Masking of temporal activity for video quality control, measurement and assessment

Ali Akbar Siddique¹, M Tahir Qadr² and Zia Mohy-Ud-Din³

Abstract
Every video stream possesses temporal redundancy based on the amount of motion presenting in it. An ample amount of motion in a video sequence may cause distorting artifacts, and in order to avoid them, there is a possibility to mask the motion or temporal activity that is not noticeable to a human eye in real time. The artifacts such as blockiness and blurriness are instigated in the video sequence as soon as it is subjected to the process of compression, and they tend to become more and more intense with the increase in temporal activity. In this paper, an algorithm is proposed to mask the temporal activity using temporal masking coefficient (q) that is unnoticeable by a human eye to bring down the distortion levels. It is possible to adjust the quality of the video sequence by varying the q parameter and thus controlling its overall quality index. Frames are extracted from the video sequence, and displacement or motion vectors are also calculated from the consecutive frames using a bi-directional block matching algorithm. These motion vectors are used to estimate the quantity of motion present between consecutive frames of the same scene. Video sequences used for this purpose are basically H.264 format. Temporal masking is performed on a video sequence with and without the implementation of motion vector. Structural similarity index and peak signal-to-noise ratio are the quality measurement tools used to assess the performance of the proposed algorithm. A bit rate of 1.2% was saved by implementing proposed algorithm at q = 1 in contrast to the standard H.264/Advanced Video Coding.

Keywords
Temporal masking, motion vector estimation, video compression, video quality measurement and assessment, structural similarity index, peak signal-to-noise ratio

Date received: 5 November 2019; accepted: 4 June 2020

Introduction
In recent times, usage of multimedia information has increased exponentially, due to which there is an enormous amount of data available on Internet and not enough bandwidth to transmit it over the wireless channel; due to these limitations in modern communication medium, a compression process is required to reduce the size of a video for an available bandwidth. This may lead to certain disturbing artifacts in a coded video stream. Video quality depends on the amount of motion present in a coded stream, every video after compression possessing disturbing artifacts that degrade its quality dramatically.¹ The compression process is classified into two different categories: lossy and lossless. Both these algorithms are able to remove unnecessary information to reduce the size of an image or video content.² Each person can have a different perception of the distortion depending on the information masked during the process. Motion or temporal masking (TM) can help adjust the relevance-distortion before quantifying its total amount.

Researchers over the years have introduced different algorithms to verify the compression ratio in the coded video sequences.³ Video quality assessment meters are also designed to assess the quality of the video after the compression process. Wang et al.⁴ proposed a video quality metrics using full reference video at the receiver.
end for an efficient video quality assessment tool. High-Efficiency Video Coding (HEVC) standard was also used to reduce the bit rate. By comparing HEVC and Advanced Video Coding (AVC) standards, Tan et al. demonstrated that about 59% bit rate can be saved with same visual quality compared to the conventional AVC standard. Video sequences contain an array of frames, and researchers have used them to predict motion using a present and an upcoming frame. An extensive amount of work has been done on block matching algorithm (BMA) to estimate motion present in the video. Temporal activity is an important feature in video sequence and it is imperative to know its quantity to apply appropriate compression algorithms. Discrete cosine transform (DCT) is also used in the literature to convert an incoming frame into spatial domain from where the information is quantized. Motion vectors (MVs) are required of each dual sequence frame to determine individual frame quality.

In this paper, the process of motion masking is applied on acquired MVs using TM parameter \( q \). Our aim is to reduce the amount of bit rate to transmit the video information over wired or wireless channel. In this regard, MVs are used to reduce the temporal redundancy by introducing the TM parameter \( q \). Masked motion is small and can be optimized. MV and residual image are the two main components required to reconstruct the encoded frame at the receiving end. MVs are coded in such a way that any motion that is greater than or less than the \( q \) parameter is reduced to zero and thus only requires a single bit to encode as discussed in section “Proposed TM parameter \( q \) for motion masking.” The motion within the frames is non-linear and the object can move anywhere along the \( XY \) plane. Due to this reason, it possesses negative values as discussed in section “H.264 video coding model with TM.” By the implementation of the proposed algorithm, the bit rate is reduced up to 13% for the Sky video sequence at \( q = 4 \) without losing its visual quality as discussed in section “Results.” An opinion score (OS) is also included in section “Results,” which we have acquired from the masses by asking them to view the given video sequences and share their valuable response for subjective analysis of the proposed algorithm. Five different video sequences are used for this purpose, and all of them scored more than 50% in OS as given in Table 7, which validates the quality index and further proves the importance of the proposed algorithm. The video data set used in this paper is provided by Seshadrinathan et al.

