inter-frame prediction are commonly used to mitigate the effects of packet loss on video quality [17, 18].
- In H.265/HEVC, the traditional SEC techniques are not applicable in real scenarios since the errors damage multiple consecutive CTUs, which are wide regions of a frame [16].
- While various EC algorithms have been proposed in the literature that rely on the side information, they are often unsuitable for real-world scenarios because they are non-standard and require modifications to the codec. Consequently, these algorithms have limited practical applications [17].

This paper introduces a novel TEC method specifically designed for estimating lost blocks in H.265/HEVC encoded video data. Our proposed block-based approach aims to strike a balance between computational complexity and video quality preservation. To achieve this, we leverage a PCA-based hierarchical clustering algorithm to identify highly correlated MVs and improve the accuracy of MV candidates. The proposed method considers each mutually orthogonal unit eigenvector as an axis of an ellipsoid. In other words, it develops a classification scheme that categorizes correlated temporally neighboring MVs of the erroneous regions of the frame. The proposed approach can also discover essential features of the most correlated motion information of the frame's faulty parts and reveal previously unsuspected relationships. Additionally, the overall area of the ellipsoid can be interpreted as a parameter of computational complexity, i.e., there is a positive relationship between the MVs' nonuniformity and the area of the ellipsoid.

The proposed method offers several advantages over the previous algorithms. Our method is content adaptive, which means it can adapt to the characteristics of the video being processed, resulting in more accurate and visually pleasing results. In addition, our method has lower complexity than the previous methods, which can lead to faster processing times and lower computational requirements. Another significant advantage of our method is its applicability to the H.265/HEVC video compression standard, which is becoming increasingly popular. Unlike many previous methods, our method can generate new MVs, which can be especially useful in cases where there are insufficient MVs available. This can improve the accuracy of the reconstructed frames and reduce the visual artifacts. We evaluated the performance of our method using several benchmark video sequences, and compared the results with several previous methods. The experiments show that our method outperforms compared methods in terms of visual quality and computational efficiency. Overall, our proposed method for video error concealment offers a comprehensive set of advantages that make it a valuable tool for a wide range of applications in video processing and compression. The major contribution of this paper are given as follows:

- Our proposed method makes a significant contribution by utilizing the directional correlation among an object's MVs in a video scene, interpreting them as elongated clusters within a frame's region.
- Our paper introduces a novel approach by utilizing Principal Component Analysis (PCA) for modeling ellipsoidal motion data, which enables us to limit candidate regions in directions with lower diversification, thereby enhancing the algorithm's efficiency.
- In our paper, we employ a divisive hierarchical clustering technique to classify MVs, leveraging the Sum of Square Error (SSE) to minimize Euclidean distance between clusters and decrease the order of complexity, thereby improving the method's performance.
- Our proposed method presents a comprehensive methodology for generating new MV candidates using the axis-aligned ellipse, where the magnitude and angle of the ellipse are determined by the variance and covariance of the MVs in each cluster.

Table 1 The description of key symbols in our proposed method

Symbol	Description
n	number of data points or observations in a dataset
Θ_{SSE}	threshold that determins stopping criterion
S	set of temporally neighboring MVs
$\mathbf{X}_l$	the l^{th} observation
N	number of elements in the set S
$C_{i,j}$	the j^{th} cluster in the i^{th} level of hierarchical clustering
$R_{i,j}$	the SSE value for the cluster $C_{i,j}$
j	index of cluster number in divisive hierarchical clustering
i	index of cluster level in divisive hierarchical clustering
L	set of all clusters
$stop_{flag}$	stopping flag for the divisive hierarchical clustering algorithm
k	a temporary value that shows the cluster number in each level
$\mathbf{MV}$	a motion vector of a block
MV_x	the x component of a **MV**
MV_y	the y component of a **MV**
(x_i, y_i)	the i^{th} inner point of an ellipse
$[x_{min}, x_{max}]$	solution interval on the x-axis
$[y_{min}, y_{max}]$	solution interval on the y-axis
λ_1, λ_2	the largest and second-largest eigenvalues of the covariance matrix
R_x	the major semi-axis length of an ellipse
R_y	the minor semi-axis length of an ellipse
σ_x	the standard deviation of motion data along first principal component
σ_y	the standard deviation of motion data along second principal component
α	the orientation of an ellipse
$\mathbf{u}_1$	the largest eigenvector of a covariance matrix
(x_d, y_d)	displacement of an ellipse
a, b, c	coefficients of a quadratic equation

- The results and discussion in our paper highlight the exceptional efficiency and performance of our proposed scheme, demonstrating the significant contribution of our work.

This paper is organized as follows. The state-of-the-art EC algorithms are reviewed in Section 2. Section 3 summarizes the problem statement, discussing the MV modeling and the divisive hierarchical clustering algorithm. Section 4 describes the proposed algorithm. Detailed experimental results and computational complexity analysis are presented in Section 5. Finally, Section 6 concludes the paper with final remarks.

Table 1 summarizes the main symbols utilized in our proposed method.

2 Related work

TEC algorithms mostly use the available motion data surrounding the damaged area. A novel nonlinear and non-Gaussian motion estimation technique is proposed in [28]. MV recovery should also consider MVs, which are strongly correlated with the lost MV, as presented in [14]. The authors find the tendency of the MV in each 4×4 block by processing the co-located MV in the previous frame with successfully received blocks. Moreover, the method provides information about the closest MV to the motion information of the lost block. Although this method presents an appropriate strategy for motion recovery, it only exploits the upper blocks for tendency estimation, which may not be valid in all scenarios. Furthermore, the recovered MV for the more inner blocks is not sufficiently trustable for large damaged areas, and the result may not be accurate.

A trained PCA is used to conceal damaged parts of a frame using information from the previous frames' buffer [5]. The authors proposed a novel method for detecting scene changes using dynamic thresholding and a similarity metric with a novel scene change detection algorithm. Additionally, an updated PCA model is presented by combining scene change features with index transformation buffers, and missing images are concealed by Projection Onto the Convex Set (POCS). Due to scene change, all previous frames may not be suitable and should be discarded from the buffer. Furthermore, the success of this method depends on the correlation of the lost area with the surrounding pixels, which is not the case for H.265/HEVC since the available correct pixel data is far from the lossy blocks.

Data hiding is a technique that can embed auxiliary or watermark data into digital video, which has a high degree of data redundancy. This can enable video error concealment by recovering the lost parts of the video frames from the embedded data. For example, the authors in [4] present a scheme that uses compressed sensing to hide auxiliary data into video frames over lossy channels. Compressed sensing is a method that can reconstruct sparse signals from fewer measurements than the conventional methods. Data hiding can also enhance image authentication by embedding watermark data [10].

Li et al. model neighboring MVs of the lost block by plane fitting, which shows how they change in small regions of the damaged frame [21]. However, this approach is not meaningful and needs further modifications for heavily corrupted video frames. Similarly, Lin et al. use the nearest available MVs to predict the missing ones. They propose a novel TEC method to estimate MVs' weighting using disparities among MVs of available neighboring blocks [23]. This method does not work well for large, lossy regions where the neighboring MVs are missing for inner blocks. In [13], the authors propose a shape-preservation technique that uses different EC strategies depending on the position of the corrupted blocks. However, this method is limited by the difficulty of measuring the object's boundaries.

