Video Data Compression by Progressive Iterative Approximation

M. J. Ebadi[1]*, A. Ebrahimi[2]

[1] Department of Mathematics, Chabahar Maritime University, Chabahar (Iran)
[2] Computer Geometry and Dynamical Systems Laboratory, Faculty of Mathematical Sciences, Yazd University, Yazd, (Iran)

uniR
LA UNIVERSIDAD
EN INTERNET

Abstract

In the present paper, the B-spline curve is used for reducing the entropy of video data. We consider the color or luminance variations of a spatial position in a series of frames as input data points in Euclidean space R or R3. The progressive and iterative approximation (PIA) method is a direct and intuitive way of generating curve series of high and higher fitting accuracy. The video data points are approximated using progressive and iterative approximation for least square (LSPIA) fitting. The Lossless video data compression is done through storing the B-spline curve control points (CPs) and the difference between fitted and original video data. The proposed method is applied to two classes of synthetically produced and naturally recorded video sequences and makes a reduction in the entropy of both. However, this reduction is higher for syntactically created than those naturally produced. The comparative analysis of experiments on a variety of video sequences suggests that the entropy of output video data is much less than that of input video data.

Keywords

I. Introduction

THE technology of video compression has been a fundamental tool in video fields and multimedia communication for many years. The main objective of video compression is making a reduction in the volume of data by prospecting the correlations of video frames in such a way that a digital video file can be broadcast almost entirely over the network and stored on the computer disks. According to the required reconstruction, video compression techniques can be categorized into two large groups of lossy compression and lossless compression [1]–[3].

The increased use of high quality videos reveals the need for decreasing the volume of compressed video for transmission and storage, especially in social media networks. The lower the entropy of the data, the smaller the number of bits is required to encode them. Thus, this study aims to provide a practical procedure to reduce the entropy of the video data. Each color plane in the RGB space and subsequently RGB color image are respectively indicated using 8 bits/pixel and 24 bits/pixel. The inter and intra frame codings are exerted on the image sequences to decrease the temporal and spatial redundancy of the data in the image sequences.

The study of curve construction from a data point set is widely employed as a modeling instrument in many areas such as image processing, computer graphics, computer aided design (CAD), reverse engineering, object shape detection, and scientific visualization. According to its application, curves of implicit, parametric, and subdivision type are applied to data fitting. Converting data points into parametric curves including B-spline or Bézier curves is extremely desired in engineering applications. Most of the papers in the literature used motion estimation for video data compression. In more recent ones, the parametric curves such as the Bézier curve or natural cubic spline have been applied to compress video data into small storage space.

The Bézier curve is constructed by Bernstein basis that has limited flexibility. The degree of curve is directly relative to the number of control points (CPs). For a complicated shape and data, a large number of CPs may be required. To overcome this shortcoming and provide more flexibility and control, the B-spline curve is suggested as the generalization of the Bézier curve. In case the number of CPs is high, the use of lower degree parametric curves is possible. In order to prevent the additional cost of computations for solving a large linear equations system, the progressive iterative approximation (PIA) is used that is computationally efficient and simple to implement.

The PIA method is a direct and intuitive way of generating curve series of high and higher curve fitting accuracy. The PIA method refrains from solving a large linear equations system with an additional computational cost. The PIA technique begins with an initial curve and adjusts the curve CPs in an iterative process. Then, the resulted point cloud is interpolated and approximated by the limit curve. In this paper, we propose a technique for lossless video data compression making use of the B-spline curve The color or luminance variations of a spatial position in a series of frames are considered as data points in Euclidean space $\mathbb{R}$ or $\mathbb{R}^3$. The data points are approximated using progressive and iterative approximation for least square fitting (LSPIA). The proposed method reduces entropy and has efficient computational complexity.

* Corresponding author.

E-mail address: ebadi@cmu.ac.ir

In particular, our contributions are the following:

- We use the B-spline curve with remarkable flexibility to approximate the video data.
- The PIA method is applied to find the optimal CP of the B-spline curve with no need of solving a large linear equations system.
- Our method can be considered as a lossless video compression method that reduces the entropy of video data.

The organization of this paper is as follows. A brief summary of the related works is given in Section II. Section III provides a simple overview of LSPIA using the B-spline curve. The procedure of the proposed method using the B-spline curve to fit video data is explained in Section IV. Section V describes in detail the methodology adopted to design the video data compression. Section VI is dedicated to the study of the experimental results for various videos. A brief discussion on the proposed method in video compression is provided in Section VII. In the end, a conclusion is made in Section VIII.