### H.264 video coding model with TM

A block diagram of an encoder of H.264 video compression model is displayed in Figure 1 in which the TM process is also included for motion reduction. H.264 encoder acquires an input video signal \( s(r,c,n) \), where \( r \) and \( c \) represent the number of rows and columns, and since it is a video sequence, \( n \) represents the frame number. The feedback signal \( s(r,c,n) \), also known as the predicted signal, is subtracted from the input signal and induces an error signal \( e(r,c,n) \) which may contain far less information compared to an original input frame. The error signal is actually a difference or residual image acquired from the difference between feedback signal and an input signal. The error signal is subjected to the process of domain transformation, widely known as DCT, where the residual signal is transformed into the frequency domain to remove the high-frequency components. Only low-frequency components contribute to the process of image or video reconstruction; thus, by introducing the process of quantization, it is possible to effectively remove high-frequency components without compromising video quality. \( e(r,c,n) \) is gained after quantizing the DCT-applied frame, and it is a key component required to reconstruct frame in order to gain MVs; hence, it is encoded using lossless coding into a binary bit-stream which can be transmitted over wireless channel. \( e(r,c,n) \) is also de-quantized and transformed back using inverse discrete cosine transform (IDCT) to gain the original error response \( e(r,c,n) \), but due to quantization it is not fully recovered, but most of its components are unchanged. The error signal \( e(r,c,n) \) along with the motion-compensated frame \( s(r,c,n) \) is used to reconstruct the original frame \( s(r,c,n) \); this signal is the one applied at the input, but due to the process of quantization it is not fully restored and has a high correlation index close to unity. \( s(r,c,n) \) is stored and considered to be a previous frame \( s(r,c,n−1) \).

Both upcoming or new frame and previous (Stored) frame are used to acquire MVs with the help of the BMA. TM is applied on the acquired MVs to gain temporally masked displacement vectors (TMDV) which are used in conjuncture with the stored frame \( s(r,c,n−1) \) to make a frame prediction denoted by \( s(r,c,n−1) \). This signal is also called the temporally masked displacement-compensated frame (TMDCF), and the prediction requires temporally masked displacement-compensated prediction (TMDCP). The whole process repeats itself until the video is completed; the initial frame is transmitted without undergoing motion compensation and thus becomes a reference frame for the next one.

To reconstruct or predict the frame at the decoder end, only two components are required for such purpose; first is the acquired MVs between the present frame \( s(r,c,n) \) and the previous frame \( s(r,c,n−1) \). The error or difference signal \( e(r,c,n) \) is the other component needed for reconstruction. Both these components undergo entropy coding to convert the information into the binary bit-stream to prepare it for transmission or storage. At the decoding end, the whole process is repeated in reverse. The received binary bit-stream is decoded into the two components that were encoded and regains MVs along with the difference frame.
Figure 2(a) and (b) represent both past and present frames. These two are used to acquire the difference frame in Figure 2(c) that can be transmitted. Figure 2(d) displays the DCT-applied image with quantized response, which actually holds the lossy information needed to reconstruct an image at the decoder end. It is a fact that two consecutive frames possess a small amount of temporal activity that can be coded with a simple difference. The DCT-applied image in Figure 2(d) possesses a lot of redundant information, thus saving a huge amount of space. It generates high-frequency components with values close to zero. Quantization is applied on the DCT output, which will further reduce the near-zero values to actual zero. Table 1 represents the quantized macro-block of $8 \times 8$ pixels. All zeros in Table 1 represent the amount of redundant information an image possesses. The quantity of zeros in the DCT-applied image determines the space it requires to be stored.

Low-frequency components are located on the top left part of the $8 \times 8$ macro-block given in Table 1, and the rest of the values are high-frequency components reduced to zero by quantization. Only these components contributed toward the reconstruction of the frame at the decoder end, and the rest of the values are deemed unnecessary and can be easily coded while occupying less memory space in case of storage. They also require less bit rate if they are to be transmitted over wireless channel or uploaded on Internet.