In [37], the authors propose a novel weighted boundary matching EC scheme for H.265/HEVC based on the Coding Unit (CU) depths and Prediction Unit (PU) partitions in reference frames. The proposed method uses the information of CU depths in reference frames for lost slices, and for each Largest Coding Units (LCU) in a lost slice. The LCUs surrounding the co-located LCU are used to calculate the summed CU-depth weight to determine the conceal order of each CU. The co-located partition decision from the reference frame is adopted for PUs in each lost CU, and the sequence of PUs to conceal is sorted based on the texture randomness index weight. Finally, the best estimated MV for the lost PU is selected for concealment. Although experimental results show that the proposed method achieves higher PSNR gains compared to the other methods, this approach may face challenges when dealing with videos that have high motion. In such scenarios, the co-located CUs and PUs in reference frames may not provide accurate or reliable motion information, particularly when the MVs are large. As a result, the proposed method cannot determine the conceal order of each CU since the neighboring LCUs may not provide sufficient motion information to accurately estimate the CU depths. In addition, the use of MV prediction for lost PUs may result in suboptimal concealment when the MVs are unreliable or ambiguous due to high motion.

In [22], the authors propose an EC algorithm that improves upon a commonly used algorithm that extrapolates MVs from the previous frame. The proposed method considers partition decision information from the previous frame to improve the concealment process. The results of the study demonstrate that the proposed method performs better than the conventional algorithm. However, the proposed method relies on the partition decision information from the previous frame, which may not be accurate or reliable in scenarios with fast object movement, as the object boundaries and sizes may change rapidly. Moreover, it is worth noting that the proposed method may also require additional computational resources to account for the partition decision information from the previous frame, which may impact its real-time performance in certain scenarios. In [3], the paper proposes a motion-compensated EC method for H.265/HEVC. The proposed method refines the MV from the co-located block for motion compensation and merges blocks based on the reliability of these MVs. New MVs are then assigned to the merged blocks. The experimental results demonstrate a significant gain in PSNR and an improvement in visual quality compared to the conventional EC methods. However, any motion-compensated EC method may face challenges in scenarios where the lost data is significant or the MVs are unreliable, which can result in suboptimal concealment performance. Additionally, the proposed method relies on the reliability of the MVs from the co-located block, which may not be accurate or reliable in scenarios when there are fast object movements.

In [36], a trained Generative Adversarial Network (GAN) is exploited for EC. This method trains completion and discriminator networks with the temporally and spatially surrounding pixels as inputs. Thus the generator is used for loss recovery. However, training the GAN network is very time-consuming. It is not applicable since the loss area should be known apriori, which is impossible for most EC scenarios. In [30], the authors propose a convolutional Long Short-Term Memory (LSTM) layer and a simple convolutional layer for optical flow estimation. There are two rows of MBs above and below each slice for three frames which are exploited for recovery purposes. Furthermore, only using optical flow and spatial information cannot be sufficient. Training the network for potential loss locations in real scenarios is necessary, but it limits its practical application. In addition, LSTM has the ability to perform intelligent feature extraction from the data packets. For instance, in [34], the authors demonstrates a deep learning approach for feature extraction from the captured data that can handle limited sample sizes effectively.

In [6], the authors present a homography-based registration approach for the pixels surrounding the corrupted region and their corresponding pixels in the reference frame. This method aims to identify the registration with the least distortion regarding the whole frame and the area around the lossy region. Further, this technique is performed forward and backward, with the remaining points being filled by SEC. It enhances the registration quality when the correlation is sufficient for determining the registration points. However, finding good matching points is impossible for all contents, especially for H.265/HEVC, since the lossy areas are large.

There is a parallelogram partitioning approach for H.265/HEVC in [12]. Corrupted areas are divided into smaller parallelograms, and MV recovery and filling are applied. Moreover, it is used for the various sizes and angles of parallelograms. Despite this algorithm's high quality, it is more complex since it should iterate for different angles and sizes. In [27], the two spatially adjacent MBs of each lost 8 × 8 block are used to identify the most correlated MV in the previous frame. Then, if any block with a sufficiently close motion trajectory is found in the few previous frames, its MV is used to recover the lost 8 × 8 block. A weighted average of two MVs of spatially adjacent MBs is calculated if the MV is not found. Then, one of these three MVs is selected, which produces the least boundary distortion. The algorithm is not appropriate for H.265/HEVC due to its reliance on nearby MVs that are not always available.

3 Problem statement

The EC methods attempt to improve the video quality by exploiting the most correlated available information for the corrupted area and replacing the damaged CU. The lossy area in H.265/HEVC is relatively large, and the available Prediction Block (PB) is far from each damaged block in the temporal domain. Specifically, the data is encapsulated in an integer number of Coding Tree Units (CTUs), and each CTU can be up to 64×64 pixels in size. Thus, packet loss can result in losing a significant amount of frame data. A CU of H.265/HEVC varies from LCU to Smallest Coding Unit (SCU). In LCUs and SCUs, the sizes are 64×64 pixels and 8×8 pixels, respectively [7]. A lost LCU is split into several SCUs for further processing.

3.1 Hierarchical MV clustering using divisive approach

The divisive hierarchical clustering algorithm is a top-down technique. First, the data points are associated with one cluster, and then it branches into smaller sets as it goes down the hierarchy. Finally, each cluster splits into two new clusters iteratively until the stopping criterion is satisfied. There are ways of breaking each cluster [39]. The proposed algorithm exploits the SSE and minimizes it at each stage of cluster analysis to avoid the increasing order of complexity [26]. In addition, there is a limited distance among the MVs since they are highly correlated in a region. Furthermore, there are $O(2^n)$ ways of dividing the clusters. The variable n represents the number of data points or observations in the dataset. In our proposed model, n refers specifically to the temporally neighboring MVs in the set S. Also, a fixed threshold determines the stopping criterion, which is determined by Θ_{SSE}. The smaller the threshold, the greater the complexity and accuracy [29].

To apply this method, we assume the set S such that $\mathbf{X}_l \in S$ where $l \in N$ and N is the number of temporally neighboring MVs. Each cluster is represented by $C_{i,j}$ and is a member of the set L, which contains all clusters. The variable i denotes the level of the cluster while j specifies its position within that level. We will proceed with the following steps to apply this method.

1. Build the unique set S, which is $C_{0,0}$.
2. Partition each cluster into two distinct clusters.
3. Reiterate step 2 until the SSE for each cluster $R_{i,j}$ becomes smaller than a predefined threshold.

Algorithm 1 describes the divisive clustering method in detail. In this algorithm, Θ_{SSE} is the threshold for the SSE of each cluster $C_{i,j}$ and $stop_{flag}$ sets the stopping criterion for the algorithm. Initially, the set L consists only of the set S. At each iteration of the while loop, the members of all $C_{i,j} \in L$ are used to measure the SSE, denoted by $R_{i,j}$. If the SSE value is greater than or equal to Θ_{SSE}, then $stop_{flag}$ is set to zero, indicating that $C_{i,j}$ must be divided, and the algorithm should proceed to the next iteration. Additionally, $C_{i,j}$ should be

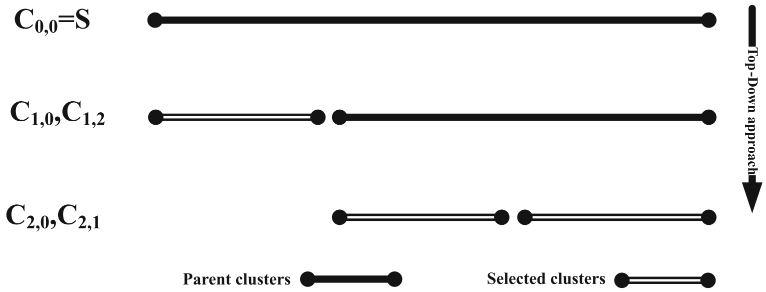


Fig. 1 The divisive hierarchical clustering algorithm is visualized. Clusters are split into two separate clusters at each level if the SSE exceeds the threshold value Θ_{SSE}. The l^{th} data point is denoted as $\mathbf{X}_l \in C_{i,j}$. Initially, the parent cluster is defined as the set S. As the algorithm progresses, new clusters are formed by splitting existing clusters. Each cluster is labeled with a subscripted index, such as $C_{i,j}$, where i denotes the level of the cluster in the hierarchy and j denotes its position within that level

removed from L after being divided. This iterative process continues until the SSE value of all members of L is less than Θ_{SSE}. Finally, L contains all the clusters.