II. Review of Literature

In recent decades, multiple processes have been developed in the field of curve fitting by the use of Bézier and B-spline curves. Biswas [4] used the quadratic Bézier curve for compression of the grayscale images. Bézier curve has been used to capture the outline of planar generic images. An outline capturing technique was presented in [5] to estimate the appropriate location of CPs by the utilization of the cubic Bézier curve properties. In [6], a method is designed to capture the outline of 2D shapes using the cubic Bézier curve with the emphasis given to local control of data points rather than the global error of square fitting. A novel outline capturing scheme for 2D shapes was introduced in [7] based on the Nelder-Mead simplex method.

In [8], the L-BFGS optimization is exerted on data points to which B-spline curve is fitted. Ebrahimi and Loghmani [9] used approximation BFGS methods to make optimization of the foot and CPs of the B-spline curve. The complexity per step in [9] is $O(n)$, requiring only $O(n)$ memory allocations. In [10], a practical approach to curve fitting is presented for the specification of the initial B-spline curve which is near to the target curve. A length parameter is presented by this method which allows adjustment to the number of CPs. This makes the initial B-spline curve more precise. The scaled BFGS algorithm is then employed for simultaneous optimization of control and foot points.

Lin et al. [11] introduced the phrase "progressive iterative approximation" in 2005. The standard PIA procedure is not feasible for curve fitting with plenty of the data points when control and data points are equal in number [12]. Delgado and Pena [13] proved that the normalized B-basis is a totally positive basis with the fastest convergence rate. A local PIA format is designed in [14] and showed the convergence of the local format for the normalized totally positive based blending curve. An approach is proposed for weighted PIA of data points using normalized totally positive basis in [15] with a faster convergence rate. In [16], an extended PIA is introduced in which the number of given data points with storage requirement $O(n)$ is higher than the number of CPs, where n is the number of the CPs. An adaptive data point fitting based on the PIA is proposed in [17]. Zhang et al. [18] developed a progressive T-spline method of fitting large-scale datasets such as images of high precision. Deng and Lin [19] introduced the LSPIA where the number of data points is more than that of CPs. LSPIA provides a set of fitting curves making adjustments of the CPs and leading to the given data points through least square (LS) fitting as the final curve. Ebrahimi and Loghmani [20] presented the composite iterative method for LSPIA with a fast convergence rate. This method constructs a series of matrices applied to the adjusting

vector on the base of the Schulz iterative method. A comprehensive survey on PIA methods has been provided in [21].

Motion Estimation (ME) is the most popular in removing the temporal redundancy in video compression that can be arranged into pixel and block motion estimations [22]. The motion vector in pixel motion estimation is computed for every pixel in the frame. The block motion estimation method divides frames into blocks and then the motion vector is computed for every block. In the interframe coding method, block motion estimation plays a key role in reducing temporal redundancy in the image sequence. A block-matching approach can be developed to modify the coding efficiency and video quality. In the past three decades, some improvements have been made in motion estimation techniques such as pelrecursive methods, optical flow, block matching algorithms, and parametric-based models [23].

By making some attempts, quick application and simple comprehension of block matching algorithms make them fundamental methods of motion estimation in video compression. The full search algorithm (FSA) is the easiest method in the block matching algorithms that has high computational cost. To accelerate the search procedure and decrease the computational complexity, several fast block matching algorithms, such as diamond search (DS) [24], hexagon search (HS) [25], three step search (TSS) [26], [27], four step search (FSS) [28] have been proposed.

To perform a method of fast motion estimation, Koga et al. [29] introduced TSS as a primary attempt. Compared to the full search, the TSS method has a less computational cost in terms of average search point and mean absolute difference.

The computational cost TSS method is less in average search point and mean absolute difference as compared to the full search method. The modified TSS algorithm is presented in [27] for weighted finite automata coding and block matching motion estimation methods to reduce the encoding time.

Video and image compression using parametric curves explored by many authors. Fu et al. [30] has been explored a video object encoding method pursuant to the data fitting trajectory of video object moving edges pixels that is suitable for the slowly moving video/ video data. The cubic spline interpolation is used in [31] to modify medical image compression for medicine applications. The cubic convolution spline interpolation is proposed on the basis of the LSs method to compress the image data in [32]. In [33], based on the natural cubic spline and parametric line fitting, a method for lossy compression is presented in order for compression of digital video data in the temporal dimension. The linear Bézier curve is used in [34] for the approximation of temporal video sequence in Euclidean space. Khan [35] has proposed an algorithm for lossless video compression which was based on the quadratic Bézier curve and least square technique.

III. The Progressive and Iterative Approximation for Least Square Fitting (LSPIA)

Here, we first formulate the blending curve and review the LSPIA (readers are referred to [19] for details).