**Bi-directional MV estimation using BMA**

BMA is a process to identify the movement of an object with the two consecutive frames, it is also considered to be the most widely used method of motion detection among the researchers in the field of video coding.\textsuperscript{11,14} A macro-block that holds a certain object in an image will be slightly moved in the next frame due to the object’s motion. It searches a macro-block of the past frame that holds a similarity with the one in the next or
upcoming frame. Bi-directional motion estimation process will require both the past frame and the next frame to predict a present frame.

It is imperative for the search macro-block to have same size as that of the reference frame; otherwise, there will be mismatch error. BMA requires a specific matching criterion to identify the matching block in the previous or next frame. It is a fact that BMA is a task that is computationally intensive and requires intense hardware configuration to perform, but the quantity of motion among the adjacent frames is not that significant; thus, it is a complete waste of computing resources to search the whole image for the specific macro-block. The search for the specific macro-block is limited to the search window that is of smaller size compared to the entire image frame as displayed in Figure 3.

Figure 3(a) represents the $8 \times 8$ macro-block that possesses an object that is moved to another location within the next frame in Figure 3(b). Red arrow in Figure 3(b) is basically the MVs that hold the displaced information of the macro-block. The frame in Figure 3(a) has a resolution of $M_1 \times N_1$ and $M_2 \times N_2$ for Figure 3(b). The search area is given in equation (1):

$$S_a = (M_1 + 2M_2) \times (N_1 + 2N_2)$$  \hspace{1cm} (1)

Equation (2) represents the mean absolute difference (MAD) where $d_x$ and $d_y$ are the displacement vectors. MAD is considered to be the best matching criterion among the research community in the respective field. MAD($d_x$, $d_y$) is calculated for each macro-block, which makes it possible for the closely matched block to be identified. Equation (3) identifies the minimum valued macro-block within $S_a$.

$$MAD(d_x, d_y) = \frac{1}{MN} \sum_{(r,c) \in W} [s(r, c, n) - s(r - d_y, c - d_x, n - 1)]$$  \hspace{1cm} (2)

$$\left[ \begin{array}{c} D_x \\ D_y \end{array} \right] = \min(MAD(d_x, d_y))$$  \hspace{1cm} (3)

Figure 4(a) and (c) is used to predict the frame in Figure 4(b). In this case, both forward and backward predictions are necessary because a red object in the past frame is in front of the blue object due to which a small part of it is being hidden. The hidden part is recovered from the next frame in which the red object has passed and revealed the hidden part of the blue object.

Proposed TM parameter (Q) for motion masking

MV$s$ are the most essential components required for video coding. They hold the motion activity and are used to reconstruct the required frame. Since video is the input signal for the proposed algorithm, it is safe to say that besides rows and columns, time is the third factor. MV$s$ obtained are of two types, in which one denotes the displacement vector in the $X$ direction and the other one in the $Y$ direction. These MV$s$ determine the location of a displaced object from the previous frame with respect to the next frame. Decoder at the receiving end aligns the macro-block using these MV$s$ and reconstructs the frame.

A human eye is unable to follow and identify the less motion activity in a video sequence because high motion activity superimposes the less motion activity. This feature of human visual system can be easily exploited and reduces the less motion activity to save the bit rate required for its transmission. In this paper, a rule is proposed to reduce the low motion activity expressed in equations (4) and (5). The $q$ parameter is used for TM and it is a variable quantity that has a range of 1 to 4. If the value of $q$ is set to be greater than 4, it is possible that it may mask some significant motion activity, which might dampen the experience of the viewer. Equation (4) is used to mask MV of the $X$ direction, while equation (5) is used to mask MV of the $Y$ direction. Table 2 represents the sample of $6 \times 6$ extracted MV values from frames given in Figure 2. Table 3 represents the $6 \times 6$ motion masked blocks on which the proposed rule in equations (4) and (5) is applied.

$$Q_X = \frac{1}{256} \sum_{(r,c) \in W} |s(r, c, n) - s(r - d_y, c - d_x, n - 1)|$$  \hspace{1cm} (4)

$$Q_Y = \frac{1}{256} \sum_{(r,c) \in W} |s(r, c, n) - s(r - d_y, c - d_x, n - 1)|$$  \hspace{1cm} (5)

Table 2: Sample of extracted MV values from frames given in Figure 2.