The divisive hierarchical clustering method is illustrated in Fig. 1. At each level, the algorithm splits a cluster into two subclusters if the SSE is greater than a predefined threshold Θ_{SSE}.

Algorithm 1 Divisive hierarchical clustering algorithm

```
Input:
$S = \{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \ldots, \mathbf{X}_N\}$, $\Theta_{SSE}$
Output:
The dendogram for the set $S$
Initialization:
    $i \leftarrow 0$
    $j \leftarrow 0$
    $C_{i,j} \leftarrow S$
    $L \leftarrow \{C_{i,j}\}$
    $k \leftarrow 0$
    $stop_{flag} \leftarrow 0$
while $stop_{flag} = 0$ do
    $k \leftarrow 0$
    $j \leftarrow 0$
    for each $C_{i,j} \in L$ do
    $stop_{flag} \leftarrow 1$
    $R_{i,j} \leftarrow \sum_{\mathbf{X}_l \in C_{i,j}} (\mathbf{X}_l - \overline{\mathbf{X}})^2$
    if $R_{i,j} > \Theta_{SSE}$ then
        $stop_{flag} \leftarrow 0$
        $k \leftarrow k + 1$
        Split $C_{i,j}$ into $C_{i+1,k-1}$ and $C_{i+1,k}$
        Remove $C_{i,j}$ from $L$
        Add $C_{i+1,k-1}$ and $C_{i+1,k}$ to $L$
    end if
    $j \leftarrow j + 1$
  end for
    $i \leftarrow i + 1$
end while
```

3.2 Motion vector modeling using PCA

A key challenge for EC algorithms is to accurately model the MVs of erroneous regions in a video frame. EC methods attempt to infer motion patterns to achieve better performance. The proposed approach develops a motion tracking model for the corrupted block. Also, it creates a model for each of these corrupted blocks by utilizing motion data clustering to improve the accuracy and efficiency of estimating the missing MVs. We define MV of a block as a 2D vector **MV**, which is denoted in (1):

$$\mathbf{MV} = (MV_x, MV_y) \tag{1}$$

where MV_x and MV_y are x and y components, respectively, of the considered missing MV.

Furthermore, the definition of a candidate region within a frame suggests more accurate predictions for MV recovery. Thoroughly determining this region requires considering the characteristic properties of the ellipsoid. In this paper, we propose to embed each MV into its specific cluster and investigate these clusters to find the most appropriate region for correctly predicting the lost motion information of the corrupted blocks. To achieve this, we use PCA, a statistical technique widely used for dimensionality reduction, to compute linearly uncorrelated Principal Components (PCs) from the input training data [2, 9, 19]. Using the parameters obtained from PCA, we generate new data points from the modeled ellipsoids, which allows us to increase the size of the training data set without collecting new data. This approach is particularly useful in applications where the cost of collecting new data is high or where training data availability is limited.

Assume that the data are distributed in a 2D plane. Figure 2 visualizes the iso-contour representation of the ellipses in the 5th frame of the 'Stefan' sequence for the H.265/HEVC coding scenario. This experiment illustrates clusters of MVs in the third Picture Order Count (POC). The MVs are chosen from the set S, which is made up of the adjacent motion information of the considered corrupted block. This experiment shows that the temporally neighboring motion information can be effectively classified into multiple ellipses. Figure 2 (c) depicts such scenarios where a rotated ellipse bounds two MVs.

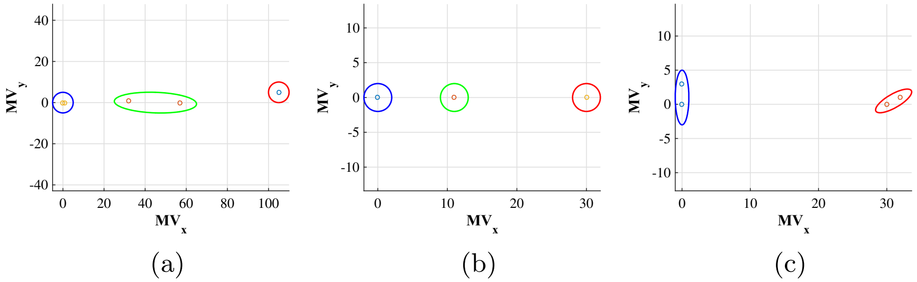


Fig. 2 Iso-contour visualization of the ellipses. This figure represents the MV clustering in the HEVC standard for the fifth frame of the 'Stefan' sequence. In addition, this experiment shows the ellipsoidal clustering regions in the third POC. (a) Region: $P_x \in [312, 336]$, $P_y \in [264, 288]$ (b) Region: $P_x \in [88, 112]$, $P_y \in [248, 272]$ (c) Region: $P_x \in [304, 328]$, $P_y \in [192, 216]$

4 Proposed algorithm

The Generative PCA-based Hierarchical (GPH) method is an innovative clustering EC approach that utilizes the PCA technique to specify the MVs for each cluster. The main objective of MV clustering is generating the correlated MVs inside a cluster. Moreover, the commonly used criterion for clustering is to minimize the Euclidean distance between data points. In addition, the object's MVs are directionally correlated to each other in a video scene that can be intuitively interpreted as elongated clusters in a frame's region. The MVs reflect the motion of the camera and the objects in the scene. We argue that MVs are correlated depending on the type and extent of motion in the video. When there is a global motion (such as panning or zooming), all MVs have a similar direction and magnitude. When there is a local motion (such as an object moving independently), some MVs have a different direction and magnitude.

Furthermore, the proposed algorithm enhances accuracy and speed due to innovative ellipsoidal motion data modeling. It derives more correlated MV candidates for the corrupted area. Also, to avoid any confusion, we would like to emphasize that our proposed video EC technique does not address consecutive frame losses, and that the use of an IDR frame at the start of a GOP is a potential solution to recover from such losses.

Also, this classification technique allows us to restrict the candidate region in the directions with more limited diversification. In other words, the candidate region is related to the motion disparity and variance of the correctly received MVs. GPH proposes a comprehensive methodology to achieve the MV candidates utilizing the axis-aligned ellipse. Also, the covariance of the MVs determines the ellipse's angle.

On the other hand, the ellipse magnitudes depend on the variance of the MVs, i.e., the larger the variance, the more extended the ellipse's axis. Moreover, the covariance of the MVs defines the angle of the ellipse. Hence, it can be a nonzero number since the MVs are highly correlated to each other. If the covariance is zero, the corresponding MVs are uncorrelated, and the ellipse is axis-aligned in this case. Figure 3 depicts the ellipse with a major axis $2R_x$ and a minor axis $2R_y$, which holds all the correctly received motion data.