A nonnegative basis $\{N_i(t); i = 0, 1, ..., n\}$ defined on a set I with $\sum_{i=1}^{n} N_i(t) = 1$ for all $t \in I$ is taken as a blending basis.

A totally positive blending basis is defined as normalized totally positive (NTP) basis. Let $\{N_i(t); i = 1, 2, \cdots, n\}$ be an NTP blending basis. Then, assuming a sequence of the CPs $\{P_i\}_{i=0}^{n}$ in $\mathbb{R}$ or $\mathbb{R}^3$, a blending curve as

$$C(t) = \sum_{i=0}^{n} P_i N_i(t)$$

(1)

can be considered. Suppose that $\{q_j; j = 1, 2, \cdots, m\}$ is an ordered data point sequence on a target curve to be fitted and

$$\Gamma = \{0 = t_0 < t_1 < \cdots < t_m = 1\}$$

is the location parameters of $\{q_j; j = 1, 2, \cdots, m\}$. Taking $\{P_i^0\}_{i=0}^n$ form $\{q_j; j = 1, 2, \cdots, m\}$ similar to the CPs, the initial blending curve $C^0(t)$ is defined:

$$C^0(t) = \sum_{i=0}^n P_i^0 N_i(t), \quad t \in [t_0, t_m] \tag{2}$$

The $(m+1) \times (n+1)$ collocation matrix of the NTP blending basis $\{N_i(t); i = 1, 2, \cdots, n\}$ on Γ is

$$A = \begin{bmatrix} N_0(t_0) & N_1(t_0) & ... & N_n(t_0) \\ N_0(t_1) & N_1(t_1) & ... & N_n(t_1) \\ ... & ... & ... & ... \\ N_0(t_m) & N_1(t_m) & ... & N_n(t_m) \end{bmatrix} \tag{3}$$

At the beginning of the iteration, let

$$\delta_j^0 = q_j - C^0(t_j), \quad j = 0, 1, \cdots, m \tag{4}$$

$$\Delta_i^0 = \mu \sum_{j=0}^m N_i(t_j)\delta_j^0, \quad i = 0, 1, ..., n \tag{5}$$

where μ is a non zero real scalar and

$$0 < \mu < \frac{2}{\lambda_0}$$

where λ_0 is the largest eigenvalue of $A^T A$. By the movement of the CPs P_i^0 along the regulating vector Δ_i^0, i.e.

$$P_i^1 = P_i^0 + \Delta_i^0, \quad i = 0, 1, ..., n \tag{6}$$

and the new curve,

$$C^1(t) = \sum_{i=0}^n P_i^1 N_i(t), \quad t \in [t_0, t_m] \tag{7}$$

Similarly, obtaining the k-th blending curve C^k after the k-th iteration, we suppose

$$\delta_j^k = q_j - C^k(t_j), \quad j = 0, 1, \cdots, m \tag{8}$$

$$\Delta_i^k = \mu \sum_{j=0}^m N_i(t_j)\delta_j^k, \quad i = 0, 1, ..., n \tag{9}$$

we can generate the $(k+1)$-th blending curve as follows

$$C^{k+1}(t) = \sum_{i=0}^n P_i^{k+1} N_i(t), \quad t \in [t_0, t_m] \tag{10}$$

The mentioned iterative process produces a curve sequence $\{C^k(t), k = 0, 1, ...\}$ whose limit is the LS fitting curve of the original data points $\{q_j\}_{j=0}^m$ [19].

The initial situation of the CPs ($\{P_i^0\}_{i=0}^n$) may be selected as

$$P_0^0 = q_0, \ P_n^0 = q_m$$

$$P_i^0 = q_{g(i)}, \ i = 1, 2, ..., n - 1$$

where $g(i) = \left[\frac{(m+1)i}{n}\right]$. In addition, we adopt the uniform parametrization to assign the parameters $\{t_j\}_{j=0}^m$ for $\{q_j\}_{j=0}^m$.