Table 3: Motion masked blocks on which the proposed rule is applied.
applied. For Table 3, \( q \) is set to be equal to 1; hence, all values in the MV field of Table 2 will be masked to zero if they are less than 1. The same scenario is repeated for \( q \) up to the value of 4; in this case, all the values that are less than 4 are masked to zero as given in Table 4.

\[
\begin{align*}
MVX_{Reduced} &= \begin{cases} 
0 & \text{if } q < MVX_{Actual} \leq q \\
MVX_{Actual} & \text{Otherwise}
\end{cases} \\
MVY_{Reduced} &= \begin{cases} 
0 & \text{if } q < MVY_{Actual} \leq q \\
MVY_{Actual} & \text{Otherwise}
\end{cases}
\end{align*}
\]

Results

Figure 5(a) represents a reconstructed frame without using the TM parameter \( q \), where Figure 5(c) is an extracted MV field used to reconstruct the frame in Figure 5(a). Figure 5(b), (e) and (f) denotes the reconstructed frames using the TM parameter \( q \) where the

| Table 2. Sample of 6 × 6 motion vectors along the X direction (MVX_actual). |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| 0 0 | -1.3000 | -1.3000 | -1.8201 | -1.8201 |
| 0 0.8595 | 0.8595 | 0.8721 | 0.8595 | 1.1210 |
| 0 0.9498 | 0.8595 | 0.8104 | 1.8027 | 1.1210 |
| 0 0.9498 | 0.8990 | 0.8937 | 1.8315 | 0.2520 |
| 0 1.0545 | 1.2794 | 2.2943 | 2.999 | 0.2026 |
| 0 1.0545 | 1.3612 | 2.2680 | 2.998 | 0.2368 |

| Table 3. Application of temporal masking \( q \) parameter on extracted (MVX_actual) values, where \( q = 1 \). |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| 0 0 | -1.3000 | -1.3000 | -1.8201 | -1.8201 |
| 0 0 | 0 | 0 | 0 | 1.1210 |
| 0 0 | 0 | 0 | 1.8027 | 1.1210 |
| 0 0 | 0 | 0 | 1.8315 | 0 |
| 0 1.0545 | 1.2794 | 2.2943 | 2.999 | 0 |
| 0 1.0545 | 1.3612 | 2.2680 | 2.998 | 0 |

Note: Applied for Table 3, \( q \) set to be equal to 1; hence, all values in the MV field of Table 2 will be masked to zero if they are less than 1. The same scenario is repeated for \( q \) up to the value of 4; in this case, all the values that are less than 4 are masked to zero as given in Table 4.

| Table 4. Application of temporal masking \( q \) parameter on extracted (MVX_actual) values, where \( q = 4 \). |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| 0 0 | 0 | 0 | 0 | 0 | 0 |
| 0 0 | 0 | 0 | 0 | 0 | 0 |
| 0 0 | 0 | 0 | 0 | 0 | 0 |
| 0 0 | 0 | 0 | 0 | 0 | 0 |
| 0 1.0545 | 1.2794 | 2.2943 | 2.999 | 0 |
| 0 1.0545 | 1.3612 | 2.2680 | 2.998 | 0 |

Figure 5. Sky video sequence: (a) reconstructed frame without TM; (b) reconstructed frame with TM at \( q = 4 \); (c) MV field of (a); (d) MV field of (b); (e) reconstructed frame with TM at \( q = 2 \); (f) reconstructed frame with TM at \( q = 1 \); (g) MV field of (e); and (h) MV field of (f).
value of $q$ is 4, 2, and 1, respectively. Figure 5(d), (g), and (h) represents the MV fields of the reconstructed frames using the TM parameter ($q$). Small blue arrows in the MV field point to the direction of the moving object contained by the macro-block. The MV field in Figure 5(d) has the least amount of temporal activity where most of the values are reduced to zero as the TM parameter ($q$) is equal to 4. If the value of $q$ is reduced to 2 or 1, the MV field possesses more temporal activity as given in Figure 5(g) and (h). White spaces in all MV fields in Figure 5(c), (d), (g), and (h) are zero, which means that there is no motion activity in this section of a frame. Figure 6(a) is a reconstructed frame using the standard (H.264/AVC) format, and Figure 6(c) represents its MV field. Figure 6(b) is a reconstructed frame using TM at $q = 4$, and Figure 6(d) is its MV field. The algorithm is applied on five different video sequences discussed in Tables 5 and 6 to increase the data set.