The axis directions are where motion data varies the most, characterized by a linearly transformed matrix called the covariance matrix. Furthermore, the ellipse axes are eigenvectors of the covariance matrix, and the length of each axis is proportional to the square root of the corresponding eigenvalue, i.e., $\sqrt{\lambda_1}$ and $\sqrt{\lambda_2}$. These eigenvalues indicate the amount of variation in the motion data along the corresponding eigenvectors. Moreover, each cluster can be modeled as a confidence ellipsoid shape with a 95% confidence region, where the major and minor semi-axis lengths are given by $R_x = \sigma_x\sqrt{5.991\lambda_1}$ and $R_y = \sigma_y\sqrt{5.991\lambda_2}$, respectively, where σ_x and σ_y are the standard deviations of the motion data along the first and second PCs, and λ_1 and λ_2 are the largest and second-largest eigenvalues of the covariance matrix, respectively.

One can measure the orientation of the ellipse using the largest eigenvector towards the x-axis:

$$\alpha = \tan^{-1}\frac{\mathbf{u}_1(y)}{\mathbf{u}_1(x)} \tag{2}$$

where $\mathbf{u}_1$ is the largest eigenvector of the covariance matrix and $\alpha \in [0, \pi]$. Accordingly, the general equation of the ellipse is:

$$\frac{((x-x_d)cos(\alpha)+(y-y_d)sin(\alpha))^2}{R_x^2}+\frac{((x-x_d)sin(\alpha)-(y-y_d)cos(\alpha))^2}{R_y^2}=s \tag{3}$$

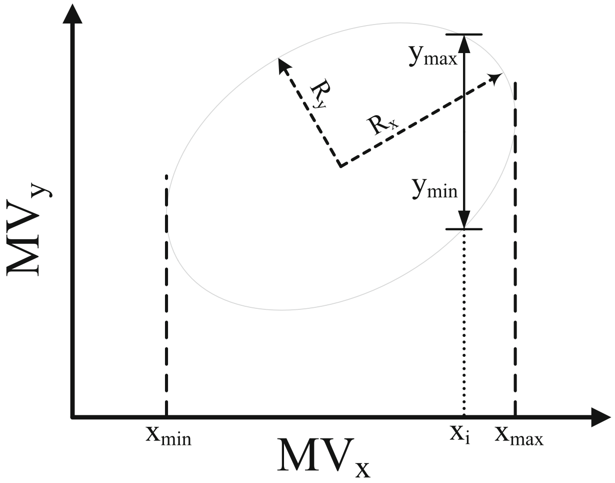


Fig. 3 This illustration shows a 2D ellipse for the MV points of a corresponding erroneous block. It displays the solution intervals on both the x and y axes. For each value of x_i, there exists a corresponding value of y_i that falls within the interval $[y_{min}, y_{max}]$

where (x_d, y_d) is the displacement. Also, R_x and R_y are semi-axes in x and y directions, respectively. To simplify the (3), assume that a, b, and c are defined as follows:

$$a = \frac{cos(\alpha)^2}{R_x^2} + \frac{sin(\alpha)^2}{R_y^2} \tag{4}$$

$$b = (\frac{1}{R_x^2} - \frac{1}{R_y^2})sin(2\alpha) \tag{5}$$

$$c = \frac{sin(\alpha)^2}{R_x^2} + \frac{cos(\alpha)^2}{R_y^2} \tag{6}$$

Hence, the ellipse equation in (3) is reformulated as:

$$a(y - y_d)^2 + (y - y_d)(x - x_d)b + (x - x_d)^2c - s = 0 \tag{7}$$

where a, b and c are the coefficients of the quadratic (7). It has a closed-form solution. It is categorized into four cases that affect the major and minor axes of the ellipse and its shape:

1. $R_x \neq 0$, $R_y \neq 0$, $0 \leq \alpha \leq \pi$: In this case, the solution written in terms of its coefficients is:

$$y = \frac{-b' \pm \sqrt{b'^2 - 4ac'}}{2a} \tag{8}$$

where b' and c' are:

$$b' = -2ay_d + (x - x_d)b \tag{9}$$

$$c' = ay_d^2 - y_d(x - x_d)b + (x - x_d)^2c - s \tag{10}$$

Figure 3 illustrates the solution intervals for the x and y axes. $x_i \in [x_{min}, x_{max}]$ is selected based on the quarter pixel resolution for the entire video sequence since higher resolution can better provide accuracy. Hence, b' and c' are determined after obtaining x_i using (9) and (10), and then $y_i \in [y_{min}, y_{max}]$ is specified by applying (8).

2. $R_x = 0, R_y \neq 0, \alpha = k\pi \pm \frac{\pi}{2}$: In the present case, the ellipse transforms into a vertical segment limited between y_{max} and y_{min}. It can commonly occur if there is a fast vertical camera movement in a scene or the objects shift vertically in the video frames. It is depicted in Fig. 4a.
3. $R_x \neq 0, R_y = 0, \alpha = k\pi$: This case is illustrated in Fig. 4b. The MVs only vary in the x-axis, which explains the nonuniformity of motion in the x-direction. It can also be the result of the object's displacement or camera movement only in the x-axis.
4. $R_x = 0, R_y = 0$: This case indicates that the temporally neighboring MVs are uniform around the erroneous region. Figure 4c depicts this scenario where MV_{x_0} and MV_{y_0} are the x and y components, respectively, of the correctly received MV.

Furthermore, this paper recommends two methods to determine the relevant cluster for extracting the MV candidates: 1)Maximum Likelihood (ML) and 2)Nearest Mean (NM). The former selects the cluster with the highest probability density function (PDF) value, while the latter finds the cluster with a mean value more closely related to the average of the temporally neighboring correctly received MVs. Both of these approaches are used to select a single cluster from the clusters obtained from the clustering process. The ML approach is a standard method used in many clustering algorithms due to its ease of implementation. On the other hand, the NM approach is suitable for applications where there is a strong correlation between the data, such as neighboring MVs in a local region of a video frame. However, in the case of H.265/HEVC, the erroneous region is often large, and spatially neighboring MVs are not available, which results in a decrease in performance of the NM approach compared to the ML approach.

Figure 5 demonstrates the extraction of MV candidates utilizing the GPH method. The ellipse equation is estimated for each cluster exploiting the PCA approach and the correctly received MVs, i.e., the set S. Furthermore, the resolution of x is Δx, with a minimum value of 0.25, and $n_{\Delta x}$ represents the number of iterations required to compute solutions for $x_i \in [x_{min}, x_{max}]$. Additionally, the y_i component of the MV candidates are derived by applying (7) where $y_i \in [y_{min}, y_{max}]$. GPH attempts to classify the MVs field and estimates the cluster for MV recovery. The clusters with ellipsoidal shapes are in the direction with more disparity represented in two radii extremes, the most extended set of MVs. The GPH technique exploits the selected cluster for MV candidate recovery, refined utilizing the Boundary Matching

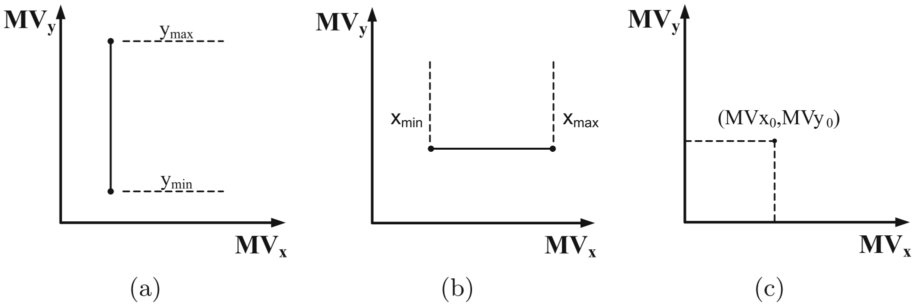


Fig. 4 The special cases of the cluster. (a) $R_x = 0, R_y \neq 0, \alpha = k\pi \pm \frac{\pi}{2}$ (b) $R_x \neq 0, R_y = 0, \alpha = k\pi$ (c) $R_x = 0, R_y = 0$