In this study, having numerical computation stability and extensive use in image processing, we wxamine the LSPIA by B-spline curve. Having B-spline basis functions, B-spline is a blending curve. Let $\{P_i\}_{i=0}^n$ be $n + 1$ CPs and $N_{i,r}(t)$ be the B-spline basis functions of degree r (or order $r + 1$) defined on a given nondecreasing real-number knot vector $U = \{u_0, u_1, \cdots, u_{n+r+1}\}$, then a B-spline curve of degree r will be as follows

$$C(t) = \sum_{i=0}^n P_i N_{i,r}(t) \tag{11}$$

where the B-spline basis functions $N_{i,r}(t)$ is defined recursively by the Boor formula

$$N_{i,0}(t) = \begin{cases} 1, & u_i \leqslant t \leqslant u_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,r}(t) = \frac{t - u_i}{u_{i+r-1} - u_i} N_{i,r-1}(t) + \frac{u_{i+r} - t}{u_{i+r} - u_{i+1}} N_{i+1,r-1}(t), r \geqslant 1$$

The proposed method works for any number of CPs and any degree of B-spline curve but from our experimental results, we notice that $r = 3$ and $n = 7$ are appropiate for degree of curve and number of CPs respectively. Further, the cubic B-spline basis are constructed on the knot vectors

$$u_0 = u_1 = ... = u_3 = 0$$

$$u_{j+3} = (1 - \beta)t_{i-1} + \beta t_i, \quad j = 1, ..., n - 3$$

$$u_{n+1} = ... = u_{n+4} = 1$$

where $i = [jd], \beta = jd - i, d = \frac{m+1}{n-2}$. According to the B-spline curve definition, it is clear that the properties of the B-spline basis function are passed t the B-spline curve. These properties are as follows:

- Partition of unity
- Affine invariance
- Convex hull property
- Local control
- Multiple knots

IV. Video Data Fitting With LSPIA

In this section, the process of the video data fitting using LSPIA is presented. Let a video include a sequence of m frames, and each frame possesses $W \times H$ pixels, where H and W respectively are the height and width of video frames. The value of each pixel in a frame is a data point in Euclidean space $\mathbb{R}^1$ or $\mathbb{R}^3$ for luminance or 3-D RGB, respectively. The temporal data of a spatial location

$$(x, y), 1 \leqslant x \leqslant W, 1 \leqslant y \leqslant H$$

in m frames are $\{q_1, q_2, \cdots, q_m\}$, i. e., $\{q_j = I_j\}_{j=0}^m$ for luminance or $\{q_j = (r_j, g_i, b_i)\}_{j=0}^m$ for 3-D RGB. Then, we approximate the m values of each spatial location $\{q_1, q_2, \cdots, q_m\}$ by the LSPIA method. Fig. 1 illustrates the RGB variation of a spatial position (50, 50) in 96 Mobile and Calendar video sequence frames.

The video data from each spatial location in an sequence of frames (input data) is approximated with much less number of control points (output data) of the B-spline curve. This process is separately used to intensify RGB variations in the temporal dimension of each spatial position. The luminance values of a spatial position (50, 50) in 96 Foreman video sequence frames are fitted using a cubic B-spline basis and LSPIA in Fig. 2.