The performance of the proposed algorithm is measured and compared with the standard H.264/AVC video using PSNR and SSIM as quality measuring tools. For Sky video sequence, the lower threshold value of PSNR and SSIM is 35 dB and 0.8, respectively, which means that this is an acceptable range of the quality index. Bit rate actually shows the amount of bit saved using the proposed algorithm, which can be seen in Figure 7(a). If the value of $q$ is set to 1 and the bit rate is 1750 kbps, we saved 1.2% bit rate, while we maintain the PSNR at 38.72 dB which is extremely close to the PSNR of the standard H.264/ACV video coding, that is, 38.81 dB. SSIM for the proposed algorithm at $q = 1$ is 0.934, while it is 0.941 for the standard H.264/AVC. In a real-time application such as this one, this difference is considered to be negligible. If the value of $q$ is greater than 4, then the difference between the standard H.264 and the proposed algorithm will become prominent; thus, the range of $q$ is set to be from 1 to 4. At $q = 4$, the PSNR value is 35.3 dB.

| Video sequence | PSNR (dB) at $q$ | SSIM at $q$ |
|----------------|-----------------|--------------|
|                | $q = 1$ | $q = 2$ | $q = 4$ | $q = 1$ | $q = 2$ | $q = 4$ |
| Sky            | 38.7   | 37.6   | 35.3   | 0.93   | 0.89   | 0.83   |
| Pedestrian     | 34.6   | 34.1   | 32.9   | 0.88   | 0.83   | 0.8    |
| Traffic        | 33.2   | 32.4   | 31     | 0.90   | 0.83   | 0.78   |
| Station        | 36     | 35.2   | 33.8   | 0.91   | 0.84   | 0.72   |
| Tractor        | 34.3   | 33.7   | 32.5   | 0.93   | 0.86   | 0.77   |

$TM$: temporal masking; $SSIM$: structural similarity index; $PSNR$: peak signal to noise ratio.

Table 6. SSIM and PSNR of five video sequences without implementing TM (standard H.264/AVC or MPEG-10) and bit rate saved using TM at $q = 1$, 2, and 4.

| Video sequence | PSNR  | SSIM  | Bit rate saved (%) using TM |
|----------------|-------|-------|----------------------------|
|                |       |       | $q = 1$ | $q = 2$ | $q = 4$ |
| Sky            | 38.81 | 0.941 | 1.2     | 2.62    | 13.04   |
| Pedestrian     | 35.6  | 0.922 | 2.02    | 2.96    | 10.62   |
| Traffic        | 33.96 | 0.931 | 1.9     | 3.4     | 11.57   |
| Station        | 36.83 | 0.933 | 1.26    | 2.99    | 9.43    |
| Tractor        | 35.19 | 0.947 | 2.3     | 3.97    | 14.12   |

$AVC$: Advanced Video Coding; $SSIM$: structural similarity index; $PSNR$: peak signal to noise ratio; $TM$: temporal masking.
and SSIM is found to be 0.834, which are acceptable for real-time application. Threshold values are not same and depend on the video content. If the temporal activity in the video is extensive, then it may require more bit rate than normal in order to transmit. For example, threshold values of PSNR and SSIM of the Traffic Video sequence are 30 dB and 0.75, respectively.

Tables 5 and 6 represent the compression of five different video sequences after applying the proposed TM algorithm and were compared with the results of the existing state-of-the-art video coding standard (H.264/AVC) format. The values of PSNR and SSIM in Table 5 for the TM parameter $q = 1$ are really close to those of the standard H.264/AVC format. At $q = 2$ and 4, these values have small difference, but at the same time it is saving more bit rate while losing some of its quality features as displayed in Table 5. This difference is extremely small and cannot be noticed in video sequences that are played at 30 frames per second (fps) rate. For example, PSNR and SSIM values of a Traffic Video sequence are 33.96 dB and 0.931, respectively, when they are compressed using the standard H.264/AVC format as shown in Table 5. After implementing the proposed TM algorithm, PSNR and SSIM at $q = 1$ are 33.2 dB and 0.9, respectively, while saving 1.9% bit rate at 1750 kbps speed as shown in Table 6. This difference is almost negligible when a video is played at 30 or 60 fps rate.

