Journal of Ambient Intelligence and Humanized Computing (2024) 15:2215–2225
https://doi.org/10.1007/s12652-023-04748-w

ORIGINAL RESEARCH

Fuzzy-based video compression using bilinear fuzzy relation equations

Barbara Cardone[1] · **Ferdinando Di Martino**[1,2]

Received: 20 January 2023 / Accepted: 16 December 2023 / Published online: 25 January 2024
© The Author(s) 2024

Abstract
We present a novel color video compression method using the greatest solution of a system of bilinear fuzzy relation equations to assess the similarity between frames. The frames in each band are treated separately and each frame is classified as an Intra frame or a Predictive frame. A frame is labelled as Predictive frame, and compressed more than an Intra-frame, if the similarity value with the previous Intra frame is higher than a selected threshold; A pre-processing activity is performed to select the optimal threshold value of the similarity between frames. The proposed method allows to supply a high quality of the reconstructed frames and has the advantage of not requiring high CPU time and memory storage for its execution; it was tested on color videos of the Fast-Moving Objects dataset; the results show that it produces better performances than the Lukasiewicz similarity-based video compression method and comparable with those achieved by MPEG-4 and the deep learning video compression method DVC_pro. The results show that the quality of the reconstructed frames obtained with BFRE is comparable with that of DVC Pro, but has a lower computational complexity, providing better performances in terms of video encoding speed.

Keywords Video compression · I-frame · P-frame · BFRE · BFRE similarity · PSNR

1 Introduction

The exponential evolution of digital technologies and the need to use and archive video streams in numerous applications has led to the development of standard video compression methods and approaches used today in devices and video communication applications (Abomhara et al. 2010; Beach and Owen 2018).

Recently some authors present new video compression algorithms aimed to balance information loss with computational speed. In particular, some deep neural network-based autoencoders for video compression were proposed. These methods minimize the rate distortion of frames, setting or maximizing the number of bits (or bit rate) for the compression, in order to obtain a higher quality of the reconstructed frame. Their main benefits are their adaptiveness to the input and their non-linearity which makes them flexible to detect spatial variations in frames. However, even if deep learning video compression algorithms improve the rate-distortion by providing a high quality of the reconstructed frames, they have the disadvantage of being slower than traditional video compression algorithms and require a high memory capacity in the learning phase.

For this purpose, in this research we propose a new video compression algorithm based on bilinear fuzzy relation equations (for shorts, BFRE) which improves the quality of the reconstructed frames compared to traditional algorithms and requires lower CPU time and stored capacity than those required for deep learning video compression algorithms.

In this paper we propose a new video compression method based on BFRE, used to measure the similarity between frames.

The main goal of this research is to build an efficient and computationally fast video compression method by measuring the similarity between the frame and previous frames applying solutions of BFRE systems. We consider

✉ Ferdinando Di Martino
fdimarti@unina.it

Barbara Cardone
b.cardone@unina.it

[1] Dipartimento Di Architettura, Università Degli Studi Di Napoli Federico II, Via Toledo 402, 80134 Naples, Italy

[2] Università Degli Studi Di Napoli Federico II, Centro Interdipartimentale Di Ricerca "A. Calza Bini", Via Toledo 402, Naples, Italy

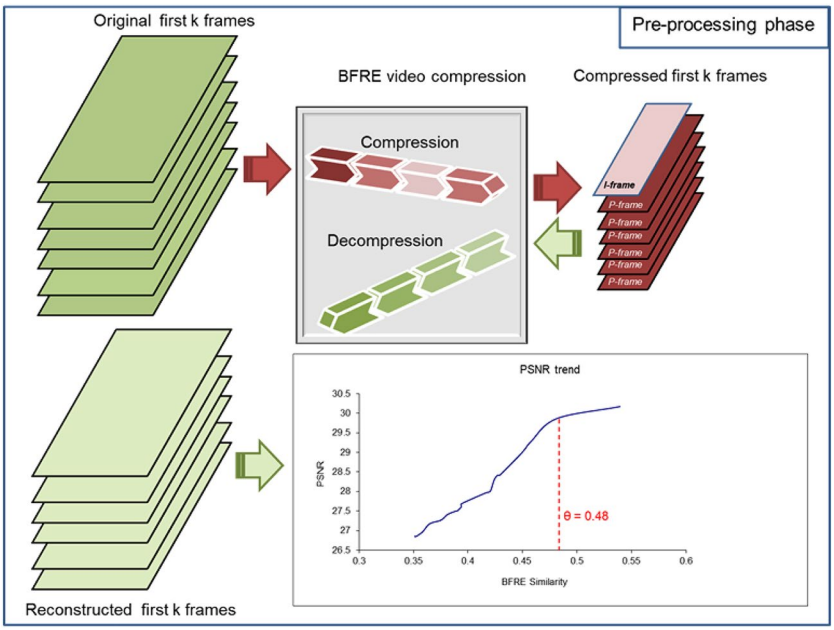

Fig. 1 Architectural scheme of the BFRE video compression pre-processing phase

the decomposition of the video frames into two types of frames, called Intra-frames (for short, I-frames) and Predictive Frames (for short, P-frames) as proposed in (Nobuhara 2006). A P-frame is a frame whose similarity with the previous I-frame is higher than a prefixed threshold. Therefore, it can be compressed at a higher compression rate than that used to compress an I-frame.

We apply a similarity measure based on the bilinear fuzzy relation equations described in (Di Martino and Sessa 2017, 2018). The authors prove that the BFRE similarity index is more robust to noise with respect to similarity measures based on the Lukasiewicz t-norm.