V. Methodology

The main purpose of the proposed method is video data compression by reducing the entropy of output data. A smaller number of bits is needed to store the video data with lower entropy. In the first step, the color or luminance variations of a spatial position in series of frames are considered the input data in Euclidean space $\mathbb{R}$ or $\mathbb{R}^3$. In the next step, we use the B-spline curve and approximate the input video data with a considerably smaller number of CPs. In addition, LSPIA fits the input video data with low approximation errors and without solving a system of equations. In this step of the work, we need to store only the CPs of the B-spline curve to approximate the input data. In the
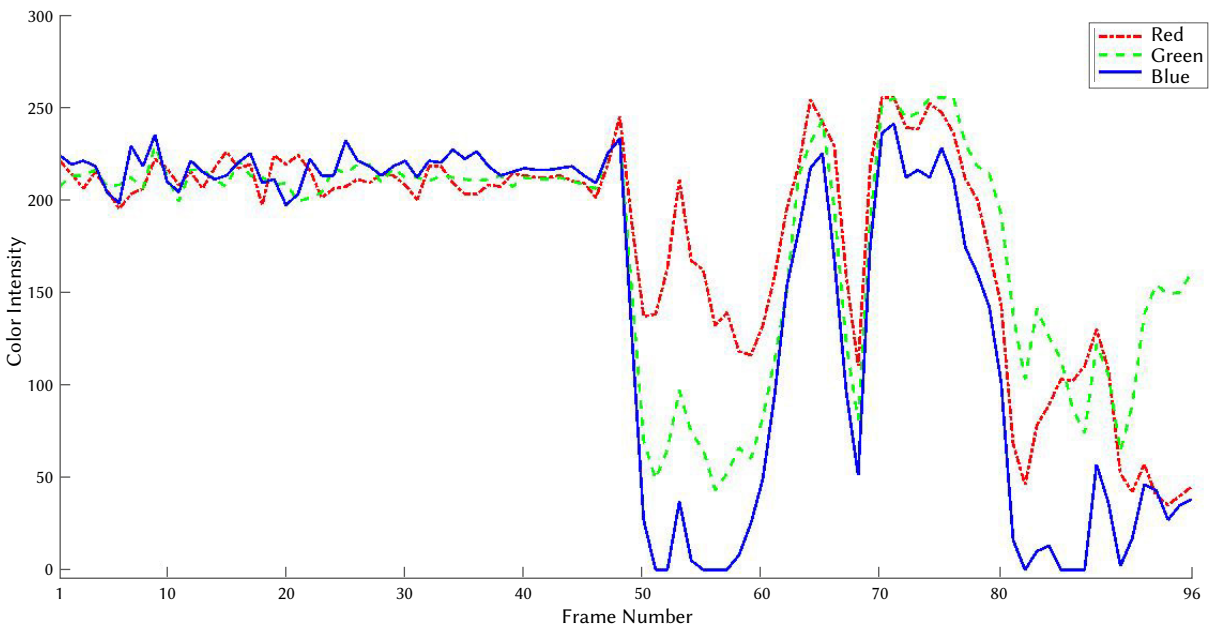
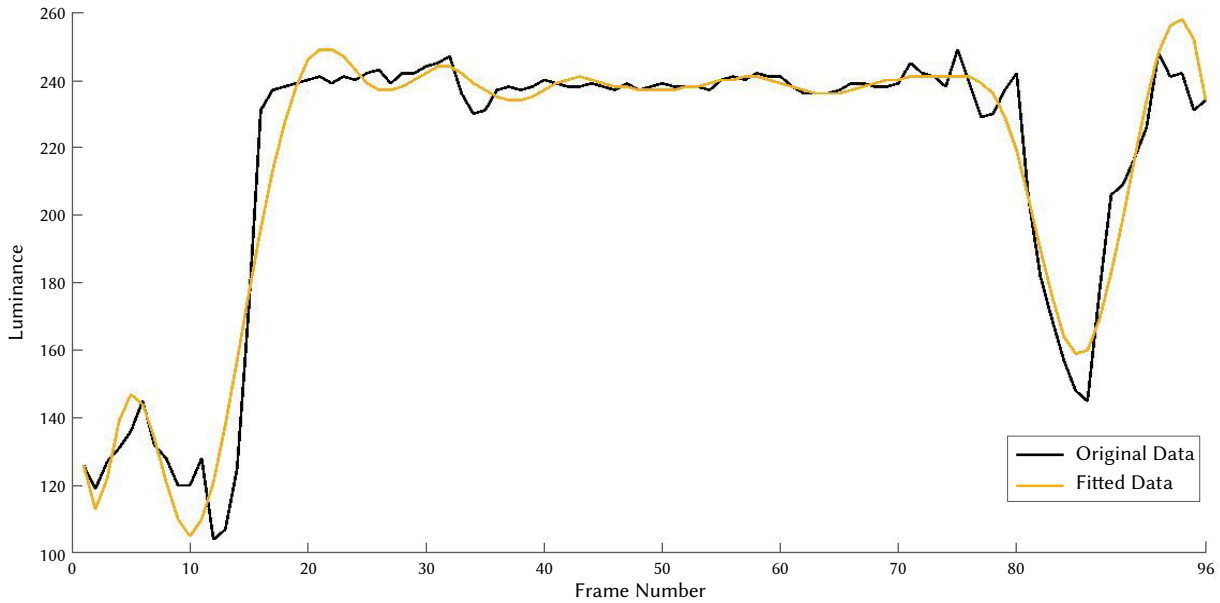Fig. 1. The RGB variation of a spatial position (50, 50) in 96 Mobile and Calendar video sequence frames.

Fig. 2. B-spline curve fitting to the luminance values of a spatial position (50, 50) in 96 Foreman video sequence frames using LSPIA.

final step, the difference between primary and approximated video frames (DF) are also stored for lossless video compression. According to the curve fitting method used in this work, the difference between primary and cubic B-spline approximated data has limited values in the short-range in comparison to primary values in the primary video sequence. Therefore, the entropy of CP and DF in the proposed method is far less than that of the primary video sequence. It is worth mentioning that our method can be used for lossy video compression. The basic foundation of our method is explained in Algorithm 1.

We use the CPs of the B-spline curve to create the approximated video frames and then add the frame difference (FD) to reconstruct the original video. In contrast to the most existing methods that use the neighbor's pixels to reduce spatial redundancy, our method merely uses temporal redundancy.