Table 7 is the OS, which was acquired from the masses by showing them 20 videos of 10 s each. For each video sequence in Table 5, four different videos were displayed, for instance, video 1 (standard H.264/AVC) format of Sky video sequence, video 2 (TM applied at $q = 1$), video 3 (TM applied at $q = 2$), and video 4 (TM applied at $q = 4$). People were told that video 1, video 5, video 9, video 13, and video 17 are the reference videos. We acquired the opinions of 50 people to generate OS. It can be seen from Table 7 that video 2 scored 83%, which means that 83% people identified this video as a reference video or they were unable to point the difference between them. Reference video was played first for the people and they were asked to comment on the three compressed videos that closely resemble the reference.

**Conclusion**

The process of compression is the most important and essential part of video coding. It is key component that makes it possible for multimedia information to be delivered wirelessly within the given bit rate. Communication channel can be wired or wireless in nature. In this paper, an algorithm is proposed that utilizes TM in MVs to reduce bit rate. A video sequence is composed of multiple individual frames, and most of these consecutive frames in a same video sequence will retain a large chunk of information from its predecessor and it will possess temporal activity. Less temporal activity within the consecutive frame is masked using the proposed TM parameter ($q$) that is insignificant to human eyes in real time. By varying $q$, it is possible to control the bit rate required for the transmission. SSIM and PSNR are the two measurement tools used for assessing the performance of the proposed algorithm. At the bit rate of 1750 kbps, the SSIM was found to be 0.934 at $q = 1$, and for the standard H.264/AVC, it is 0.941. This difference is almost negligible for a human eye if the video is playing in real time, and in return 1.2% of the bit rate is saved. The

| Video no. | Applied TM                                      | OS (%) |
|-----------|-------------------------------------------------|--------|
| Video 1   | Sky sequence (standard H.264/AVC)                | Reference |
| Video 2   | Sky sequence at $q = 1$                         | 83     |
| Video 3   | Sky sequence at $q = 2$                         | 78     |
| Video 4   | Sky sequence at $q = 4$                         | 62     |
| Video 5   | Pedestrian sequence (standard H.264/AVC)        | Reference |
| Video 6   | Pedestrian sequence at $q = 1$                  | 79     |
| Video 7   | Pedestrian sequence at $q = 2$                  | 76     |
| Video 8   | Pedestrian sequence at $q = 4$                  | 58     |
| Video 9   | Traffic sequence (standard H.264/AVC)           | Reference |
| Video 10  | Traffic sequence at $q = 1$                     | 89     |
| Video 11  | Traffic sequence at $q = 2$                     | 71     |
| Video 12  | Traffic sequence at $q = 4$                     | 64     |
| Video 13  | Station sequence (standard H.264/AVC)           | Reference |
| Video 14  | Station sequence at $q = 1$                     | 77     |
| Video 15  | Station sequence at $q = 2$                     | 76     |
| Video 16  | Station sequence at $q = 4$                     | 62     |
| Video 17  | Tractor sequence (standard H.264/AVC)           | Reference |
| Video 18  | Tractor sequence at $q = 1$                     | 81     |
| Video 19  | Tractor sequence at $q = 2$                     | 78     |
| Video 20  | Tractor sequence at $q = 4$                     | 60     |

MOS: mean opinion score; AVC: Advanced Video Coding; TM: temporal masking; OS: opinion score.
acceptable range of PSNR is 35 dB, and if it is less than the given PSNR threshold value, the difference within the frames becomes more prominent. For this reason, the range of \( q \) is set to be from 1 to 4. The proposed algorithm performed slightly better in terms of saving bit rate by varying the \( q \) parameter.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) received no financial support for the research, authorship, and/or publication of this article.

**ORCID iD**

Ali Akbar Siddique [https://orcid.org/0000-0002-9641-7067](https://orcid.org/0000-0002-9641-7067)

**References**

1. Kaur A, Sidhu JS and Bhullar JS. Artifacts reduction based on separate modes in compressed images. *J Intel Fuzzy Syst* 2018; 35(2): 1645–1656.