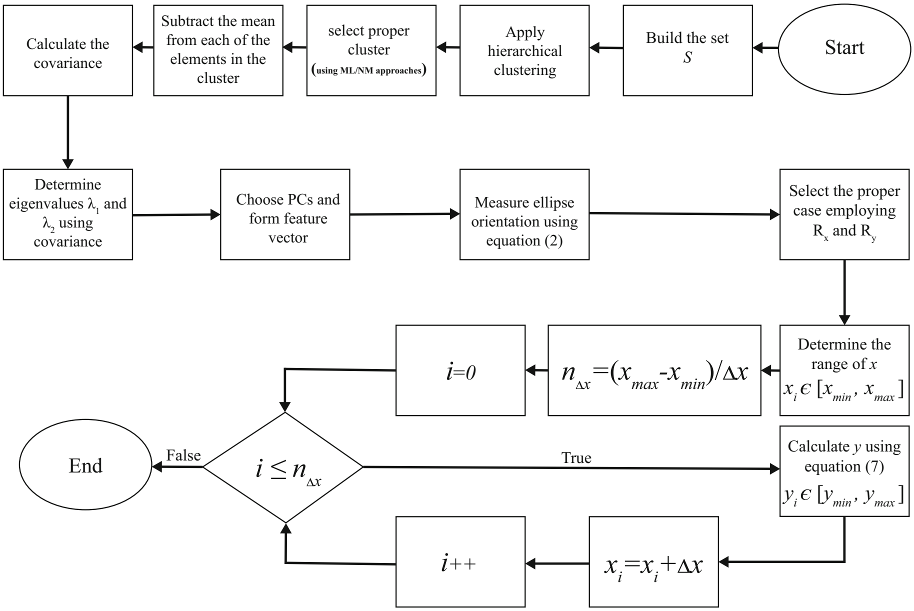

Start
Build the set S
Apply hierarchical clustering
select proper cluster (using ML/NM approaches)
Subtract the mean from each of the elements in the cluster
Calculate the covariance
Determine eigenvalues λ1 and λ2 using covariance
Choose PCs and form feature vector
Measure ellipse orientation using equation (2)
Select the proper case employing Rx and Ry
Determine the range of x
xi ϵ [xmin, xmax]
nΔx=(xmax-xmin)/Δx
i=0
i ≤ nΔx
True
False
End
Calculate y using equation (7)
yi ϵ [ymin, ymax]
xi=xi+Δx
i++

Fig. 5 Diagram of the proposed algorithm. After building the set S, the mean of the set is subtracted from each element. Next, the set's covariance is measured, and the eigenvalues are extracted, which results in finding the PCs and the ellipse orientation. Based on the R_x and R_y, one of the four cases is determined, and the range of y is measured, leading to obtaining the MV candidates

Algorithm (BMA) procedure. The proposed scheme can be broken down into the following steps:

- Error detection: The first step involves decoding the received video stream and checking for errors. Any errors found are flagged and sent to the EC module for further processing.
- Data extraction: In this step, the set S is built using available temporally neighboring MVs from the previous frame. These MVs are extracted from co-located LCU in the previous frame and its spatially neighboring MVs are added to the set S. These data points are aggregated, and also zero and average MVs are added to the set.
- Data clustering: The data points are associated with a cluster in this step, and the cluster branches into smaller sets as it goes down the hierarchy. The process iteratively splits each set into smaller ones until a stopping criterion is satisfied. ML or NM algorithms are then used to determine the relevant cluster.
- Model training: In this step, the PCA procedure is performed, and the PCs are calculated using existing data in the chosen cluster. To do this, the mean is first subtracted from each element in the cluster. Then, the covariance is calculated, and the eigenvalues are derived.
- Data generation: Once the model is trained, it can be used to generate new data points. This involves deriving the ellipse orientation using the trained model's parameters, as well as determining the major-axis and minor-axis lengths of the ellipse. Then, the x and y components of the data points, which are the MV candidates, are calculated.
- Concealment: In this final step, the erroneous blocks are concealed using the MVs derived in the data generation step, utilizing the BMA approach.

5 Experimental results

5.1 Experimental environment

The simulations for EC were carried out on an Intel core i5 Dual-Core, 2.6 GHz processor with 8GB RAM of a MacBook laptop. The experiments were conducted using the H.265/HEVC reference software HM 16.20 [8]. The video sequence was encoded using the main profile with an input bit depth of 8. The maximum width and height of coding units were limited to 64 pixels, while the maximum partition depth was set at 4. The decoding refresh type was not utilized, and a Group of Pictures (GOP) size of 4 was selected. Motion estimation was performed using the TZ search method with a search range of 64.

In this paper, we use a video dataset from Xiph.org [25]. The dataset consists of various video sequences that covers a wide range of content. We tested the proposed method on five video sequences: ‘Shields,’ ‘Ducks Take-Off (DTO),’ ‘Park run (PR),’ and ‘Johnny (JO),’. All sequences have a resolution of 1280 × 720 pixels and use the 720 progressive (720p) HD signal format. We use 4:2:0 color subsampling on these video signal. The frame rates for all the video sequences are 50 Frames per Second (fps), except for ‘Johnny’, which has a frame rate of 60 fps.

In our experimental setup, we specified a specific Packet Loss Rate (PLR) for each simulation and applied a concealment procedure after detecting errors in the decoded video. The video sequences were encoded in Slice Mode, with each slice containing an integer number of CTU. We set the maximum slice size to 1500 bytes, which aligns with the Maximum Transportation Unit (MTU) of the network for a single packet. We tested two PLR values of 10% and 20%, and modeled the transmission channel using the Elliot-Gilbert model. We generated 20 instances of packet loss, with an average burst length of three packets. Moreover, we only considered P-frames and did not include B-frames. The coding structure used for testing the performance is “IPPP", which consists of only one intra-frame (I) after 51 inter-frames (P). There are 150 frames used in each video sequence. Quantization Parameters (QP)s were set to 25 and 35.

We evaluated the performance of the proposed GPH method and its two approaches, GPH_{ML} and GPH_{NM}, on a set of benchmark datasets. The experimental results show that both GPH_{ML} and GPH_{NM} achieve superior performance compared to the state-of-the-art methods in terms of accuracy and computational efficiency. Notably, the two approaches offer robust and efficient solutions to the problem across a range of datasets.

5.2 Performance evaluation

The paper compares the performance and complexity of the proposed algorithm with those of other related EC schemes such as Xu method [37], Lin method [22], and Chang method [3] for H.265/HEVC. In experiments, we use PSNR as an objective metric for quality estimation and evaluate the video frames using Structural Similarity Index Measurement (SSIM), which captures essential perception-based facts or inter-dependent pixel information better than the PSNR [31].

$$PSNR = 10\log_{10}\frac{M^2}{MSE} \tag{11}$$

where M represents the most significant value of the signal, which in our case was 255, and MSE represents the Mean Square Error difference between two video frames.