The iterative frame classification process proposed in (Nobuhara 2006; Di Martino et al. 2010) is used to classify frames; we treat separately the three bands Red (R), Green (G) and Blue (B), then, for each band we apply the BFRE similarity measures to compare a frame with the previous I-frame, where each frame is normalized in [0,1] to represent a fuzzy relation. Instead of setting a-priori the similarity threshold, we execute a pre-processing phase in which the first k frames are analyzed; the first frame is marked as an Intra frame (I-frame) and the following frames as Predictive frames (P-frames). Each of the (k -1) P-frames is subsequently reconstructed and the trend of the PSNR is analyzed as the BFRE similarity index changes, choosing as the similarity threshold that value of the similarity index below which the PSNR trend decreases rapidly.

In Fig. 1 is schematized the pre-processing phase.

The pre-processing phase ends after selecting the similarity threshold value; then, is executed the BFRE compression process, in which each frame is labelled and compressed as I-frame or P-frame depending on whether the BFRE similarity with the previous I-frame is less than the selected threshold or not. The BFRE decompression process reconstructs the original frame; finally, three reconstructed frames in the three bands are merged to obtain the frame of the reconstructed color video.

The main performance benefits that characterize the proposed color video compression method are the following:

- it uses an algorithm based on the BFRE similarity between images to label frames; it allows to reconstruct color videos with less information loss than that obtained in (Di Martino et al. 2010), in which a similarity measure based on Lukasiewicz's triangular norm is used. Indeed, in (Di Martino & Sessa 2017, 2018;) it was demonstrated that the BFRE similarity is more robust to noise with respect Lukasiewicz t-norm similarity;
- the adopted pre-processing phase allows to determine for each band an optimal value of the similarity threshold, guaranteeing a lower loss of information in the reconstructed frames compared to that obtained by selecting other threshold values;
- the proposed method is independent of the image compression algorithm used to compress the frames;
- unlike video-compression deep learning algorithms, it does not require high CPU time and memory storage.

1.1 Related work

In recent years the need to improve the quality of the reconstructed frames has prompted many researchers to propose

video compression methods that minimize the RDO of the reconstructed frames.

In (Jridi and Meher 2017) a video compression technique based on Discrete Cosine Transform (for short DCT) is applied; it has better performances than other DCT-based video compression techniques, but it has high processing times. In (Wang et al. 2017) a video-compression method based on a quadtree plus binary tree block partition structure is proposed; this approach represents a tradeoff between encoding complexity and quality performance, but it engenders a not negligible decrease in the quality of the reconstructed videos.

In (Haitham 2018) a new video-compression technique based on a Contourlet Transform used with a motion compensation technique (MC) is proposed.

Recently, deep neural network-based autoencoders for video compression were proposed. These methods minimize the rate distortion (RDO), setting or maximizing the number of bits (or bit rate) for the compression, in order to obtain a higher quality of the reconstructed frame. Their main benefits are their adaptiveness to the input and their non-linearity which makes them flexible to detect spatial variations in frames.

A neural video compression algorithm is proposed in (Chen et al. 2020) to predict spatiotemporal relationship between blocks in video frames, minimizing the residual block error. In (Honggui and Trocan 2018) a deep neural framework is proposed in which multiple deep neural networks are trained to predict pixels of intra-frames based on pixels of previous frames.

An end-to-end network neural scheme is proposed in (Lu et al. 2018) to replace the standard codec structure. In (Lu et al. 2021) a hybrid variation of this end-to-end neural video compression framework is proposed to improve the compression efficiency and optimize the rate distortion. The results show that this hybrid video compression method provides better quality reconstructed frames than those obtained using traditional algorithms, but at the expense of high computational complexity.

A Deep Convolutional Neural Network with three hidden layers is used in (Putra et al. 2022) for intra-frame-based video compression; test results performed on color videos show that the quality of the reconstructed frames is better than that obtained using discrete cosine transform and fractal video-compression algorithms.

Surveys of deep learning video compression algorithms are provided in (Zhang et al. 2020; Bidwe et al. 2022). These studies show that deep learning video compression approaches improve the rate-distortion; however, they are slower than traditional video compression algorithms and require higher memory consumption.

In order to reach a trade-off between quality of the reconstructed frames and CPU times, we propose a new color videos compression method based on BFRE where the solutions of bilinear fuzzy relation equations are applied to measure the similarity e between frames.

(Loia and Sessa 2005) applied fuzzy relation equations in an iterative video compression method in which the Lukasiewicz fuzzy triangular norm measures the similarity between frames. The first frame is labelled as a I-frame, then, a t-norm based similarity measure is applied to compare the next frame to the previous I-frame; if this similarity value is higher than the selected threshold, then the current frame is labelled as a P-frame, otherwise it is labelled as a I-frame and becomes the I-frame to be compared with the next frame. The frames are compressed by applying the fuzzy relation equations image compression method (Di Martino et al. 2003a, b; Nobuhara 2006). In (Di Martino et al. 2010) the video compression method proposed in (Loia and Sessa 2005) is applied using the Fuzzy transform image compression algorithm (Perfilieva 2006; Di Martino and Sessa 2007; Di Martino et al. 2008) (for short F-transform) to compress each frame. The authors compare their method with the video compression method in (Loia and Sessa 2005) showing that the better results in terms of quality of the reconstructed video are obtained by using the F-transform algorithm.