Algorithm 1. Video data compression using LSPIA

Input: A video includes a sequence of m frames, and each frame possesses $W \times H$ pixels;

Output: The CPs of B-spline curve and the difference between primary and approximated video frames (DF);

for $i = 1$ **to** W **do**

    for $j = 1$ **to** H **do**

        Consider the data of spatial location (i, j) in m frames $\{q_k\}_{k=1}^{m}$;

        Approximate $\{q_k\}_{k=1}^{m}$ using LSPIA;

        Store the CPs of B-spline curve;

        Store the difference between $\{q_k\}_{k=1}^{m}$ and B-spline curve (DF);

    end for

end for

a) Mobile and calendar sequence   b) Dinosaur sequence   c) Cloud sequence

d) Foreman sequence   e) Hall and monitor sequence

Fig. 3. One of the frames of video sequences.

VI. Experiments and Results

The introduced technique described in the previous section has been applied to some synthetically produced and naturally recorded video sequences and its results have been compared with those obtained with the method in [35]. We compare our method with the technique proposed by khan [35] because it used the quadratic Bézier curve and entropy criterion.

According to the required reconstruction, the methods of video compression can be classified into two groups of lossless compression methods, in which the output video is identical to input video, and lossy compression methods, with generally provide much higher compression in which the output video is different from the input video.

Some innovative improvements have been recently made to lossy video compression to which interested readers can refer [27] and the references therein. The introduced method in this study is a lossless video compression and hence instead of PSNR, we use the entropy criterion to evaluate the efficiency of the compression method. The entropy is a scale of the required mean number of binary symbols for coding the source output. Encoding source output with the bit mean number equal to the source entropy is indicative of a desired lossless compression method.

Suppose a source (frame) of information has M symbols (pixel values) with individual probability P_i and

$$\sum_{i=1}^{M} P_i = 1$$

The entropy of a single video frame can be defined by:

$$H = -\sum_{i=1}^{M} P_i \log P_i$$

(12)

We calculate the entropy of video using the mean entropy of all frames that construct the video sequence.

To evaluate the proposed method efficiency, five standard video sequences of different resolutions with sufficient complexity are selected for the simulation as listed in Table I and one of the frames of each input video sequence is represented in Fig. 3.

TABLE I. Schematic of the Test Video Sequences

Test video sequences	Format	Resolution	Frames
Mobile and calendar	RGB	352 × 240	96
Dinosaur	RGB	352 × 288	96
Cloud	RGB	352 × 240	96
Foreman	Luminance	352 × 240	96
Hall and monitor	Luminance	352 × 28896	96

The entropy of the videos is simply calculated using Equation (12). The output data in our method for computing entropy consists of the CPs and the difference between primary and approximated video frames (DF). The output data produced by algorithm [35] need to be stored and used in computing entropy which includes: (1) the end CPs of Bézier curve, (2) the middle CPs of Bézier curve, and (3) the difference between the quadratic Bézier approximated and original video sequences.

Table II compares the introduced method with algorithm [35] in terms of entropy. It can be seen that the significantly lower entropy is produced by the proposed method than those generated by [35].

TABLE II. Performance Comparison in Terms of Entropy

Video name	Original video	Method [35]	Our method
Mobile and calendar	7.627	6.653	6.431
Dinosaur	7.163	2.736	2.334
Cloud	7.567	4.032	3.849
Foreman	7.228	5.124	4.576
Hall and monitor	7.233	3.918	3.243

VII. Discussion

The videos tested in the previous section are classified into two groups: (a) naturally recorded video sequences; and (b) synthetically created video sequences. Among them Hall, Mobile and Foreman video sequences are naturally recorded, while Cloud and Dinosaur video sequences are synthetically produced. Mobile and Calendar, Dinosaur and Cloud video sequences have RGB components while Foreman and Hall have a single component of luminance. Although our proposed method makes a decrease in the entropy of both classes of video sequences, the entropy of naturally recorded is more decreased than that of synthetically produced. In fact, the proposed algorithm performs significantly better for the synthetically created video sequences. It can be justified that the synthetically created video sequences have less temporal fluctuations and can be approximated with a small number of CPs. The number and degree of CPs in the B-spline curve are two factors that must be determined in our method.

The causes for the performance of our method are as follows:

1. Instead of the Bézier curve, the B-spline curve is used in our proposed method which has better interactive flexibility and local control property. Also, the number of CPs can be changed with no need of changing the degree of the B-spline curve. Hence, the introduced method creates a better approximation with desirable precision.

2. The input data in [35] are divided into segments based on the breakpoints and each segment is then approximated by a quadratic Bézier curve. This is while our method fits the input data without segmentation using a B-spline curve. This makes simpler computations for the method proposed compared to the method [35].

3. The LSPIA method used in this study approximates the video data with low fitting errors and without solving a system of equations.