Tables 2 and 3 show the PSNR and SSIM results for a PLR of 10% and 20%, respectively. These tests were conducted on the entire test sequence. The table demonstrates that the proposed method outperforms the others by more than 4.99 dB and 4.19 dB in PSNR of 10% and 20%, respectively. Additionally, it achieves a total gain of 0.0208 and 0.029 in terms of SSIM for a PLR of 10% and 20%, respectively. Accordingly, the improved performance of the proposed approaches is consistent with our expectation that the clustering hypothesis experiment would outperform the motion-compensated method. Due to the high compression ratio of HEVC, error propagation is entirely severe in the succeeding frames in a video sequence which causes rapid degradation for all EC techniques since even minor error results in a high degradation in the following GOP frames. PSNR gain is consistent in our proposed schemes, and visual quality is superior to the algorithms we compared. Also, we have evaluated the performance of both the ML and NM approaches for selecting a single cluster. Our experimental results demonstrate that the GPH_{ML} approach outperforms the GPH_{NM} approach in terms of both concealment quality and computational complexity. This finding is consistent

Table 2 Performance comparison of H.265/HEVC in terms of PSNR, SSIM, and run time for packet loss rate 10%

QP	video		Xu [37]	Lin [22]	Chang [3]	GPH_{ml}	GPH_{nm}
25	Shields	PSNR(dB)	24.54	22.02	31.40	38.95	33.21
		SSIM	0.9104	0.7744	0.9690	0.9935	0.9714
		Time(s)	0.17	0.16	0.16	0.12	0.12
	DTO	PSNR(dB)	29.53	23.26	30.51	36.02	32.76
		SSIM	0.9518	0.8204	0.9672	0.9872	0.9766
		Time(s)	0.16	0.19	0.14	0.13	0.13
	PR	PSNR(dB)	28.09	27.27	29.03	34.00	30.05
		SSIM	0.9534	0.9589	0.9581	0.9777	0.9680
		Time(s)	0.16	0.19	0.18	0.16	0.13
	JO	PSNR(dB)	35.26	31.26	36.05	38.92	37.79
		SSIM	0.9837	0.9654	0.9843	0.9882	0.9853
		Time(s)	0.05	0.07	0.05	0.05	0.05
35	Shields	PSNR(dB)	23.57	20.15	28.96	37.57	31.46
		SSIM	0.8893	0.7321	0.9620	0.9937	0.9631
		Time(s)	0.05	0.04	0.04	0.04	0.03
	DTO	PSNR(dB)	22.91	21.41	28.88	34.29	32.34
		SSIM	0.8011	0.6630	0.9547	0.9808	0.9732
		Time(s)	0.08	0.09	0.07	0.07	0.07
	PR	PSNR(dB)	26.47	24.88	28.60	32.12	29.65
		SSIM	0.9446	0.8969	0.9442	0.9678	0.9564
		Time(s)	0.08	0.08	0.07	0.07	0.06
	JO	PSNR(dB)	29.26	27.16	34.36	35.80	35.75
		SSIM	0.9474	0.9374	0.9690	0.9863	0.9787
		Time(s)	0.04	0.04	0.03	0.04	0.03
	Average	PSNR(dB)	27.45	24.68	30.97	35.96	32.88
		SSIM	0.9227	0.8436	0.9636	0.9844	0.9716
		Time(s)	0.09	0.11	0.09	0.08	0.08

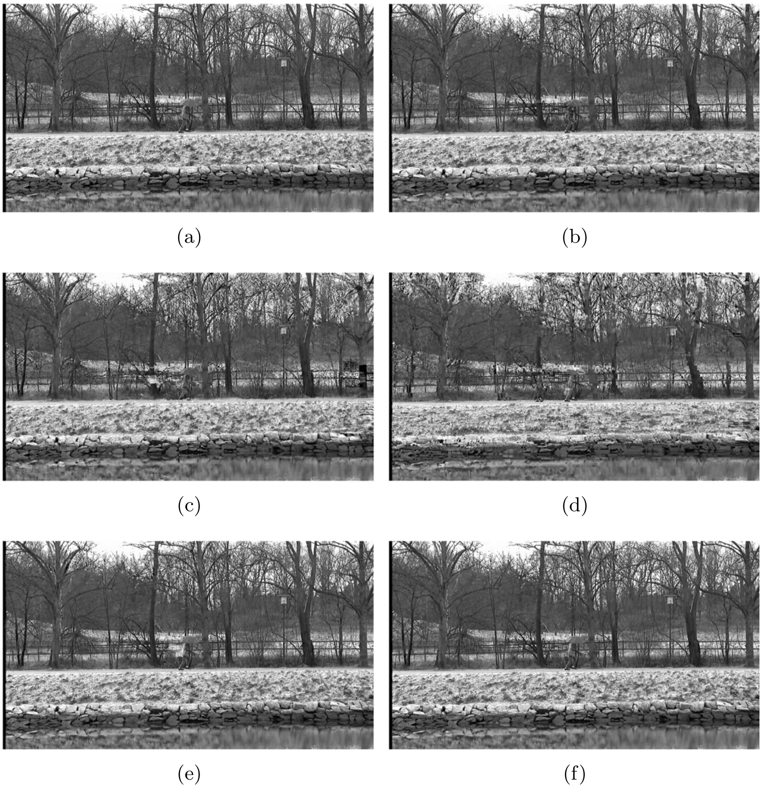
Table 3 performance comparison of H.265/HEVC in terms of PSNR, SSIM, and run time for packet loss rate 20%

QP	video		Xu [37]	Lin [22]	Chang [3]	GPH_{ml}	GPH_{nm}
25	Shields	PSNR(dB)	22.59	19.68	28.97	37.21	30.78
		SSIM	0.8747	0.7020	0.9520	0.9905	0.950
		Time(s)	0.20	0.18	0.19	0.13	0.13
	DTO	PSNR(dB)	23.49	20.00	27.88	32.65	28.10
		SSIM	0.8331	0.6595	0.9407	0.9732	0.9311
		Time(s)	0.20	0.22	0.19	0.15	0.14
	PR	PSNR(dB)	26.38	26.24	27.14	30.14	27.29
		SSIM	0.9234	0.9149	0.9355	0.9500	0.9421
		Time(s)	0.16	0.20	0.18	0.17	0.14
	JO	PSNR(dB)	29.15	28.53	32.10	34.03	33.53
		SSIM	0.9416	0.9367	0.9620	0.9747	0.9638
		Time(s)	0.09	0.16	0.08	0.05	0.05
35	Shields	PSNR(dB)	20.56	18.61	28.30	33.95	30.06
		SSIM	0.7842	0.6334	0.9472	0.9851	0.9490
		Time(s)	0.05	0.04	0.04	0.04	0.03
	DTO	PSNR(dB)	20.55	19.84	26.95	31.57	27.71
		SSIM	0.6995	0.6462	0.9305	0.9627	0.9295
		Time(s)	0.10	0.11	0.10	0.09	0.07
	PR	PSNR(dB)	24.48	23.90	25.87	27.99	26.55
		SSIM	0.8892	0.8631	0.8956	0.9380	0.9302
		Time(s)	0.08	0.12	0.07	0.07	0.06
	JO	PSNR(dB)	28.84	25.48	30.55	33.75	30.82
		SSIM	0.9584	0.9330	0.9549	0.9761	0.9582
		Time(s)	0.06	0.05	0.03	0.04	0.03
	Average	PSNR(dB)	24.50	22.79	28.47	32.66	29.35
		SSIM	0.8630	0.7861	0.9398	0.9688	0.9442
		Time(s)	0.12	0.13	0.11	0.09	0.08

with the fact that in the erroneous region of a frame in H.265/HEVC, the availability of spatially neighboring MVs is limited, and the correlation between them decreases dramatically. Therefore, we recommend using the GPH_{ML} approach in our proposed algorithm or in similar applications.

Moreover, the proposed method requires fewer computations for video frames with more uniform motion vectors. For example, in Table 2, the 'Shields' sequence requires about 30% fewer computations for the proposed algorithms than the other compared ones. This video sequence depicts a man walking uniformly in front of a wall. The results show that the proposed methods have less computational complexity than the other compared methods.