(Di Martino and Sessa 2013) proposed a new video compression method in which the frames are classified in I-frames, P-frames, and Bi-directional frames (for short, B-frames), following the MPEG-4 algorithm classification (Pereira and Ebrahimi 2002; Richardson 2010). A B-frame is reconstructed by using either the previous or successive I-frame. The authors compare their method with the MPEG algorithm and the video compression method (Loia and Sessa 2005) using the Peak Signal to Noise Ratio (for short PSNR) index to measure the quality of the reconstructed frames. They show that the performances of the three algorithms are similar.

One of the main critical issues of the fuzzy-based videos compression algorithms is the selection of the similarity threshold between frames. In (Nobuhara et al. 2000; Loia & Sessa 2005) the Lukasiewicz's t-norm operator is used to calculate the similarity between two frames considered as fuzzy relations; the similarity threshold is experimentally set at 0.95 as the quality of the reconstructed frames decreases exponentially when the similarity measure is below this value; however, this threshold value could vary according to the characteristics and type of frames. In the proposed method a preprocessing phase is adopted to optimize the choice of the similarity threshold in each band.

In Sect. 2 we introduce the BFRE image similarity measure, synthetizing the method to measure the similarity measure between images based on the BFRE greatest solution. In Sect. 3 the proposed BFRE color video compression method is presented. The results of experimental tests applied on color videos are shown and discussed in Sect. 4. Final considerations are discussed in Sect. 5.

2 BFRE Similarity between images

Let F and G be two M × N grey images. We denote with $F_{p,q}$ and $G_{p,q}$ the values of the pixels of F and G at the p^{th} row and at the q^{th} column with p = 1,…,M and q = 1,…,N. Let L the number of grey levels of the two images.

We construct a 3 × 3 window around the pixel F_{pq} with components:

$$a_{i,j} = \frac{F_{p+i-2,\, q+j-2}}{L-1} \; i = 1,..,3 j = 1,\ldots,3 \tag{1}$$

and a correspondent 3 × 3 window around the pixel $G_{p,q}$ with components:

$$b_{i,j} = \frac{G_{p+i-2,\, q+j-2}}{L-1} \; i = 1,..,3 j = 1,\ldots,3 \tag{2}$$

Let $A = [a_{ij}]$ and $B = [b_{ij}]$, where i = 1,…,3 and j = 1,…,3, be two 3 × 3 fuzzy relations; we construct the 3 × 3 BFRE system:

$$\bigcup_{j=1}^{3}(a_{ij} \wedge x_j) = \bigcup_{j=1}^{3}(b_{ij} \wedge x_j) i = 1,..,3 \tag{3}$$

where $\mathbf{x} = (x_1,x_2,x_3)$ is a vector of three fuzzy variables and the union and intersection operators are given, respectively, by the max s-norm and by the min t-norm.

In (Li 1992) is proposed an algorithm, called the BFRE algorithm, to find the greatest solution of a BFRE system in the form (3).

The BFRE algorithm is applied in (Di Martino and Sessa 2018) to find the greatest solution of the (3), given by $\hat{\mathbf{x}}^{\mathbf{pq}} = (\hat{x}_1^{pq}, \hat{x}_2^{pq}, \hat{x}_3^{pq})^T$.

If $\mathbf{x}_s^{\mathbf{pq}} = (0,0,0)^T$ is the smallest solution of (3), in (Di Martino and Sessa 2018) is defined the following measure of similarity between the two fuzzy relations A and B:

$$Sim_{BFRE}^{pq} = \frac{1}{3}\sqrt{\sum_{j=1}^{3}\left(\hat{x}_j^{pq} - x_{sj}^{pq}\right)^2} = \frac{1}{3}\sqrt{\sum_{j=1}^{3}\left(\hat{x}_j^{pq}\right)^2} \tag{4}$$

The index Sim_{BFRE}^{pq} varies between 0 and $\frac{\sqrt{3}}{3}$. Its value is 0 if $a_{ij} = 0$ (1) and $b_{ij} = 1$ (0) for each i = 1,2,3 and j = 1,2,3 (maximum difference between the two windows) and is $\frac{\sqrt{3}}{3} \approx 0.577$ if the greatest solution of (9) is $\hat{\mathbf{x}}^{\mathbf{pq}} = (1,1,1)^T$ (the two 3 × 3 windows are identical).

In (Di Martino and Sessa 2018) is defined the following similarity index between the images F and G as

$$Sim_{BFRE}(F,G) = \frac{1}{M \times N}\sum_{p=1}^{M}\sum_{q=1}^{N} Sim_{BFRE}^{pq} \tag{5}$$

where M and N are the number of rows and columns of the two images.

If F and G are color images in (Di Martino and Sessa 2018) the similarity between the two images is given by the average of the similarities in the three bands Red (R), Green (G) and Blue (B):

$$Sim_{BFRE}(F,G) = \frac{1}{3}\left(Sim_{BFRE}^{R}(F,G) + Sim_{BFRE}^{G}(F,G) + Sim_{BFRE}^{B}(F,G)\right) \tag{6}$$

This similarity measure (6) is compared in (Di Martino and Sessa 2018) with other image similarity measures based on fuzzy relations. The authors measure the similarity between images distorted by Gaussian noise, proving the similarity measure (6) is more robust to noise with respect to other image similarity indices.

Below is schematized in pseudocode the algorithm used to calculate the similarity between two grey levels images in which the formula (5) is used.