4. The reduction in entropy is higher for synthetically produced than naturally recorded video sequences.

5. In comparison with block level fitting, the pixel level fitting provides more control over accuracy.

The other PIA methods such as composite iterative method with fast convergence rate [20] can be further used to find the optimal CP of the B-spline curve. The weighted parametric curves like the NURBS curve can be applied instead of the B-spline curve. This is a topic of interest for our future work. The authors plan to use the proposed method in H.264 coding that is a modern video compression method with lossless macro-block coding features.

VIII. Conclusion

A practical method for lossless video compression with a B-spline curve has been introduced. The purpose of our method was to fit the data obtained from the color or luminance variations of a spatial position in series of frames. The LSPIA found the optimal CPs and approximated the input data. The introduced method can be used for 3-D color spaces such as *RGB*, $Y C_b C_r$ or *HSV*. The experimental results demonstrated an easier implementation of our proposed algorithm and substantially reduced entropy of video sequences. The superiority of our study lies behind the fact that it causes a reduction in the entropy of all video sequences, particularly the synthetically created ones.

References

[1] K. Sayood, *Introduction to data compression*. Morgan Kaufmann, 2017.

[2] E. V. Pérez, M. Sánchez, R. G. Crespo, *et al.*, "A system to generate signwriting for video tracks enhancing accessibility of deaf people.,"

International Journal of Interactive Multimedia & Artificial Intelligence, vol. 4, no. 6, 2017.

[3] R. C. Joshi, A. G. Singh, M. Joshi, S. Mathur, "A low cost and computationally efficient approach for occlusion handling in video surveillance systems," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 5, no. 7, pp. 28–38, 2019.

[4] S. Biswas, "One-dimensional b–b polynomial and hilbert scan for graylevel image coding," *Pattern Recognition*, vol. 37, no. 4, pp. 789–800, 2004.

[5] M. Sarfraz, A. Masood, "Capturing outlines of planar images using bézier cubics," *Computers & Graphics*, vol. 31, no. 5, pp. 719–729, 2007.

[6] G. Barid Loghmani, A. Ebrahimi, M. Sarfraz, "Capturing outlines of planar generic images by simultaneous curve fitting and subdivision," *Journal of AI and Data Mining*, 2019.

[7] A. Ebrahimi, G. Loghmani, M. Sarfraz, "Capturing outlines of generic shapes with cubic bézier curves using the nelder–mead simplex method," *Iranian Journal of Numerical Analysis and Optimization*, vol. 9, no. 2, pp. 103–121, 2019.

[8] W. Zheng, P. Bo, Y. Liu, W. Wang, "Fast b-spline curve fitting by l-bfgs," *Computer Aided Geometric Design*, vol. 29, no. 7, pp. 448–462, 2012.

[9] A. Ebrahimi, G. B. Loghmani, "B-spline curve fitting by diagonal approximation bfgs methods," *Iranian Journal of Science and Technology, Transactions A: Science*, vol. 43, no. 3, pp. 947–958, 2019.

[10] A. Ebrahimi, G. B. Loghmani, "Shape modeling based on specifying the initial b-spline curve and scaled bfgs optimization method," *Multimedia Tools and Applications*, vol. 77, no. 23, pp. 30331–30351, 2018.

[11] H. W. Lin, H. J. Bao, G. J. Wang, "Totally positive bases and progressive iteration approximation," *Computers & Mathematics with Applications*, vol. 50, no. 3-4, pp. 575–586, 2005.

[12] Y. Kineri, M. Wang, H. Lin, T. Maekawa, "B-spline surface fitting by iterative geometric interpolation/approximation algorithms," *Computer-Aided Design*, vol. 44, no. 7, pp. 697–708, 2012.

[13] J. Delgado, J. M. Peña, "Progressive iterative approximation and bases with the fastest convergence rates," *Computer Aided Geometric Design*, vol. 24, no. 1, pp. 10–18, 2007.

[14] H. Lin, "Local progressive-iterative approximation format for blending curves and patches," *Computer Aided Geometric Design*, vol. 27, no. 4, pp. 322–339, 2010.

[15] L. Lu, "Weighted progressive iteration approximation and convergence analysis," *Computer Aided Geometric Design*, vol. 27, no. 2, pp. 129–137, 2010.

[16] H. Lin, Z. Zhang, "An extended iterative format for the progressive-iteration approximation," *Computers & Graphics*, vol. 35, no. 5, pp. 967–975, 2011.

[17] H. Lin, "Adaptive data fitting by the progressive-iterative approximation," *Computer aided geometric design*, vol. 29, no. 7, pp. 463–473, 2012.