2. Pathak KC, Sarvaiya JN and Darji AD. Adaptive prediction methods for medical image/video compression for telemedicine application. In: Dey N, Ashour AS, Kalla H, et al. (eds) Histopathological image analysis in medical decision making. Hershey, PA: IGI Global, 2019, pp. 244–275.

3. Cheng L and Zarki ME. Systems and methods for video compression for low bit rate and low latency video communications. US9237347 Patent, USA, 2016.

4. Wang Z, Sheikh HR and Bovik AC. Objective video quality assessment. In: Furht B and Marque O (eds) *The handbook of video databases: design and applications*. Boca Raton, FL: CRC Press, 2003, pp.1041–1078.

5. Tan TK, Weerakkody R, Mrak M, et al. Video quality evaluation methodology and verification testing of HEVC compression performance. *IEEE T Circuit Syst Video Technol* 2016; 26(1): 76–90.

6. Daribo I, Florencio D and Cheung G. Arbitrarily shaped motion prediction for depth video compression using arithmetic edge coding. *IEEE T Image Proc* 2014; 23(11): 4696–4708.

7. Mantiu RK and Myszkowski K. Perception-inspired high dynamic range video coding and compression. In: Höflinger B (ed.) *CHIPS 2020. The frontiers collection*, vol. 2. New York: Springer, 2016, pp. 211–220.

8. Watson AB, Hu J and McGowan JF. Digital video quality metric based on human vision. *J Electron Imag* 2001; 10(20): 20–29.

9. Suresh N, Palaniappan R, Mane P, et al. Testing of a no-reference VQ metric: monitoring quality and detecting visible artifacts, 2009, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.518.5024&rep=rep1&type=pdf

10. McGuinness CD, David W, Clark T, et al. Evaluation of H. 264 and H.265 full motion video encoding for small UAS platforms. *Proc SPIE* 2016; 9841: 12.

11. Bachu S. A new diamond search algorithm for adaptive pixel based motion estimation in video coding. *Int J Signal Process Image Proc Pattern Recogn* 2016; 9(11): 319–330.

12. Kamble SD, Thakur NV and Bajaj PR. Modified three-step search block matching motion estimation and weighted finite automata based fractal video compression. *Int J Interact Multimed Artif Intell* 2017; 4: 27–39.

13. Hossen MK and Tuli SH. A surveillance system based on motion detection and motion estimation using optical flow. In: *5th international conference on informatics, electronics and vision (ICIEV)*, Dhaka, Bangladesh, 13–14 May 2016.

14. Suliman A and Li R. Video compression using variable block size motion compensation with selective subpixel accuracy in redundant wavelet transform. In: Latifi S (ed.) *Information technology: new generations*. New York: Springer, 2016, pp. 1021–1028.

15. Sridhar K, Sattar SA and Chandra Mohan M. Review on data hiding in motion vectors and in intra-prediction modes for video compression. In: *Proceedings of the international conference on data engineering and communication technology*, Pune, India, 15–16 December 2017. Singapore: Springer, 2017.

16. Sebastian T and Anitha J. Hybrid hierarchical search motion estimation for video compression. In: *2nd international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB)*, Chennai, India, 27–28 February 2016. New York: IEEE.

17. Rüfenacht D and Taubman D. Temporally consistent high frame-rate upsampling with motion sparsification. In: *IEEE 18th international workshop on multimedia signal processing (MMSP)*, Montreal, QC, Canada, 21–23 September 2016.

18. Chan CH, Ji HS and Tasi WJ. Adaptive accordion transformation based video compression method on HEVC. In: *Visual communications and image processing (VCIP)*, Chengdu, China, 27–30 November 2016. New York: IEEE.

19. Seshadrinathan K, Soundararajan R, Bovik AC, et al. Study of subjective and objective quality assessment of video. *IEEE T Image Proc* 2010; 19(6): 1427–1441.

20. Seshadrinathan K, Soundararajan R, Bovik AC, et al. A subjective study to evaluate video quality assessment algorithms. In: *SPIE proceedings human vision and electronic imaging*, San Jose, CA, 18–21 January 2010.