Algorithm 1: BFRESimilarity

Input: MxN gray images F and G with L grey levels

Output: Index of similarity between F and G $SIM_{BFRE}(F,G)$

```
SIM_BFRE := 0  // initialize to 0 the similarity index
For i = 1 to M
  For j = 1 to N
    Construct the 3x3 fuzzy relation A by (1)
    Construct the 3x3 fuzzy relation B by (2)
    Construct the 3x3 BFRE system (3)
    Calculate the greatest solution x̂ = (x̂1,x̂2,x̂3)^T
    Calculate Sim^ij_BFRE by (4)
    Sim_BFRE := Sim_BFRE + Sim^ij_BFRE
  Next j
Next i
SIM_BFRE := SIM_BFRE / (M·N)
Return SIM_BFRE
```

3 The proposed video compression method

We propose a method based to the BFRE similarity measure (6) to compress color videos. We use the BFRE similarity measure to classify individual frames as I-frames or P-frames, treating videos in the three color bands R, G and B separately.

The first frame is always labeled as I-frame; then, the similarity (6) between the I-frame and the next frame is measured. If this similarity value is greater than a fixed threshold the next frame is labeled as P-frame, otherwise it is labeled as I-frame and is considered in the comparisons with the following frames. This process is iterated until the last frame of the video has been analyzed.

Following (Di Martino et al. 2010), we apply the Fuzzy Transform image compression method to compress the frames. The I-frames are compressed with a compression rate with a weak com-pression rate ρ_I and the P-frames are compressed by applying a strong compression rate ρ_P.

In order to compress the P-frames with a strong compression rate, in (Loia and Sessa 2005; Di Martino et al. 2010) an approach was used in which the P-frame is transformed in a difference frame D, called Δ-frame, where:

$$D(i,j) = \frac{1}{2}\left[F(i,j) - G(i,j)\right] \; i = 1,\ldots,M \; j = 1,\ldots,N \tag{7}$$

As the Δ-frame D has a low quantity of information it can be compressed with a strong compression rate. If $\widehat{F}$ is the decoded I-frame and $\widehat{D}$ the decoded Δ-frame, the P-frame G can be reconstructed by the following formula:

$$\hat{G}(i,j) = \frac{\max\left\{0, \hat{F}(i,j) - 2\hat{D}(i,j) + 1\right\}}{\max\left\{1, \hat{F}(i,j) - 2\hat{D}(i,j) + 1\right\}} \; i = 1,\ldots,M \; j = 1,\ldots,N \tag{8}$$

Below is schematized the proposed video compression algorithm.

Algorithm 2: BFRE video compression

Input: A single-band MxN video with L grey levels

Output: Compressed video

```
1.  Label the first frame as an I-frame
2.    F:= the first frame
3.    Repeat
4.      G:= next frame
5.      Sim = BFRESimilarity(F,G,L)
6.      if Sim < Sim_th then
7.        Label the current frame as an I-frame
8.        F:= the current frame
9.        Compress the current frame with the compression rate ρ_I
10.     else
11.       Label the current frame as a P-frame
12.       Construct the Δ-frame D by (7)
13.       Compress the Δ-frame D with the compression rate ρ_P
14.     end if
15.   Until all frames are labeled
16.   Store the compressed frames
```

Fig. 2 **a** darts_window1 – Frame 1. **b** ping_pong_side – Frame 1

The following algorithm schematize the decompression process in which a P-frame is reconstructed by (8).

Algorithm 3: BFRE video decompression

Input: Compressed single-band video level frames compressed
Output: Decompressed video

1. F:= the first compressed frame
2. Decompress F using the compression rate ρ_I obtaining $\hat{F}$
3. ***Repeat***
4. D:= next compressed frame
5. Decompress D using the compression rate ρ_P obtaining $\hat{D}$
6. Reconstruct G by (8) obtaining $\hat{G}$
7. ***Until*** all frames are reconstructed
8. Store the decompressed video

To fix the similarity threshold Sim_{th} is executed a preprocessing phase in which the first k frames of the single band video is analyzed; in this phase the first frame is marked as an I-frame and the following k-1 frames are considered P-frames. Each compressed P-frame is subsequently reconstructed and is calculated the PSNR measure, adopted to evaluate the quality of the reconstructed frame $\hat{G}$ compared to the correspondent original frame G. Then, the trend of the PSNR index with respect to the BFRE similarity between the frame and the first frame is constructed. The PSNR index between the original frame G and the reconstructed frame $\hat{G}$ is given by

$$\mathrm{PSNR}\left(G,\hat{G}\right) = 20\log_{10}\frac{L-1}{\mathrm{RMSE}} \tag{9}$$

where:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left[G(i,j)-\hat{G}(i,j)\right]^2}{M\times N}} \tag{10}$$

The threshold Sim_{th} is selected as the value of the similarity index below which the PSNR trend decreases rapidly.

In Algorithm 4 is schematized in pseudocode the preprocessing method applied to select the similarity threshold.