Figure 6 illustrates a visual comparison of the 50th frame of the 'Park Run' sequence using the algorithms by Xu [37], Lin [22], and Chang [3]. In this experiment, QP is 35 and PLR is set to be 20%. The frames in Fig. 6b, c, and d are concealed using Xu's, Lin's, and Chang's algorithms, respectively. Figure 6a shows the original frame of the 50th frame of the 'Park Run' sequence. Figure 6e and f show the frames recovered using the proposed methods, i.e.,



(a) (b) (c) (d) (e) (f)

Fig. 6 Error concealment results of 50th frame over ‘Park run’ sequence at the PLR of 20% and QP=35. (a)Original frame (b)Xu approach [37] (c)Lin approach [22] (d)Chang approach [3] (e)The proposed method by $\mathrm{GPH_{nm}}$ approach. (f)The proposed method by $\mathrm{GPH_{ml}}$ approach

$\mathrm{GPH_{nm}}$ and $\mathrm{GPH_{ml}}$, respectively. These figures show that the proposed methods significantly enhance the visual quality compared to the other algorithms. The video sequence depicts a man running in a park with an umbrella in his hand. The proposed approaches accurately predict the shape of the man. Moreover, there are trees, snow, and water in the background with lots of intricate patterns. The proposed approaches, specifically $\mathrm{GPH_{ml}}$, preserve the details more accurately than the other compared methods.

Figure 7 shows the subjective comparison of the 70th frame of the ‘Johnny’ sequence. In this experiment, QP is 25 and PLR is set to be 20%. Figure 7a shows the original frame, which contains a man moving his head and a static background with several details. The other figures, namely Fig. 7b, c, and d, demonstrate the results of using Xu’s [37], Lin’s [22], and Chang’s [3] algorithms to conceal parts of the scene. Errors occurring on moving parts, particularly the left side of the head, greatly affect the visual quality of the scene. This is because sudden movements in these areas result in more non-uniform motions that are challenging for EC methods to recover. Our proposed methods, $\mathrm{GPH_{nm}}$ and $\mathrm{GPH_{ml}}$, demonstrate superior motion information estimation compared to the other methods we tested. This is evidenced by the results in Fig. 7e and f. Overall, our findings demonstrate the superiority of our algorithms over the other compared methods.

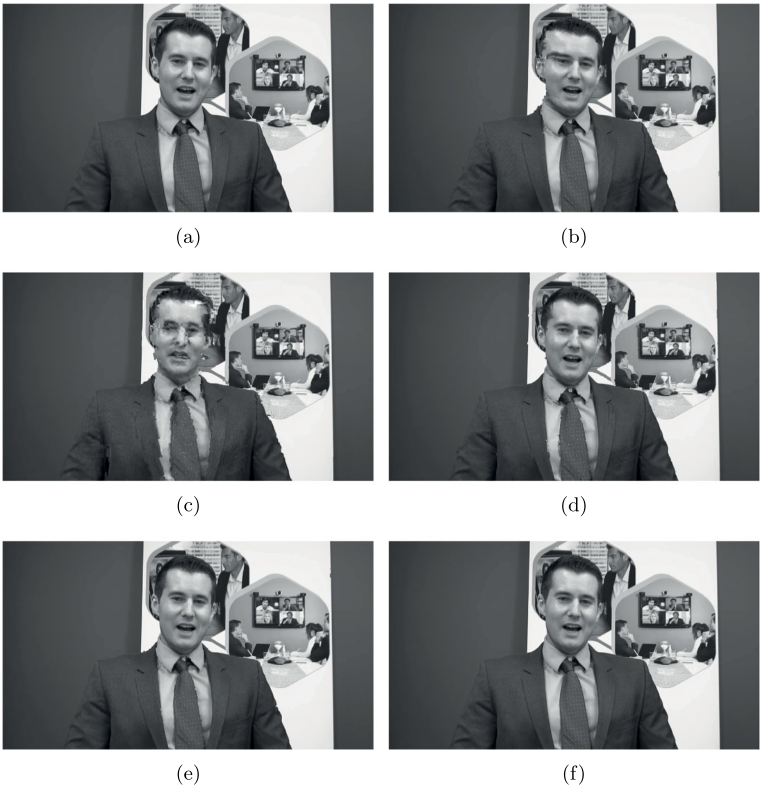


Fig. 7 Error concealment results of 70th frame over ‘Johnny’ sequence at the PLR of 20% and QP=25. (a)Original frame (b)Xu approach [37] (c)Lin approach [22] (d)Chang approach [3] (e)The proposed method by GPH$_{\mathrm{nm}}$ approach. (f)The proposed method by GPH$_{\mathrm{ml}}$ approach

We attribute this exceptional performance to two reasons that mitigate the obstacles:

1. The EC algorithms, such as the Xu method [37], employ limited motion information, and hence they cannot predict the missed MVs considerably.
2. In the absence of motion information in adjacent blocks, EC techniques employing only neighboring MVs cannot provide a good estimation of the MV.

Furthermore, the moving facial region of the frame does not provide precise PU/TU information at the slice level. Thus, applying only the neighboring MVs can result in successive slice errors and blocking artifacts as motion increases. In our study, we used the “IPPP” coding structure, which means that each frame (except the first one) is dependent on the previous frames for reconstruction. This can make the video more vulnerable to the effects of packet loss. However, we did not introduce any I-frames to the coding structure to reduce the dependence on previous frames. As a result, our results may represent the worst-case scenario in an error-prone environment. Furthermore, future studies could explore the use of I-frames in the coding structure to improve the video quality in such environments.

5.3 Computational complexity analysis

Tables 2 and 3 compare the computational complexity of decoding video frames concealed in different EC techniques. They show that the EC algorithms demand post-processing procedures that increase the overall decoding time. The required decoding time for the proposed algorithms depends on the degree of MV uniformity around the corrupted region, i.e., the smaller the MV range, the fewer computations are needed. In contrast, other EC algorithms usually consume the same amount of decoding time in different video sequences. In the proposed GPH method, the first PC determines the number of computations and the special cases of the quadratic (7) which defines the ellipse transformation. It is illustrated in Fig. 4. Consequently, these conditions automatically reduce the computational complexity based on the local information of the MV uniformity.

6 Conclusion

This paper proposes a novel approach for lost MV recovery in H.265/HEVC using a PCA-based hierarchical clustering technique to address the challenge of dealing with large lossy areas where the available PBs are far from the damaged blocks. Our study utilizes divisive hierarchical clustering to classify temporally and spatially adjacent MVs. Through several experiments, we employed the PCA method to estimate the magnitude and direction of each cluster. Additionally, our proposed procedure generates an ellipsoidal model that limits the diversification of motion data for loss concealment.

Furthermore, our proposed method introduces two strategies for selecting the best cluster: maximum likelihood and nearest mean. The former utilizes the PDF, while the latter relies on the average MV for each cluster. Other similar approaches could also be studied to select the best cluster. The proposed algorithm generates related motion data based on the variance of the selected cluster, making it suitable for processing different video sequences as its computational complexity is determined by motion diversity and variance. Thus, our method provides a new framework for video EC, and further research in this area is needed.

The experimental results demonstrate that the proposed method outperforms other tested EC algorithms in both PSNR and SSIM. Specifically, for PLR of 10%, the proposed method improves PSNR by up to 16.11% and SSIM by up to 2.1586% on average. Similarly, for PLR of 20%, the proposed method improves PSNR by up to 14.72% and SSIM by up to 3.0857% on average, compared to the other EC algorithms.