[18] H. Lin, Z. Zhang, "An efficient method for fitting large data sets using t-splines," *SIAM Journal on Scientific Computing*, vol. 35, no. 6, pp. A3052–A3068, 2013.

[19] C. Deng, H. Lin, "Progressive and iterative approximation for least squares b-spline curve and surface fitting," *Computer-Aided Design*, vol. 47, pp. 32–44, 2014.

[20] A. Ebrahimi, G. B. Loghmani, "A composite iterative procedure with fast convergence rate for the progressive-iteration approximation of curves," *Journal of Computational and Applied Mathematics*, vol. 359, pp. 1–15, 2019.

[21] H. Lin, T. Maekawa, C. Deng, "Survey on geometric iterative methods and their applications," *Computer-Aided Design*, vol. 95, pp. 40–51, 2018.

[22] H. Amirpour, M. Ghanbari, A. Pinheiro, M. Pereira, "Motion estimation with chessboard pattern prediction strategy," *Multimedia Tools and Applications*, vol. 78, no. 15, pp. 21785–21804, 2019.

[23] I. Chakrabarti, K. N. S. Batta, S. K. Chatterjee, *Motion Estimation for Video Coding*. Springer, 2015.

[24] S. Zhu, K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE transactions on Image Processing*, vol. 9, no. 2, pp. 287–290, 2000.

[25] C. Zhu, X. Lin, L. P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE transactions on circuits and systems for video technology*, vol. 12, no. 5, pp. 349–355, 2002.

[26] X. Jing, L. P. Chau, "An efficient three-step search algorithm for block

motion estimation," *IEEE transactions on multimedia*, vol. 6, no. 3, pp. 435–438, 2004.

[27] S. D. Kamble, N. V. Thakur, P. R. Bajaj, "Modified three-step search block matching motion estimation and weighted finite automata based fractal video compression.," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 4, no. 4, pp. 27–39, 2017.

[28] L. M. Po, W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE transactions on circuits and systems for video technology*, vol. 6, no. 3, pp. 313–317, 1996.

[29] T. Koga, "Motion compensated interframe coding for video-conferencing," in *Proc. Nat. Telecommun. Conf.*, 1981, pp. G5–3.

[30] X. Fu, D. Liang, D. Wang, "A new video compression algorithm for very low bandwidth using curve fitting method," in *International Conference on Advances in Visual Information Systems*, 2007, pp. 223–229, Springer.

[31] T. K. Truong, S. H. Chen, T. C. Lin, "Medical image compression using cubic spline interpolation with bit-plane compensation," in *Medical Imaging 2007: PACS and Imaging Informatics*, vol. 6516, 2007, p. 65160D, International Society for Optics and Photonics.

[32] T. K. Truong, L. J. Wang, I. S. Reed, W. S. Hsieh, "Image data compression using cubic convolution spline interpolation," *IEEE Transactions on Image Processing*, vol. 9, no. 11, pp. 1988–1995, 2000.

[33] M. A. Khan, Y. Ohno, "Compression of video data using parametric line and natural cubic spline block level approximation," *IEICE transactions on information and systems*, vol. 90, no. 5, pp. 844–850, 2007.

[34] M. A. Khan, "An automated algorithm for approximation of temporal video data using linear b'ezier fitting," *The International Journal of Multimedia and Its Applications*, vol. 2, no. 2, pp. 81–94, 2010.

[35] M. A. Khan, "A new method for video data compression by quadratic bézier curve fitting," *Signal, Image and Video Processing*, vol. 6, no. 1, pp. 19–24, 2012.

M. J. Ebadi

M. J. Ebadi received his BSc degree in Applied Mathematics from Shahid Bahonar University of Kerman, Iran in 2003 and his MSc degree in Applied Mathematics from S&B University, Iran in 2006. In the same year, he joined the Department of Mathematics at Chabahar Maritime University, Iran, as a faculty member. In 2018, he received his Ph.D. in Applied Mathematics from Yazd University, Iran. He is a member of editorial board of two international reputed journals and a reviewer of more than 20 international reputed journals. He has published several international papers in high rank journals. Currently, he is an Assistant Professor in Applied Mathematics at the Department of Mathematics, Chabahar Maritime University, Iran. His research interests include numerical optimization, deep learning, neural networks, numerical analysis, optimal control, fuzzy optimization, numerical optimization, EEG signals classification, and image processing.

A. Ebrahimi

A. Ebrahimi is a researcher in computeraided geometric design at the computer geometry and dynamical systems laboratory of Yazd University, Yazd, Iran. He received his Ph.D. degree in Applied Mathematics from Yazd University in 2019. His research interest includes geometric modeling and image processing.