Algorithm 4: BFRE similarity threshold selection

Input: The first k frames of a MxN single band video with L grey levels
Output: The similarity threshold Sim_{th}

1. Label the first frame as an I-frame
2. F:= the first frame
3. ***For*** i = 2 to k
4. G:= the i^{th} frame
5. Sim[i] = *BFRESimilarity*(F,G,L)
6. Compress G with the compression rate ρ_P
7. Decompress G with the compression rate ρ_P
8. PSNR[i] = PSNR(G, $\hat{G}$)
9. ***Next*** i
10. Fix Sim_{th} as the similarity value below which the PSNR trend decreases rapidly
11. Return Sim_{th}

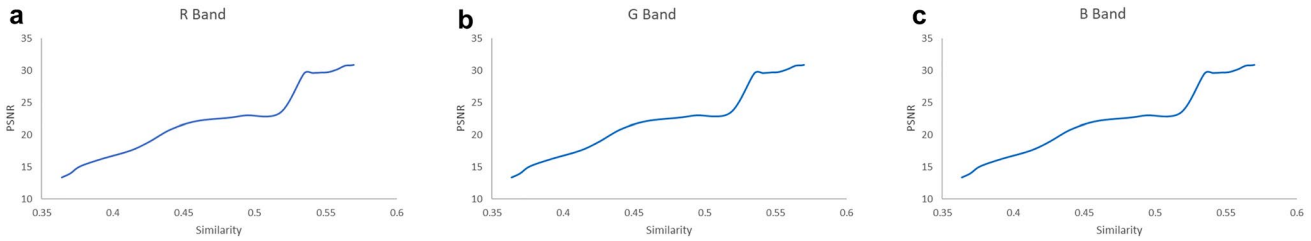

Fig. 3 **a** PSNR trend—R Band. **b** PSNR trend – G Band. **c** PSNR trend—B Band

Fig. 4 **a** darts_window1– reconst. Frame 1. **b** darts_window1 – reconst. Frame 2

We tested our method by applying it to color videos with different sizes. To measure the performance of the BFRE video compression algorithm we compared the results with those obtained using the MPEG-4 video compression method, the video com-pression method based on fuzzy transform (Di Martino and Sessa 2007) and the end-to-end Deep Video Compression DVC_Pro in (Lu et al 2021). The results are shown and discussed in the next section.

4 Results

To test the BFRE video compression algorithm, we have executed it on a set of 16 color videos extracted from the Fast-Moving Object (FMO) web page http://cmp.felk.cvut.cz/fmo. The video frames have sizes 1920×1080, 1280×820 and 960×540 pixels. An Intel(R) Core(TM) i7 processor with a CPU clock speed of 2.90 GHz was used for all tests.

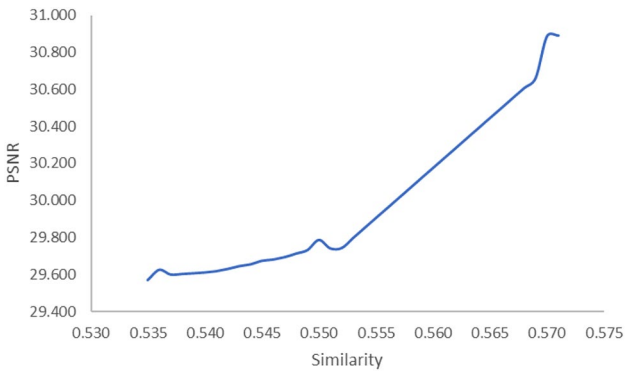

Fig. 5 darts_window1 – trend of PSNR with respect to the BFRE similarity

For the sake of brevity, the results obtained on the color video darts_window1, made up of 179 frames, and ping_pong_side, made up of 445 frames, are shown in detail. In Fig. 2a and b are shown the first frames of the two-color videos.

All the frames were decomposed in the three bands Red, Green and Blue and the frames in each band have been treated separately. In these tests the values $\rho_I = 0.25$ and $\rho_P = 0.0625$ were selected respectively for the compression rate of I-frames and P-frames.

Now are shown the results obtained for the color video darts_window1. The first 40 frames are used in the pre-processing phase to fix the similarity threshold. The first frame is considered an I-frame and the other frames are all considered P-frames. In Fig. 3 is show the trend of the PSNR index with respect to the BFRE similarity in the three bands.

All the three trends in Fig. 4 show a rapid decrease below the similarity value 0.535, for which the selected similarity threshold is $Sim_{th} = 0.535$ in any band.

Table 1 darts_window1– PSNR ccomparison results

	BFRE	FTR	MPEG	DVC_Pro
Average	29.996	29.945	29.988	30.012
Standard dev	0.401	0.466	0.461	0.495
Min	29.571	29.544	29.567	29.030
Max	30.889	30.870	30.893	31.075

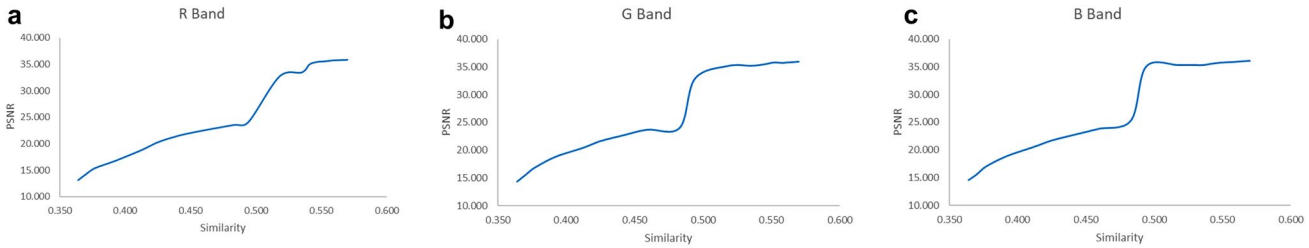

Fig. 6 **a** PSNR trend—R Band. **b** PSNR trend – G Band. **c** PSNR trend—B Band

Fig. 7 **a** ping_pong_side – reconst. Frame 1 **b** ping_pong_side – reconst. Frame 2

After selecting the similarity threshold were executed the BFRE video compression and BFRE video decompression algorithms to obtain the reconstructed video. The reconstructed first two frames are shown in Fig. 4a–b.