Our study proposes a PCA-based approach for error concealment in H.265/HEVC, leveraging MV clustering to generate accurate MVs for missing regions of damaged frames. While our findings are promising, there is still much to explore in this area. Future research could focus on evaluating other effective clustering methods for different types of video frames beyond the Euclidean distance measure used in this paper, potentially leading to increased homogeneity within clusters. Additionally, investigating alternative strategies for proper cluster selection could also be valuable. Finally, exploring properties of generative models, such as hidden Markov models and mixture of Gaussians, could help identify more suitable approaches for motion estimation.

In addition, our proposed method effectively preserves the structure of moving objects and edges in concealed regions. We also conducted experiments to determine the decoding time for our approach and compared it with other related techniques. The results indicate that our method outperforms the other EC schemes, achieving high video quality while reducing computational complexity.

Acknowledgements The authors would like to acknowledge the funding support of Babol Noshirvani University of Technology through grant program No. BNUT/389059/1401.

Data Availability Statement All data generated or analyzed during this study are included in this published article, and its supplementary information files are available in the Zenodo repository, 10.5281/zenodo.7756564.

Declarations

Funding and Conflicts of interests/Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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(22):31733–31755
2. Balzano L, Chi Y, Lu YM (2018) Streaming PCA and Subspace Tracking: The Missing Data Case. Proc IEEE 106(8):1293–1310
3. Chang YL, Reznik YA, Chen Z, Cosman PC (2013) Motion Compensated Error Concealment for HEVC Based on Block-Merging and Residual Energy. In 20th International Packet Video Workshop, pp 1–6
4. Chen Y, Wang H, Wu H et al (2018) A video error concealment method using data hiding based on compressed sensing over lossy channel. Telecommun Syst 68(2):337–349
5. Choe G, Nam C, Chu C (2018) An effective temporal error concealment in H.264 video sequences based on scene change detection-PCA model. Multimedia Tools and Applications 77(24):31953–31967
6. Chung B, Yim C (2020) Bi-Sequential Video Error Concealment Method Using Adaptive Homography-Based Registration. IEEE Trans Circuits Syst Video Technol 30(6):1535–1549
7. Coding HEV (2021) High Efficiency Video Coding: Recommendation ITU-T H.265. International Standard ISO/IEC 23008-23002. https://handle.itu.int/11.1002/1000/14660 Accessed 7 March 2023
8. Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute(HHI). https://hevc.hhi.fraunhofer.de/ Accessed 18 Mar 2023
9. Fujiwara T, Chou JK, Shilpika S, Xu P, Ren L, Ma KL (2020) An Incremental Dimensionality Reduction Method for Visualizing Streaming Multidimensional Data. IEEE Transactions on Visualization and Computer Graphics 26(1):418–428
10. Gutub A (2022) Boosting image watermarking authenticity spreading secrecy from counting-based secret-sharing. CAAI Trans. on Intelligence Technology
11. Hameed A, Dai R, Balas B (2016) A Decision-Tree-Based Perceptual Video Quality Prediction Model and Its Application in FEC for Wireless Multimedia Communications. IEEE Trans Multimed 18(4):764–774
12. Hojati S, Kazemi M, Moallem P (2020) Error concealment with parallelogram partitioning of the lost area. Multimed Tools Appl 79(11):7449–7469
13. Hsia SC, Hsiao CH (2016) Fast-efficient shape error concealment technique based on block classification. IET Image Process. 10(10):693–700
14. Huang Z, Cai Q, Xiao X (2018) A Video Data Recovery Algorithm in Wireless Communication Networks. In 2018 IEEE 18th Int Conf Commun Technol (ICCT), pp 727–731
15. Kazemi M (2021) In favor of fully intra coding for HEVC video transmission over lossy channels. Signal, Image and Video Processing 15(1):165–173
16. Kazemi M, Ghanbari M, Shirmohammadi S (2020) The Performance of Quality Metrics in Assessing Error-Concealed Video Quality. IEEE Transactions on Image Processing 29:5937–5952
17. 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(8):12685–12730
18. Korhonen J (2018) Study of the subjective visibility of packet loss artifacts in decoded video sequences. IEEE Trans. Broadcast 64(2):354–366
19. Lei Y (2022) Research on microvideo character perception and recognition based on target detection technology. J Comput Cog Eng 1(2):83–87

20. Li ZN, Drew MS, Liu J (2021) Multimedia Over Wireless and Mobile Networks. Fundamentals of Multimedia, pp 627–669
21. Li Y, Chen R (2017) Motion vector recovery for video error concealment based on the plane fitting. Multimed. Tools Appl. 76(13):14993–15006
22. Lin TL, Yang NC, Syu RH, Liao CC, Tsai WL (2013) Error concealment algorithm for HEVC coded video using block partition decisions. In 2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013), pp 1–5
23. Lin T, Wei X, Su T, Chiang Y (2018) Novel pixel recovery method based on motion vector disparity and compensation difference. IEEE Access 6:44362–44375
24. Ma R, Li T, Bo D, Wu Q, An P (2020) Error sensitivity model based on spatial and temporal features. Multimed Tools Appl 79(43–44):31913–31930
25. Montgomery C (2023) Xiph.org. Derf's Test Media Collection. Available at: https://media.xiph.org/video/derf/. Accessed 7 March 2023
26. Murtagh F, Contreras P (2017) Algorithms for hierarchical clustering: an overview, II. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 7(6):e1219
27. Nam C, Chu C, Kim T, Han S (2020) A novel motion recovery using temporal and spatial correlation for fast temporal error concealment over H.264 video sequences. Multimedia Tools and Applications 79(1):1221–1240
28. Radmehr A, Ghasemi A (2016) Error concealment via particle filter by Gaussian mixture modeling of motion vectors for H.264/AVC. Signal, Image, and Video Processing 10(2):311–318
29. Ros F, Guillaume S (2019) A hierarchical clustering algorithm and an improvement of the single linkage criterion to deal with noise. Expert Syst Appl 128:96–108
30. Sankisa A, Punjabi A, Katsaggelos AK (2018) Video Error Concealment Using Deep Neural Networks. In 2018 25th IEEE International Conference on Image Processing (ICIP), pp 380–384
31. Sara U, Akter M, Uddin MS (2019) Image Quality Assessment through FSIM, SSIM, MSE and PSNR-A Comparative Study. J Comput Commun 7(3):8–18
32. Tiwari V, Bhatnagar C (2021) A survey of recent work on video summarization: approaches and techniques. Multimed Tools Appl 80(18):27187–27221
33. Usman M, He X, Xu M, Lam KM (2015) Survey of Error Concealment techniques: Research directions and open issues. In 2015 Picture Coding Symposium (PCS), pp 233–238
34. Wani A, Khaliq R (2021) SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL). CAAI Trans. on Intelligence Technology 6(3):281–290
35. Wu J, Cheng B, Wang M, Chen J (2017) Priority-Aware FEC Coding for High-Definition Mobile Video Delivery Using TCP. IEEE Trans Mob Comput 16(4):1090–1106
36. Xiang C, Xu J, Yan C, Peng Q, Wu X (2019) Generative adversarial networks based error concealment for low resolution video. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp 1827–1831
37. Xu J, Jiang W, Yan C, Peng Q, Wu X (2018) A Novel Weighted Boundary Matching Error Concealment Schema for HEVC. In 25th IEEE International Conference on Image Processing (ICIP), pp 3294–3298
38. Zhang B, Cosman P, Milstein LB (2019) Energy Optimization For Wireless Video Transmission Employing Hybrid ARQ. IEEE Trans. Veh Technol 68(6):5606–5617
39. Zhou S, Xu Z, Liu F (2017) Method for Determining the Optimal Number of Clusters Based on Agglomerative Hierarchical Clustering. IEEE Trans Neural Netw Learn Syst 28(12):3007–3017

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