The plot in Fig. 5 shows the trend of the PSNR index with respect to the BFRE simi-larity measure obtained by executing the BFRE video compression and decompression algorithms. The PSNR value computed for each frame is given by the average of the PSNR measured in the three bands. The PSNR index rises as the similarity increases, oscillating between the values 29.6 and 30.9.

To analyze the performance of the BFRE video compression method we compared the results obtained with those obtained executing the fuzzy transform-based video compression algorithm in (Di Martino et al. 2010) (FTR), the ISO/IEC standard MPEG-4 (Pereira and Ebrahimi 2002) (MPEG) and the end-to-end Deep Video Compression DVC_Pro in (Lu et al 2021). In the comparison, the ρ_I and ρ_P compression rates selected for I-Frame and P-frame are used by performing FTR. A bit rate of 10 mbps has been selected in the MPEG-4 compression. The learning-based optical flow model Spynet (Ranjan and Black 2017) is applied in DVC_Pro to assess motion information. In Table 1 are shown the mean, standard deviation, min and max values of the PSNR index measured for all the reconstructed frames by executing BFRE, FTR MPEG and DVC_Pro.

These results show that BFRE provides better results with respect to FTR and PSNR values similar to those obtained by running MPEG and DVC_Pro. Moreover, BFRE shows better stability than FTR, MPEG and DVC_Pro, also providing the best measure of the standard deviation of the PSNR value for all frames.

Now are shown the results obtained for the color video ping_pong_side. In Fig. 6a–c are shown the trend of the PSNR index with respect to the BFRE similarity measure in the three bands.

The trend of PSNR in the bands G and B (Fig. 6b and c) show a rapid decrease below the similarity value 0.500, for which we select a similarity threshold $Sim_{th} = 0.500$ in these two bands. The trend in the R band show a rapid decrease below a BFRE similarity 0.540; this value is selected as similarity threshold in the R band.

The reconstructed first two frames are shown, respectively, in Fig. 7a and b.

In Table 2 are shown the mean, standard deviation, min and max values of the PSNR index measured for all the reconstructed frames by executing BFRE, FTR and MPEG.

The results in Table 2 confirm that the BFRE algorithm provide better performance than FTR in terms of mean PSNR. Moreover, the mean PSNR is comparable with the one measured executing MPEG-4 and DVC_Pro. In addition, BFRE shows better stability than FTR, MPEG and DVC_Pro, showing the lowest value of the standard deviation measure.

Table 2 ping_pong_side – PSNR comparison results

	BFRE	FTR	MPEG	DVC_Pro
Average	35.280	35.056	35.285	35.312
Standard dev	0.407	0.462	0.449	0.473
Min	33.665	32.872	33.694	33.567
Max	36.012	35.791	36.072	36.098

Table 3 Mean PSNR and standard deviation obtained for the color videos in the FMO dataset

Video	Frame size	Mean PSNR				PSNR Standard dev			
		BFRE	FTR	MPEG	DVC_Pro	BFRE	FTR	MPEG	DVC_Pro
VolleyBall1	1920×1080	41.256	40.854	41.169	**41.301**	**0.368**	0.401	0.395	0.415
Volleyball_passing	1920×1080	38.315	37.936	38.010	**38.654**	**0.326**	0.344	0.352	0.360
Darts1	1920×1080	39.387	39.286	39.379	**39.542**	**0.399**	0.437	0.423	0.435
Darts_window1	1920×1080	**30.332**	29.945	29.988	30.309	**0.401**	0.466	0.461	0.475
Softball	1280×720	36.281	36.078	36.167	**36.348**	**0.354**	0.380	0.373	0.378
Wiliam_tell	1920×1080	46.504	46.279	46.310	**46.853**	**0.536**	0.560	0.548	0.562
Tennis_serve_side	1280×720	37.878	37.832	37.884	**37.962**	**0.338**	0.366	0.359	0.372
Tennis_serve_back	1280×720	33.669	33.557	33.596	**33.734**	**0.391**	0.438	0.426	0.435
Tennis1	960×540	**29.175**	29.070	29.117	29.158	**0.393**	0.434	0.429	0.417
Hockey	960×540	30.637	30.567	30.592	**30.661**	**0.441**	0.489	0.481	0.476
Squash	960×540	31.195	31.107	31.186	**31.233**	**0.464**	0.492	0.488	0.499
Frisbee	1920×1080	**38.592**	37.389	38.589	**38.592**	**0.369**	0.405	0.407	0.416
Blue	1280×720	39.892	39.849	39.892	**39.936**	**0.333**	0.373	0.354	0.362
Ping_pong_paint	1280×720	30.349	30.291	30.355	**30.378**	**0.366**	0.395	0.388	0.392
Ping_pong_side	1920×1080	33.695	32.872	33.694	33.728	**0.407**	0.462	0.449	0.473
Ping_pong_top	1920×1080	34.828	34.626	34.807	**34.855**	**0.504**	0.537	0.525	0.518

The highest values of mean PSNR and the lowest values of PSNR standard deviation are shown in bold

Table 4 Encoding speed measured for the color videos in the FMO dataset

Video	Frame size	Encoding speed (fps)			
		BFRE	FTR	MPEG	DVC_Pro
VolleyBall1	1920×1080	**1.78**	1.75	1.46	1.32
Volleyball_passing	1920×1080	**1.84**	1.77	1.42	1.35
Darts1	1920×1080	**1.87**	1.83	1.47	1.38
Darts_window1	1920×1080	1.83	**1.84**	1.45	1.36
Softball	1280×720	**2.36**	2.35	2.14	2.08
Wiliam_tell	1920×1080	1.84	**1.87**	1.48	1.36
Tennis_serve_side	1280×720	**2.39**	2.34	2.20	2.09
Tennis_serve_back	1280×720	**2.38**	2.37	2.16	2.12
Tennis1	960×540	**3.65**	3.58	3.22	3.20
Hockey	960×540	**3.61**	3.60	3. 28	3.25
Squash	960×540	3.60	**3.64**	3.19	3.19
Frisbee	1920×1080	**1.89**	1.85	1.44	1.39
Blue	1280×720	2.36	**2.37**	2.14	2.08
Ping_pong_paint	1280×720	**2.41**	2.38	2.17	2.16
Ping_pong_side	1920×1080	**1.85**	1.78	1.39	1.34
Ping_pong_top	1920×1080	**1.86**	1.81	1.42	1.38

The highest values of encoding speed are shown in bold

In Table 3 are shown the mean and the standard deviation of the PSNR values obtained for all the FMO videos executing BFRE, FTR, MPEG-4 and DVC_Pro. These measures are given by the average of the mean and standard deviation of PSNR obtained in each band. In bold are highlighted the best results.

These results confirms that the quality performances of BFRE are better than the ones obtained executing FTR and comparable with MPEG-4 and DVC_PRO. In addition, for all videos the lowest value of the standard deviation is obtained executing the BFRE algorithm, showing that BFRE provides greater stability compared to FTR, MPEG-4 and DVC_PRO in the oscillation of the PSNR measured for each frame.

We compare the computational complexity of the four video compression methods measuring the encoding speed in frames per second (fps). Table 4 show the encoding speed measured executing the four methods for each color video.

These results highlight that BFRE provide better encoding speed values with respect to MPEG and DVC_Pro; the encoding speed values obtained executing BFRE are similar than the ones obtained executing FTR.

5 Conclusions

We propose a new color video compression method based on the BFRE greatest solution algorithm. Following the frame type classification in (Loia and Sessa 2005; Nobuhara et al. 2006; Di Martino et al. 2010) we partition the set of frames in I-frame and P-frame. To classify a frame a new similarity measures based of BFRE is applies; a frame is labelled as P-frame if its similarity with the previous I-frame is greater than a fixed threshold. The source color video is treated separately in the three bands Red, Green and Blue. A pre-processing phase is executed in each band to select the optimal value of the similarity threshold in that band; in this phase the first k frames are analyzed labelling as an I-frame the first frame and as a P-frame any other frame. The similarity threshold is given by the value of the similarity below which the PSNR trend decreases rapidly.

We execute comparative test of the proposed method with the method based on the F-transforms proposed in (Di Martino et al. 2010), MPEG-4 and the learning video compression method DVC-Pro proposed in (Lu et al. 2021). The results show that our method provides better performance than those provided by the F-transforms and MPEG-video compression method and comparable with the ones provided by DVC-Pro. In addition, the PSNR values measured for each frame obtained by performing BFRE show a lower standard deviation than those obtained by performing the other three video compression methods; this result suggests that the proposed method may be more robust than FTR, MPEG-4 and DVC_Pro. Finally, encoding speed comparisons were performed to analyze the computational complexity of the four methods. The results show that BFRE provides encoding speeds higher than those provided by DVC_Pro and comparable to those provided by FTR. Thus, BFRE provides overall optimal performance in terms of reconstructed frame quality and video encoding speed.

In the future we intend to carry out further experimental tests on video datasets of different sizes, frame rates and resolutions. Furthermore, we intend to test our method on videos with noisy frames and sequences of frames to measure its robustness.

Author contributions Conceptualization, B.C. and F.D.M.; methodology, B.C. and F.D.M.; software, B.C. and F.D.M.; validation, B.C. and F.D.M.; formal analysis, B.C. and F.D.M.; investigation, B.C. and F.D.M.; re-sources, B.C. and F.D.M.; data curation, B.C. and F.D.M.; writing—original draft preparation, B.C. and F.D.M.; writing—review and editing, B.C. and F.D.M.; visualization, B.C. and F.D.M.; supervision, B.C. and F.D.M. All authors have read and agreed to the published version of the manuscript.

Funding Open access funding provided by Università degli Studi di Napoli Federico II within the CRUI-CARE Agreement.

Data availability The data that support the findings of this study are available on request from the corresponding author, [F.D.M].

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethical approval This research does not contain any studies involving human participants performed by any of the authors.

Informed consent Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Abomhara M, Khalifa OO, Zakaria O, Zaidan AA, Zaidan BB, Rame A (2010) Video compression techniques: an overview. J Appl Sci 10:1834–1840. https://doi.org/10.3923/jas.2010.1834.1840

Beach A, Owen A (2018) Video compression Handbook 2nd Edition, Peachpit Press, 336 pp., ISBN: 978–0134866215

Bidwe RV, Mishra S, Patil S, Shaw K, Vora DR, Kotecha K, Zope B (2022) Deep learning approaches for video compression: a bibliometric analysis. Big Data Cogn Comput 6:44. https://doi.org/10.3390/bdcc6020044

Chen Z, He T, Jin X, Wu F (2020) Learning for video compression. IEEE Trans Circuits Syst Video Technol 30(2):566–576. https://doi.org/10.1109/TCSVT.2019.2892608

Di Martino F, Sessa S (2007) Compression and decompression of images with discrete fuzzy transforms. Inf Sci 177:2349–2362. https://doi.org/10.1016/j.ins.2006.12.027

Di Martino F, Sessa S (2018) Comparison between images via bilinear fuzzy relation equations. J Ambient Intell Humaniz Comput 9:1517–1525. https://doi.org/10.1007/s12652-017-0576-3

Di Martino F, Loia V, Perfilieva I, Sessa S (2008) An image coding/decoding method based on direct and inverse fuzzy transforms. Int J Approxim Reason 47:110–131. https://doi.org/10.1016/j.ijar.2007.06.008

Di Martino F, Loia V, Sessa S (2010) Fuzzy transforms for compression and decompression of color videos. Inf Sci 180(20):3914–3931. https://doi.org/10.1016/j.fss.2009.08.002

Haitham A-A (2018) Video compression based on motion compensation and contourlet transform. Third Scientific Conference of Electrical Engineering (SCEE) 2018:90–94. https://doi.org/10.1109/SCEE.2018.868407

Honggui L, Trocan M (2018) Deep neural network based single pixel prediction for unified video coding. Neurocomputing 272:558–570. https://doi.org/10.1016/j.neucom.2017.07.037

Jridi M, Meher PK (2017) A scalable approximate DCT architectures for efficient HEVC compliant video coding. IEEE Trans Circuits Syst Video Technol 27(8):1815–1825. https://doi.org/10.1109/TCSVT.2016.2556578

Li JX (1992) A new algorithm for the greatest solution of fuzzy bilinear equations. Fuzzy Sets Syst 46:193–210. https://doi.org/10.1016/0165-0114(92)90132-n

Loia V, Sessa S (2005) Fuzzy relation equations for coding/decoding processes of images and videos. Inf Sci 171:145–172. https://doi.org/10.1016/j.ins.2004.04.003

Lu G, Zhang X, Ouyang W, Chen L, Gao Z, Xu D (2021) An end-to-end learning framework for video compression. IEEE Trans Pattern Anal Mach Intell 43(10):3292–3308. https://doi.org/10.1109/TPAMI.2020.2988453

Lu G, Ouyang W, Xu D, Zhang X, Cai C, Gao Z (2018) DVC: an end-to-end deep video compression framework. arXiv preprint arXiv:1812.00101

Di Martino F, Sessa S (2013) Coding B-Frames of Color Videos with Fuzzy Transforms. Adv Fuzzy Syst Article ID 652429, 9 pp. https://doi.org/10.1155/2013/652429

Di Martino F, Sessa S (2017), Bilinear equations and fuzzy image comparison, 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 9–12 July 2017, Naples (Italy), pp. 1–6. https://doi.org/10.1109/FUZZ-IEEE.2017.8015397

Di Martino F, Loia V, Sessa S (2003a) A method in the compression/decompression of images using fuzzy equations and fuzzy similarities, in: Proceedings of Conference IFSA 2003, (29/6–2/7/2003, Istanbul, Turkey) In: Bilgic T, De Baets B and Kaynak O (Eds.), Lecture Notes in Artificial Intelligence, Vol. 2715, Springer, Berlin, Germany, pp. 524–527

Di Martino F, Loia V, Sessa S (2003b) A method for coding/decoding images by using fuzzy relation equations, Selected papers from IFSA (29/6–2/7/2003, Istanbul, Turkey), In: Bilgic T, De Baets B and Kaynak O (Eds.), Lecture Notes in Artificial Intelligence, Vol. 2715, Springer, Berlin, Germany, 2003, pp. 436–441. https://doi.org/10.1007/3-540-44967-1_52

Nobuhara H, Hirota K, Pedrycz W (2000) Fast solving method of fuzzy relational equations and its application to lossy image compression. IEEE Transact Fuzzy Syst 8(3):325–334. https://doi.org/10.1109/91.855920

Nobuhara H, Hirota K, Pedrycz W, Sessa S (2006) A motion compression/reconstruction method based on max T-norm composite fuzzy relational equations. Inf Sci 176(17):2526–2552. https://doi.org/10.1016/j.ins.2005.12.004

Pereira F, Ebrahimi T (2002), MPEG-4 Book, Prentice Hall Professional, Upper Saddle River, NJ, USA, 849 pp., ISBN: 978–0–13–061621–0

Perfilieva I (2006) Fuzzy transforms: theory and applications. Fuzzy Sets Syst 157:993–1023. https://doi.org/10.1016/j.fss.2005.11.012

Putra ABW, Gaffar A, Sumadi M, Setiawati L (2022) Intra-frame based video compression using Deep Convolutional Neural Network (DCNN), 6(3), 650659, https://doi.org/10.30630/joiv.6.3.1012

Ranjan A, Black M J (2017) Optical flow estimation using a spatial pyramid network, in Proc. IEEE Conf. Comput. Vis. Pattern Recognition, Art. no. 2, pp. 4161–4170

Richardson IE (2010) The H. 264 advanced videos compression standard. John Wiley & Sons, Hoboken, NJ, USA, 316 pp., ISBN: 978–0470516928

Wang Z, Wang S, Zhang J, Wang S, Ma S (2017) Effective quadtree plus binary tree block partition decision for future video coding, 2017 Data Compression Conference (DCC), 4–7 April 2017, Snowbird, UT, USA, pp. 23–32, https://doi.org/10.1109/DCC.2017.70

Zhang Y, Kwong S, Wang S (2020) Machine learning based video coding optimizations: a survey. Inf Sci 506:395–423. https://doi.org/10.1016/j.ins.2019.07.096

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